DemoGraphDB - A Novel Knowledge Graph Approach to Predicting Inpatient Mortality

by Shon T. Mohsin

B.A in Psychology, February 2013, Brooklyn College
M.S. in Data Science, December 2019, Southern Methodist University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Engineering

January 9, 2026

Praxis directed by

Titus Hei Yeung Fong
Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Shon T. Mohsin has passed the Final Examination for the degree of Doctor of Engineering as of November 4, 2025. This is the final and approved form of the dissertation.

DemoGraphDB - A Novel Knowledge Graph Approach to Predicting Inpatient Mortality

Shon T. Mohsin

Dissertation Research Committee:

Titus Hei Yeung Fong, Professorial Lecturer of Engineering Management and Systems Engineering, Praxis Director

John M. Fossaceca, Adjunct Professor of Engineering and Applied Science, Praxis Chair

Jonathon Ng, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Shon T. Mohsin
All rights reserved

Dedication

I dedicate this praxis to my father Dr. Mohsin Ali Khan, my mother Nargis Begum, my wife Justina Mohsin and my children Willow and River Mohsin.

Abboo, your lifelong dedication to learning and reinventing yourself has instilled in me a resilience and fortitude that I only realized after you were gone. I can still see you smoking your pipe and reading a book in the back of my mind. Amma, your brilliance in creating businesses, and your command of people inspired me to lead without title. Your sunny disposition in life has passed down to me and is my greatest treasure. You are both missed.

Tina, you are the love of my life. Nothing I've done would have been possible without you. You've dealt with my highs and my lows. My impulsiveness and my obsessiveness have all been tempered by your even hand. With every day that passes, you are more beautiful to me. Especially through this degree, you've moved heaven and earth to support me. You've earned this more than I have.

To my babies River and Willow; I am sorry I spent so much time away from you to get this done. I want to show you that you can begin anything at any time, and achieve it. Time goes, whether you use it or not.

Acknowledgments

I wish to acknowledge my wonderful cohort. Dan Koster, thank you for your endless positivity. I needed it more than you know. To Jeremiah Lowhorn, your brilliance is what inspired me to strive, and your straightforward evaluation is just what I needed. To Ayo Akinkugbe, thank you for introducing me to Obsidian, it has profoundly changed how I learn. Your drive and determination are an inspiration to me. To the rest of my cohort, thank you for being there through all the highs and lows, for our study groups and flash cards, and the brilliant notes on the praxis. This was a group effort, and I could not have asked for a better group.

To Dr. Fong, thank you for your kind and inspiring words and guidance. Your evaluation of my progess was always a lot rosier than mine, but you were right in the end!

Abstract

DemoGraphDB - A Novel Knowledge Graph Approach to Predicting Inpatient Mortality

This praxis explores the integration of Social Determinants of Health (SDOH) with healthcare data using Knowledge Graph (KG) embeddings to predict inpatient mortality at the time of admission. The research is motivated by the significant impact of SDOH on health outcomes and the potential for KGs to provide a structured representation of complex relationships between entities.

The methodology involves creating a novel framework, DemoGraphDB, which combines healthcare data from the 2021 New York State Inpatient Database with SDOH data from the American Census Survey geospatially. The SDOH data is structured into a KG and transformed into embeddings that captures all information present for a specific FIPS code using the StarE graph neural network algorithm. This embedding is then appended to each patient's FIPS code and used as input into a transformer model. The full framework improves mortality prediction accuracy by 86 basis points compared to a baseline logistic regression model.

The research demonstrates the feasibility of using KGs to integrate diverse datasets and enhance predictive power in healthcare applications. The study also highlights the potential of using KGs to inject domain knowledge, improve data integration, and enable causal reasoning in machine learning models. The findings suggest that incorporating SDOH into predictive models leads to increased prediction accuracy that may enable proactive healthcare interventions, ultimately benefiting patient outcomes and reducing healthcare costs.

Table of Contents

Dedication iv

Acknowledgments v

Abstract vi

List of Figures x

List of Tables xi

List of Abbreviations xii

Chapter 1: Introduction 1
1.1 Background 1
1.1.1 Social Determinants of Health and the Healthcare System . . . 1
1.1.2 Knowledge Graphs and Social Determinants of Health Integration 3
1.2 Research Motivation 4
1.3 Problem Statement 4
1.4 Thesis Statement 5
1.5 Research Objectives 6
1.6 Research Questions and Hypothesis 7
1.7 Scope of Research 8
1.8 Research Limitations 9
1.9 Organization of Praxis 9

Chapter 2: Literature Review 11
2.1 Introduction 11
2.2 The Regulatory Landscape and Innovation Sharing Challenges in Healthcare 14
2.2.1 Healthcare Regulations 15
2.2.2 Innovation Sharing Challenges 16
2.3 Social Determinants of Health 18
2.3.1 SDOH influence on health outcomes 18
2.3.2 Social Determinant Categories 19
2.4 Knowledge Graphs 20
2.4.1 History of Knowledge Graphs 20
2.4.2 Graph Theory and Knowledge Graphs 23
2.4.3 Advantages of knowledge graphs 25
2.5 Neural Networks 27
2.5.1 Transformers 29
2.6 Graph Embeddings 32
2.6.1 Graph Neural Networks 33
2.6.2 StarE 36

2.7 The current state of ML in healthcare 38
2.7.1 Diagnostic Applications 38
2.7.2 Personalized Treatment Strategies 39
2.7.3 Predictive Analytics 39
2.7.4 Hybrid Models 40
2.7.5 Electronic Health Records Analysis 40
2.7.6 Recent ML Approaches Using KG and SDOH 40
2.8 Conclusion 42

Chapter 3: Methodology **44**
3.1 Introduction 44
3.2 Model Architecture 45
3.3 Compute 46
3.4 Datasets 46
3.4.1 Healthcare Cost and Utilization Project and the State Inpatient Database 46
3.4.2 Geospatial Graph 53
3.4.3 American Community Survey 54
3.4.4 GEOACSKG — The Knowledge Graph of ACS data 55
3.5 GEOACSKG StarE Embedding 58
3.5.1 Dataset Preparation 59
3.5.2 StarE Algorithm on GEOACSKG 61
3.6 Model Iterations 64
3.6.1 Logistic Regression Baseline 64
3.6.2 H1 Model Architecture 67
3.6.3 H2 Model Architecture 71
3.6.4 H3 Model Architecture 72
3.7 Conclusion 75

Chapter 4: Results and Analysis **77**
4.1 Introduction 77
4.2 Embedding Evaluation 77
4.2.1 Node2vec Embedding of HCUP data 78
4.2.2 StarE Embedding of ACS data 79
4.3 Model Descriptions 80
4.4 Hypothesis Evaluation 81
4.4.1 Hypothesis 1 Evaluation 81
4.4.2 Hypothesis 2 Evaluation 82
4.4.3 Hypothesis 3 Evaluation 83
4.5 All Model Performance Summary 84
4.5.1 Confusion Matrix of All Models 86
4.5.2 McNemar's test comparing Asymmetry within models 87
4.5.3 McNemar's test against baseline LR + OHE 88
4.6 Conclusion 89

Chapter 5: Conclusion **91**
5.1 Overview of the Study 91
5.2 Summary of Key Findings 91
5.3 Discussion of Research Questions and Hypotheses 92
5.4 Contributions to Knowledge 93
5.5 Implications for Practice and Policy 93
5.6 Limitations of the Study 94
5.7 Recommendations for Future Research 95
5.8 Closing Remarks 95

Appendix A: Github Link **107**

List of Figures

1 Euler's Seven Bridges of Königsberg. Source: Wikimedia Commons [Public Domain] 21
2 Example of a Triple. 24
3 A simple neural network with two hidden layers. 29
4 The transformer architecture. Adapted from Vaswani et al. 30
5 The attention mechanism. 31
6 Hyper-relational facts. 36
7 StarE hyper-relational fact encoder. 37

8 Architecture of H1, H2, and H3 models with shared components. 45
9 Age distribution of non-negative records from the 2021 CORE file. 48
10 `DIED` counts present in the 2021 CORE file. 49
11 Average Length of Stay for patients that died vs. those who did not in the 2021 CORE file. 50
12 Count of columns with null values in joined CORE and DXPR file. 51
13 Base geospatial graph of New York State with counties and FIPS code connected. 53
14 A sample of the ACS mapping starting with Root Node USA. . . 57
15 StarE Algorithm implementation. 58
16 Logistic regression model architecture. 64
17 H1 Model high-level archticture. 67
18 Node2Vec search bias illustration. Adapted from Labonne, 2023. 68
19 H2 Model high-level archticture. 71
20 H3 Model high-level archticture. 72

21 t-SNE of HCUP Embedding. 78
22 t-SNE of GEOACSKG Embedding. 79
23 ROC Curve — All Models. 85
24 Precision-Recall — All Models. 86

List of Tables

3.1 Summary statistics comparing died vs. not died. 50
3.2 Data composition of train and test split. 52
3.3 Example of Census Block Group components. 54
3.4 Sample of CBG file with the first 5 columns. 55
3.5 H3 Transformer stages and shapes. 74
3.6 H3 Transformer: training hyperparameters. 75

4.1 Models and their input datasets. 81
4.2 H1 — KG methods yield > chance accuracy on HCUP Data. . . 82
4.3 H2 — Does TRANS + OHE improve LR accuracy? 83
4.4 H3 — Adding SDOH KG and using a Transformer improves accuracy over LR-only 84
4.5 Model performance summary. 85
4.6 Confusion Matrix Summary by Model. 87
4.7 Row-normalized Confusion Matrix Percentages by Model. . . . 87
4.8 McNemar's test for FP–FN asymmetry within each model. . . . 88
4.9 Pairwise Comparison of Models vs. Baseline (LR + OHE). 88

List of Abbreviations

SDOH Social Determinants of Health

KG Knowledge Graph

GEOACSKG Geospatial ACS Knowledge Graph

GEOKG Geospatial Knowledge Graph

ACSKG American Community Survey Knowledge Graph

HCUP Healthcare Cost and Utilization Project

SID State Inpatient Database

HCUPKG HCUP Knowledge Graph

FHIR Fast Healthcare Interoperability Resources

FIPS Federal Information Processing Standards

CBG Census Block Group

ZIP Zone Improvement Plan (ZIP code)

ICU Intensive Care Unit

MODS Multiple Organ Dysfunction Syndrome

EHR Electronic Health Record

HIPAA Health Insurance Portability and Accountability Act

ACA Affordable Care Act

PCORI Patient-Centered Outcomes Research Institute

RDF Resource Description Framework

OWL Web Ontology Language

ML Machine Learning

GNN Graph Neural Network

GCN Graph Convolutional Network

StarE Hyper-relational KG embedding framework

NLP Natural Language Processing

LLM Large Language Model

RAG Retrieval-Augmented Generation

GPU Graphics Processing Unit

CNN Convolutional Neural Network

SVM Support Vector Machine

TRANS Transformer-based model

OHE One-Hot Encoding

AUC Area Under the Curve

AUPR Area Under the Precision–Recall Curve

ROC Receiver Operating Characteristic

PR Precision–Recall

TP True Positive

TN True Negative

FP False Positive

FN False Negative

LOS Length of Stay

OHCA Out-of-Hospital Cardiac Arrest

DIED Inpatient mortality indicator (HCUP)

Chapter 1: Introduction

1.1 Background

1.1.1 Social Determinants of Health and the Healthcare System

Social Determinants of Health (SDOH) are the conditions where people are born, grow, live and age. They include larger systems and forces that influence the daily conditions of life. Factors at the individual, local, national and global level influence the overall health and well-being of people (Salehnia et al., 2022). Many studies have shown direct links between health outcomes and income, education, nutrition, housing, environmental exposures, and other circumstances. There is evidence that clinical care impacts 20 % health outcomes, while SDOH affects up to 50 % (Whitman et al., n.d.). As an example, people experiencing homelessness at the time of HIV diagnosis have a 27-fold increase in mortality compared to people diagnosed while housed (Spinelli et al., 2019). The literature makes it evident that the impacts of a patient's associated SDOH explicitly express themselves in health outcomes, and warrant consideration when evaluating treatment options.

SDOH differences can impact the cost of healthcare. Healthcare costs are a growing concern of individuals and governments. In 2022, the cost of healthcare in the United States is estimated at 4.4 trillion dollars and represents 17. 3% of the gross domestic product (GDP), down from a peak of 19.7% in 2020 ("U.S. Health expenditure as GDP share 1960-2022," n.d.). Cost analysis in Canada shows that the cost per day for a patient in an Intensive Care Unit (ICU) bed can be up to $18,000 more, adjusted for

inflation, than a regular bed (Noseworthy et al., 1996). In the US between 2005 to 2013, medical debt was the single biggest cause of bankruptcy for individuals (Ciulla et al., 2023). However, these high costs do not lead to better outcomes. In fact, a report from the Commonwealth Fund's analysis of 10 developed countries ranking healthcare systems found the United States in the last place ("Mirror, Mirror 2024," 2024).

Moving to the clinical setting, intensive care unit (ICU) staff need quick decision-making to facilitate patient care. To do so, operating rooms and emergency departments use a framework in which an intensivist is present at the same place as the patient. An intensivist is responsible for overseeing the care of patients with severe, life-threatening illness or injuries. The intensivist monitors the patient through alarms, patient history information, and other feedback to react accordingly as patient care needs evolve. Unlike scheduled procedures, where deliberation panels can convene to determine the best course of action for a patient, intensivists in the ICU must make quick decisions in reaction to immediate patient care needs. Decisions are often based on notifications and alarms from connected monitoring equipment. Repeated exposure to alarms with a high false-positive rate often leads to being ignored until additional events occur. This kind of attention and decision-making skill requires years of experience in that specific setting (Murias et al., 2009).

Unfortunately, there is a shortage of physicians in general, and intensivists in particular in ICUs. The American Hospital Association states that 48% of the hospitals surveyed were not fully staffed by critical care teams. In 2020, the US healthcare system faced a shortage of 7,900 critical care physicians (Jatoi et al., 2022). This shortfall likely reduces the time intensivists can spend with patients in the ICU, and increases volume of

rapid decisions made. Solutions that proactively identify high-risk patients can help intensivists direct attention to those that need it most, among other uses.

1.1.2 Knowledge Graphs and Social Determinants of Health Integration

This praxis approaches predicting patients with higher mortality risk at admission time by integrating a Knowledge Graph (KG), which is a kind of semantic network, into a machine learning pipeline. A semantic network is a graphical representation of knowledge where entities (concepts, objects) are represented as nodes, and relationships between these entities are represented as edges. In computer science, semantic networks were first implemented by Richard H. Richens to formalize the use of semantic meaning in a structured format (Richens, 1958). The term KG was first popularized with the announcement of the "Google Knowledge Graph" in 2012 (Haslhofer et al., 2018). An advantage of KGs over traditional databases is the ability to express relationships between entities in a structured manner.

A specific kind of KG is a directed graph. In a directed graph, each edge has a direction, providing additional semantic information. The edges themselves can also have labels. When used as recommender systems, KGs can introduce semantic relatedness between nodes. These special properties of semantic relatedness and directionality provide additional information not easily discoverable in traditional relational database structures (Mao et al., 2021).

This praxis proposes a novel approach to mortality prediction from patient hospital records for the year 2021 from New York in combination with SDOH from the American Census Survey (ACS) at the patient's US Federal

Information Processing Standards (FIPS) Code level. The ACS data is mapped to a KG, and an embedding is extracted using a transformer-based Graph Neural Network (GNN). This embedding is then joined with the hospital admission record by FIPS to augment the record, which is then fed into another transformer Neural Network (NN) to classify patient mortality.

1.2 Research Motivation

Utilizing SDOH to augment the admissions data when training a classification model has the potential to increase the accuracy of the model when compared to the same model without data augmentation. The ability to predict mortality with an increasing degree of accuracy may allow proactive measures that can potentially save lives and prevent additional cost from the treatment of multiple organ dysfunction syndrome (MODS) and other negative outcomes (Asim et al., 2020). The cognitive burden of patient evaluation for critical care specialists may be reduced if mortality events can be predicted beforehand. This can allow more time for research, consultation and deliberation of the best treatment options based on current patient status. A predictive system can also provide information to make decisions proactively versus decisions made reactively from alarm systems. Alarm systems may be ignored for too long, and in some cases when critical alarms are activated, intervention times are reduced and care becomes a high-pressure situation.

1.3 Problem Statement

Excluding socioeconomic factors, which drive 80% of health outcomes (Payne et al., 2021) leads to 22% reduction in mortality classification accuracy

in predictive models. (Segar et al., 2022).

Studies have shown links between SDOH and health outcomes. Factors like a parent's education, years worked, income and insurance parity explain disparities in psychological health and well-being better than individual attributes like age, sex, and healthcare utilization (Jester et al., 2022). Income inequality, social policies and maternal socioeconomic status have been shown to affect infant mortality rates as well (Kim & Saada, 2013).

Since the links between SDOH and health outcomes have been established, researchers have been able to combine SDOH with patient health records to predict future risk in different ways. For example, machine learning models were able to improve the prediction of major adverse cardiovascular events (MACEs) within two years of breast cancer diagnosis for non-Hispanic Black patients (Stabellini et al., 2023). For out-of-hospital cardiac arrest (OHCA) survival, machine learning clustering analysis was able to identify distinct SDOH measures associated with mortality after 1 year (Abbott et al., 2023). These studies demonstrate the usefulness of incorporating SDOH into ML models for downstream predictions.

1.4 Thesis Statement

DemoGraphDB, a framework that creates a knowledge graph of socioeconomic data, creates an embedding out of the knowledge graph using Graph Neural Networks, augments inpatient admission records with the embedding and trains a transformer model, enables that model to improve mortality classification by 86 basis points.

Any improvement in the predictive power of ML models to predict health outcomes will naturally have benefits that can be realized in either policy or clinical settings. For this reason, multiple ML approaches utilizing SDOH

have been proposed in the literature. This praxis proposes a framework that utilizes SDOH in conjunction with KGs to enhance predictive power for mortality classification from emergency room admissions records.

1.5 Research Objectives

This paper primarily proposes a novel multi-model approach to predicting inpatient mortality that increases predictive power. First, SDOH data is mapped to a KG. Then, using a GNN, an embedding is extracted at the FIPS code level. This embedding is then joined by the FIPS code to patient admission data and fed to a transformer model to classify mortality events. The overall goal is to demonstrate a repeatable set of architectural decisions that researchers can use to set up their architecture and execute specific steps when creating predictive healthcare models to incorporate publicly available SDOH data to maximize the predictive power of their models.

The detailed objectives are to first demonstrate the feasibility of loading healthcare data, specifically the New York 2021 State Inpatient Database (SID) from the Healthcare Cost & Utilization Project (HCUP) into a KG. Next, output that KG embedding and use it to create a classification model that can predict mortality events using logistic regression (LR) with higher than random probability. This demonstrates that a KG embedding is able to encode useful information. Then, demonstrate the feasibility of using a transformer model to increase predictive power.

For the final model, create an embedding of SDOH from a knowledge graph of the American Community Survey using the StarE GNN. Finally, join the embedding with HCUP data on patient FIPS code to train a transformer model to classify mortality events. This will demonstrate that the final approach can increase by 86 basis points the prediction accuracy of

mortality events compared to a baseline LR model.

The detailed objectives are:

1. Demonstrate the feasibility of loading healthcare data, specifically the New York 2021 SID from the HCUP into a KG.

2. Output an embedding from the KG and use it to train a classification model that predicts inpatient mortality using LR, showing performance better than random chance.

3. Demonstrate the feasibility of loading SDOH data from the ACS into a KG, creating an embedding, and mapping individual patient records by FIPS code to census data.

4. Create a transformer model that trains on HCUP data only to demonstrate gains from using transformer architecture.

5. Use the same transformer architecture created previously, but train on HCUP data augmented with SDOH embeddings to demonstrate performance gain from SDOH data augmentation.

6. Show that the final approach increases prediction accuracy by 86 basis points compared to the baseline model, and performs better than all other models created.

1.6 Research Questions and Hypothesis

The research questions addressed by this praxis are:

- **Research Question 1.** Can Knowledge Graph Theory be applied to predict mortality from inpatient admissions data?

- **Research Question 2.** Can an advanced neural network model increase predictive power on inpatient mortality detection?
- **Research Question 3.** Is there a particular architecture that combines SDOH and ML that performs best classifying inpatient mortality?

Based on the research questions, the following hypotheses are proposed:

- **Hypothesis 1.** Knowledge Graph Theory can be applied directly to inpatient admissions data to classify mortality events with more than random accuracy.
- **Hypothesis 2.** Incorporating a neural network increases classification accuracy by 60 basis points in predicting mortality from hospital admissions data.
- **Hypothesis 3.** Incorporating Social Determinants of Health using a Knowledge Graph embedding with hospital admissions data and using transformer architecture increases mortality classification accuracy compared to a baseline logistic regression model trained on only healthcare data.

1.7 Scope of Research

The scope of this research is first, to explore the viability of using KG as a data repository for healthcare data. Next, to prove that using a transformer model yields increases in performance over a simpler logistic regression model on HCUP data. Next, to create an embedding from a KG of SDOH data, and use that embedding to augment healthcare data to train a transformer model, proving that creating this kind of multi-model architecture creates an increase in predictive power. The hope of this praxis is to demonstrate

that SDOH data provides a measurable increase to predictive power, and should be used to augment healthcare data when available.

1.8 Research Limitations

The exact results of the research are limited by the sources of data. Specifically, the HCUP data is limited to the state of NY for the year 2021 and has only inpatient data records from hospitals. No other information about patients, their health records, or lifestyles is provided. The ACS data is from 2019, and does not capture the changes from 2019 to 2021, the year the HCUP data is available. It also makes many approximations to arrive at population-level statistics and is limited by user response for tabulation. The ACS data is also filtered on the State of NY to align with patient FIPS codes, and may present unique latent characteristics that are only present in the NY population. There may be inherent latent factors present or absent in both of these datasets that are not captured during the embedding and training process due to differences in year and collection methods. Therefore, the results are only demonstrative and should not be used in a clinical setting without research into these considerations and mitigation of any inherent biases present or omissions as a result of the data collection methods.

1.9 Organization of Praxis

The remainder of the Praxis is structured as follows:

Literature Review (Chapter 2): Chapter 2 provides a curated overview and exploration of existing literature on several topics of interest. First, an elaboration and deep dive into SDOH and how it affects populations and

individuals. Then, the state of ML in healthcare and some published results and approaches. Next, a look at fundamental and advanced KG theory and application in various industries. Finally, a deep dive and discussion of transformers architecture, and how it is used in multiple problem spaces for both predictive and generative tasks.

Methodology (Chapter 3): Chapter 3 describes the HCUP and ACS data, and the methods implemented to create the KG and ML models. First, the collection methods and an evaluation of the American Community Survey Census Data are discussed. Then, the details of the 2021 HCUP SID for the State of New York are explored with descriptive statistics. Finally, the architecture for the 3 main models proposed in the hypotheses and 2 supplemental models is provided. The associated execution details, along with the software and hardware used to train and obtain results, are provided for reproducibility.

Results (Chapter 4): In Chapter 4, the embeddings from the HCUP data and the ACS data are analyzed using t-SNE to reduce dimensionality and visualize data grouping and separation patterns. The results of the 3 main models and 2 supplemental models are documented. An evaluation of each hypothesis with confidence intervals, z-tests and a decision to reject the null hypothesis are provided. Comparisons between the models using McNemar's tests and confusion matrices are performed to detect differences in model performance.

Discussion and Conclusion (Chapter 5): Chapter 5 provides a summary of the key findings and their significance for healthcare. The contributions made to the field are documented here, along with implications for practice and policy. The limitations of the study, recommendations for future research, and closing remarks conclude the chapter.

Chapter 2: Literature Review

2.1 Introduction

In 2025, we are witnessing expansion and rapid adoption of Artificial Intelligence (AI) and machine learning (ML) capabilities. Incredible investments in compute infrastructure are announced nearly every day. World governments are actively granting subsidies to tech companies to build compute infrastructure for AI (Center for Security and Emerging Technology & Hwang, 2020). The rapid pace of technical breakthroughs in AI architecture, open access models, cost-effective cloud compute and open data initiatives are converging to create a proliferation of adoption (Gosselink et al., 2024). This increase in innovation and investment is cascading down to almost all industries.

In the healthcare domain however, organizational entities are taking a more cautious approach. Healthcare is governed by strict privacy regulations. Consequently, data acquisition and sharing are slowed by privacy requirements. ML research that can benefit from data sharing are often conducted internally without open sourcing due privacy concerns. In addition, research by some organizations is conducted in silos to preserve competitive advantage. These factors lead to slower dissemination of innovation across the industry (Marjanovic et al., 2020). An alternative data source that holds promise in aiding research applications and is available to all entities is Social Determinants of Health (SDOH) data.

Incorporating SDOH data to supplement healthcare datasets can increase relevant information for patient health records, as the results of this praxis show. SDOH data is publicly available and collected by various public and

private agencies across the world, and the American Census Bureau in the United States. Numerous studies have shown the impacts of how different socioeconomic factors affect patient health outcome, making this publicly available information a valuable input source for ML applications. Research using SDOH in ML applications demonstrate to various degrees the efficacy in predictive tasks (Al Saidi et al., 2025; Salehnia et al., 2022; Spinelli et al., 2019).

Due to the public nature of this data, entities can adopt usage across the industry. Components built using SDOH can easily be shared without privacy concerns. In addition, SDOH data are inherently geospatial in nature, consisting of aggregate statistics specifically tied to geographical areas. This property makes SDOH data an ideal candidate to map to a graph network representing a geographical region. Mapping information in a graph network creates a Knowledge Graph (KG). A KG is a structured representation of information that captures relationships between entities in a graph format. Once SDOH is mapped to create a KG, several key properties are unlocked that can be used in ML applications.

Using KGs to represent information provides specific advantages over traditional relational databases, like the ability to inject domain knowledge into downstream models, feature enrichment of existing data, information gain from inductive bias, and causal reasoning injection from structured knowledge mapping. Knowledge transfer using shared ontology and relations provides another key advantage over traditional databases, but is not used in this study.

There are applications that can be used directly on a KG without further transformation. For example, KGs are used for recommender systems, rank answers to questions, and drug discovery using biomedical KGs to predict

drug-target interaction (Liang et al., 2023; Mohamed et al., 2025; Q. Wang et al., 2017). A drawback of representing data in a graph is the loss of the tabular structure present in traditional databases. To overcome this limitation, KGs can be converted to graph embeddings to capture the latent information and structure present in the KG for downstream applications that require tabular data.

A graph embedding maps nodes and relations along with the structural composition of local and global neighborhoods, to an n-dimensional latent space, yielding a normalized matrix suitable for standard models. Common methods to create embeddings include random-walk and matrix-factorization, as well as deep learning approaches such as node2vec and StarE (Goyal & Ferrara, 2018). KG embeddings preserve the graph's distinctive relational signals and neighborhood structure while projecting the the information down into a latent space in a matrix format that is ready to use in downstream models like Neural Networks (NN).

NNs have largely supplanted traditional models in domains such as natural language processing, computer vision, and biomedical prediction, especially in instances where data is plentiful and model performance optimization is the main objective (Rudolph et al., 2023). The complexity of the information that is projected into the latent space from a KG embedding along with the numerous columns present in the HCUP dataset make using a transformer model the optimal choice to create the final inferences for DemoGraphDB.

In the upcoming sections, information about the various regulations in the US healthcare industry frame the current state of healthcare compliance and challenges. A review of SDOH highlights the importance of this dataset for DemoGraphDB. An overview of the current state of ML research in

healthcare illustrates the various approaches being taken in the domain, and highlights the gaps in literature that DemoGraphDB framework can address. A brief history of KGs, and foundational information about Graph Theory relevant to this study build the theoretical backbone of the algorithms used for DemoGraphDB. A deep dive into embedding techniques provides justification on why the StarE deep learning embedding is chosen as the final embedding method. A brief overview of the theory and application of NNs, transformer model architecture, and the Attention mechanism create the context for how the KG embedding and healthcare data are used as input and processed to train and serve predictions for mortality events in the complete DemoGraphDB framework.

2.2 The Regulatory Landscape and Innovation Sharing Challenges in Healthcare

Healthcare organizations like providers, insurers and government agencies generate a tremendous amount of healthcare data in their day to day operations. Patient privacy needs and protected health record regulations require strict control to ensure this data is not exposed publicly. Consequently, cutting-edge machine learning research that could benefit from training on this data often overlooks the healthcare space (Mendelevitch & Lesh, 2021). For companies that do invest in research, the data is internal, and the results are often siloed within the organization for those same privacy and regulatory concerns. This leads to disjointed innovation-sharing. Despite these concerns, finding a way to create a space for shared innovation in healthcare can lead to real positive impact for patients and society at large.

2.2.1 Healthcare Regulations

To protect patient privacy, the privacy rule standards set forth by the Health Insurance Portability and Accountability Act of 1996 (HIPAA) cover patient protected health information (PHI) and how entities like healthcare providers, health plans, healthcare clearinghouses, and others can use and share a patient's information. Specifically, HIPAA mandates limited dataset usage for research, public health initiatives, or the administration of research and investigation for healthcare operations (CDC, 2024).

These mandates from HIPAA have created friction for organizations that wish to share data for research. The US government recognizes this issue and has created additional regulations to facilitate smoother data sharing among entities. The Patient Protection and Affordable Care Act (ACA) enacted in March 2010 created the Patient-Centered Outcomes and Research Institute (PCORI) and the Health Information Technology (Health IT) initiative. PCORI provides grants and financial support to comparative effectiveness research. It also mandates public access to research findings within 90 days. A successful implementation example of a PCORI-funded project expanded the use of community health workers, showing improved quality of care and reduced hospital stays for socioeconomically disadvantaged patients ("Expanding the Use of IMPaCT," 2019). The Health IT initiative mandates the adoption of certified Electronic Health Record (EHR) systems to facilitate interoperable Health IT infrastructure and encourages physicians to adopt standardized EHR data practices (U.S. Congress, 2010). Adopting EHR means data is recorded with uniform standards, facilitating interoperability between organizations.

The 21st Century Cures Act of 2016 also designates a Data Standards mandate that discourages information blocking practices, which are actions

that interfere with the access, exchange, or use of EHR, as well as providing funding for high-risk, high-reward research as detailed in SEC. 2036 (Commissioner, 2025). However, this data sharing is mostly for patients that request their data or ask for their records to be transferred from one provider to another.

These initiatives and specifically the Healthcare Utilization Project (HCUP) dataset publications, which this praxis uses, have eased the use of data sharing with reasonable protections for patient privacy. Although the data-sharing aspect has eased slightly, healthcare data is still very hard to obtain, and organizations still silo research for competitive advantage.

2.2.2 Innovation Sharing Challenges

To maintain competitive advantage, entities in healthcare often create organizational research silos, intellectual property frameworks, and focus on problem-specific applications. This is understandable, as fundamentally, most healthcare organizations are businesses that can benefit from having a competitive advantage.

Healthcare companies often focus on specific medical issues that directly concern them and create problem-specific applications, which leads to the allocation of resources to highly targeted solutions (Marjanovic et al., 2020). These practices, however, promote work in isolation while the resulting fragmentation hinders interdisciplinary collaboration. This often leads to inefficiencies and duplicated effort. Although this type of focus can lead to advances in specific research categories, the often narrow research scope overlooks broader systemic innovations that could be realized from more interdisciplinary research collaborations. Another issue with this kind of targeted research is the impetus to create intellectual property (IP). With

these different problems grouped together, the combined effect limits the spread of innovative solutions throughout the healthcare system and wastes the potential for comprehensive advancements that could be gained from collaboration (Lau et al., 2024). A shift in focus from these targeted solutions to the broader challenges at the population level can lead to solutions that address the root causes of health inequities holistically (Pettigrew & Cooke, 2022).

In other domains, deep research funded by private organizations often lead to innovations that are registered as IP. Registering IPs can inadvertently stifle collaborative research regardless of industry. Patents granting exclusive rights can lead to "Tragedy of the Anticommons," a type of market failure from fragmented property rights and coordination breakdown (R. Wang et al., 2024). An implementation of a larger idea that encompasses multiple IP components would require assembling IP rights in a piecemeal fashion. Often, the effort to secure multiple agreements exceeds the benefit of doing so (Major et al., 2016).

To summarize, researchers in the healthcare domain must conform to regulatory requirements, be cognizant of data sharing and privacy agreements, and navigate complex IP laws and configurations. With the additional impetus of entities to silo research, it becomes difficult for researchers to utilize solutions across the board. Although broad efforts to facilitate collaboration have been implemented, the scale of difficulty still remains daunting. The proposed framework of DemoGraphDB seeks to mitigate some of these concerns by demonstrating a reusable solution using publicly available SDOH data that can be used for appropriate ML predictive tasks in healthcare, with multiple downstream use cases possible.

2.3 Social Determinants of Health

A key component of the DemoGraphDB is publicly available SDOH data. SDOH has demonstrated to be a remarkably reliable influencing factor in predicting health outcomes in various settings. Broad and refined categories of SDOH have emerged to keep this collection of data organized. A review of literature on the uses of SDOH in various healthcare and healthcare-adjacent applications, including selected applications of ML, shows how organizations today are using SDOH to further research and clinical applications.

2.3.1 SDOH influence on health outcomes

Although the numbers vary, some studies demonstrate SDOH can account for 30% to 55% of health outcomes. SDOH has been shown to be a statistically significant factor in conjunction with the availability of health resources in infant mortality (Al Saidi et al., 2025). SDOH can also influence patient behavior and predispose them to either make positive or negative choices regarding their health. Differing levels of income, education levels, housing stability, and access to nutritious food can increase or decrease a patient's engagement and adherence to care plans provided by health insurance, causing direct and indirect impacts to care costs over short and long time periods (Adeogun & Faezipour, 2025). Research shows that SDOH can be used to predict health-related quality of life (HRQOL), with studies demonstrating that just four SDOH indicators of being in a relationship, the level of education, occupational status, and net income per household can increase or decrease HRQOL (Abegaz et al., 2025).

2.3.2 Social Determinant Categories

To understand what data actually qualifies as SDOH, some classification and standardization is required, with several frameworks proposed (Barros et al., 2010; Novilla et al., 2023). Broadly, SDOH factors can be classified into four levels: individual, local, national, and global. Individual factors comprise education, income, lifestyle choices like diet and exercise, access to healthcare, and occupation. Local factors include neighborhood conditions like access to housing and clean water, community resources like grocery store availability and access to parks, local economic conditions like job opportunities and employment rates, crime and safety conditions, and access to transportation. National factors include the healthcare system and access to coverage, economic policies like the minimum wage and taxation, education policies like access and quality of schools, social programs like housing support and food assistance, and environmental regulations like policies on pollution and clean air and water standards. Global factors include global trade policies like access to medicine and food supplies, climate change impacts, pandemics and diseases, global inequities like disparities in technology between countries, and the influence of international aid and development (Murray et al., 2020; Salehnia et al., 2022). For the purposes of DemoGraphDB, the availability of data is the driving factor, and not necessarily the finer details of SDOH classification methodology. A recurring theme in these levels is the anchoring of data to a specific location. This geospatial property provides a base to "hang" the data in a KG and is used as the foundational KG 'lattice' to construct both the HCUP and ACS datasets, since both include FIPS codes as a data column.

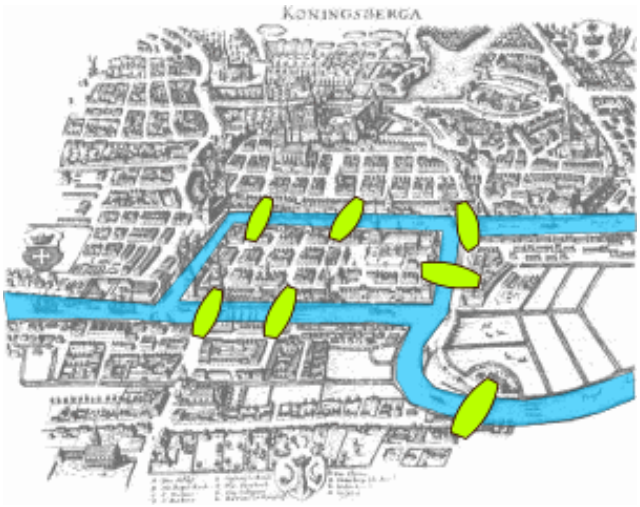
2.4 Knowledge Graphs

KGs are a core component of the DemoGraphDB framework. A KG is used because it has distinct advantages over traditional architectures as a store of information. Using a KG for data means data forms a semantic structure from linking nodes via edges that mimics the properties of semantic relations between words in language. This is a crucial detail of why KG embeddings perform well when used with transformers both to create embeddings, and also when using embeddings for downstream applications. The presence or absence of connecting edges also forms local and global structures within the graph. In the next section, a brief history, a review of graph theory, and some interesting properties of KGs are covered.

2.4.1 History of Knowledge Graphs

The first description of graph theory was loosely proposed by Leonhard Euler in 1736 describing the solution to the seven-bridge problem of Königsberg (now Kaliningrad, Russia) which had two large islands connected to each other and the mainland by seven bridges as depicted in Figure 1. Euler wanted to see if there was an optimal way to traverse the island using the 7 bridges in the town. The solution he found was a walk through the city that crosses each bridge exactly once.

Figure 1
Euler's Seven Bridges of Königsberg.
Source: Wikimedia Commons [Public Domain]



Sadly, from Euler's time to the 20th century, there was little notable research in the field, until graph theory became a formal branch of mathematics (Abu Saleem, 2020). In 1958, Richens proposed the concept of a network of bonded semantic elements, known as a Semantic Net, to solve the problem of language translation via the foundational rules underpinning language, rather than through an intermediary language, for example from Chinese to Japanese without English as a translation medium (Richens, 1958). In 1968, Marvin Minsky from MIT published one of the first books defining the core concept behind what would later become known as Knowledge Graphs. This book is a compendium of his work and papers from his students at the time. His student Bertram Raphael expanded the concept of a semantic network and created one of the first computer programs to utilize these networks using Semantic Information Retrieval (SIR), a dynamic semantic model that learns from user input and answers novel questions from the learned knowledge (Minsky, 1968).

The 1970s and onward saw an acceleration of discovery and increasing

interest in KGs. In the beginning of the 1970s, Expert Systems with AI programs mimicking decision-making abilities using manually coded rules-based inference engines were first created. This process was somewhat standardized in 1983 with the introduction of the Knowledge Engineering Environment, a tool for building expert systems using frames and rules, a frame-based knowledge representation with object-oriented features. This was further refined with Frame-based Languages in the late 1980s, culminating in KL-ONE, which provides hierarchical classification with defined semantics. 1984 saw the initiation of the Cyc Project, an ambitious AI project to encode common sense knowledge in a formalized knowledge base. Unfortunately, the technology to achieve that grand ambition did not exist at the time.

In 1999, the Resource Description Framework (RDF), a standardized model for data interchange on the web was introduced by the World Wide Web Consortium ("RDF - Semantic Web Standards," n.d.). In 2004, W3C introduced Web Ontology Language (OWL) for the RDF to introduce best practices for ontology (the formal representation of a set of concepts, entities, and their relationships in a specific domain) creation in collaboration with researchers and organizations in computer science and technology. In 2012, Google released the first commercial-scale knowledge graph based on an entity-centric graph to improve their search query results (Liang et al., 2023; Mohamed et al., 2025).

Although the study of graph theory and KGs stalled since Euler's bridge up to the early 20th century, modern applications in ML and AI are rediscovering the potential of KGs and this is now an area of intense research.

2.4.2 Graph Theory and Knowledge Graphs

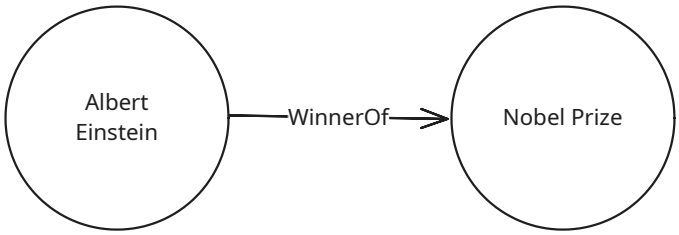
Graph Theory is a branch of discrete mathematics concerned with the study of structures that model pairwise relationships between objects. A graph G consists of a set of vertices or nodes V and a set of edges or relations E. The edges can be undirected in a graph indicating two-way relationships, or directed in a digraph indicating one-way relationships. Graphs can be used to represent a wide variety of real-world systems, from social networks, computer networks, city road systems, biological pathways, elemental relationships, and more (Abu Saleem, 2020).

Definition 1 ((Q. Wang et al., 2017))**.** A knowledge graph is a multi-relational graph composed of entities and relations which are regarded as nodes and different types of edges, respectively.

When real information is depicted in this way, we label the graph structure as a KG. KGs come in 3 flavors: static, temporal and multi-modal. Static KG comprise of static uni-modal facts. Temporal and multi-modal KG are constructed using extra temporal and/or multi-modal information, with their foundation anchored in static KG concepts (Liang et al., 2023).

A KG is denoted as $KG = \{E, R, T\}$, where E, R, T are the set of entities, relations, and triples, respectively. A triple is a fundamental concept of a KG that represents a relationship between entities (a snippet of information), consisting of a head entity, relation, and tail entity in the form $(h, r, t) \in T$ where $h, t \in E$, and $r \in R$ (Xu et al. 2024).

Figure 2
Example of a Triple.



Property graphs are KGs where relations used to connect entities have properties or attributes. In this type of graph, a fact is represented as a triple. (Ji et al., 2022). An example of a factual triple can be (*Albert Einstein*, 'WinnerOf', *Nobel Prize*) as shown in Figure 2. Triples can be used to represent almost any type of relationship between entities, like the connection between people, places, organizations, and concepts. Practically, they can be used to answer questions, reason over data, and perform tasks like entity recognition, relation extraction, and knowledge graph completion (Xu et al., 2024). A triple is a very human-readable way to present information; the descriptive relation between two entities provides context that can be inferred without creating explicit sentences to explain the triple.

KGs are used in various domains, such as finance, e-commerce, education, and manufacturing. Financial firms use KG for risk assessment and fraud detection through first linking disparate financial datasets and then discovering patterns and relationships through various imputation and discovery methods specific to KGs. In manufacturing, KGs enable automation and optimization by integrating different types of data sources based on their ability to link different entities together through relation mapping (Shang, He, et al., 2024). Specific applications of KG use in healthcare are provided in later sections of this chapter. In the next section, some

advantages of using KGs are highlighted.

2.4.3 Advantages of knowledge graphs

KGs provide a few unique benefits over traditional tabular data architectures. Disparate data sizes and sets can be integrated with less overhead. Semantic parsing allows the capturing of knowledge in a structured format. The task of mapping knowledge using triples encodes inductive bias into the structure, which can be leveraged downstream. Data can be mapped with explicit guidance during the graph creation process to encode causal reasoning into the structure, and preserve it during the embedding phase with careful embedding model selection.

2.4.3.1 Improved Data Integration

One of the key strengths of KGs is their ability to integrate multiple sources of data. In a traditional tabular databases, each new dataset might need to be table normalized, and an extensive extract, transform and load process plan implemented to integrate the new data source. The ability to integrate different size datasets without the need to explicitly create schemas eases the time technical teams need to spend on database architecture, as well as providing business users a way to integrate many different data sources intuitively. KGs have a flexible and expandable structure, which reduces the time needed for data integration. Since KGs can represent almost any kind of information as entities or relations, researchers can make intuitive decisions on how to arrange new data into existing KGs. In applications such as question answering and recommendation systems, KGs use this flexible data storage capability to combine diverse data sources like text, images, video and audio, leading to enhanced performance (Von

Rueden et al., 2021). For DemoGraphDB, this flexibility allows for the seamless integration of location data and the SDOH from the ACS to be mapped as a holistic geospatial graph.

2.4.3.2 Injecting Domain Knowledge

KGs are constructed using semantic parsing, which is the process of converting natural language expressions or structured information into triples (Vogt, 2024). This triple can then be encoded into a lower-dimensional latent space via graph embedding methods. The structural information like neighborhood structure and node proximity and semantic information like relationship description and node content is encoded in the embedding. This structural and semantic information captured in the embedding injects domain-specific knowledge into downstream models, improving model performance and accuracy (Dash et al., 2022). In DemoGraphDB, the knowledge graph might encode relationships between different data points such as housing availability and the number of children in the area to make more informed predictions.

2.4.3.3 Inductive Bias in Knowledge Graphs

KGs can introduce inductive bias via their embeddings when used with NNs, allowing them to learn more effectively from structured data. In graph networks, if two entities have a connection between them, the assumption is that they have some sort of interaction. The absence of an edge implies isolation (Cheng et al., 2021). This trait of structuring knowledge by indicating the presence or absence of a relationship explicitly may enhance model performance in tasks like object detection and natural language analysis, where understanding complex relationships is crucial (Von Rueden

et al., 2021). This kind of structure may also guide models towards more effective learning outcomes (Dash et al., 2022). Ultimately, it is hard to prove inductive bias is encoded into a KG embedding, as the embedding process produces a matrix that is not interpretable even when information is captured in the latent space. In the proposed framework, the integration of geospatial structure and the explicit modeling of semantic relationships from the ACS data may introduce inductive bias to enhance the information projected onto the embedding for consumption in by the downstream models.

2.4.3.4 Enabling Causal Reasoning

Causality refers to the relationship between cause and effect, where one event directly influences another event. Although creating a KG does not automatically impart causal reasoning as a property, the act of explicitly connecting facts with data annotation may encode causal information into the relationships between entities in a KG. This capability allows KGs to encode potential causal interactions into embeddings (Narendra & Nagaraj, 2025). The DemoGraphDB pipeline integrates SDOH as structured semantic relationships with geographical location, enabling opportunities for causal reasoning to be encoded in the embedding. Much like inductive bias encoding, causal reasoning encoding is also difficult to discern from an embedding. The hope is that the embedding created captures both inductive bias and causal reasoning in the latent space for DemoGraphDB.

2.5 Neural Networks

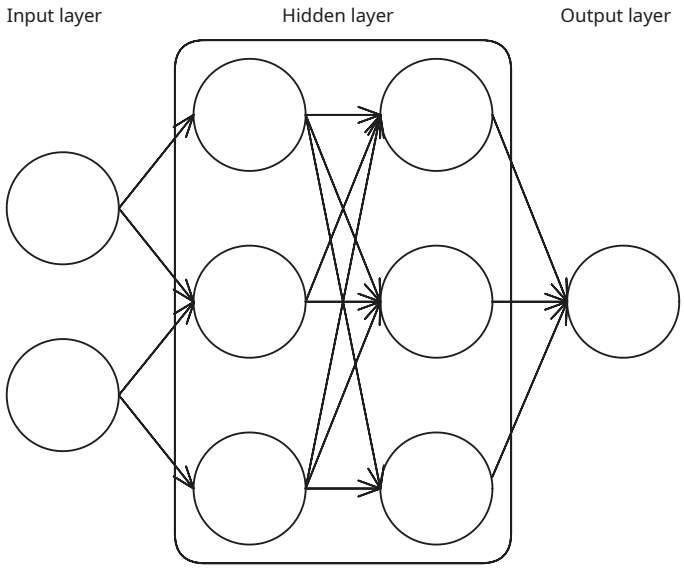
In the framework, the output from the KG is used as the training input for the transformer, which is a type of NN. NNs are deep learning models

consisting of neurons stacked and layered in various shallow or deep configurations as shown in Figure 3. Neurons here are the basic computational unit, taking some input vector of values, multiplying that vector by another learned vector and adding a bias value. Then, it applies a nonlinear function like ReLU or sigmoid to normalize the output between zero and one.

Many of these neurons are stacked together to process a single input vector, each producing a slightly different output from the input using each neuron has own weight parameters. These stacked neurons can be layered as well, with different arrangements connecting one layer of neurons to another layer, with additional processes like dropouts and data augmentation from residual/skip connections creating learned weights that are diverse and capable of capturing intricate latent information, which approximate complex functions that can recreate the latent property with updates as training progresses.

The training process updates these approximate functions by iteratively adjusts the learned weights using gradient descent to continuously search for the global minima that reduces the loss between the learned weight and the actual vector value via the process of backpropagation, which goes back and updates the learned weights after each training pass with respect to the updated gradient (Alves et al., 2024).

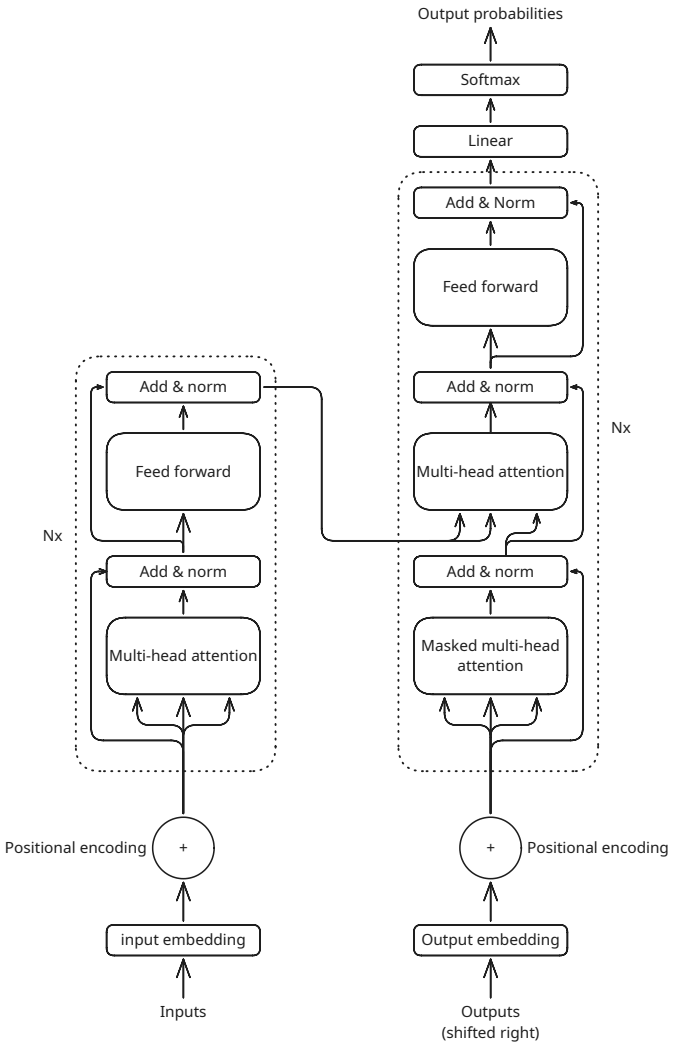
Figure 3
A simple neural network with two hidden layers.



2.5.1 Transformers

Transformers are particularly effective for tasks involving high dimensional, unstructured, or complex data because their architecture, as depicted in Figure 4 allows them to approximate the previously mentioned arbitrary functions with constant updating of the learned weights. The attention mechanism in transformers and the application of residual connections improve the learning efficiency and the expressiveness of the model. The ability to capture, or at least approximate, highly nonlinear functions to represent the shape of data gradients supports the selection of transformer models for the task of predicting inpatient mortality from the HCUP dataset (Rodriguez et al., 2025; Rudolph et al., 2023).

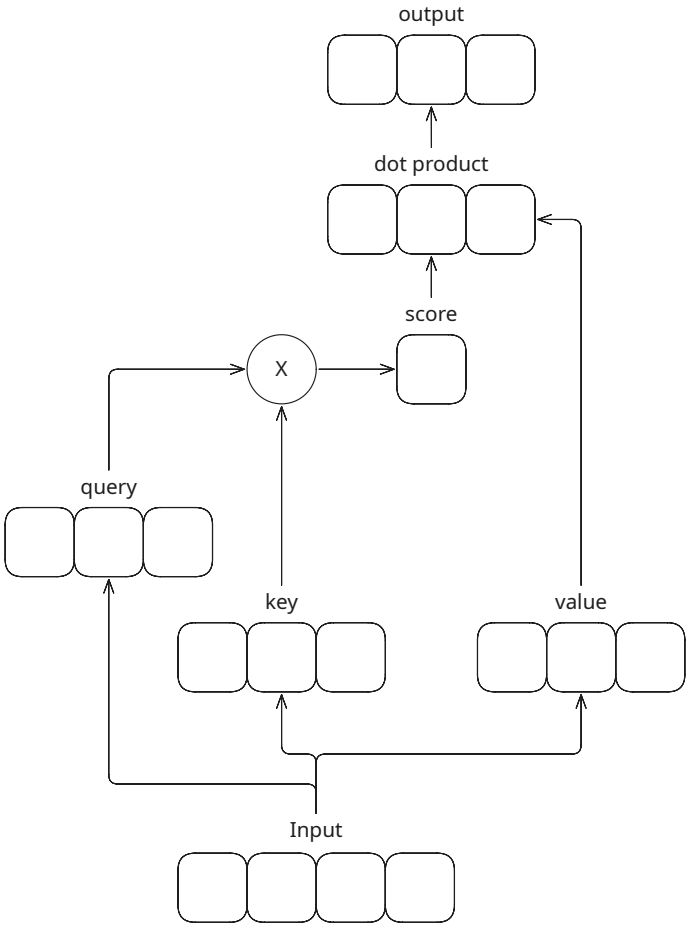
Figure 4
The transformer architecture. Adapted from Vaswani et al.



Transformers are widely used in various tasks, including natural language processing (NLP), computer vision, and graph neural networks (Vaswani et al., 2017). The main components of the transformer are the encoder and the decoder. The encoder layer has a multi-head self-attention layer, a feed forward network, residual connections and layer normalization. The decoder has the same components along with masked self-attention and an encoder-decoder connection. At a high level, the encoder converts input into a contextual representation, and the decoder takes that representation

and generates an output sequence.

Figure 5
The attention mechanism.



The self-attention mechanism in Figure 5 computes the scaled dot product attention. For every input token, the model learns three vectors: Query (Q) - the information the token is looking for, Key (K) - the kind of information each token has, and Value (V) - the actual information using Equation 2.1.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.1}$$

Here, d_k is the dimension of the key vectors, and the Softmax function

ensures that the attention scores sum to 1 Using multi-head attention, each head learns different representations by projecting the input into different subspaces. The outputs of all the heads are concatenated and linearly transformed to produce the final output. Each layer in the transformer has a feed-forward network applied to each position separately and identically. It consists of two linear transformations with a ReLU activation in between. Since transformers process the input at once, sequential information is lost during this calculating process. Positional embeddings are created using sinusoidal functions to provide a continuous representation of the positions of the input using sine and cosine functions. Once the positional embeddings are generated, they are added back to the input embedding, which is then fed into the first layer of the transformer.

Transformers are typically trained using a variant of stochastic gradient descent, such as Adam or AdamW. Cross-entropy is often used as the loss, and binary cross-entropy will be the loss used for our classification task.

Transformers have revolutionized NLP by enabling parallelization during training and capturing long-range dependencies more effectively than previous architectures like Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM). They are the backbone of many advanced models, including BERT, GPT, and T5.

2.6 Graph Embeddings

The KG created from the mapping of the HCUP and the ACS data needs to be converted into an embedding for the transformer. The embeddings can represent the nodes in a graph as vectors in a continuous space, enabling the capture of complex relationships, local and global neighborhoods, and structural dependencies. Some techniques to create embeddings are

Node2vec, DeepWalk, Graph Convolutional Networks, GraphSAGE, Laplacian eigenvectors, and pre-computed singular value decomposition (SVD) vectors (Q. Wang et al., 2017). Conveniently, transformer models can also be used to create these embeddings. In graph transformers, embeddings are typically initialized using node features and positional encodings and then refined through layers of attention mechanisms. This process transforms the input graph into a dense lower-dimensional representation that preserves structural and semantic information (Min et al., 2022). StarE, a type of graph neural network (GNN) framework that uses transformers to encode the embedding, is selected as the embedding method based on its performant characteristics compared to alternative embedding techniques (Q. Wang et al., 2017).

2.6.1 Graph Neural Networks

This section details the math behind how a GNN captures information from a graph to form an embedding. In an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where $\mathscr{V}$ represents a set of nodes and $\mathscr{E}$ represents a set of edges, each node $v \in \mathscr{V}$ has a vector h_v and a neighborhood $\mathscr{N}(v)$. A GNN learns a representation of this graph iteratively via aggregating messages from their neighbors as denoted in Equation 2.2.

Let the node update at iteration $k+1$ be defined as:

$$h_v^{k+1} = \text{UPD}\left(h_v^k, \text{AGGR}_{u \in \mathscr{N}(v)}\, \phi\left(h_v^k, h_u^k, e_{vu}\right)\right) \tag{2.2}$$

Where:

$h_v^{(k)}$ feature vector of node v at layer k;

$h_u^{(k)}$ feature vector of node u at layer k;

$\mathcal{N}(v)$ neighborhood of v;

e_{vu} edge feature from node u to v;

ϕ message-passing function;

AGGR aggregation function;

UPD update function.

The embedding is learned by passing the node and the neighborhood along with the edge features, which are then aggregated via the aggregate function (AGGR), and finally the update function (UPD) updates the embedding with the new information (Gilmer et al., 2017).

Different GNN architectures like Graph Convolutional Networks (GCN) and Graph Attention Network (GAT) modify the AGGR and UPD strategy based on their architecture. In GCN architectures, representations of neighbors are transformed via a weight matrix W and then combined and passed through a non-linear function $f(\cdot)$ like ReLU. a GCN layer k is represented in Equation 2.3.

$$h_v^{(k)} = f\left(\sum_{u \in \mathcal{N}(v)} W^k h_u^{(k-1)}\right) \tag{2.3}$$

On a directed multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ where $\mathcal{R}$ represents the set of relations r, and $\mathcal{E}$ denotes a set of directed edges (s, r, o) where nodes $s \in \mathcal{V}$ and $o \in \mathcal{V}$ are connected via relation r. A GCN assumes information in a directed edge flows both ways. Therefore, for each edge (s, r, o) an inverse edge (o, r^{-1}, s) is added to $\mathcal{E}$ along with any self-loops (v, r^{self}, v) for each node $v \in \mathcal{V}$. To account for the set of relations, Equation 2.4 can be extended by introducing a relation-specific weight matrix W_r in Equation 2.5.

$$h_v^{(k)} = f\left(\sum_{u \in \mathcal{N}(v)} W_r^{(k)} h_u^{(k-1)}\right) \tag{2.4}$$

However, these networks are known to be overparameterized. CompGCN instead learns specific edge-type vectors (Vashishth et al., 2020):

$$h_v^{(k)} = f\left(\sum_{(u,r) \in \mathcal{N}(v)} W_{\lambda(r)}^{(k)} \phi\left(h_u^{(k-1)}, h_r^{(k-1)}\right)\right) \tag{2.5}$$

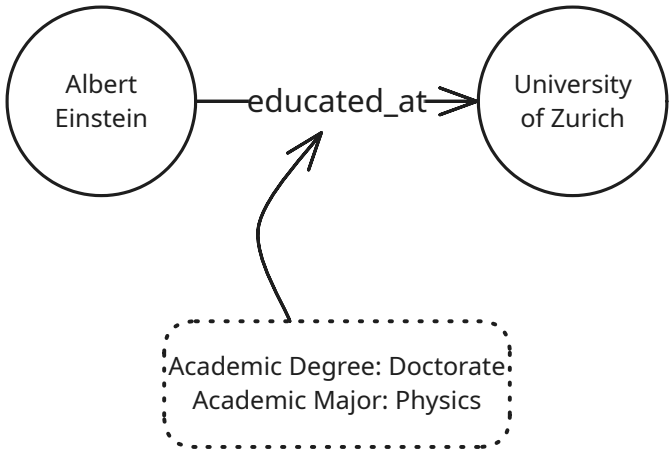
In Equation 2.5, the $\sum_{(u,r) \in \mathcal{N}(v)}$ summation iterates over the set of neighbors $\mathcal{N}(v)$ of node v. Each neighbor is represented as a pair (u,r), where u is a neighboring node and r is a relation or edge type connecting v and u. Then a weight matrix $W_{\lambda(r)}^{(k)}$ specific to the k-th layer and the relation r is updated where the function $\lambda(r)$ maps the relation r to a specific index or category, allowing for relation-specific transformations. $\phi\left(h_u^{(k-1)}, h_r^{(k-1)}\right)$ computes a feature representation based on the embeddings of the neighboring node u and the relation r from the previous layer $(k-1)$. The function ϕ could be a simple concatenation, a more complex interaction function, or another neural network layer.

Overall, Equation 2.5 describes how the embedding of a node v is updated by aggregating information from its neighbors, taking into account both the neighboring nodes and the types of relations connecting them. This process is repeated iteratively across layers to capture increasingly complex patterns in the graph.

A hyper-relational knowledge graph (HKG) is a type of KG where the edges have semantic information like educated at in Figure 6. Each edge in a hyper-relational graph not only connects two nodes with a primary relation but also includes a set of additional attributes that provide more detailed information about the relationship. This structure allows for richer

and more complex representations of relationships in the graph. This information is represented by qualifiers, also known as triples metadata, to capture and represent hyper-relational facts (Galkin et al., 2020).

Figure 6
Hyper-relational facts.

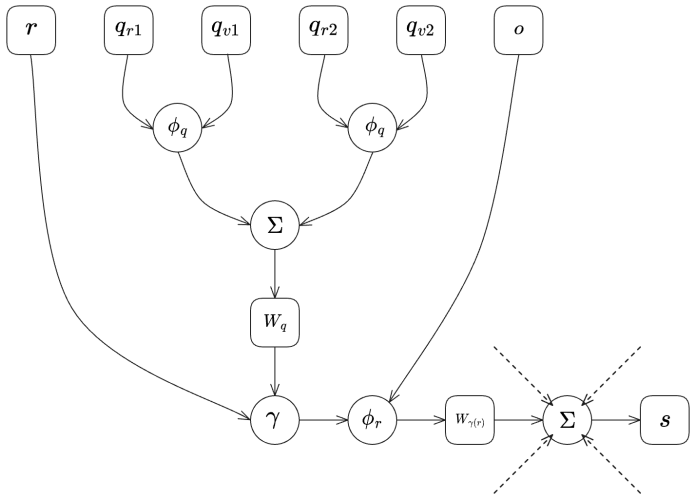


In a HKG $\mathscr{G} = (\mathscr{V}, \mathscr{R}, \mathscr{E})$, $\mathscr{E}$ is a list $(e_1, ..., e_n)$ of edges where $e_j \in \mathscr{V} \times \mathscr{R} \times \mathscr{V} \times \mathscr{P}(\mathscr{R} \times \mathscr{V})$ for $1 \leq j \leq n$, and $\mathscr{P}$ is the power set. A hyper-relational fact $e_j \in \mathscr{E}$ is written as a tuple $(\int, r, \wr, \mathscr{Q})$ where $\mathscr{Q}$ is a set of qualifier pairs $\{(qr_i, qv_i)\}$ with qualifier relations $qr_i = \mathscr{R}$ and qualifier values $qv_i = \mathscr{V}$. Here, (s, r, o) is the main triple. Q_j denotes the qualifier pairs of e_j.

2.6.2 StarE

Galkin et al. (2020) propose StarE, a graph embedding technique that can encode an arbitrary number of qualifiers along with the main triple, while keeping the semantic role of qualifiers and triples intact. This approach is superior to traditional GNN architecture, allowing for the capture of qualifier pairs along with edge pairs. The embedding is accomplished using a modified graph neural network (GNN) framework.

Figure 7
StarE hyper-relational fact encoder.



In Figure 7, Qualifier pairs q_{r1}, q_{r2} are passed through the composition function , summed, and transformed by W_q and merged with r via γ. The relation vector o is merged via ϕ_r to form the final $W_\gamma r$ matrix which is passed and summed along with other hyper-relational edges to aggregate to node .The STARE hyper-relational edge encoding mechanism incorporates the qualifiers $\{(qr_i, qv_i)\}$ with the main triple (s, r, o) into the message passing process. This extends Equation 2.5 to combine edge embedding h_r with a fixed-length vector h_q, representing the qualifiers associated with a specific relation r between nodes u and v.

$$h_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} W_{\lambda(r)}\,\phi_r\big(h_u,\ \gamma\big(h_r, h_q\big)_{vu}\big)\right) \tag{2.6}$$

In Equation 2.6, $\gamma(\cdot)$ combines the main relation representation with a representation of its respective qualifiers which are the concatenation of $[h_r, h_q]$. Then an element-wise multiplication of $h_r \odot h_{rq}$ is performed. A weighted sum as shown in Equation 7 can also be used.

$$\gamma(h_r, h_q) = \alpha \odot h_r + (1 - \alpha) \odot h_q \tag{2.7}$$

In Equation 2.7 α is a hyperparameter that controls the flow of information from qualifier vector h_q to h_r. The qualifier vector h_q here is obtained from a composition ϕ_q of a qualifier relation h_{qr} and qualifier entity h_{qv}. The composition function can be any entity-relation function similar to Equation 5. The representation of different qualifiers pairs are aggregated in a position-invariant summation and projected onto W_q:

$$\mathbf{h}_q = \mathbf{W}_q \sum_{(qr,qv) \in Q_{jrvu}} \phi_q(\mathbf{h}_{qr}, \mathbf{h}_{qv}) \tag{2.8}$$

Equation 2.8 allows the representation of the qualifier to incorporate an arbitrary number of qualifier pairs, and can account for whether entities and relations occur in the main triple or qualifier pairs. This novel method of representation presents the ideal embedding choice for DemoGraphDB.

2.7 The current state of ML in healthcare

The healthcare sector is witnessing major shifts in care management and research applications with the integration of ML technologies. These advancements enhancing diagnostic accuracy and personalizing treatment strategies using patient records to serve patient care plans. This section describes some recent deployments and use of ML in clinical and research settings.

2.7.1 Diagnostic Applications

Computer vision models are being used to interpret X-rays, CT, and MRI at scale and flag candidates that deserve a closer look, which shortens

review times for practitioners for many conditions (Yadav, 2024). In practice they work best as triage and second-reader systems, not replacements. In clinical settings, these applications should incorporate tight feedback loops with model builders to flag errors to see the most reliable gains.

2.7.2 Personalized Treatment Strategies

Patient-level modeling blends genetics, prior history, vitals, and medication data to recommend treatment options backed by research that can be referenced at scale (Yadav, 2024). Typical applications include pharmacogenomic dosing, risk-stratified care plans, and dynamic care adjustments after new labs arrive or with added clinical notes.

2.7.3 Predictive Analytics

Forecasting tools can predict readmission rates, sepsis onset, and deterioration rates among other uses. These kinds of predictive tools use longitudinal EHR along with recent health patterns to anticipate risk (Yadav, 2024). Optimized and calibrated scores like the ones produced by DemoGraphDB support decision thresholds, and can turn a prediction into a concrete next step like additional monitoring, ordering new labs, etc.

Machine learning is also being used to improve the accuracy and reliability of length of stay (LOS) predictions in hospitals. By analyzing vast amounts of patient data, ML applications can discern complex patterns and relationships that may influence LOS (AlMuhaideb et al., 2024). An inpatient stay is orders of magnitude more expensive than outpatient treatment. Being able to predict LOS can help hospitals, doctors and insurance companies plan treatment strategies and cost estimates to find opportunities to maximize treatment effectiveness and reduce the length of stay where

possible.

2.7.4 Hybrid Models

Hybrid models that combine the strengths of different algorithms to achieve high accuracy in disease classification are now being considered as a viable option due to the lowering cost of compute. Assembling multiple models to create predictive outcomes are now less onerous to run reliably in production. For instance, a hybrid model combining a Convolutional Neural Network (CNN) with a Support Vector Machine (SVM) achieved an accuracy of 89%, sensitivity of 81%, and specificity of 93% in detecting cardiovascular diseases (Ogunpola et al., 2024). These hybrid models highlight the potential of integrating diverse ML techniques to enhance diagnostic accuracy and reliability.

2.7.5 Electronic Health Records Analysis

The analysis of electronic health records (EHR) using ML is yielding valuable insights and patterns that enhance disease diagnosis, treatment prediction, personalized medicine, and overall patient management (Ogunpola et al., 2024). By extracting meaningful information from EHR structured data like hospital forms, along with free form text like doctor notes, fine-tuned models using Retrieval Augmented Generation (RAG) and models with low rank adaptation-based fine-tuning are providing reliable care management recommendations to providers.

2.7.6 Recent ML Approaches Using KG and SDOH

KGs are used in some approaches to enhance the performance of ML models in healthcare applications. For instance, integrating KGs with RAG

techniques has improved the quality of responses in healthcare settings by providing domain-specific content generation on local, verified data. In addition, chain-of-thought reasoning has been used to process information in the KG by articulating and exploring intermediate steps and rationale before arriving at conclusions when generative models provide clinical care recommendations. The use of synthetic and feature-engineered text included in KGs has been proposed to further fine-tune and instruct large language models (LLMs) with high-quality inputs and reduce hallucinations (Lee & Lindsey, 2024).

SDOH data has been leveraged in ML applications to improve healthcare outcomes, such as predicting 30-day hospital readmission rates in heart failure patients. Studies have shown that incorporating additional domain-specific SDOH variables beyond the census-tract level to the Healthcare Research and Quality (AHRQ) dataset enhances the performance of downstream ML models. Open-source large language models (LLM) have also been used to automatically annotate over 700 SDOH variables from multiple data sources, facilitating efficient integration with individual patient data (Fensore et al., 2024).

Several recent advances have focused on the incorporation of SDOH data into predictive models to improve accuracy. However, the inclusion of SDOH data can introduce bias and fairness issues, particularly when certain demographic groups are underrepresented or misrepresented in the data. Data collection methods can overlook certain sections of the population. For example, the ACS survey is mailed out to households, but this means the survey cannot be administered to the homeless population without additional steps like sending surveyors to the locations where the homeless are. It is difficult to locate transient people, and location-based

statistics would be difficult to apply to people constantly on the move. To address these challenges, methodologies have been developed to augment the data that is collected as well as to detect and mitigate bias in ML models trained on KGs (Shang, Yang, et al., 2024). Bias detection and mitigation is outside the scope of this praxis, but is a worthwhile area of further research as a refinement step to DemoGraphDB.

2.8 Conclusion

Healthcare is not the easiest domain to innovate in. Various privacy regulations hinder data sharing. Protectionist research by siloed organizations encourage knowledge and innovation hoarding. Narrow focus on solving specific business problems create solutions that are hard to apply broadly. Navigating complex IP regulations hinder broad research that span large problem spaces.

However, innovation in healthcare is a worthwhile pursuit. The potential societal benefits are broad; whether the innovation is an advancement in a clinical setting like helping intensivists make more informed decisions about which patients are in most dire need of attention in Intensive Care Units, or health insurance companies better assessing risk in the patient pool to provide tailored services and reduce cost. There is much to be gained from research and innovation in healthcare, and the benefits if shared can span multiple problem spaces.

The integration of machine learning into the healthcare sector is driving advances in various settings, from diagnostics to personalized treatment strategies. The ability of ML to analyze complex medical imaging data, tailor treatment plans based on comprehensive patient profiles, and predict disease progression and hospital LOS is transforming patient care and resource

management. The development of hybrid models highlights the potential of combining multiple ML techniques to enhance diagnostic accuracy and reliability as the cost of serving predictions decreases.

The combination of KG and SDOH data into ML models is a promising approach for improving healthcare outcomes. By representing complex relationships and integrating diverse data sources that may otherwise prove difficult to incorporate using traditional database methods, KGs offer a structured framework for information integration that has flexibility and scalability. The KGs can then be used directly for applications like question-answer knowledge bases, or transformed into embeddings that encode information from entities, relations, and neighborhood structure. The embeddings may also encode inductive bias and inject domain knowledge into a form suitable for downstream ML applications to use as data augmentation.

This praxis aims to create a framework that can enhance the predictive power of an inference in a healthcare setting using SDOH represented in a KG as a data supplement to existing data. The framework, dubbed DemoGraphDB, utilizes publicly available SDOH data and the inherent advantages of KGs, creates an embedding of the KG using StarE, and combines it with a healthcare dataset to increase predictive power of a transformer model. The interchangeable piece here is the healthcare dataset, and the target variable selected for inference. Specifically, the framework in the context of this praxis is used to predict a mortality event from patient admission data. In practice, the inferences could be used to provide additional information to intensivists in the ICU to focus attention on the patients most at risk. Generally, using this framework allows researchers and organizations to leverage publicly available SDOH data to augment their internal data when building predictive applications using patient EHR.

Chapter 3: Methodology

3.1 Introduction

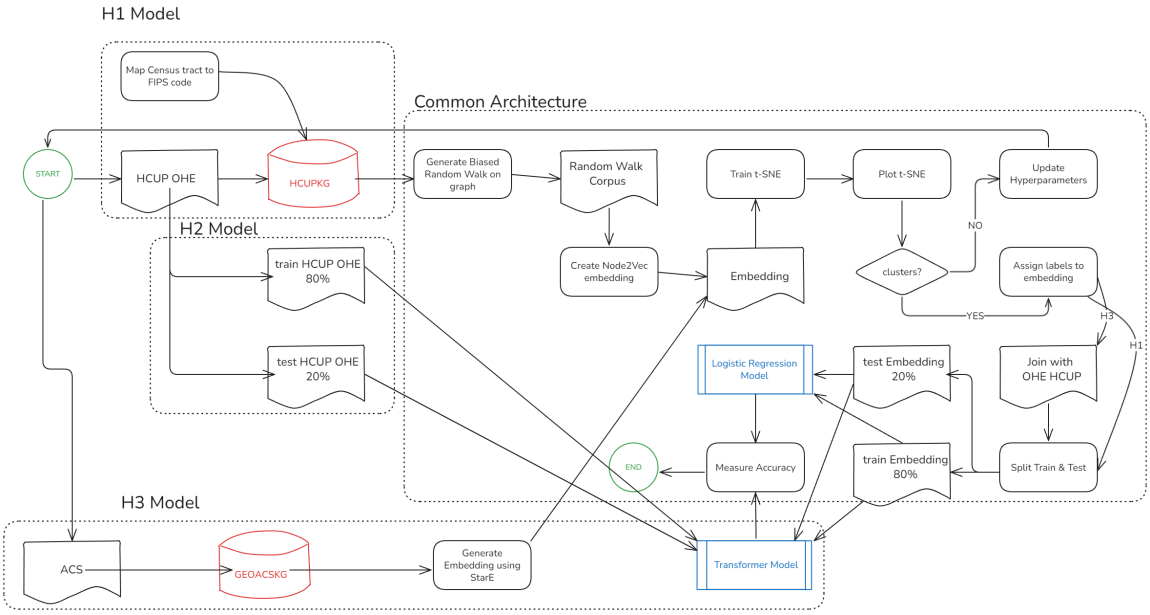
This chapter provides a description of the datasets along with the data processing steps required for each dataset in preparation for the inference tasks. To establish a baseline from the healthcare dataset, a simple logistic regression (LR) model is used. To test the 3 hypotheses, 2 knowledge graphs (KG), and 2 separate model types are created. Finally, a conclusion leads to Chapter 4, where the results are discussed in detail.

The work presented in this chapter provides the technical details to achieve two key objectives when paired with the link to the github repository provided in Appendix A. First, to introduce a general framework to output an inference for a target variable in the healthcare domain using a combination of graph theory and transformer-based neural networks. Second, to demonstrate that using Social Determinants of Health (SDOH) can enhance the predictive power of an existing dataset in healthcare predictive applications.

The framework used introduces some key benefits. First, it demonstrates that using KGs to house healthcare data is a feasible alternative to using traditional tabular data storage methods. Second, it demonstrates that using KGs to house data injects new information that would otherwise not be present if using tabular storage methods. Third, that complicated preprocessing work to feature-engineer datasets for downstream machine learning (ML) can be simplified. These benefits can be incorporated into the inference pipeline to realize immediate value from information gain and engineering resource reduction when building predictive applications in the healthcare domain for appropriate use cases.

3.2 Model Architecture

Figure 8
Architecture of H1, H2, and H3 models with shared components.



To establish a baseline accuracy metric, a Logistic Regression (LR) to predict `DIED` from hospital admission data is set up using One Hot Encoded (OHE) data derived from the Healthcare Cost and Utilization Project (HCUP) 2021 New York State Inpatient Database (SID). Then the testing of each hypothesis from H1 to H3 is enabled by the use of unique components in combination with reusable components of a common architecture as illustrated in Figure 8. The unique components of H1 are the use of the Healthcare dataset mapped to a HCUPKG geospatial graph. It then uses the common architecture components to iterate through a biased random walk algorithm to create a random walk corpus. This corpus is embedded using node2vec, verified for clustering using t-SNE, split into train and test, and trains the LR with the same configuration as baseline to get an accuracy score for H1. For H2, the HCUP OHE data is split, and trained

using a transformer model. For H3, the StarE algorithm is used to create the embedding of GEOACSKG. It is then combined with the HCUP OHE data at the FIPS code level, and run through the same transformer model in H2 to get an accuracy score.

3.3 Compute

All processing and compute are performed on a Windows 11 PC with the following specifications:

- Operating system: Microsoft Windows 11 Pro, Version 10.0.26100
- Processor: Intel(R) Core(TM) i9-14900KF 3.20 GHz
- Memory: 192 GB (192 GB usable)
- Graphical Processing Unit: NVIDIA GeForce RTX 4090

3.4 Datasets

3.4.1 Healthcare Cost and Utilization Project and the State Inpatient Database

The Healthcare Cost and Utilization Project (HCUP) is a collection of healthcare databases and software tools provided for use through a partnership between Federal, State, and Industry participants. It contains the largest collection of longitudinal hospital care data in the United States (*HCUP-US Overview*, n.d.). The State Inpatient Database (SID) contains inpatient discharge abstracts from participating states sorted by year. The SID provides the ideal real-world data of hospital visits and outcomes required to demonstrate the efficacy of the framework.

3.4.1.1 Data Summary

The specific dataset used is the 2021 New York State Inpatient data (NYSID) from the SID, which contains clinical and non-clinical discharge information on all visits to 207 hospitals in New York as defined by the Agency for Healthcare Research and Quality (AHRQ) regardless of payer. The data includes Medicare, Medicaid, private insurance, self-pay and "no charge" visits (*Introduction to the SID*, n.d.). Individual entries can have missing elements depending on the source of the original data. The data is provided in ASCII format, available through the submission of a formal request for use in a research setting.

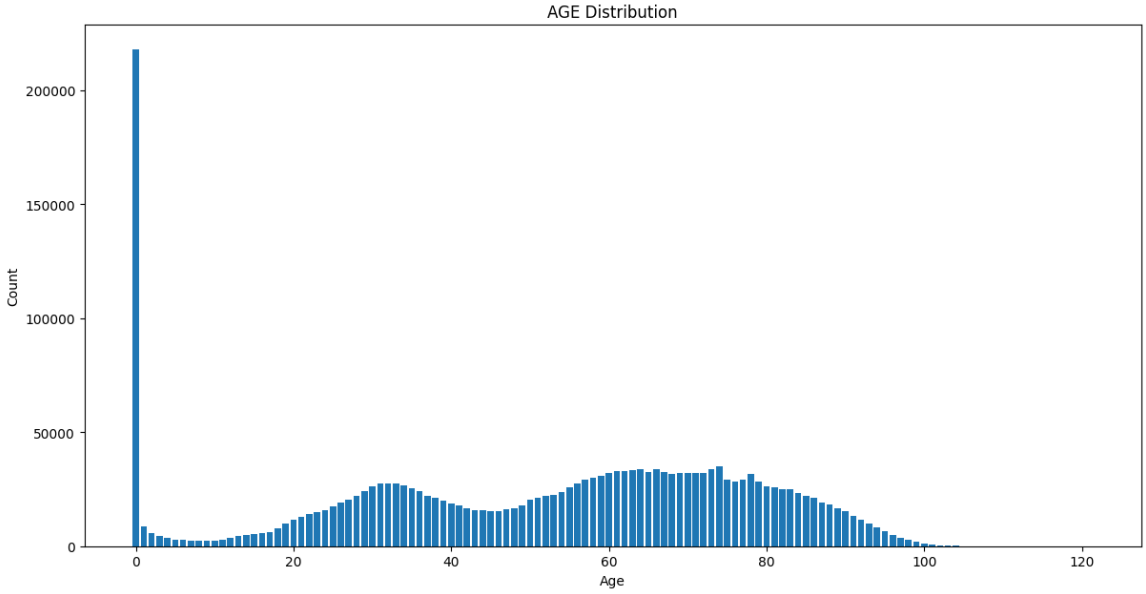
The 2021 NYSID consists of 4 files: Core, Charges, AHA Linkage, and Diagnosis and Procedure Groups. From the SID-provided files, only the Core and Diagnosis and Procedure Group files are used, since this praxis is only concerned with mortality events ("DIED").

3.4.1.2 Core File

The Core file contains one discharge-level observation per discharge. The file contains 2,031,943 unique entries and 260 columns. Column entries consist of measurements taken during intake if available, along with demographic information, time of admission, etc. Each unique entry represents a discrete hospital stay. Individual patients can have multiple entries for unique visits. The median age for entries in the file is 50.56 years.

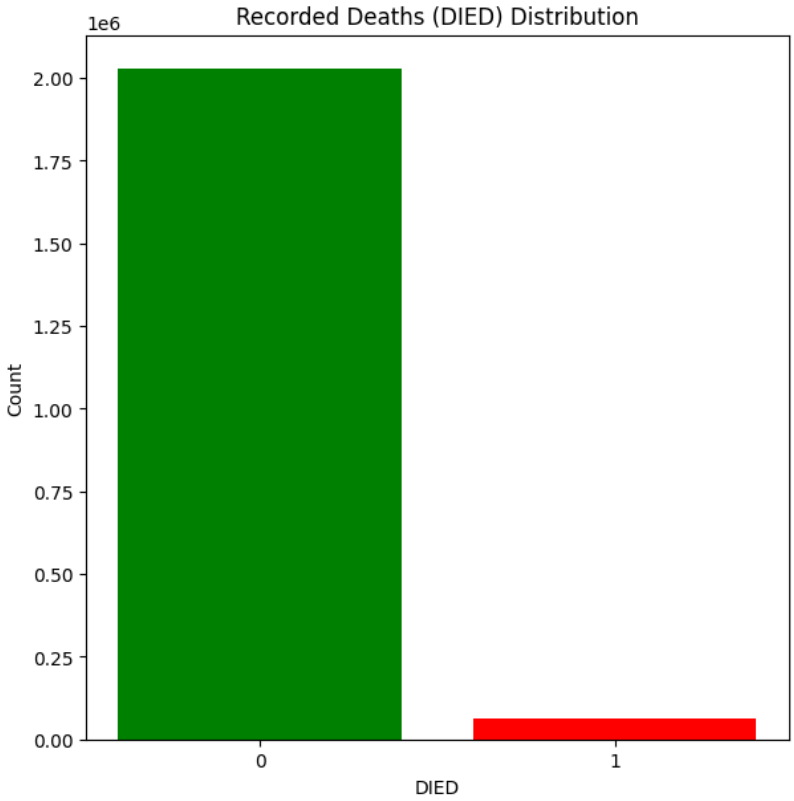
Figure 9

Age distribution of non-negative records from the 2021 CORE file.



There are 44 records with a negative age value, which indicate a data recording error and are dropped. The age distribution excluding negative values is shown in Figure 9.

Figure 10
DIED counts present in the 2021 CORE file.



The target variable chosen for inference is DIED. In the dataset, there are 62,941 mortality events and 2,028,959 non-mortality events recorded as depicted in Figure 10.

Figure 11
Average Length of Stay for patients that died vs. those who did not in the 2021 CORE file.

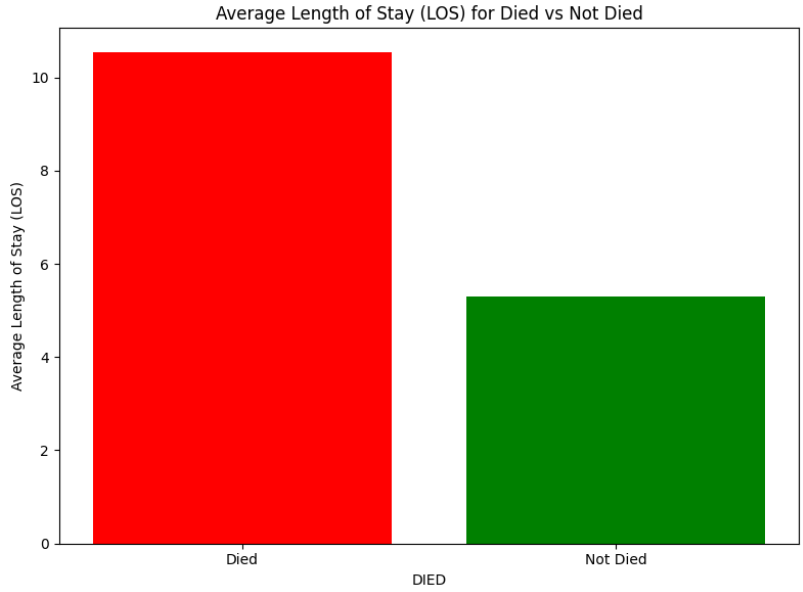


Table 3.1 *Summary statistics comparing died vs. not died.*

	DIED	NOT DIED
Length of Stay	10.53 days	5.29 days
Average Age	72.50 years	49.89 years
Median Age	75 years	55 years

The average length of stay (LOS) in Figure 11 shows the difference in LOS along with the average and median age of patients by LOS in Table 3.1.

3.4.1.3 Diagnostic and Procedure Group file

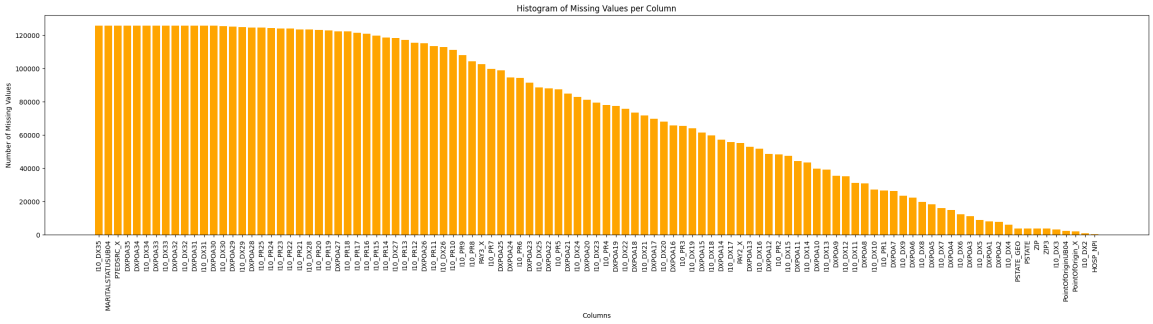
The Diagnostic and Procedure Group (DXPR) file contains discharge-level elements, with the unit of observation being the inpatient stay record. The file uses ICD-10-CM/PCS diagnostic and procedure information from the HCUP database and is linked to the CORE file via the `KEY` column. The file contains 2,031,943 entries and 1032 columns.

3.4.1.4 Data Processing

The files are provided in fixed-width ASCII format. HCUP provides database creation and uploading scripts for SAS, SPSS, and STATA. Unfortunately, these programs are not available to use for this praxis, and an alternative ingestion method had to be created. For this framework, the STATA load program is refactored into Python to create usable dataframes for both the CORE and DXPR files.

Next, the CORE and DXPR files are merged on the `KEY` column to create the master file for downstream use. With 62,941 "`DIED`=1" labels and 2,028,959 "`DIED`=0" labels, the dataset is unbalanced. The dataset is balanced by undersampling, which reduces the instances of the majority class to match the minority class. Random sampling with shuffling is performed with a fixed seed of 42. This results in a balanced dataset of 125,884 entries with 1291 columns.

Figure 12
Count of columns with null values in joined CORE and DXPR file.



A total of 18 columns where $> 99\%$ of values are missing (Figure 12) are dropped. Eight additional columns are dropped based on analysis. Further, the framework requires a location for each record, so 3,744 records without location are dropped. In addition, 3,809 records with FIPS code -9999 are dropped.

To prevent data leakage, discharge information that would not be available at the time of admission is removed after reviewing HCUP documentation and correlation analysis. The final column count (excluding location and target `DIED`) is 25 categorical (Y/N/null) and 9 numeric columns. The following conversions are applied to the categorical variables: Y=1, N=0, null=2.

3.4.1.5 Train–Test Split

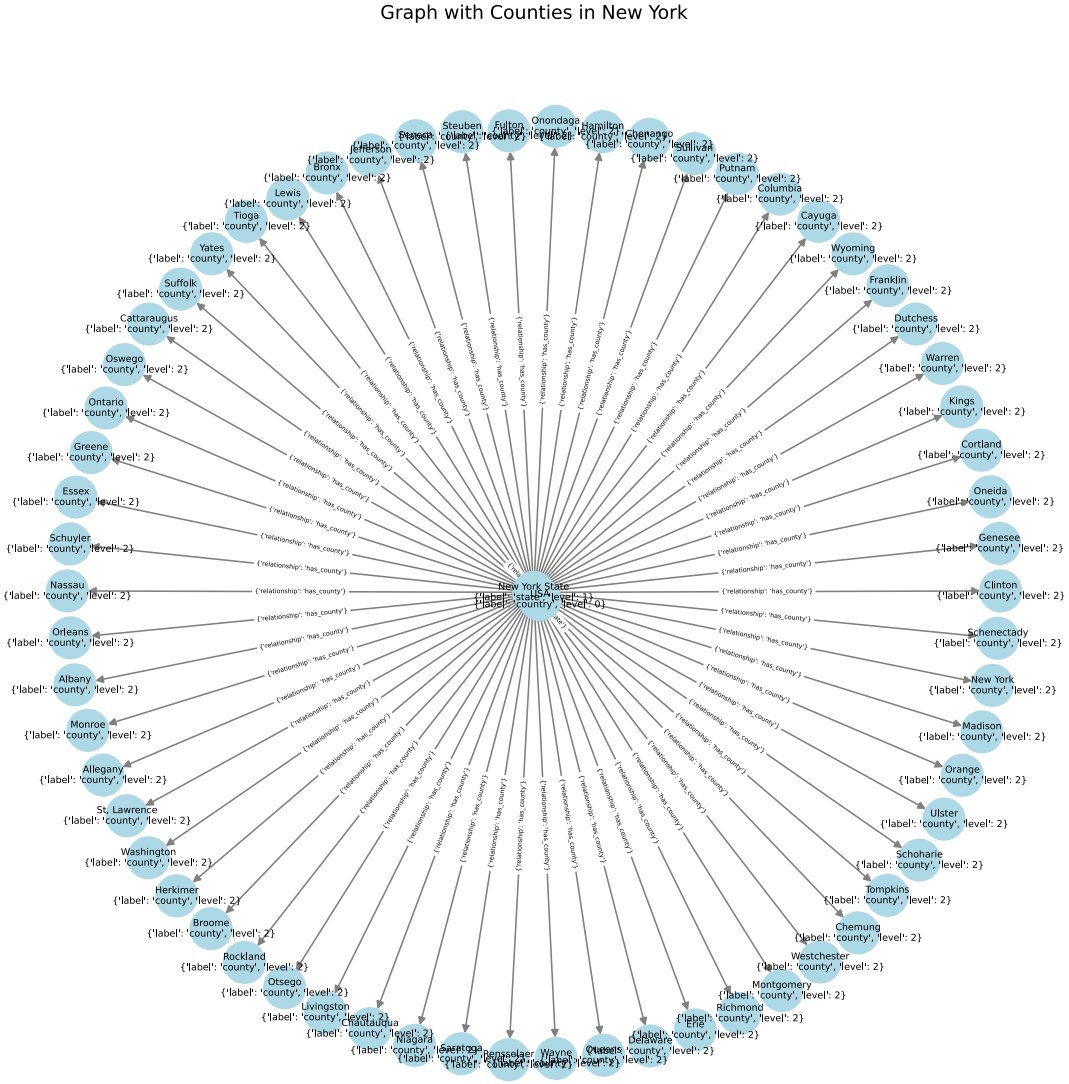
The data is split into train and test, with `X_train` having 95,137 records and `X_test` having 23,783 records. `Y_train` and `Y_test` use `DIED` as the target. One-Hot Encoding is applied to `X_train` and `X_test` resulting in 285 columns each (Table 3.2).

Table 3.2 *Data composition of train and test split.*

Term	Description	Example Values
X_train	Input features used to train the model.	Y=1 / N=0 / null=2
Y_train	Targets for `X_train`.	0/1 for `DIED`
X_test	Input features for evaluation.	Y=1 / N=0 / null=2
Y_test	True targets for `X_test`.	0/1 for `DIED`

3.4.2 Geospatial Graph

Figure 13
Base geospatial graph of New York State with counties and FIPS code connected.



SDOH are primarily location-based, and the statistics change depending on the neighborhood and geographic area being analyzed. A bare geospatial graph depicted in Figure 13 is created with root node USA, the state of New York, its counties, and the FIPS code associated with each county. This

base graph provides the FIPS code anchors needed to map the ACS data by FIPS code, and the patient records by FIPS code from the HCUP dataset. This base graph named GEOKG is built according to the pseudocode below:

Build GEOKG (Geospatial Skeleton)

- Create a directed graph with core geography:
  - Add `USA` (`label=country`, `level=0`) and `New York State` (`label=state`, `level=1`); edge `has_state`.
  - For each NY county: add county node (`label=county`, `level=2`); edge `has_county` from state.
  - For each county FIPS: add FIPS node as 5-digit string (`label=fips`, `level=3`); edge `has_fips` from county.
- Persist GEOKG.

3.4.3 American Community Survey

The 2019 ACS data as shown in Table 3.4 from the SafeGraph collection is used for this praxis. The 26 files consist of over 7,500 attributes by census block group (CBG) with an example in Table 3.3 from which FIPS codes are derived by combining State FIPS and County FIPS.

Table 3.3 *Example of Census Block Group components.*

State	County	Tract	State FIPS	County FIPS	Tract Code	Block Group
California	Los Angeles	06037264000	06	037	264000	1
New York	New York	36061000100	36	061	000100	2
Illinois	Cook	17031010100	17	031	010100	3
Texas	Harris	48201130900	48	201	130900	4
Florida	Miami-Dade	12086000100	12	086	000100	1

Table 3.4 *Sample of CBG file with the first 5 columns.*

Census Block Group	B01001e1	B01001m1	B01001e2	B01001m2	B01001e3
010010201001	730	222	293	87	10
010010201002	1263	291	614	148	24
010010202001	835	222	393	137	40
010010202002	1124	166	665	132	39
010010203001	2774	330	1423	225	20

3.4.3.1 Data Processing

The 26 ACS files are merged and processed into a single dataframe, CBG identifiers are normalized to 12 characters, geographic components split, and then filtered to New York State (`state==36`). The resulting `ny_merged_cbg_2019` is prepared for loading into a KG.

3.4.4 GEOACSKG — The Knowledge Graph of ACS data

To use the ACS data as a supplement to the HCUP data, the ACS data is mapped to a geospatial knowledge graph. This is accomplished using NetworkX. NetworkX is a Python library that creates and manipulates complex networks like KGs (Hagberg et al., 2008). NetworkX allows nodes in a network to be any hashable Python object, and edges can contain arbitrary data. This flexibility allows for creating KGs without rigid data constraints using variable sized datasets that might require extensive processing in other applications. The `ny_merged_cbg_2019` is used as the source data for the ACS. The GEOACSKG is created from this source data according to the pseudocode outline below.

1. **Load GEOKG**

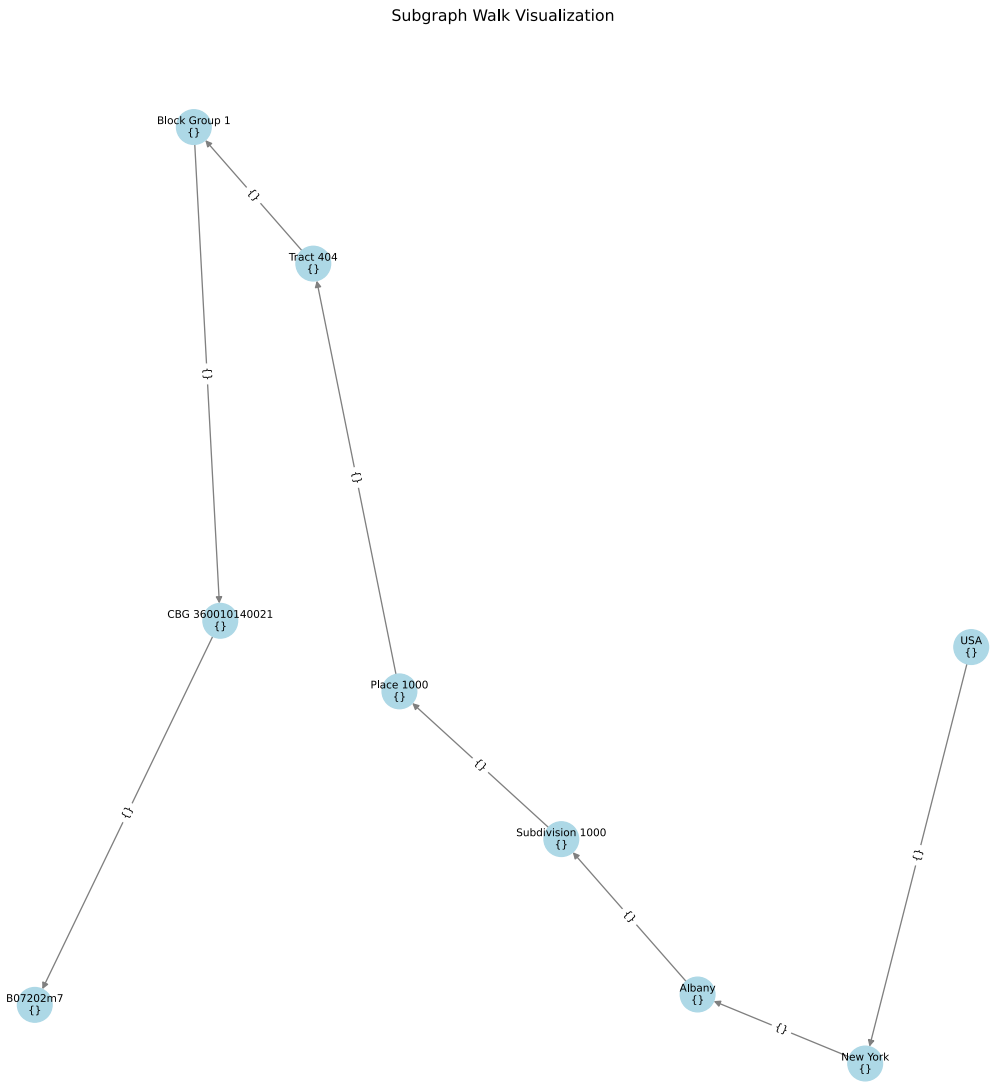
2. **Build ACSKG (ACS-Enriched CBG Graph)**

- From the merged CBG table (NY subset with `state_fips='36'`), derive `fips_code = state_fips + county_fips` and keep geographic components (`tract_code`, `block_group`).
- Add CBG nodes (label=`census_block_group`) and connect them to their county FIPS via `has_cbg` edges.
- Attach ACS measurements as structured attributes using keys such as `table_id` and `value` (e.g., nodes like `360870105031::C24020m10::13` and table nodes such as `C24020m10`).
- Persist ACSKG.

3. **Join ACSKG with GEOKG to create GEOACSKG**

- Load GEOKG and ACSKG from its pickle.
- Initialize `geoacskg = nx.DiGraph(name='geoacskg')` and add all nodes and edges from both GEOKG and ACSKG.
- Node-ID join: because county FIPS are the same 5-digit string in both graphs (labelled `fips` in GEOKG and `fips_code` in ACSKG), the merge occurs by shared node IDs.
- Persist GEOACSKG.

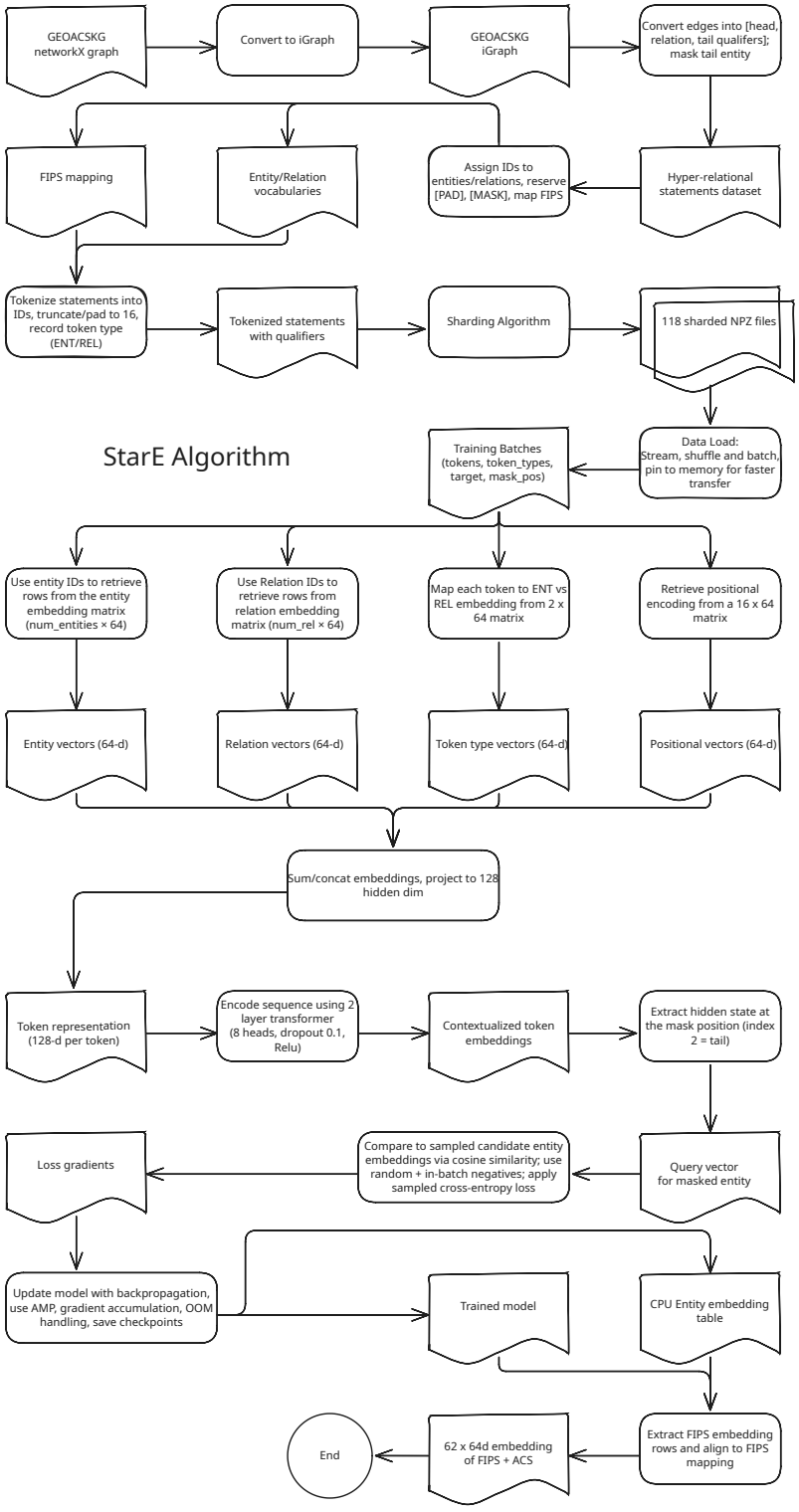
Figure 14
A sample of the ACS mapping starting with Root Node USA.



The GEOACSKG graph created incorporates all data points and descriptions available from the ACS for the state of New York into a geospatial knowledge graph (Figure 14), resulting in a graph with 98.4 million nodes and 234.6 million edges. The resultant graph is roughly 140GB when loaded into memory and creates a few challenges for data processing and requires several data partitioning steps and sequenced processing steps.

3.5 GEOACSKG StarE Embedding

Figure 15
StarE Algorithm implementation.



The final GEOACSKG graph with 98 million nodes is significantly larger than the HCUPKG used to create the embedding for H1, which has roughly 110 thousand nodes. This praxis is limited to using only local compute resources in order to comply with the privacy and data protection requirements of the HCUP data agreement which bars the use of cloud compute resources. Although a fairly powerful Windows PC is used as described in the compute resources section, the data processing and computation times have created significant bottlenecks to generating the embedding output, and have required several workarounds. The data processing steps and reasoning, along with the final output, are described in the following sections and depicted in Figure 15.

3.5.1 Dataset Preparation

Recall that each record in the HCUP data has a FIPS code per patient. The underlying assumption is that the patient's neighborhood, as represented by the statistical information from the ACS data for their FIPS code, affects their health by some factor. To validate the assumption, the embedding extracted from the GEOACSKG needs to compress the information available for each FIPS code into a latent representation. To achieve this, the information in the GEOACSKG needs to be embedded at the FIPS code level. Due to compute limitations, a 64-dim embedding is chosen, and the embedding's latent space is created to represent each FIPS code. There are 62 FIPS codes in NY state, resulting in an embedding matrix of size '62*64' with FIPS codes as the index.

To facilitate faster processing, NetworkX GEOACSKG is converted to an iGraph, which supports memory-efficient graph traversal at scale. The embedding learning is framed as a masked entity prediction over the hyper-

relational statements derived from graph edges. The statements are structured `[head, relation, tail, qual_k1, qual_v1, qual_k2, qual_v2, qual_k2, ...]` with a fixed MAX_LEN=16.

The training target tail entity is masked at position 2 with [MASK] token. The original tail index is the target. The Edge attributes and other ACS attributes are represented as key/value token pairs to provide context beyond simple triples. This hyper-relational tokenization allows the capture of rich attribute context without the need to construct large tensors. The Masked tail prediction task provides a dense self-supervised signal for the the embedding to learn entities.

The igraph of the GEOACSKG is first preprocessed to create a vocabulary of entities and relations. All unique node identifiers for entities are created in the vocabulary, and PAD=0 and MASK=1 are reserved. All unique relationship values are created for each entity as `qual_*` and added to the vocabulary. The FIPS code and their entity indices are preserved. A single global vocabulary guarantees consistent reproducibility and consistency when sharding and the special tokens enable masked prediction and padding.

To create the vocabulary, hyper-relational tokenization process to enrich the data is performed. For every edge, a base `[head, relation, tail]` sequence is created. The `[qual_k, value]` qualifier pairs are appended for each edge attribute. For connected edges, the node's immediate neighbors are traversed, and the data appended as additional qualifier pairs with a MAX_LEN=16. The created sequences are either truncated or padded to a fixed length. This ensures ACS information is captured while capping at the fixed length limits compute usage and reduces overfitting from contexts that may be too long. The final vocabulary file is $\tilde{4}$.3 GB.

From this corpus, a sharding algorithm is applied to create manageable

file sizes for downstream processing. The final output creates arrays with columns for tokens, token_types, and targets with chunk_edges of $\tilde{1}0$ million, and 118 shard files of $\tilde{2}$ million rows in Numpy zip containers (NPZ). Crucially, this sharding process allows streaming-based training without loading all statements from the vocabulary into memory, which is infeasible at $\tilde{2}00$ million scale for the full vocabulary. With the sharded files created, the StarE algorithm can be implemented.

3.5.2 StarE Algorithm on GEOACSKG

A custom dataset class is created and used to stream one shard file at a time and read rows sequentially. This yields small chunks for processing. As training progresses, rows are pulled from disk in order. A shuffle window is applied to the roughly few million rows being processed, and randomizes the samples inside the buffer. This ensures that batches are shuffled without requiring the full dataset randomization. For each batch, rows are grouped into fixed size and include the tokens, token_types, target and mask_pos. Before sending to GPU, the batches are moved to pinned (page-locked) memory to enable faster transfer across the PCIe bus into GPU VRAM. This process reduces CPU to GPU bottlenecks, which saves time at this large scale. Multiprocessing with 8 workers is launched, and each worker loads and preprocesses shard sliced in parallel. This balances I/O with GPU training throughput.

Next, entity IDs, relation IDs, the token type vectors, and positional vectors are created using individual processes that consume the input from the streaming shards. These are then summed and concatenated and projected on a 128-dimensional hidden layer. The output is a token representation of 128-d per token.

From each token, a Query (Q), Key (K), and Value (V) is generated using the learned linear projection. The scaled dot product attention softmax is computed, aggregated across the 2 layers and 8 heads with dropuout=0.1 and ReLU. This creates a 128-d contextualized token embedding for each FIPS code. Recall the tail entity in position 2 is replaced by a [MASK] token earlier. This masking forces the model to predict the missing tail entity based on the surrounding context of head, relation, and qualifiers. The vector at position 2 now encodes the information from the entire statement because the QKV attention allows it to attend to the head, relation, and qualifiers. This vector is pulled out explicitly and treated as the query representation. Over the course of training, the hidden state iterates guessing what position 2 should be given this context.

The training objective's goal is to estimate the probability of the masked true tail entity given the statement context using the sampled softmax (since the full vocabulary softmax is intractable at this scale). The candidate here is the true tail plus the negatives drawn from the uniform random negatives and in-batch negatives. Scoring using cosine similarity provides a measure of the similarity between what the model predicts and the actual masked value. The sampled softmax loss over the sampled set retains discriminative power while keeping compute resources manageable. The in-batch negatives provide negatives at no additional compute cost. In summary, A candidate set is built, the normalized queries are computed and become logits, loss is calculated, and backpropagation updates the encoder as training progresses. The training environment consists of 8 shard workers, batch size of 256, gradient accumulator of 2, and the number of negatives at 1024 run over 3 epochs.

During the forward pass, PyTorch builds a graph of the embedding

lookups, the linear projections, attention (QKV), normalization, and loss calculations. Each of these operations computes the gradients of their inputs. Autograd from PyTorch records the operations during the forward pass and applies the chain rule to compute gradients in the backward pass. This tells the model how to update the weights to better aggregate context into the mask position for the next pass. The AdamW optimizer is used to update the entity embedding table parameters from the accumulated gradients in the backward pass.

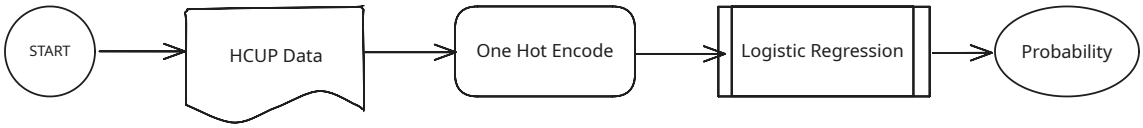
Resource constraints require the use of compute optimization techniques. Automatic Mixed Precision (AMP) is utilized resulting in 2 to 3 times faster training and $\tilde{5}$0% less GPU memory. During the forward pass, matrix multiplications are performed at Floating Point 16 for speed. During the backward pass, gradients are scaled to avoid underflow. Normalization and loss calculations are done in Floating Point 32 to ensure precision. Out of Memory (OOM) errors can occur if the GPU runs out of memory during these calculations. If this occurs, the model pipeline could fail and may need to be restarted. To avoid this, an algorithm is created where if an OOM is thrown, the batch in operation is skipped, and the gradients cleared. Then the training continues from the next batch.

The final output is a 62x64-d embedding matrix with the ID being the FIPS code that represents all of the ACS information present in the GEOAC-SKG graph. This embedding is used in conjunction with data for the H2 and H3 models.

3.6 Model Iterations

3.6.1 Logistic Regression Baseline

Figure 16
Logistic regression model architecture.



The values in the 'DIED' column of the HCUP are binary 0 or 1. Logistic Regression (LR) is chosen to establish a baseline to compare the results of the hypotheses against as shown in Figure 16. LR is a statistical method often used in binary classification problems. It uses a Sigmoid function to output values between 0 and 1 (Yedida et al., 2018). The general form of the model in Equation 9 is:

$$P(Y \mid X, W) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}} \tag{9}$$

Where

$P(Y \mid X, W)$ Predicted probability of outcome Y given features X and weights W.

w_0 Intercept (bias), a constant added to shift the decision boundary

w_i Weight associated with feature x_i learned from the data.

x_i The i-th input feature

$\sum_i w_i x_i$ The dot product of weights and features from the linear combination

e Euler's number. items[$\frac{1}{1+e^{-(\cdot)}}$] Sigmoid function, which maps any real-valued number to a value between 0 and 1

3.6.1.1 Dataset Preparation

The dataset after the cleanup is still extremely unbalanced, with only 59,460 DIED records and 1,972,483 who have not died. Not accounting for this would lead to several detrimental effects in the LR prediction task. First, majority-class dominance means the loss is driven almost entirely by negative samples, leading to the optimizer achieving low loss by predicting negatives and ignoring positives, resulting in poor generalization. Second, gradients from a few positive samples are overwhelmed by negative samples, leading to poor weight estimates for features predictive of the minor class. Last, if a model learns something about the minority, the default decision threshold of 0.5 will rarely classify positives because the predicted probabilities cluster near zero.

Balancing the date leads to equal loss contribution, where each class contributes comparably to the log loss objective, so the optimizer must fit both classes. Balancing positive samples also yields reliable coefficient estimates and better calibration of the probability. Lastly, the decision boundary becomes meaningful, boosting minority class detection.

A 'balance_dataset' function is created that computes the value_counts of the DIED column, finds the minority class count, and undersamples the majority class to match the minority class count, and returns a new balanced dataframe of 118,920 records and 36 columns. Finally, missing values for the 'PSTCO' column are dropped if any are left from the previous drops to ensure data validity.

3.6.1.2 Train–Test Split with OHE

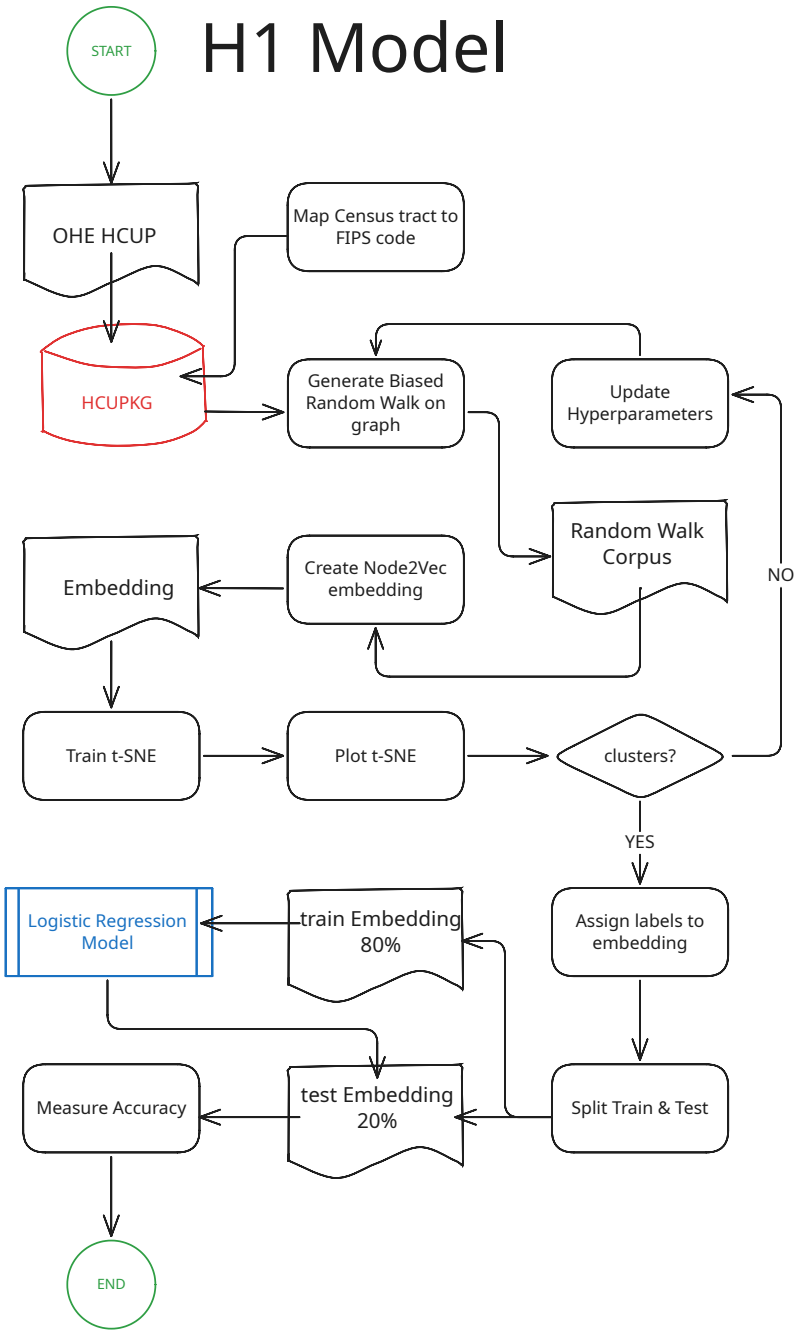
The data is then split into train and test at 80% and 20% respectively, with ‘X_train‘ having 95,137 records and ‘X_test‘ having 23,783 records. ‘Y_train‘ and ‘Y_test‘ have ‘DIED‘ as the target variable. One Hot Encoding using scikit-learn’s OneHotEncoder is fit exclusively on the training data to avoid data leakage and then applied to ‘X_train‘ and ‘X_test‘. This results in 285 columns for each of the final datasets as shown in Table 5.

3.6.1.3 Logistic Regression Settings

A logistic regression model from the scikit learn package with 1000 iterations and L2 regularization using the default c=1.0 value balances overfitting and underfitting. This allows for moderate regularization while still capturing meaningful patterns.

3.6.2 H1 Model Architecture

Figure 17
H1 Model high-level archticture.

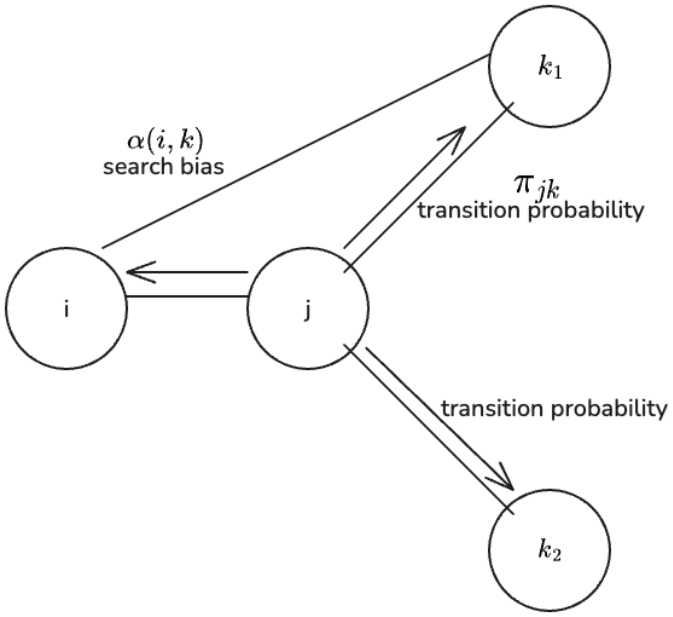


The first question of interest is if simply loading the HCUP OHE data into a KG, and then making inferences yields results that are better than chance

i.e. better than 50% accuracy. To test Hypothesis 1, a geospatial graph is manually created utilizing the County and FIPS information for New York State, and root node USA, as described in the Geospatial Graph section. Then, the HCUP OHE post-processed dataset is added to the geospatial graph by joining on the FIPS code, creating the HCUPKG, the first KG as shown in Figure 17. The embedding of the KG is created using Node2Vec.

Node2Vec is an algorithm that learns continuous feature representations for nodes in a network or graph and maps them to a lower dimensional space, maximizing the likelihood of preserving network neighborhoods using a Biased Random Walk. A Random Walk is an algorithm that simulates walking from node to connected node to create a matrix representation of the network. It starts at a random node and has a predefined length of connections through which it can walk.

Figure 18
Node2Vec search bias illustration. Adapted from Labonne, 2023.



The search bias illustrated in Figure 18 is defined in Equation 3.1 as:

$$\pi_{jk} = \alpha(i,k) \cdot \omega_{jk} \tag{3.1}$$

Where:

- j is the current node
- i is the previous node
- k is the future node
- π_{jk} is the unnormalized transition probability from j to k
- ω_{jk} is the weight of the edge from j to k
- $\alpha(i,k)$ is the search bias

For any nodes i and j in node2vec, the value of $\alpha(i,k)$ is defined based on the distance between the nodes and two additional parameters p, the return parameter, and q, the in-out parameter. The value of the search bias is defined in Equation 3.2 as:

$$\alpha(j,k) = \begin{cases} \frac{1}{p} & \text{if } d_{jk} = 0 \\ 1 & \text{if } d_{jk} = 1 \\ \frac{1}{q} & \text{if } d_{jk} = 2 \end{cases} \tag{3.2}$$

Where:

- d_{jk} is the shortest path distance between node j and node k
- p is the return parameter
- q is the in-out parameter

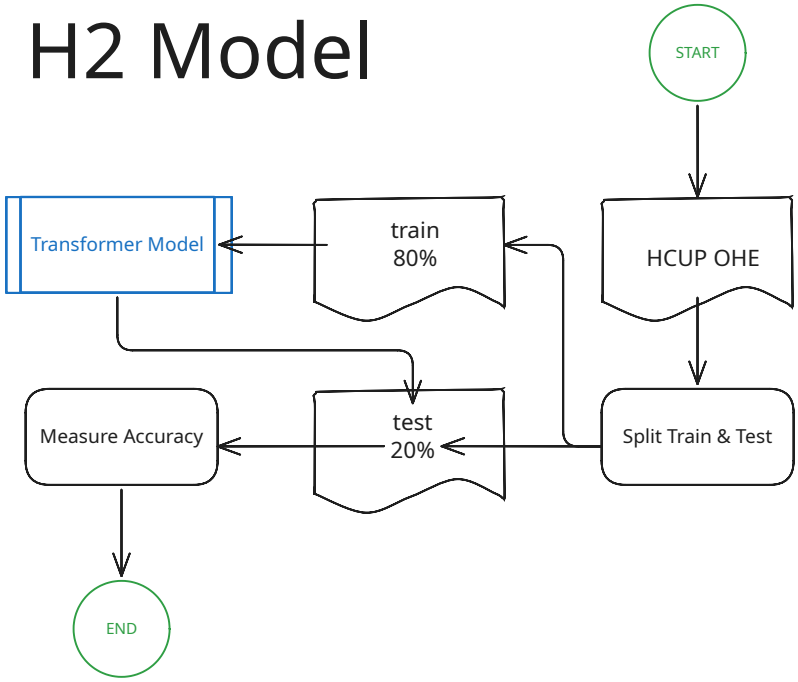
The return parameter p determines the probability that the walk will return to the previous node. The higher the probability, the more the walk will explore new nodes. The in-out parameter q controls the probability of going to k_2. Increasing this probability increases the probability that the walk will focus on nodes that are close to the previous node.

The core concept of node2vec is the assumption that the connected nature of a network is similar to the semantic connections between words in sentences and large text corpus. The output from the biased random walks is encoded similarly to how text corpus for a natural language processing (NLP) task would be encoded into an embedding. The embedding method used for node2vec is word2vec, a skip-gram based embedding method originally created for NLP (Mikolov et al., 2013). Node2Vec is a transductive learning algorithm, requiring both training and test data to create the embedding. Transductive learning focuses on making predictions for a specific set of test instances instead of learning a generalizable model that can make predictions on unseen data, and uses both train and test data to learn. This is an acceptable tradeoff since the H1 model is only used as a proof of concept to confirm predictive power is present in the embedding derived from the KG.

The output from this embedding is used to train a LR with the same settings as the baseline.

3.6.3 H2 Model Architecture

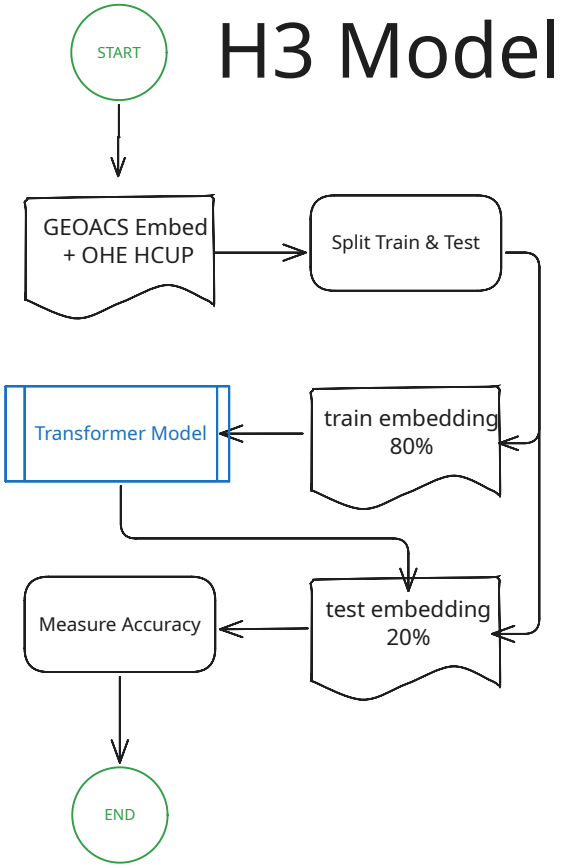
Figure 19
H2 Model high-level archticture.



For H2 shown in Figure 19, the praxis aims to understand if there is underlying predictive power in HCUP data that can be unlocked using a transformer model architecture. The same HCUP OHE data that is used to train the LR model is used to train the transformer model and run through statistical tests to detect differences in performance.

3.6.4 H3 Model Architecture

Figure 20
H3 Model high-level archticture.



To test Hypothesis 3 and arrive at the final H3 model dubbed DemoGraphDB in Figure 20, the GEOACSKG embedding and HCUP OHE dataset is joined at FIPS code, matching the FIPS code embedding row from the embedding to each patient in the HCUP OHE. The embedding at this stage has been validated for clustering using the t-SNE and the train and test split is performed, keeping the same split as the previous models for later comparability.

H3 Transformer Model Configuration

The concatenated GEOACSKG embedding and the HCUP OHE data are trained on a new H3 transformer model. The concatenated input is projected onto the model dimension of the batch size, sequence length of 16, and 256-d. 4 of the same encoding layers are stacked sequentially.

Each layer consists of 8 multi-head self-attention components that splits the 256-d vector into 32-d, and processes them in parallel. The output is then concatenated. A dropout=0.1 layer randomly sets 0.1 percent of the neuron units to zero during training and rescales the remaining units. The residual connection sublayer adds the original input back to the attention output.

The normalization sublayer then stabilizes the training. The feed-forward network applies a Multi-Layer Perceptron, which is a fully connected network, to each token, expands it to 1024-d, applies the ReLU non-linear activation function, and then projects it back to 256-d. This allows the model to transform the token embeddings in higher dimensions than simple linear combinations, allowing each token to adjust its representation after the attention step. Another dropout=0.1 sets a fraction of the output to zero.

The next sublayer again applies a residual skip connection and normalization to stabilize the training. After the 4 layer stack, another layerNorm normalizes the final hidden state across all tokens. A mean pooling of the 16 tokens averages the values across the sequence row dimension to a single vector. The classifier head converts from 256-d to 128-d to 1-d to produce the final binary classification output using binary cross-entropy with the raw logits, and applies a sigmoid to finally predict mortality as a YES/NO output per patient as seen in Table 3.5.

Table 3.5 *H3 Transformer stages and shapes.*

Stage	Subcomponent	Process	Shape
Input Projection	Linear projection	Concatenate OHE features + 64-d GeoACSKG embedding, project into model dimension	(batch, seq_len=16, 256)
Encoder Layer 1–4	Multi-Head Self-Attention (8 heads)	Q, K, V: (batch, 16, 256) $\rightarrow$ split heads (8×32-d) $\rightarrow$ attention $\rightarrow$ concat $\rightarrow$ linear projection	(batch, 16, 256)
	Dropout (0.1)	Applied after attention and after FFN	(batch, 16, 256)
	Residual + LayerNorm	Stabilizes training by adding input back and normalizing	(batch, 16, 256)
	Feed-Forward Network (FFN)	Linear: $256 \rightarrow 1024 \rightarrow 256$ with ReLU in between	(batch, 16, 256)
	Dropout (0.1)		
	Residual + LayerNorm	Preserves information and stabilizes gradients	(batch, 16, 256)
After Encoder Stack	LayerNorm	Normalizes hidden states across tokens	(batch, 16, 256)
Pooling	Mean pooling	Average across the sequence dimension (16 tokens) $\rightarrow$ single vector per sample	(batch, 256)
Classifier Head	Linear + ReLU + Dropout(0.1) + Linear	$256 \rightarrow 128 \rightarrow 1$	(batch, 1)
Output	BCEWithLogitsLoss	Binary classification (mortality yes/no)	(batch,)

The final Parameters used to train the model are documented in Table 3.6.

Table 3.6 *H3 Transformer: training hyperparameters.*

Parameter	Value
TOKENS	16
D_MODEL	256
NHEAD	8
NLAYERS	4
DROPOUT	0.1
ACTIVATION	ReLU
LR	5e-4
WEIGHT_DECAY	1e-4
EPOCHS	50
BATCH_TRAIN	2048
BATCH_TEST	4096
USE_UNDERSAMPLE	True
USE_AMP	True
LOSS	BCEWithLogitsLoss
OPTIMIZER	AdamW

3.7 Conclusion

In this chapter, the logical flow of the models built is documented. A diagram and explanation of the high-level model architecture provide an overview of the models. The Datasets section provides a data summary which includes sources and descriptions, and the data processing steps required to make the data usable from the State Inpatient Database provided by the Healthcare Cost and Utilization Project and the American Community Survey from the U.S Government. A detailed description of the creation of the GEOACSKG; the knowledge graph of the American Community Survey data is documented.

The StarE embedding of the GEOACSKG details the steps and compute optimization strategies to enable processing on local compute resources. In the Model Iterations section, a baseline Logistic Regression model is documented with initial results on the HCUP data. The H1 Knowledge

Graph documents the steps to create a knowledge graph of the HCUP data. The H2 Model Architecture section describes joining of the HCUP OHE with the GEOACSKG embedding and the model configuration used to test hypothesis 2. The H3 Model Architecture section describes the configuration of the downstream transformer model, and an additional section to test only the HCUP data in the same transformer model to compare the effect of ACS data enrichment.

This chapter aims to provide the steps necessary to recreate the models when used with the code in the appendix, and provide context on the choices made to execute the steps necessary to achieve the results documented in Chapter 4.

Chapter 4: Results and Analysis

4.1 Introduction

The goal of this praxis is to answer the following key questions of interest:

- **Research Question 1.** Can Knowledge Graph Theory be applied to predict mortality from inpatient admissions data?
- **Research Question 2.** Can an advanced neural network model increase predictive power on inpatient mortality detection?
- **Research Question 3.** Is there a particular architecture that combines SDOH and ML that performs best on classifying inpatient mortality?

To answer these questions, 3 hypotheses have been proposed. The methodology of how the models are implemented are described in chapter 3. In this chapter, an overview of the output of the two embeddings used are provided, and the individual models are compared. The results for each model are compared to examine their performance on the same test set using accuracy, confusion matrices, ROC curves, and precision-recall. A McNemar test is performed to compare false positive (FP) and false negative (FN) within each model to detect asymmetry. Another McNemar test is performed to compare all models with the baseline logistic regression model.

4.2 Embedding Evaluation

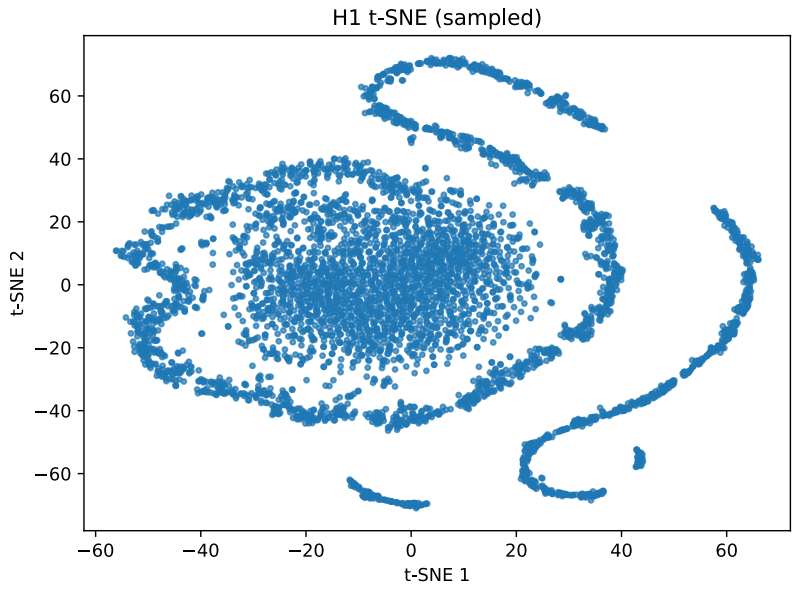
This paper focuses on the creation of Knowledge Graphs (KG) and then using their embeddings for downstream models. Two KG embeddings are

created to facilitate model implementation, the HCUP Embedding and the GEOACSKG Embedding.

4.2.1 Node2vec Embedding of HCUP data

First, a KG is created of the HCUP data, and then node2vec is used to create a 64-d embedding of all the patient KEYS, and used for downstream processing, called the HCUP Embedding. This embedding is a direct representation of each patient's record as an embedding row as captured by the KG.

Figure 21
t-SNE of HCUP Embedding.

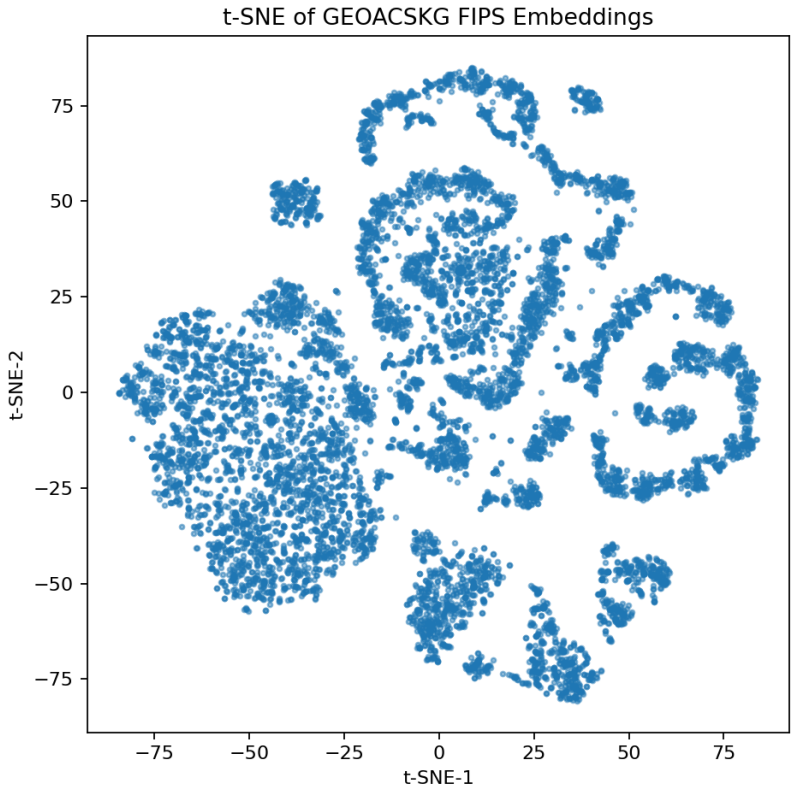


A 500 sample t-SNE of the HCUP data extracted from a knowledge graph, shown in Figure 21 shows a dense central cluster with loosely organized spiral and some peripheral bands. The central region indicates that most patients occupy a narrow latent space, suggesting limited variance and poor separability between classes. The scattered outlying arms indicate that outliers are also encoded.

4.2.2 StarE Embedding of ACS data

The Social Determinants of Health (SDOH) data is derived from the 2019 American Community Survey (ACS) and filtered on New York State to create the GEOASCKG. The KG used to represent this information is orders of magnitude larger than the HCUPKG. Therefore, using node2vec to encode this data becomes unreasonable given the local compute limitations and the much slower compute algorithm of node2vec. The StarE algorithm with sharding is used to create this 64-d embedding to capture connected information of the 62 FIPS codes for New York.

Figure 22
t-SNE of GEOACSKG Embedding.



A 500 point sample of the GEOASCKG StarE embedding in Figure 22 shows clear local structure, and multiple separated clusters compared to

the HCUPKG embedding. The distinct groups displayed indicates that the embedding captures meaningful latent relationships, likely reflecting socioeconomic and geospatial similarities encoded in the FIPS code relationships.

The spiral and island formations create clear non-linear separations, suggesting distinct boundaries between the data. Because this separation is still visible from the compression of the 64-d to the 2-d representation of the t-SNE, we can assume these separations exist in the higher dimensions of the embedding as well. The relatively even density in the clusters indicate a consistent embedding, with low noise and low sparsity.

4.3 Model Descriptions

To demonstrate the feasibility of Social Determinants of Health (SDOH) efficacy, several models are developed for this praxis and listed in Table 4.1.

1. **Logistic Regression Baseline (LR + OHE)** — a baseline logistic regression model to establish the feasibility of the state inpatient database from the Healthcare Cost and Utilization Project (HCUP) One Hot Encoded (OHE) data.

2. **HCUPKG node2vec Embedding + LR (LR + HCUP Embedding)** — The baseline LR configuration trained on a node2vec embedding of HCUP data.

3. **GEOACSKG StarE Embedding + Logistic Regression (LR + GEOACSKG)** — The Logistic Regression model trained on the HCUP OHE and GEOACSKG StarE embedding.

4. **Transformer Model (TRANS + OHE)** — A transformer model trained only on the HCUP OHE data.

5. **GEOACSKG StarE Embedding + Transformer (TRANS + GEOACSKG or DemoGraphDB)** — The TRANS model trained on HCUP OHE + GEOACSKG StarE embedding. This is the final model, dubbed **DemoGraphDB**, and the focus of this praxis.

Table 4.1 *Models and their input datasets.*

Model Name	**Data 1**	**Data 2**
LR + OHE	HCUP OHE	—
LR + HCUP Embedding	HCUP node2vec emb.	—
LR + GEOACSKG	HCUP OHE	GEOACSKG emb.
TRANS + OHE	HCUP OHE	—
TRANS + GEOACSKG (DemoGraphDB)	HCUP OHE	GEOACSKG emb.

4.4 Hypothesis Evaluation

This section evaluates each hypothesis stated in Chapter 1 based on the results obtained. For each hypothesis, a Wilson confidence interval is calculated A determination is made on whether to accept the reject the null hypothesis.

4.4.1 Hypothesis 1 Evaluation

- **Hypothesis 1.** Knowledge Graph Theory can be applied directly to healthcare data to classify mortality events with more than random accuracy.

To test Hypothesis 1, the LR + HCUP Embedding model is created to test if mapping the HCUP data to a knowledge graph, and extracting that via an embedding can yield predictive power that is greater than random i.e. more than 50% accuracy.

Table 4.2 *H1 — KG methods yield > chance accuracy on HCUP Data.*

Metric	**Value**
Model	LR + HCUP Embedding (node2vec)
Sample size n	23,783
Correct	17,622
Accuracy $\hat{p}$	0.74095
95% CI (Wilson)	[0.736, 0.746]
Hypotheses	$H_0: p \leq 0.5$ vs $H_1: p > 0.5$
Test statistic z (under $p_0 = 0.5$)	≈ 74.3
p-value	$< 10^{-300}$
Decision	**Significant**

The LR + HCUP Embedding model results documented in Table 4.2 has an accuracy of 74.10%. The Wilson confidence interval provides a statistically reliable range for the true accuracy of the model based on the large sample size of 23,783. The test confirms with 95% confidence that the accuracy of the model lies between 73.6% and 74.6%. Using $p_0 = 0.5$, the test statistic was $z \approx 74.3$, corresponding to a one-sided p-value $< 10^{-300}$. Given this extremely small p-value, there is sufficient evidence to **reject the null hypothesis at the** $\alpha = 0.05$ **level**.

4.4.2 Hypothesis 2 Evaluation

- **Hypothesis 2.** Incorporating a neural network increases classification accuracy by 60 basis points in predicting mortality from hospital admissions data.

To test Hypothesis 2, The TRANS + OHE model, a transformer neural network model using the HCUP OHE data, is compared with the performance of the LR + HCUP OHE model to determine if there is any gain in performance, and if that gain is statistically significant.

Table 4.3 *H2 — Does TRANS + OHE improve LR accuracy?*

Metric	**Value**
Comparator	TRANS + OHE vs LR + OHE
Accuracy (TRANS + OHE)	0.83085 [0.8261, 0.8356]
Accuracy (LR + OHE)	0.82370 [0.8189, 0.8285]
Difference Δ	$+0.00715$ ($\approx +0.715$ pp)
95% CI for Δ	$[0.0004, 0.0139]$
One-sided test	$H_0: \Delta \leq 0$ vs $H_1: \Delta > 0$
Test statistic z	≈ 2.06
p-value	≈ 0.0196
Decision	**Significant**

The TRANS + OHE model outperforms the baseline LR + OHE model by 0.72% as documented in Table 4.3, or 72 basis points. The confidence interval $[0.0004, 0.0139]$ for this statistically significant result does not include zero, meaning the improvement is unlikely due to random variation. A one-sided z-test ($H_0: \Delta \leq 0$ vs. $H_1: \Delta > 0$) produced a test statistic of $z \approx 2.06$ with $p \approx 0.0196$, providing sufficient evidence to **reject the null hypothesis at the** $\alpha = 0.05$ **level**.

4.4.3 Hypothesis 3 Evaluation

- **Hypothesis 3.** Incorporating Social Determinants of Health using a Knowledge Graph embedding with hospital admissions data and using transformer architecture increases mortality classification accuracy compared to a baseline logistic regression model trained on only healthcare data.

Hypothesis 3, the final hypothesis and the basis for DemoGraphDB, tests the difference in performance between the TRANS + GEOACSKG (DemoGraphDB) model and the baseline LR + OHE model.

Table 4.4 *H3 — Adding SDOH KG and using a Transformer improves accuracy over LR-only*

Metric	**Value**
Comparator	DemoGraphDB vs LR + OHE
Accuracy (DemoGraphDB)	0.83228 [0.8275, 0.8370]
Accuracy (LR + OHE)	0.82370 [0.8189, 0.8286]
Difference Δ	$+0.00858$ ($\approx +0.858$ pp)
95% CI for Δ	$[0.0018, 0.0154]$
One-sided test	$H_0 : \Delta \leq 0$ vs $H_1 : \Delta > 0$
Test statistic z	≈ 2.48
p-value	≈ 0.0066
Decision	**Significant**

The results comparing DemoGraphDB to the baseline are compared in Table 4.4. The accuracy of DemoGraphDB is 83.23%, which is ≈ 86 basis points above the LR + OHE model. With a $p \approx 0.0066$, the results are statistically significant. The confidence interval here does not include zero, meaning the true improvement is very likely positive. A one-sided z-test ($H_0 : \Delta \leq 0$ vs. $H_1 : \Delta > 0$) produced a test statistic of $z \approx 2.48$ with $p \approx 0.0066$, providing sufficient evidence to **reject the null hypothesis at the** $\alpha = 0.05$ **level**.

4.5 All Model Performance Summary

The TRANS + GEOACSKG (DemoGraphDB) outperforms all other models with the highest accuracy at 83.23%, the ROC area under the curve coverage at 90.81%, and the precision-recall at 89.05% versus all models tested on a sample size of 23,783 as seen in Table 4.5. A visual representation of the ROC is displayed in Figure 23. A visual representation of the precision-recall is displayed in Figure 24.

Table 4.5 *Model performance summary.*

Model	Acc	ROC AUC	PR AUC	N
LR + OHE	0.823698	0.901544	0.885458	23,783
LR + HCUP Embedding	0.740949	0.817950	0.790559	23,783
LR + GEOACSKG	0.823950	0.901567	0.885494	23,783
TRANS + OHE	0.830846	0.901899	0.881672	23,783
DemoGraphDB	**0.832275**	**0.908085**	**0.890481**	23,783

Figure 23
ROC Curve — All Models.

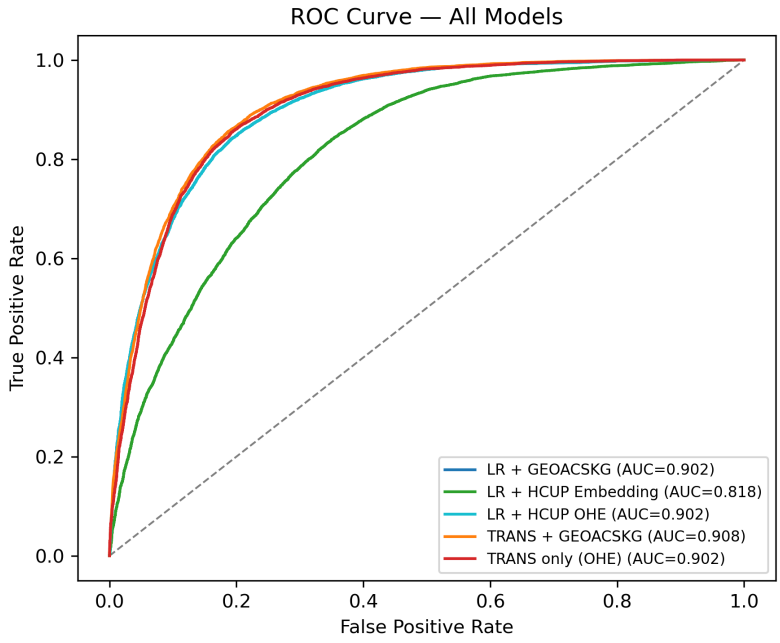
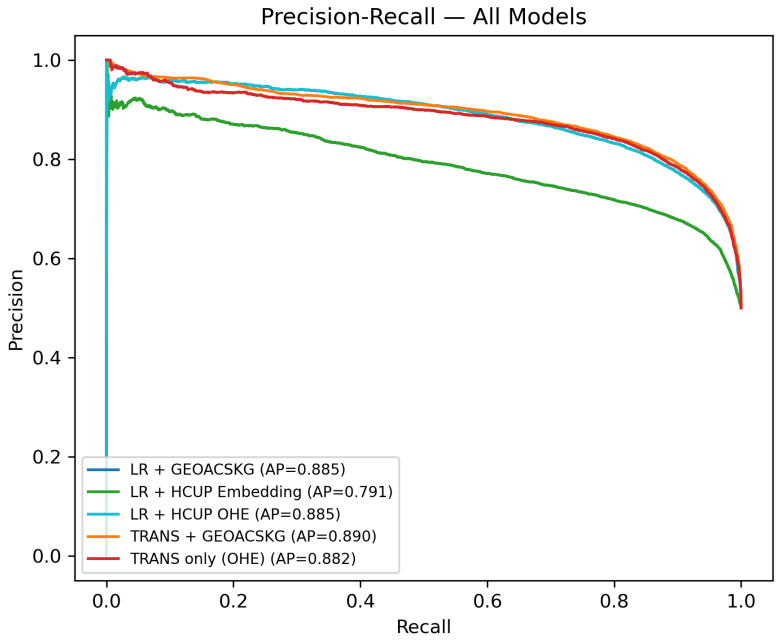


Figure 24
Precision-Recall — All Models.



4.5.1 Confusion Matrix of All Models

The confusion matrix is a useful tool to measure binary outcome model performance tied to the actual model output. In table 4.6 we see the raw classification output from the same samples provided to all models, as well as the numbers expressed as percentages in table 4.7. DemoGraphDB achieves the highest true negative (TN) at 83.22% and lowest false positive (FP) at 16.78% among the models. The TRANS + OHE model achieves the lowest false negative (FN) at 14.64% and highest true positive (TP) at 85.36%.

Table 4.6 *Confusion Matrix Summary by Model.*

Model	TN	FP	FN	TP
LR + OHE	9,496	2,396	1,797	10,094
LR + HCUP Embedding	8,472	3,420	2,741	9,150
LR + GEOACSKG	9,499	2,393	1,794	10,097
TRANS + OHE	9,610	2,282	**1,741**	**10,150**
DemoGraphDB	**9,897**	**1,995**	1,994	9,897

Table 4.7 *Row-normalized Confusion Matrix Percentages by Model.*

Model	TN%	FP%	FN%	TP%
LR + OHE	79.85	20.15	15.11	84.89
LR + HCUP Embedding	71.24	28.76	23.05	76.95
LR + GEOACSKG	79.88	20.12	15.09	84.91
TRANS + OHE	80.81	19.19	**14.64**	**85.36**
DemoGraphDB	**83.22**	**16.78**	16.77	83.23

4.5.2 McNemar's test comparing Asymmetry within models

Within the models, McNemar's test evaluates whether a model exhibits significant asymmetry between false positives (FP) and false negatives (FN), which indicates a model favoring one class over the other in its misclassification pattern. The null hypothesis (H_0) assumes that FP and FN occur with equal probability, implying a balanced error profile (Leon, 1998). Table 4.8 shows DemoGraphDb having an almost perfect balance between FN and FP misclassification rate. Therefore, there is no significant asymmetry, and thus, is the only model displaying class-balanced performance. Even though the TRANS + OHE model acheives higher FN and TP rates, the McNemar's test provides evidence of misclassification bias in this model.

Table 4.8 *McNemar's test for FP–FN asymmetry within each model.*

Model	b (FP)	c (FN)	b+c	χ^2_{cc}	p-value	($\alpha = 0.05$)
LR + OHE	2,396	1,797	4,193	85.9	$\approx 3.2 \times 10^{-19}$	Sig
LR + HCUP Emb	3,420	2,741	6,161	74.6	$\approx 6.0 \times 10^{-17}$	Sig
LR + GEOACSKG	2,393	1,794	4,187	85.4	$\approx 2.8 \times 10^{-19}$	Sig
TRANS + OHE	2,282	1,741	4,023	72.6	$\approx 2.0 \times 10^{-16}$	Sig
DemoGraphDB	**1,995**	**1,994**	**3,989**	**0.0**	**1.000**	**Not sig**

4.5.3 McNemar's test against baseline LR + OHE

Table 4.9 presents each model's relative performance against the baseline LR + OHE model. Improvements or declines are expressed as percentage point (*pp*) changes for accuracy, ROC AUC, and PR AUC, alongside the difference in total misclassification count (ΔErrors).

Table 4.9 *Pairwise Comparison of Models vs. Baseline (LR + OHE).*

Model	ΔAcc (pp)	ΔROC AUC (pp)	ΔPR AUC (pp)	ΔErrors (count)	ΔError Rate (pp)
LR + GEO	+0.0252	+0.0023	+0.0036	-6	-0.0252
LR + HCUP Emb	-8.2748	-8.3595	-9.4898	+1968	+8.2748
TRANS + OHE	+0.7148	+0.0354	-0.3786	-170	-0.7148
DemoGraphDB	**+0.8578**	**+0.6540**	**+0.5023**	**-204**	**-0.8578**

The **LR + GEOACSKG** model shows a negligible but positive gain across all metrics, reducing total errors by 6 and modestly increasing discrimination performance with ROC AUC $+0.0023$. This indicates that augmenting logistic regression with SDOH via the GEOACSKG embedding yields small but consistent gains over the baseline. However, this gain is performance is not statistically significant. The confidence interval includes zero for this model, indicating that the LR model may not realize any performance gain from augmenting the data with SDOH.

The **LR + HCUP Embedding** model performs substantially worse, with

an accuracy drop of -8.27% and nearly 2,000 additional errors. Node2vec, the algorithm used to embed this data is one of the first GNN embedding algorithms created. Since its creation, several newer embedding techniques have been developed that more efficiently capture the KG structure and semantic relationships. The algorithm is also very resource intensive when computing the embedding. Therefore, it could not be used to create an embedding of the GEOACSKG on the hardware used. The results indicate that the choice of algorithm plays an important role in embedding information from a KG, and embedding algorithms must be tested and optimized for the downstream task while also considering the embedding efficiency and time to completion.

The **TRANS + OHE** model provides a moderate improvement over the baseline, achieving a $+0.71$ pp increase in accuracy and 170 fewer errors, showing that the transformer architecture alone provides some performance advantage, even without the use of embeddings.

DemoGraphDB has the strongest overall performance, improving accuracy by $+0.86$ pp and reducing errors by 204. It also has the largest ROC AUC gain at ($+0.65$ pp). The ROC AUC gain and the non-significance of the McNemar test indicate this model performs the best in balancing class discrimination among the models. The consistent improvements across almost all metrics suggest that incorporating SDOH via GEOACSKG StarE embeddings into the transformer pipeline results in both higher predictive power and greater robustness compared to the other models created.

4.6 Conclusion

This chapter provides an analysis of the two embeddings created, the first being a KG representation of the HCUP data using node2vec, and the second

being a KG representation of the ACS data using StarE. It describes the latent information captured in these two embeddings from interpretations of the t-SNE plots.

Next, the 3 hypothesis are evaluated, and each null hypothesis is rejected with statistically significant results, indicating performance gains from each subsequent model iteration.

A total of 5 models are created to effectively compare performance. The 5 models created for predicting mortality events are described with their outcomes. McNemar's test results are reviewed to determine any class bias within the models, and to test gains or reductions in performances against the baseline model of LR + HCUP OHE.

The final results show that DemoGraphDB, the transformer model that augments SDOH data by appending a StarE embedding, has the best overall performance.

The next chapter provides an overview of the study, a summary of key findings, a discussion and answering of the research questions, contributions to the body of knowledge, implications for practice and policy, limitations of the study, recommendations for future research, and finally the closing remarks.

Chapter 5: Conclusion

5.1 Overview of the Study

This chapter summarizes the research, revisits the original objectives, hypotheses, and motivation outlined in Chapter 1. The praxis proposes a novel framework, **DemoGraphDB**, which integrates Social Determinants of Health (SDOH) within a Knowledge Graph (KG) architecture, creates a location-focused embedding, and joins the embedding with healthcare data location, and leverages a transformer neural network model for mortality prediction. The praxis results show that using a KG embedding derived from SDOH to augment healthcare data can increase a transformer model's performance for healthcare predictive applications.

5.2 Summary of Key Findings

The analysis presented in Chapter 4 demonstrates that incorporating SDOH into a KG embedding and joining it to healthcare data improves mortality prediction compared to models using healthcare data alone for certain models. A logistic regression baseline on healthcare data establishes a proof of concept, while transformer models yield the highest gains in accuracy, ROC AUC and PR AUC. These findings validate first that transformers are able to extract latent information that would otherwise be missed using simpler models, and that the enrichment of healthcare data by using SDOH from a KG embedding can improve downstream classification tasks in clinical contexts. The results empirically support the integration of neural network models and socioeconomic context for stronger predictive model-

ing in healthcare, and most importantly, demonstrate an easily executable framework to utilize SDOH as a data supplement.

5.3 Discussion of Research Questions and Hypotheses

The study addressed three primary research questions:

RQ1: Can Knowledge Graph Theory be applied to predict mortality from inpatient admissions data?

The results confirm that KG architectures can effectively model healthcare data for classification, achieving above-random predictive accuracy, as demonstrated by a score of 71.28% achieved using a simple node2vec embedding of the Healthcare Cost and Utilization Project (HCUP) 2021 State inpatient data for New York from a KG alone to predict mortality events.

RQ2: Can an advanced neural network model increase predictive power on inpatient mortality detection?

Using a transformer model in place of the logistic regression used as a baseline, training on the same data produces an improvement in accuracy of 72 basis points. Though this is a modest gain, the implication of improving accuracy simply by switching models while keeping the data constant demonstrates the benefits of informed model selection in the healthcare field, and the need for knowledgeable researchers when building these applications.

RQ3: Is there a particular architecture that combines SDOH and ML that performs best classifying inpatient mortality?

The statistically significant gains in accuracy from DemoGraphDB support this research question. Enriched embeddings from SDOH using advanced Graph Neural Network algorithms, along with a downstream transformer model, increased mortality prediction accuracy by approximately 86

basis points over baseline logistic regression. In addition, this architecture provides the best class balance. These gains can translate into real and meaningful impact. If DemoGraphDB is used to classify 100,000 patients at admission, it would make 857 more correct predictions versus the baseline. Those 857 patients can then be identified to receive appropriate medical care based on their current health status. For those 857 people, that could mean the difference between life and death.

5.4 Contributions to Knowledge

This research contributes to the field in several ways:

1. Development of the **DemoGraphDB** framework combining SDOH representation in a KG, advanced KG embedding techniques, and the combining of that KG embedding via location to healthcare data as input for a transformer model to achieve gains in predictive power.

2. Empirical validation of KG-enhanced ML models for healthcare outcome prediction.

3. A reproducible architectural pipeline for integrating public and clinical data sources using semantic representation.

5.5 Implications for Practice and Policy

The integration of SDOH with clinical data offers increases in model predictive power for healthcare practitioners, policymakers, and data scientists. The DemoGraphDB framework supports proactive risk stratification, allowing intensivists and clinicians to prioritize high-risk patients with increased accuracy. For policymakers, the framework underscores the importance

of the role of socioeconomic context and the integration of this data in designing healthcare predictive models as a standard recommendation.

SDOH data is publicly available, and this study demonstrates the efficacy of using it to augment model training during build. The inherent nature of SDOH is location-based, creating an easily translatable analogy of connectedness based on geospatial location that can be optimally represented in a KG.

From a practical perspective, organizations can create this underlying structure, and the KG can be continually updated with additional datasets from disparate sources if a location is a part of the new dataset. Because the final output of this KG is an embedding, less time can be allocated to the actual administration and curation of the KG, and can be thought of as a continuously updating asset to be appended and used for almost any patient-related predictive task.

5.6 Limitations of the Study

The study is limited by several factors. It uses data from a single state (New York) for the year 2021, restricting temporal and geographic generalization. Both the HCUP and ACS datasets may contain inherent sampling biases. Fairness analysis and explainability, though essential for clinical adoption, were not explored in this phase, and can be difficult to obtain when using transformer-based architecture. Crucially, computational limitations due to data privacy agreements constrained the exploration of larger transformer variants and extensive hyperparameter optimization, along with limiting the number of epochs the SDOH data was trained for.

Finally, research deadline objectives limited the exploration of using healthcare ontology standards like the Systematized Nomenclature of Medicine

Clinical Terms (SNOMED CT) or the Fast Healthcare Interoperability Resources (FHIR) to create a semantically richer KG before embedding. Deadline objectives also limited the exploration of using link prediction and node imputation techniques which may have contributed to further gains during embedding training.

5.7 Recommendations for Future Research

Future studies should consider expanding DemoGraphDB to other states or longitudinal datasets to assess temporal robustness. Integration of FHIR or SNOMED CT when creating the KG should be explored. Link prediction and node imputation techniques are a promising path to enriching the semantic context of KGs and warrant further investigation. Increasing compute resources will most likely yield benefits from reduced compute times during the embedding and transformer training process. A more robust transformer architecture with the use of additional compute may encode additional properties from the latent space of both the SDOH embedding and the healthcare data. Additional work should address fairness and bias mitigation, evaluate alternative transformer architectures and graph neural networks on more powerful hardware, and investigate multi-modal integration (e.g., EHR text, imaging, video). Extending this framework for other predictive healthcare outcomes (e.g., readmission, complication risk, risk scoring) would further demonstrate its generalizability.

5.8 Closing Remarks

This praxis established a foundational framework for combining knowledge graphs, SDOH, and transformer architectures to improve healthcare

predictions. The integration of geospacially structured socioeconomic context within data-driven models can become standard practice when building healthcare predictive applications. Simply put, patients all live somewhere, and the place they live have a measurable impact on their health outcomes. Not leveraging this information can lead to poorer predictive models in the healthcare domain.

DemoGraphDB demonstrates that leveraging relational, structural, and contextual knowledge from SDOH can meaningfully increase the predictive capabilities of machine learning systems, offering a framework to integrate SDOH across clinical and policy domains.

Bibliography

Abbott, E. E., Oh, W., Dai, Y., Feuer, C., Chan, L., Carr, B. G., & Nadkarni, G. N. (2023). Joint Modeling of Social Determinants and Clinical Factors to Define Subphenotypes in Out-of-Hospital Cardiac Arrest Survival: Cluster Analysis. *JMIR Aging*, *6*, e51844–e51844. https://doi.org/10.2196/51844

Abegaz, T. M., Ahmed, M., Ali, A. A., & Bhagavathula, A. S. (2025). Predicting Health-Related Quality of Life Using Social Determinants of Health: A Machine Learning Approach with the All of Us Cohort. *Bioengineering*, *12*(2), 166. https://doi.org/10.3390/bioengineering12020166

Abu Saleem, M. (2020). Folding on the Chaotic Graph Operations and Their Fundamental Group (K. Shah & B. Okutmuştur, Eds.) [ISBN: 9781838800079 9781838800086 Publisher: IntechOpen]. *Functional Calculus*. https://doi.org/10.5772/intechopen.88553

Adeogun, A., & Faezipour, M. (2025). Patient-Centric Paradigm: A Systems Thinking Approach to Enhance Healthcare. *Healthcare*, *13*(3), 213. https://doi.org/10.3390/healthcare13030213

Al Saidi, M. A. N., Eltayib, R. A. A., Wirayuda, A. A. B., Al Sumri, H. H., & Chan, M. F. (2025). The Impact of Social Determinants of Health, Health Resources, and Environmental Factors on Infant Mortality Rates in Three Gulf Cooperation Council (GCC) Countries. *European Journal of Investigation in Health, Psychology and Education*, *15*(3), 26. https://doi.org/10.3390/ejihpe15030026

AlMuhaideb, S., Bin Shawyah, A., Alhamid, M. F., Alabbad, A., Alabbad, M., Alsergani, H., & Alswailem, O. (2024). Beyond the Bedside: Machine

Learning-Guided Length of Stay (LOS) Prediction for Cardiac Patients in Tertiary Care. *Healthcare*, *12*(11), 1110. https://doi.org/10.3390/healthcare12111110

Alves, G. P., Gigena, N., & Kaniewski, J. (2024). Machine learning meets the CHSH scenario.

Barros, F. C., Victora, C. G., Scherpbier, R., & Gwatkin, D. (2010). Socioeconomic inequities in the health and nutrition of children in low/middle income countries. *Revista de Saúde Pública*, *44*(1), 1–16. https://doi.org/10.1590/S0034-89102010000100001

CDC. (2024, September). Health Insurance Portability and Accountability Act of 1996 (HIPAA).

Center for Security and Emerging Technology & Hwang, T. (2020). Shaping the Terrain of AI Competition. https://doi.org/10.51593/20190029

Cheng, J., Fostiropoulos, I., & Boehm, B. W. (2021). GN-Transformer: Fusing Sequence and Graph Representation for Improved Code Summarization. *ArXiv*.

Ciulla, M., Marinelli, L., Di Biase, G., Cacciatore, I., Santoleri, F., Costantini, A., Dimmito, M. P., & Di Stefano, A. (2023). Healthcare Systems across Europe and the US: The Managed Entry Agreements Experience. *Healthcare*, *11*(3), 447. https://doi.org/10.3390/healthcare11030447

Commissioner, O. o. t. (2025, March). 21st Century Cures Act [Publisher: FDA].

Dash, T., Chitlangia, S., Ahuja, A., & Srinivasan, A. (2022). A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, *12*(1), 1040. https://doi.org/10.1038/s41598-021-04590-0

Expanding the Use of IMPaCT: A Community Health Worker Program | PCORI. (2019, February).

Fensore, C., Carrillo-Larco, R. M., Patel, S. A., Morris, A. A., & Ho, J. C. (2024). Large Language Models for Integrating Social Determinant of Health Data: A Case Study on Heart Failure 30-Day Readmission Prediction [Publisher: arXiv Version Number: 1]. https://doi.org/10.48550/ARXIV.2407.09688

Galkin, M., Trivedi, P., Maheshwari, G., Usbeck, R., & Lehmann, J. (2020, November). Message Passing for Hyper-Relational Knowledge Graphs. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 7346–7359). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.596

Gosselink, B. H., Brandt, K., Croak, M., DeSalvo, K., Gomes, B., Ibrahim, L., Johnson, M., Matias, Y., Porat, R., Walker, K., & Manyika, J. (2024). AI in Action: Accelerating Progress Towards the Sustainable Development Goals [Publisher: arXiv Version Number: 1]. https://doi.org/10.48550/ARXIV.2407.02711

Goyal, P., & Ferrara, E. (2018). Graph Embedding Techniques, Applications, and Performance: A Survey [arXiv:1705.02801 [cs]]. *Knowledge-Based Systems*, *151*, 78–94. https://doi.org/10.1016/j.knosys.2018.03.022

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX, 11–15. https://doi.org/10.25080/TCWV9851

Haslhofer, B., Isaac, A., & Simon, R. (2018). Knowledge Graphs in the Libraries and Digital Humanities Domain (S. Sakr & A. Zomaya, Eds.) [ISBN: 9783319639628 Place: Cham Publisher: Springer International

Publishing]. *Encyclopedia of Big Data Technologies*, 1–8. https://doi.org/10.1007/978-3-319-63962-8_291-1

Jatoi, N. N., Awan, S., Abbasi, M., Marufi, M. M., Ahmed, M., Memon, S. F., Farooqui, N., Khan, M. H., Saiyid, H., Husain, A., Fatima, K., Maroof, S., & Malhotra, A. (2022). Intensivist and COVID-19 in the United States of America: A narrative review of clinical roles, current workforce, and future direction. *Pan African Medical Journal*, *41*. https://doi.org/10.11604/pamj.2022.41.210.29956

Jester, D., Kohn, J., Tibirica, L., Thomas, M., Brown, L., Murphy, J., & Jeste, D. (2022). SOCIAL DETERMINANTS OF HEALTH UNDERLIE RACIAL/ETHNIC DISPARITIES IN PSYCHOLOGICAL HEALTH AND WELL-BEING. *Innovation in Aging*, *6*(Supplement_1), 506–507. https://doi.org/10.1093/geroni/igac059.1940

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications [Conference Name: IEEE Transactions on Neural Networks and Learning Systems]. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kim, D., & Saada, A. (2013). The Social Determinants of Infant Mortality and Birth Outcomes in Western Developed Nations: A Cross-Country Systematic Review. *International Journal of Environmental Research and Public Health*, *10*(6), 2296–2335. https://doi.org/10.3390/ijerph10062296

Lau, R. S., Boesen, M. E., Richer, L., & Hill, M. D. (2024). Siloed mentality, health system suboptimization and the healthcare symphony: A

Canadian perspective. *Health Research Policy and Systems*, *22*(1), 87. https://doi.org/10.1186/s12961-024-01168-w

Lee, S. A., & Lindsey, T. (2024). Can Large Language Models abstract Medical Coded Language?

Leon, A. C. (1998, January). 3.12 - Descriptive and Inferential Statistics. In A. S. Bellack & M. Hersen (Eds.), *Comprehensive Clinical Psychology* (pp. 243–285). Pergamon. https://doi.org/10.1016/B0080-4270(73)00264-9

Liang, K., Meng, L., Liu, M., Liu, Y., Tu, W., Wang, S., Zhou, S., Liu, X., & Sun, F. (2023, July). A Survey of Knowledge Graph Reasoning on Graph Types: Static, Dynamic, and Multimodal [arXiv:2212.05767 [cs]]. https://doi.org/10.48550/arXiv.2212.05767

Major, I., King, R. F., & Marian, C. G. (2016). Anticommons, the Coase Theorem and the problem of bundling inefficiency. *International Journal of the Commons*, *10*(1). https://doi.org/10.18352/ijc.614

Mao, Y., Mokhov, S. A., & Mudur, S. (2021). Application of Knowledge Graphs to Provide Side Information for Improved Recommendation Accuracy. *ArXiv*.

Marjanovic, S., Altenhofer, M., Hocking, L., Chataway, J., & Ling, T. (2020). Innovating for improved healthcare: Sociotechnical and innovation systems perspectives and lessons from the NHS. *Science and Public Policy*, *47*(2), 283–297. https://doi.org/10.1093/scipol/scaa005

Mendelevitch, O., & Lesh, M. (2021). Fidelity and Privacy of Synthetic Medical Data. *ArXiv*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September). Efficient Estimation of Word Representations in Vector Space [arXiv:1301.3781 [cs]]. https://doi.org/10.48550/arXiv.1301.3781

Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., Zhao, P., Huang, J., Ananiadou, S., & Rong, Y. (2022). Transformer for Graphs: An Overview from Architecture Perspective. *ArXiv*.

Minsky, M. (1968). *Semantic Information Processing* [Google-Books-ID: XPTOnQAACAAJ]. MIT Press.

Mirror, Mirror 2024: A Portrait of the Failing U.S. Health System. (2024, September). https://doi.org/10.26099/ta0g-zp66

Mohamed, S., Farah, K., Lotfy, A., Rizk, K., Saeed, A., Mohamed, S., Khouriba, G., & Arafa, T. (2025). Knowledge Graphs: The Future of Data Integration and Insightful Discovery [Publisher: arXiv Version Number: 1]. https://doi.org/10.48550/ARXIV.2502.15689

Murias, G., Sales, B., Garcia-Esquirol, O., & Blanch, L. (2009). Telemedicine in Critical Care. *The Open Respiratory Medicine Journal*, *3*(1), 10–16. https://doi.org/10.2174/1874306400903010010

Murray, C. J. L., Aravkin, A. Y., Zheng, P., Abbafati, C., Abbas, K. M., Abbasi-Kangevari, M., Abd-Allah, F., Abdelalim, A., Abdollahi, M., Abdollahpour, I., Abegaz, K. H., Abolhassani, H., Aboyans, V., Abreu, L. G., Abrigo, M. R. M., Abualhasan, A., Abu-Raddad, L. J., Abushouk, A. I., Adabi, M., . . . Lim, S. S. (2020). Global burden of 87 risk factors in 204 countries and territories, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019 [Publisher: Elsevier]. *The Lancet*, *396*(10258), 1223–1249. https://doi.org/10.1016/S0140-6736(20)30752-2

Narendra, N. C., & Nagaraj, N. (2025). Integrating Causality with Neurochaos Learning: Proposed Approach and Research Agenda [Publisher: arXiv Version Number: 2]. https://doi.org/10.48550/ARXIV.2501.13763

Noseworthy, T. W., Konopad, E., Shustack, A., Johnston, R., & Grace, M. P. E. (1996). Cost accounting of adult intensive care: Methods and human and capital inputs. *Critical Care Medicine*, *24*(7), 1168.

Novilla, M. L. B., Goates, M. C., Leffler, T., Novilla, N. K. B., Wu, C.-Y., Dall, A., & Hansen, C. (2023). Integrating Social Care into Healthcare: A Review on Applying the Social Determinants of Health in Clinical Settings. *International Journal of Environmental Research and Public Health*, *20*(19), 6873. https://doi.org/10.3390/ijerph20196873

Ogunpola, A., Saeed, F., Basurra, S., Albarrak, A. M., & Qasem, S. N. (2024). Machine Learning-Based Predictive Models for Detection of Cardiovascular Diseases. *Diagnostics*, *14*(2), 144. https://doi.org/10.3390/diagnostics14020144

Payne, R., Esse, T., Qian, J., Mohan, A., Vadhariya, A., Becho-Dominguez, G., Serna, O., Villarreal, K., & Abughosh, S. (2021). Evaluating perceptions of social determinants of health and Part D star performance of Medicare Advantage-contracted primary care providers serving a South Texas market [Publisher: Academy of Managed Care Pharmacy]. *Journal of Managed Care & Specialty Pharmacy*, *27*(5), 544–553. https://doi.org/10.18553/jmcp.2021.27.5.544

Pettigrew, R. I., & Cooke, J. P. (2022). At the nexus of science, engineering, and medicine: Pasteur's quadrant reconsidered (K. E. Nelson, Ed.). *PNAS Nexus*, *1*(3), pgac092. https://doi.org/10.1093/pnasnexus/pgac092

RDF - Semantic Web Standards. (n.d.).

Richens, R. H. (1958). Interlingual Machine Translation. *The Computer Journal*, *1*(3), 144–147. https://doi.org/10.1093/comjnl/1.3.144

Rodriguez, E. B., Diaz, L. E., & Hernandez, S. T. (2025). Phase Diagrams Construction Using Mean-Field Renormalization and Neural Network Fitting.

Rudolph, M., Kurz, S., & Rakitsch, B. (2023, December). Hybrid Modeling Design Patterns [arXiv:2401.00033 [cs]]. https://doi.org/10.48550/arXiv.2401.00033

Salehnia, N., Assari Arani, A., Olyaeemanesh, A., & Sadeghi Saghdel, H. (2022). Analysis of Casual Relationships between Social Determinants of Health in Iran: Using Fuzzy Cognitive Map. *Journal of Research in Health Sciences*, *22*(4), e00566. https://doi.org/10.34172/jrhs.2022.101

Segar, M. W., Hall, J. L., Jhund, P. S., Powell-Wiley, T. M., Morris, A. A., Kao, D., Fonarow, G. C., Hernandez, R., Ibrahim, N. E., Rutan, C., Navar, A. M., Stevens, L. M., & Pandey, A. (2022). Machine Learning–Based Models Incorporating Social Determinants of Health vs Traditional Models for Predicting In-Hospital Mortality in Patients With Heart Failure. *JAMA Cardiology*, *7*(8), 844–854. https://doi.org/10.1001/jamacardio.2022.1900

Shang, T., He, W., Chen, T., Ding, Y., Wu, H., Zhou, K., & Shen, L. (2024). Integrating Social Determinants of Health into Knowledge Graphs: Evaluating Prediction Bias and Fairness in Healthcare [Publisher: arXiv tex.version: 1]. https://doi.org/10.48550/ARXIV.2412.00245

Shang, T., Yang, S., He, W., Zhai, T., Li, D., Hou, B., Chen, T., Moore, J. H., Ritchie, M. D., & Shen, L. (2024). Leveraging Social Determinants of Health in Alzheimer's Research Using LLM-Augmented Literature Mining and Knowledge Graphs [Publisher: arXiv Version Number: 2]. https://doi.org/10.48550/ARXIV.2410.09080

Spinelli, M. A., Hessol, N. A., Schwarcz, S., Hsu, L., Parisi, M.-K., Pipkin, S., Scheer, S., Havlir, D., & Buchbinder, S. P. (2019). Homelessness at diagnosis is associated with death among people with HIV in a population-based study of a US city. *AIDS*, *33*(11), 1789. https://doi.org/10.1097/QAD.0000000000002287

Stabellini, N., Cullen, J., Moore, J. X., Dent, S., Sutton, A. L., Shanahan, J., Montero, A. J., & Guha, A. (2023). Social Determinants of Health Data Improve the Prediction of Cardiac Outcomes in Females with Breast Cancer. *Cancers*, *15*(18), 4630. https://doi.org/10.3390/cancers15184630

U.S. Health expenditure as GDP share 1960-2022 [Authority: Statista]. (n.d.).

U.S. Congress. (2010, March). Patient protection and affordable care act, public law 111–148 [tex.key: PL 111–148].

Vashishth, S., Sanyal, S., Nitin, V., & Talukdar, P. (2020, January). Composition-based Multi-Relational Graph Convolutional Networks [arXiv:1911.03082 [cs]]. https://doi.org/10.48550/arXiv.1911.03082

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need.

Vogt, L. (2024). Semantic Units: Increasing Expressivity and Simplicity of Formal Representations of Data and Knowledge in Knowledge Graphs [Publisher: arXiv Version Number: 2]. https://doi.org/10.48550/ARXIV.2407.10720

Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A., Ramamurthy, R., Garcke, J., Bauckhage, C., & Schuecker, J. (2021). Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge

into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 1–1. https://doi.org/10.1109/TKDE.2021.3079836

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, R., Lee, J.-A., & Liu, J. (2024). Unwinding NFTs in the shadow of IP law. *American Business Law Journal*, *61*(1), 31–55. https://doi.org/10.1111/ablj.12237

Whitman, A., Lew, N. D., Chappel, A., Aysola, V., Zuckerman, R., & Sommers, B. D. (n.d.). Addressing Social Determinants of Health: Examples of Successful Evidence-Based Strategies and Current Federal Efforts.

Xu, C., Li, M., Yang, C., Jiang, X., Tang, L., Qi, Y., & Guo, J. (2024). Context Graph [Publisher: arXiv Version Number: 3]. https://doi.org/10.48550/ARXIV.2406.11160

Yadav, S. (2024). Transformative Frontiers: A Comprehensive Review of Emerging Technologies in Modern Healthcare. *Cureus*. https://doi.org/10.7759/cureus.56538

Yedida, R., Reddy, R., Vahi, R., Jana, R., AbhilashG, V., & Kulkarni, D. (2018). Employee Attrition Prediction. *ArXiv*.

Appendix A: Github Link

All code and scripts are uploaded to github. Data is not provided due to the HCUP data privacy agreement signed by the author. Please email the author for additional information on datasets if needed.

Github: https://github.com/Shon247/demographdb.git