# Machine Learning and Natural Language Processing of Clinical Notes for Sepsis Mortality Prediction

by

**Tyler Kelly**

BS in Biochemistry, University of Washington, 2019

BS in Chemistry, University of Washington, 2019

Submitted to the Graduate Faculty of

the School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2025

UNIVERSITY OF PITTSBURGH

SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

**Tyler Kelly**

It was defended on

November 17, 2025

and approved by

Thesis Advisor: Lu Tang, PhD, Associate Professor, Department of Biostatistics and Health Data Science

School of Public Health, University of Pittsburgh

Yongseok Park, PhD, Assistant Professor, Department of Biostatistics and Health Data Science

School of Public Health, University of Pittsburgh

Victor Talisa, PhD, Assistant Professor, Department of Critical Care Medicine

School of Medicine, University of Pittsburgh

**Machine Learning and Natural Language Processing of Clinical Notes for Sepsis Mortality Prediction**

Tyler Kelly, M.S.

University of Pittsburgh, 2025

This thesis seeks to improve machine learning (ML) protocols for predicting sepsis mortality by integrating natural language processing (NLP) into existing frameworks. Word2Vec, a shallow neural network technique for learning word embeddings, was applied to a MIMIC-IV-derived dataset that was curated by Gao et al. This dataset originates from the MIMIC-IV electronic health record repository at Beth Israel Deaconess Medical Center in Boston, MA. Google BigQuery was used to query data from the repository.

After matching patients by subject ID, 38 structured demographic, laboratory, vital sign, and comorbidity features were combined with radiology notes for each of the 5,208 patients. With the preprocessed dataset established, an ML pipeline was built and optimized under best software engineering practices. Because sepsis mortality is imbalanced, SMOTE resampling was applied to both the structured-only and multimodal (Word2Vec + structured) training datasets to evaluate its impact on sensitivity and performance.

Multimodal Word2Vec models were compared to structured-only models using AUROC as the primary metric. $\Delta$AUROC values consistently improved for multimodal models over baseline, suggesting that NLP meaningfully enhances predictive performance.

This work demonstrates that incorporating clinical notes into ML pipelines improves both predictive accuracy and discriminative power for sepsis mortality. Although SMOTE increases sensitivity at the cost of calibration, it improves detection of rare mortality events — an important tradeoff for a rapidly progressing condition such as sepsis. Leveraging the breadth of EHR data available, this study shows that NLP-based language models can strengthen prediction accuracy while highlighting the tradeoffs introduced by different resampling schemes for sepsis mortality prediction. These findings provide a foundation for future work to integrate more advanced NLP models into early-detection or risk-stratification systems designed to identify fatal health events earlier in the period following ICU admission.

## Table of Contents

**Acknowledgments** . . . . . xii

**1.0 Introduction** . . . . . 1

1.1 Clinical Background . . . . . 1

1.2 Machine Learning in Critical Care . . . . . 1

1.3 Research Gap and Motivation . . . . . 2

1.4 Objectives and Hypotheses . . . . . 4

1.5 Thesis Outline . . . . . 5

**2.0 Data Procurement and Processing** . . . . . 6

2.1 Data Source and Cohort Selection . . . . . 6

2.1.1 Database Description . . . . . 6

2.1.2 Inclusion and Exclusion Criteria . . . . . 7

2.1.3 Outcome and Structured Features . . . . . 7

2.2 Structured Data Processing . . . . . 11

2.3 Unstructured Text Data . . . . . 12

2.3.1 Note Type and Extraction . . . . . 13

2.3.2 Missing Unstructured Data Handling . . . . . 13

2.3.3 Text Cleaning Pipeline . . . . . 13

2.3.4 Tokenization and Aggregation . . . . . 15

2.4 Word2Vec Training . . . . . 15

2.4.1 Overview . . . . . 15

2.4.2 CBOW vs Skip-Gram . . . . . 16

2.4.3 Preliminary Embedding Parameters . . . . . 17

2.5 Post-Baseline Word2Vec Training . . . . . 17

2.5.1 Cosine Similarity . . . . . 18

2.5.2 Intrinsic Evaluation . . . . . 19

2.5.3 Optimal Embedding Parameters . . . . . 20

2.6 Feature Engineering and Model Training Preparation . . . . . 20
2.6.1 Data Partitioning . . . . . 20
2.6.2 Embedding Aggregation . . . . . 21
2.6.3 Normalization and Merging . . . . . 21
**3.0 Methods** . . . . . 22
3.1 Addressing Class Imbalance . . . . . 22
3.1.1 Target Variable Distributions . . . . . 22
3.1.2 Synthetic Minority Over-Sampling Technique . . . . . 23
3.1.3 Applying SMOTE . . . . . 23
3.2 Model Development . . . . . 24
3.2.1 Model Families . . . . . 24
3.2.2 Tree-Based Methods . . . . . 26
3.2.3 Boosting-Based Methods . . . . . 26
3.2.4 Hyperparameter Optimization . . . . . 27
3.2.5 Descriptive Internal Cross-Validation . . . . . 28
3.2.6 Retraining on SMOTE Balanced Training Sets . . . . . 28
3.3 Evaluation Framework . . . . . 29
3.3.1 Metrics . . . . . 29
3.3.2 Validation and Testing Design . . . . . 30
3.3.3 Metric Visualizations . . . . . 31
3.4 Explainability and Statistical Testing . . . . . 31
3.4.1 Feature Importance . . . . . 31
3.4.2 Calibration . . . . . 32
3.4.3 Clustering Embeddings . . . . . 32
3.4.4 Comparative Significance Testing . . . . . 32
**4.0 Results** . . . . . 33
4.1 Overview of Analytical Workflow . . . . . 33
4.2 Baseline Model Development and Performance . . . . . 34
4.2.1 Structured-Only Baseline Models . . . . . 34
4.2.2 Structured + Word2Vec Baseline Models . . . . . 34

4.2.3 Comparison of Structured and Multimodal Baseline Models . . . . . 35
4.2.4 Baseline Model Performance with SMOTE Resampling . . . . . 36
4.2.5 Comparison of Structured and Multimodal SMOTE Baseline Models . 37
4.2.6 Comparison of Resampling Strategies . . . . . 38
4.2.7 Top Selected Classifiers for Optimized Training . . . . . 41
4.3 Word2Vec Embedding Optimization . . . . . 41
4.3.1 Intrinsic Evaluation of Embeddings . . . . . 41
4.3.2 Selected Optimal Embedding Configuration . . . . . 43
4.4 Optimized Word2Vec Multimodal Performance and Evaluation . . . . . 43
4.4.1 Classifier Re-Tuning on Optimized Embeddings . . . . . 43
4.4.2 Optimized Multimodal Performance (Non-SMOTE) . . . . . 45
4.4.3 Optimized Multimodal Performance (SMOTE) . . . . . 46
4.4.4 Comparison of Optimized and Baseline Multimodal Models . . . . . 47
4.4.5 Comparison of Resampling Strategy for Optimized Multimodal Models 48
4.5 Visual Summaries . . . . . 49
4.5.1 AUROC Comparison Across Variants . . . . . 49
4.5.2 ROC and PR Curves . . . . . 50
4.5.3 Dependency and SHAP Analysis . . . . . 51
4.5.4 Calibration Plots and Brier Scores . . . . . 54
4.5.5 Clustering Analysis . . . . . 57
4.6 Final Evaluation and Statistical Testing . . . . . 59
4.6.1 Holdout Evaluation Design . . . . . 59
4.6.2 DeLong Tests for Non-SMOTE Models . . . . . 59
4.6.3 DeLong Tests for SMOTE Models . . . . . 60
4.6.4 DeLong Tests for Best Models . . . . . 61
4.6.5 Bootstrapped Confidence Intervals . . . . . 63
4.6.6 Multiple Comparison Adjustment . . . . . 64
4.6.7 Cross-Model Interpretation . . . . . 64
4.6.8 McNemar's Test for $\Delta F_2$ Significance Testing . . . . . 66
4.7 Results Summary . . . . . 68

**5.0 Discussion** . . . . . 69
5.1 Principal Findings . . . . . 69
5.1.1 Hypothesis 1 – Word2Vec Improves Discrimination . . . . . 69
5.1.2 Hypothesis 2 – SMOTE Enhances Sensitivity . . . . . 69
5.1.3 Hypothesis 3 – Interpretability Maintained Across Modalities . . . . . 69
5.1.4 Summary of Statistical Testing Outcomes . . . . . 70
5.2 Clinical Implications . . . . . 70
5.2.1 Interpretability and Feature Insights . . . . . 70
5.2.2 Public Health Significance . . . . . 71
5.3 Limitations . . . . . 71
5.3.1 Data Level Limitations . . . . . 71
5.3.2 Clinical Note Preprocessing, Tokenization, and Temporality . . . . . 72
5.3.3 Modeling Limitations . . . . . 74
5.3.4 Interpretability and Bias . . . . . 75
5.4 Future Work . . . . . 75
**6.0 Conclusion** . . . . . 77
**Appendix A. Codebase** . . . . . 78
**Appendix B. Additional Tables** . . . . . 79
**Appendix C. Additional Figures** . . . . . 80
**Bibliography** . . . . . 84

## List of Tables


1 Descriptive Statistics for Structured Features Included in This Study. . 8
2 Baseline Word2Vec Parameter Configuration. . . . . 17
3 Parameter Search Space for Word2Vec Embedding Optimization. . . . . 19
4 Structured-Only Baseline Model Performance (Non-SMOTE). . . . . 34
5 Structured + Word2Vec Baseline Model Performance (Non-SMOTE). . 35
6 Performance Changes (Δ) Between Structured-Only and Structured + Word2Vec Baseline Models (Non-SMOTE). . . . . 35
7 Structured-Only Baseline Model Performance (SMOTE). . . . . 36
8 Structured + Word2Vec Baseline Model Performance (SMOTE). . . . . 37
9 Performance Changes (Δ) Between Structured-Only and Structured + Word2Vec Baseline Models (SMOTE), Ordered by ΔAUROC. . . . . 37
10 Performance Changes (Δ) Between Structured-Only Baseline Models with and without SMOTE, Ordered by ΔAUROC. . . . . 38
11 Performance Changes (Δ) Between Structured + Word2Vec Baseline Models with and without SMOTE, Ordered by ΔAUROC. . . . . 38
12 Optimized Word2Vec Parameter Configuration. . . . . 43
13 Optimized Hyperparameters for Each Classifier for Optimized Word2Vec Multimodal Model Training. . . . . 44
14 Optimized Word2Vec Multimodal Model Performance (Non-SMOTE). . 45
15 Performance Changes (Δ) Between Optimized Word2Vec Multimodal and Structured-Only Models (Non-SMOTE), Ordered by ΔAUROC. . . 45
16 Optimized Word2Vec Multimodal Model Performance (SMOTE). . . . . 46
17 Performance Changes (Δ) Between Optimized Word2Vec Multimodal and Structured-Only Models (SMOTE), Ordered by ΔAUROC. . . . . 46
18 Performance Changes (Δ) Between Baseline Word2Vec and Optimized Word2Vec Multimodal Models (Non-SMOTE), Ordered by ΔAUROC. . 47

19 Performance Changes (Δ) Between Baseline Word2Vec and Optimized Word2Vec Multimodal Models (SMOTE), Ordered by ΔAUROC. . . . 47
20 Performance Changes (Δ) Between Optimized Word2Vec Multimodal Models with and without SMOTE, Ordered by ΔAUROC. . . . . 48
21 Paired DeLong Tests Comparing Optimized Multimodal Non-SMOTE Models Against Baseline. . . . . 60
22 Paired DeLong Tests Comparing Optimized Multimodal SMOTE Models Against Baseline. . . . . 61
23 DeLong Tests for Best Models. . . . . 62
24 Bootstrap Confidence Intervals for Best Models. . . . . 62
25 Bootstrapped Comparison of Raw ΔAUROC Scores (Non-SMOTE). . . 63
26 Bootstrapped Comparison of Raw ΔAUROC Scores (SMOTE). . . . . 63
27 $\Delta F_2$ Performance for Optimized Word2Vec Multimodal SMOTE vs Non-Smote Models. . . . . 66
28 Global Best-Versus-Best DeLong Tests. . . . . 79

## List of Figures


| | | |
|---|---|---|
| 1 | Class Distributions of Sepsis Mortality. | 22 |
| 2 | Baseline ROC Curves. | 39 |
| 3 | Baseline Precision–Recall Curves. | 40 |
| 4 | Word2Vec Embedding Cosine Similarity Scores. | 42 |
| 5 | AUROC Comparison of Baseline and Optimized Models (Non-SMOTE). | 49 |
| 6 | ROC and PR Curves for Word2Vec Multimodal Models. | 50 |
| 7 | SHAP Summary Plots for Optimized Word2Vec Multimodal Models. | 52 |
| 8 | SHAP Dependence Plots for Top Word2Vec Feature Contributions (Non-SMOTE). | 53 |
| 9 | Optimized Word2Vec Multimodal Calibration Plots (Non-SMOTE). | 55 |
| 10 | Optimized Word2Vec Multimodal Calibration Plots (SMOTE). | 56 |
| 11 | Clustering Analysis Using PCA of Optimized Word2Vec Embedding Space. | 58 |
| 12 | Clustering Labels for Figure 11. | 58 |
| 13 | Comparison of Raw $\Delta$AUROC Scores. | 65 |
| 14 | Comparison of $\Delta$AUROC Scores with Holm-Bonferroni Correction. | 65 |
| 15 | Comparison of $F_2$ Performance Metrics. | 67 |
| 16 | AUROC Comparison of Baseline and Optimized Models (SMOTE). | 80 |
| 17 | SHAP Summary Plots (SMOTE). | 81 |
| 18 | SHAP Dependence Plots (SMOTE). | 82 |
| 19 | SHAP Dependence Plots Feature Contributions (Non-SMOTE). | 83 |

**Acknowledgments**

I want to thank my wife for her love and devotion to me during our transition to Pittsburgh and to graduate school and to my children who constantly reminded me that family comes before all else (to the point that most of my graduate schoolwork was completed between 9pm and 2am) — my family has been a constant source of strength by reminding me of my 'why' for attending graduate school.

I also thank my thesis advisor, Dr. Lu Tang, for his consistent guidance in my work. He helped me expand my confidence in machine learning in healthcare while providing guardrails to protect me from the ocean of possibilities when it comes to model building and optimization. His advice in advanced machine learning techniques and natural language processing integration and previous research in electronic health data provided a stable basis for this work to be built off of. I am grateful for his mentorship and persistence in helping me complete this major milestone in my academic career.

# 1.0 Introduction

## 1.1 Clinical Background

Sepsis is a life-threatening disease caused by a dysregulated host response to infection [50], with overall prevalence levels of 25.8% mortality in ICU settings. [49] Diagnosis of sepsis for this study was based on the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). Infants, children, elderly adults, vulnerable individuals with medical conditions or concurrent injuries and/or surgeries, and those on medication are most susceptible to sepsis. [43] Different types of infection exposure lead to sepsis in the ICU such as community-borne, nosocomial, or hospital-acquired infection, with medical ICUs having higher infection rates than ICU addmitting elective procedures. [4] Current clinical care for sepsis patients involves extensive care treatments involving antimicrobic therapies according to infection site. [20] Lack of professional knowledge and lack of communication between disciplines and/or sectors are two main challenges of sepsis care at the early recognition and timely treatment phases. [15] As sepsis is a fast-developing disease, efforts are needed to expand the intervention window of time for sepsis, to delay further progression of the illness, and determining early indicators for disease development is paramount. Specifically, identifying biofeatures associated with increased sepsis mortality could help to gain insight into assessing the time-varying effects of these features, resulting in changes in survival time and survival probability for sepsis patients, as well as for those susceptible to sepsis.

## 1.2 Machine Learning in Critical Care

Current state of the art methods for predicting sepsis mortality and identifying features related to the survival time of sepsis are being employed [42], including boosting and tree-based methods. [34] Machine learning consists of programmatic approaches to fitting models to data and learning connections between input and output features. In clinical care settings,

this consists of fitting models to various modes of patient data, whether at the patient, cohort, hospital level, or to data in other healthcare settings. In critical care medicine, machine learning including deep learning, transfer learning, reinforcement learning, and transformer based architectures is generally used on structured features [59], but other types of work involving transformer architectures also exist. [10]

Natural language processing (NLP) consists of sets of techniques that analyze objects like text documents, visualizations, sound-recordings, and other complex objects to learn information about them and find practical insights that explain them in more detail. In the clinical care setting, natural language processing can be used to batch process and analyze electronic health record (EHR) data to gain insight into a patients health and patterns associated with care.

Many frameworks in previous literature incorporate natural language processing techniques in multimodal modeling designs that specifically incorporate EHR data and/or clinical text data for biomedical research tasks and in clinical care medicine. [57] However, few literature reviews exist in this space for applying NLP on unstructured clinical text data for sepsis mortality prediction. [57] The work that does exist incorporates GloVe [35], custom embedding designs using representation learning [58], Clinical-BERT [2], and the SERA algorithm, which is based on the latent Dirichlet Allocation (LDA) algorithm [19].

### 1.3 Research Gap and Motivation

Most of the work demonstrated in sepsis mortality prediction using natural language processing use multimodal integrated datasets in a method similar to what this study presents. Some existing methods incorporate resampling techniques such as synthetic minority oversampling technique (SMOTE), as will be described in more detail below. However, the specifics in the integration of SMOTE resampling in previous methodologies is unclear. Upsampling imbalanced classes tends to lead to inflated area under the curve (AUC) scores when applied incorrectly, even when performing cross-validation, due to information leakage into validation folds. It is paramount to define this procedure precisely when it is applied

and to quantify the performance changes associated with implementing it.

Additionally, the works that integrate structured and unstructured features, while using more advanced learning and NLP techniques, do not quantify performance improvements. There is a need to verify the statistical validity of utilizing these more advanced modalities over baseline, which can be tested using classical statistical tests such as DeLong's test for differences in area under the receiver operating curve (AUROC), and McNemar tests for differences in $F_2$ scores.

To address these research gaps, this study builds off of the work performed by Gao et al. [18] where the researchers applied a classical machine learning pipeline to a cohort of sepsis-3 patients to assess predictive power on structured clinical data including vitals, laboratory, demographic, and comorbidity features. This work incorporated SMOTE to address class imbalance of sepsis mortality, a standard method used in clinical machine learning pipelines to increase a model's discriminative power by increasing the minority class sample size with synthetic points derived from class labels and feature distributions, thus balancing a binary event outcome with the number of non-events. While attempts were made to make fair comparisons to the reports generated in Gao et al.'s study, this study branched off in new directions using a similar but different cohort, used expanded hyperparameter tuning for modeling, and incorporated resampling under a different protocol. Therefore, valid statistical measures cannot be made to compare to the results from Gao's study, but the work remains vital for determining the effectiveness of incorporating NLP into sepsis mortality prediction workflows.

This study builds off of previous literature by integrating structured clinically relevant features to sepsis mortality with Word2Vec embeddings derived from natural language processing of radiology notes data. By providing statistical testing to assess the impact of this integration and the effects of SMOTE resampling, the research presented in this study seeks to offer a quantifiable, statistically rigorous argument for the integration of clinical text notes in sepsis mortality prediction. Further, it proposes that incorporating this logical framework into existing methods such as GloVe, ClinicalBERT and others, as well as in survival analysis methods, is a reasonable extension for future analysis.

Python is an open-source coding language that was used for this study as it has a variety

of packages useful for ML and NLP. It is fast, efficient, and allows for creating modular and reproducible pipelines. A scaffold was implemented to organize the coding portion of this study into manageable source code scripts and Jupyter notebooks. It can be found in the GitHub repository found in Appendix A Codebase.

## 1.4 Objectives and Hypotheses

In this thesis, several hypotheses are explored to understand the impact of incorporating clinical text into machine learning pipelines for model discriminative power while maintaining stable calibration and assessing the stability and interpretability of these models.

The central hypothesis this study explores is to evaluate if incorporating textual representations from clinical notes enhances prediction of in-hospital mortality for patients in the ICU with sepsis. To assess this, this study explores whether models incorporating Word2Vec embedding features improves area under the receiver operating curve compared to the models using baseline structured data only. Therefore, the central null hypothesis is described as Hypothesis 1.

**Hypothesis 1** — Word2Vec improves discrimination

$$H_0 : \Delta\text{AUROC} = \text{AUROC}_{\text{Multimodal-Word2Vec}} - \text{AUROC}_{\text{Structured}} = 0$$

Specifically, the aims are to compare the following groups of models according to area under the receiver operating curve (AUC-ROC or AUROC) after hyperparameter tuning:

1. Best Word2Vec-optimized vs Best Structured only
2. Word2Vec-baseline vs Structured only
3. Word2Vec-optimized vs Structured only
4. Word2Vec-optimized vs Word2Vec-baseline
5. SMOTE vs non-SMOTE

In this way, the best overall optimized Word2Vec multimodal model was compared against the best structured model, but more importantly, groups of models were compared against one another to test the performance impact of incorporating NLP embeddings into the feature space. The performance of traditional machine learning classifiers were compared across standard and resampling training paradigms. Additional hypotheses that this study will address are described next.

**Hypothesis 2** — SMOTE enhances sensitivity.

$$H_0 : \ \Delta \mathrm{F}_2 = \mathrm{F}_{2,\mathrm{SMOTE}} - \mathrm{F}_{2,\text{Non-SMOTE}} = 0$$

This is assessed by testing for statistically significant changes in $\mathrm{F}_2$ score using a fixed threshold of 0.5, using McNemar's Test.

**Hypothesis 3** — Interpretability maintained across modalities.

This is qualitatively assessed via clustering of Word2Vec embeddings, using SHAP summaries and dependency analyses, feature importance rankings, and calibration assessment.

## 1.5 Thesis Outline

This thesis is organized as follows:

1. **Introduction**
2. **Data Procurement and Processing**
3. **Methods**
4. **Results**
5. **Discussion**
6. **Conclusion**

The next chapter, Data Procurement and Processing, discusses the extraction and processing procedure for obtaining patient-level data and the steps taken to apply natural language processing to the clinical text documents.

# 2.0 Data Procurement and Processing

## 2.1 Data Source and Cohort Selection

The raw structured data for this study originally come from the publicly available Medical Information Mart for Intensive Care (MIMIC-IV) Version 2.2 (v2.2) database. The raw unstructured clinical text data come from the MIMIC-IV notes database. In this study, a cleaned dataset for diagnoses, score assessments, vitals, demographics, and more was used as curated by Gao et al. BigQuery was implemented to pull clinical text radiology notes from the MIMIC-IV notes database. As the purpose of this study is to advance the predictive power associated directly with including Word2Vec embeddings in the original structured clinical data, great care was taken to extract notes corresponding to the exact patients used in the original structured dataset. It was performed to match exactly by `subject_id` as presented in the cleaned dataset curated by Gao et al. in their GitHub repository. [37]

### 2.1.1 Database Description

The MIMIC-IV database is a relational database that was produced by the Massachusetts Institute of Technology Laboratory for Computational Physiology. It contains de-identified electronic health records of patients admitted to Beth Israel Deaconess Medical Center from 2008 to 2019. Access to the database was received following the procedure on Physionet.org and complying with credentialing, training, and certification requirements including Collaborative Institutional Training Initiative (CITI) human subjects research training. After being credentialed and signing data use agreements, access to the different MIMIC-IV libraries was obtained. Data handling has been adhered to following the data use and privacy agreements.

The database has multiple modules for hospital-level data, ICU-level data, emergency department-level data, and more. The information contained in these modules are relational data tables that connect patient demographics, vitals, laboratory results, and diagnosis codes, among others, to admissions data, hospital data, and more. The breadth of infor-

mation contained in these modules allows researchers to perform data analysis for human medical research purposes. For this study, structured clinical variables and unstructured clinical text were extracted from this database.

### 2.1.2 Inclusion and Exclusion Criteria

The definition of sepsis described in 1.1 Clinical Background was used to define the patient cohort eligible for this study — patients who had a Sequential Organ Failure Assessment (SOFA) score of 2 or greater, a diagnosis of Sepsis-3 indicated, and suspected infection indicated.

Gao et al. used specific selection criteria for the purpose of their study, and those criterion carry into this study. Adults 18 years and older who had their first hospital stay, with an intensive care unit (ICU) stay of 24 hours or longer were included. A PaO2/FiO2 ratio of 200 threshold was set based on previous work by Bi et al., 2023 [5], and a binary coma score was assessed based on having a coma score of 8 or greater, according to the methodology by Gao et al. [18]

### 2.1.3 Outcome and Structured Features

The outcome variable used for this study was in-ICU mortality. The binary outcome variable, `hospital_expire_flag` was used as the target variable for prediction classification.

There were several continuous variables included. Age at death or discharge was the age used, which was acceptable for this study as sepsis is a fast-developing disease and length of ICU stay, another covariate used, was no greater than 102 days. SOFA score, average urine output, glucose, sodium, and albumin levels, heart rate levels, systolic blood pressure (SBP) and diastolic blood pressure (DBP) levels, respiratory levels, and $SpO_2$ levels were also considered.

Categorical variables used in this study included binary outcome variables sex, diabetes with complication or comorbidity, diabetes without complication or comorbidity, severe liver disease, aids, renal disease, and coma as defined earlier. Other categorical variables used were race and the technique of vancomycin delivery, if any.

Table 1 displays all variables used in this study. Continuous variables were assessed for normality using Shapiro-Wilk normality test, with normal continuous variables giving mean and standard deviation (sd) and non-normal variables giving median and interquartile range [IQR]. Age, heart rate, SBP, DBP, respiratory rate, $SpO_2$, and sodium are presented as mean (sd) even though they did not pass Shapiro-Wilk at N = 5208. The p-Values for these features correspond to Mann-Whitney U tests. Categorical variables list counts and proportions (%). See notes in Table 1 for statistical tests used for p-values reported.

Table 1: Descriptive Statistics for Structured Features Included in This Study.

| Variable | Overall | Survivors | Non-Survivors | P-Value |
|---|---|---|---|---|
| n | 5208 | 4005 | 1203 | |
| **Demographics** | | | | |
| Age (years) | 64.8 (15.8) | 64.3 (15.9) | 66.5 (15.5) | $< 0.001$ |
| Sex | | | | |
| Female | 2167, 41.6% | 1687, 42.1% | 480, 39.9% | 0.181 |
| Male | 3041, 58.4% | 2318, 57.9% | 723, 60.1% | 0.181 |
| Race | | | | |
| Black or African American | 685, 13.2% | 518, 12.9% | 167, 13.9% | 0.421 |
| Hispanic or Latin | 281, 5.4% | 220, 5.5% | 61, 5.1% | 0.620 |
| Other Race | 929, 17.8% | 727, 18.2% | 202, 16.8% | 0.299 |
| White | 3313, 63.6% | 2540, 63.4% | 773, 64.3% | 0.621 |

Continued on next page

| Variable | Overall | Survivors | Non-Survivors | P-Value |
|---|---|---|---|---|
| **ICU Characteristics** | | | | |
| ICU Length of Stay (days) | 3.5 [2.0, 7.7] | 3.3 [1.9, 7.2] | 4.4 [2.2, 9.2] | $< 0.001$ |
| Coma | 1164, 22.4% | 758, 18.9% | 406, 33.7% | $< 0.001$ |
| First Hospital Stay | 5208, 100.0% | 4005, 100.0% | 1203, 100.0% | 1.000 |
| Suspected Infection | 5208, 100.0% | 4005, 100.0% | 1203, 100.0% | 1.000 |
| Sepsis-3 Criteria Met | 5208, 100.0% | 4005, 100.0% | 1203, 100.0% | 1.000 |
| **Vitals** | | | | |
| Heart Rate (Min) | 67.4 (15.8) | 67.2 (15.5) | 67.8 (16.8) | 0.164 |
| Heart Rate (Max) | 117.4 (22.8) | 116.1 (22.6) | 121.6 (23.1) | $< 0.001$ |
| Heart Rate (Mean) | 88.6 (14.7) | 88.0 (14.5) | 90.6 (15.1) | $< 0.001$ |
| Systolic BP (Min) | 80.7 (16.3) | 82.3 (15.7) | 75.4 (17.0) | $< 0.001$ |
| Systolic BP (Max) | 158.2 (27.0) | 158.7 (27.3) | 156.3 (25.9) | 0.007 |
| Systolic BP (Mean) | 115.7 (14.2) | 116.7 (14.4) | 112.3 (12.9) | $< 0.001$ |
| Diastolic BP (Min) | 40.2 (11.2) | 41.2 (11.0) | 36.9 (11.3) | $< 0.001$ |
| Diastolic BP (Max) | 98.5 (23.8) | 98.4 (23.8) | 99.1 (24.0) | 0.237 |
| Diastolic BP (Mean) | 62.3 (9.7) | 62.8 (9.7) | 60.4 (9.3) | $< 0.001$ |
| Respiratory Rate (Min) | 10.9 (3.7) | 10.9 (3.7) | 11.0 (4.0) | 0.410 |
| Respiratory Rate (Max) | 32.5 (7.7) | 32.0 (7.6) | 34.1 (7.7) | $< 0.001$ |
| Respiratory Rate (Mean) | 20.0 (3.7) | 19.7 (3.6) | 20.8 (3.8) | $< 0.001$ |

Continued on next page

| Variable | Overall | Survivors | Non-Survivors | P-Value |
|---|---|---|---|---|
| SpO2 (Min) | 88.1 (9.6) | 88.8 (8.8) | 85.9 (11.7) | $< 0.001$ |
| SpO2 (Max) | 99.8 (0.7) | 99.8 (0.6) | 99.8 (0.9) | 0.076 |
| SpO2 (Mean) | 96.9 (2.0) | 97.0 (1.8) | 96.6 (2.4) | $< 0.001$ |
| Average Urine Output (mL/day) | 153.0 [97.1, 219.4] | 159.7 [106.9, 227.1] | 129.2 [67.6, 195.2] | $< 0.001$ |
| **Laboratory Measurements** | | | | |
| Glucose (Min) | 96.0 [79.0, 116.0] | 97.0 [81.0, 117.0] | 92.0 [74.0, 112.0] | $< 0.001$ |
| Glucose (Max) | 182.0 [140.0, 256.0] | 180.0 [138.0, 253.0] | 191.0 [147.0, 266.0] | $< 0.001$ |
| Glucose (Average) | 134.5 [113.8, 170.5] | 134.0 [113.5, 170.0] | 135.8 [114.7, 171.7] | 0.321 |
| Sodium (Min) | 133.7 (6.1) | 134.0 (6.0) | 132.9 (6.4) | $< 0.001$ |
| Sodium (Max) | 141.9 (5.7) | 141.8 (5.5) | 142.3 (6.2) | 0.005 |
| Sodium (Average) | 137.9 (4.7) | 138.0 (4.6) | 137.7 (4.9) | 0.046 |
| Albumin (g/dL) | 3.4 [2.8, 4.0] | 3.5 [2.8, 4.0] | 3.2 [2.6, 3.8] | $< 0.001$ |
| **Comorbidities** | | | | |
| Diabetes (No Complications) | 1565, 30.0% | 1198, 29.9% | 367, 30.5% | 0.720 |
| Diabetes (With Complications) | 777, 14.9% | 608, 15.2% | 169, 14.0% | 0.357 |
| Severe Liver Disease | 756, 14.5% | 510, 12.7% | 246, 20.4% | $< 0.001$ |
| AIDS/HIV | 89, 1.7% | 67, 1.7% | 22, 1.8% | 0.811 |
| Renal Disease | 1650, 31.7% | 1241, 31.0% | 409, 34.0% | 0.053 |

Continued on next page

| Variable | Overall | Survivors | Non-Survivors | P-Value |
|---|---|---|---|---|
| **Vancomycin Administration** | | | | |
| IV | 4441, 85.3% | 3345, 83.5% | 1096, 91.1% | $< 0.001$ |
| Antibiotic Lock | 8, 0.2% | 6, 0.1% | 2, 0.2% | 1.000 |
| Enema | 44, 0.8% | 35, 0.9% | 9, 0.7% | 0.812 |
| Intrathecal | 0, 0.0% | 0, 0.0% | 0, 0.0% | 1.000 |
| Oral Liquid | 310, 6.0% | 252, 6.3% | 58, 4.8% | 0.069 |

Note: Continuous variables are summarized as mean (SD) if approximately normal by Shapiro–Wilk testing, or median [IQR] otherwise. Binary variables are reported as counts (%). p-values are derived from Welch's t-tests or Mann–Whitney U tests for continuous variables and chi-square or Fisher's exact tests for binary variables, as appropriate.

### 2.2 Structured Data Processing

The data available from the GitHub repository by Gao et al. were previously cleaned, so preprocessing was not necessary for the structured data. There were no missing structural data as missing values and duplicated data were filtered out during data preprocessing. They outline their selection and processing techniques in their Methods section.

The structured features were extracted using BigQuery and come from various tables contained within the MIMIC-IV database and derived by Gao et al. Their resulting dataset resulting included all sepsis3 patients who had some or all of these clinical indicators.

To ensure a stable training space, Gao et al. chose to drop duplicated row values, and removed patient records with missing data. The resulting cleaned dataset represents patient level aggregated data, demonstrated by minimum, maximum and average level laboratory and vitals data.

Categorical variables were one-hot encoded. All binary variables with yes/no or true/false outcomes were thus assigned a 1 to signify the categorical variable was present and a 0 if it was not. Categorical variables with more than two levels were one-hot encoded to signify presence of the level, and no presence of any of the other levels.

### 2.3 Unstructured Text Data

A SQL query was performed using Google Cloud Platform's BigQuery tool to identify *all* unique radiology notes per patient; see Appendix A Codebase under sql/. The manner in which these notes were extracted presents a full general radiology background for each patient. Instead of looking at a targeted window of time for each patient's ICU stay, techniques that are important to early prediction models and survival analysis for sepsis, the full background gives EHR data that capture the full radiological background of a patient, which is helpful for identifying important language features related to sepsis for NLP. Since each patient had a minimum stay in the ICU of 24 hours or greater, all of these notes are associated with patients who ended up in the ICU with a sepsis-3 diagnosis. This general approach to gathering notes is a limiting aspect of this study from an early-prediction perspective, but the statistical validity of incorporating Word2Vec embeddings for demonstrating the utility of NLP in sepsis mortality prediction remains intact.

The patients were identified using the `subject_id` from `data_after_cleaning.csv` file in the GitHub repository of Gao et al. [37]. The resulting file, `data_full_notes_interim.txt`, contained additional columns such as `note_type_1` and `text`. Each patient note was cleaned, and all notes of similar type were collapsed into a single note per patient. This was an ideal process for this study, as it kept the total sample size fixed to the number of patients in the cohort. There were limiting drawbacks to this strategy, including that this strategy may have included radiology notes for some patients outside of their time in the ICU. It is unclear how Gao et al. processed the structured data including start and end time for aggregate patient-level statistics, so no assumptions are made whether the time intervals for structured and unstructured data align or not.

Although this strategy was primarily based on improving computational efficiency by reducing model training times due to minimizing the note sample size, as well as using a fairly primitive use of Word2Vec, a more complete approach for future work will be described in 5.3.2 Clinical Note Preprocessing Tokenization and Temporality and in 5.4 Future Work.

### 2.3.1 Note Type and Extraction

Radiology plays a key role in identifying suspected infections in patients experiencing infection-like symptoms. Due to the abundance of radiology procedures performed for ICU patients, there is an abundance of radiology notes to learn from. Radiology was a note type of choice to identify key semantic structures that could provide signal for sepsis, and potentially for sepsis mortality. Therefore, the specific data for the text processing of clinical notes come from the individual radiology notes per procedure for each patient. There are risks associated with using text notes due to semantic inconsistencies between providers or the risk of rare events that cannot be identified in clinical notes, but natural language processing in healthcare is designed to navigate these risks and provide robust interpretations. More on the limitations of using radiology notes for natural language processing is addressed in 5.3.1 Data Level Limitations.

After extracting the data into long-format, cleaning and preprocessing was performed to prepare the data for Word2Vec training.

### 2.3.2 Missing Unstructured Data Handling

It was discovered late in the study design that 2 of the 5208 patients had missing or no radiology notes. As this represents less than 0.04% of the total sample size, it carries little statistical impact, but is worth mentioning for completeness. In this study, the 5208 patients were used for model training, with these two patients having blank Word2Vec embeddings.

### 2.3.3 Text Cleaning Pipeline

Several helper functions were designed to apply the cleaning procedures and can be found in Appendix A Codebase under src/data_prep.py. All text data underwent text cleaning via a cleaning pipeline prepared in python:

1. Notes per patient were loaded in from `data_full_notes_interim.csv`.
2. A grouping function was applied to group notes by `subject_id` and `note_type_1` that returned records for each patient, aggregating text entries from notes into lists.

Then, the grouped dataframe was converted into a list of dictionary records with `subject_id` as a primary key.

3. To clean each individual clinical note string, `clean_text` was developed to normalize whitespace, remove underlines, remove unnecessary characters while keeping clinical symbols, and handled missing values. A raw note text would be passed as an argument and the cleaned note string would be returned.
4. The helper function `process_group` was developed to accept a grouped record of notes, and apply `clean_text`. Therefore, each individual cleaned note per patient was processed and concatenated into a single clinical notes string for each patient.
5. Using `joblib` inside of `process_notes_in_parallel`, grouped note records were processed in parallel to apply the `process_group` function to all notes in a record. Ultimately, this step led to immense computational efficiency by utilizing the max number of cores available to process notes instead of using much less computationally efficient serial processing with 1 core. The function accepted a records list and the number of CPU cores to use, and returned a list of processed dictionaries with keys: `subject_id`, `note_type_1`, and `combined_notes`.
6. The processed notes were then converted from the list of dictionaries back to a pandas dataframe using the output from `process_notes_in_parallel`, and it was saved to csv.
7. A pivot was performed using the note type as the pivoting column and the clinical text as the value column to convert and filter the notes into a column specific to combined radiology notes per patient.
8. Finally, the processed single clinical note text string per patient column was appended to the original structured dataset to create `nlp_ready_df`, a dataframe ready to be passed into NLP transformer pipelines.

This pipeline allows for full modularity and reproducibility for downstream forks of this study, but alternative approaches for batch processing notes could be used for future work involving NLP for sepsis mortality prediction. These concatenated note strings were prepared for tokenization, which will be described next.

### 2.3.4 Tokenization and Aggregation

Word2Vec requires tokenization prior to training. This can be applied by transformers, or can simply be a tokenized sentence structure as a list of sentences. The per patient radiology notes from `nlp_ready_df` were passed to `write_radiology_notes_for_w2v`, a helper function that returns tokenized sentences. In this study, the corpus given to Word2Vec are the tokenized sentences produced and written into a text file by `write_radiology_notes_for_w2v`.

These tokenized sentences were aggregated at the patient level. Grouped records were not sorted by `charttime`, and so the notes were not concatenated chronologically. Temporality was therefore not maintained in the linguistic context of developed tokens. Word2Vec does not require these notes to be chronologically ordered, as will be discussed in the next section, so this drawback is minor. More on the notes temporality is discussed in 5.3.2 Clinical Note Preprocessing Tokenization and Temporality.

## 2.4 Word2Vec Training

### 2.4.1 Overview

Word2Vec is a collection of architectures introduced by Google in 2013 to compute continuous vectors from collections of words from large datasets. [41] It uses a shallow neural network to embed words from a semantic structure, based on words and semantics in its surrounding context, into a lower-dimensional vector space. It outputs word-vectors that have similar meanings based on contextual surroundings when they are close together in vector space. Word2Vec is used as an alternative to the classical bag-of-words models that contain vectorized representations of words, but lack semantic and temporal meaning. [61]

Word2Vec models are trained using stochastic gradient descent and back-propagation and are inspired by the Feedforward Neural Net Language Model (NNLM or NN) and the Recurrent Neural Net Language Model (RNNLM or RNN) with the main difference between the two that the NN includes a projection layer that is projected onto by the input layer, while the RNN model does not have a projection layer.

Word2Vec is time-independent in both of its architectures, meaning that it maps tokens to embeddings in a manner that is independent of time — the notes do not have to be inherently ordered. Both architectures look at windows of tokens for textual predictions and use the full corpora of tokens for determining a global embedding space, which is then used to find patient-level embeddings. Since Word2Vec is independent of time, the notes passed into it do not need to be ordered. However issues can arise when the search window crosses sentence boundaries that could be connected to one another, or contexts are mixed in concatenated notes. Temporality is lost unless tokens are set to be searched across in order by explicit design.

The goal of using Word2Vec for this study was to learn word-vector embeddings from the clinical corpora passed into the Word2Vec trainer from 2.3.4 Tokenization and Aggregation. A preliminary choice for the Word2Vec architecture was used to set baseline metrics for downstream model training, while the optimal Word2Vec architecture was determined from the a combination of hyperparameters that came from maximizing pairwise cosine similarity scores across patient-level embeddings.

### 2.4.2 CBOW vs Skip-Gram

Word2Vec incorporates two architectures for training word-vector embeddings from a collection of text data, continuous-bag-of-words and skip-grams.

**Continuous-bag-of-words (CBOW)** provides a continuous distributed representation of the context and predicts the current word based on the context. The order of words does not impact the projection, and Word2Vec uses both historical and future words to predict the current word.

**Skip-grams (SG)** try to maximize the classification of a word based on other words contained in the same sentence. Each current word is used as the input to the log-linear classifier and predicts words within a specific range before and after the current word. Increasing range improves word-vector quality, but it comes at a computational expense. Less weights are given to words far away from the current word than those closest.

### 2.4.3 Preliminary Embedding Parameters

Word2Vec offers several tuning parameters for finding the best word-vector representations for word corpora. Such parameters include min_count, vector_size, window, size, negative, alpha, and more. [60] The preliminary tuning parameters for baseline fitting are shown in Table 2. This kept the embedding feature space low (100) and provided preliminary embeddings by incorporating 10 epochs and keeping the window low and the minimum count low. It focused on collections of words that were closer together, to retain clinical contextual meanings within sentences.

These parameters were useful for quickly identifying the best model families for further tuning, but after the first grid search, the cosine similarity heuristic was incorporated to find a better hyperparameter tuning grid for Word2Vec.

Table 2: Preliminary Word2Vec Parameter Configuration Used for Baseline Embedding Fitting.

| **Parameter** | **Selected Value** |
|---|---|
| `vector_size` | 100 |
| `window` | 5 |
| `min_count` | 2 |
| `sg` | 0 (CBOW model) |
| `epochs` | 10 |
| `workers` | 4 |

## 2.5 Post-Baseline Word2Vec Training

After identifying the best models in model development using baseline Word2Vec parameters, Word2Vec embeddings were optimized for a final pass-through of the top classifiers. Running a full grid search on each model family for N-many hyperparameter combinations

of Word2Vec would have resulted in N-fold grid search rounds in model training, and with the Word2Vec parameter grid selected, this would result in 64-fold greater compute time. A lightweight heuristic using pairwise cosine similarity scores across embeddings was used to quickly identify the best parameter space for generating patient-level embeddings from the notes corpora prior to downstream model training and evaluation.

These pairwise cosine similarity scores between patient level embeddings generated from fully concatenated radiology notes data were used to identify the best tuning parameter space to determine optimal feature embeddings for training.

### 2.5.1 Cosine Similarity

The cosine of two non-zero vectors, A and B, can be defined as follows:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\|\|\mathbf{B}\| \cos\theta.$$

The resulting range leads from -1 to 1, with values of -1 having opposite similarity and values of +1 corresponding to identical similarity. A value of 0 is indicative of orthogonality between the two vectors and therefore indicates that there is no similarity between them. This metric can be utilized in Euclidean spaces to evaluate the similarity between two vectors. In NLP context, it can measure the similarity between two or more semantic structures — it can indicate similarity in semantics of two separate word documents. Semantic structures that are identical in sentence structure and/or meaning, such as radiology evaluation notes between two similar patients, would have similarity scores close to 1.

In the context of Word2Vec in this study, a pairwise cosine similarity value of 1 would mean that the word-vectors calculated between two patients are identical and would thus represent identical semantic meaning contained in each corresponding radiology note.

A parameter grid space was created to identify the best hyperparameter combination to maximize mean pairwise cosine similarity. A higher mean pairwise cosine similarity score would result in the collection of resulting word-vectors having similar directionality, a rough metric for assessing the quality of generated embeddings without incorporating extensive downstream modeling into the selection scheme. The parameter space shown in Table 3 was

kept relatively low (N=64) to reduce computational cost while retaining contextual meanings in the embeddings.

Table 3: Parameter Search Space for Word2Vec Embedding Optimization.

| **Parameter** | **Search Range / Options** |
|---|---|
| `vector_size` | [100, 200] |
| `window` | [5, 10] |
| `min_count` | [2, 5] |
| `sg` | [0, 1] (0 = CBOW, 1 = Skip-gram) |
| `negative` | [5, 10] |
| `epochs` | [15, 25] |
| `workers` | 12 |

### 2.5.2 Intrinsic Evaluation

Each of the 64 combinations was trained using a helper function `train_w2v_variant` and patient-level document embeddings were generated. A function `intrinsic_similarity` was used to compute the pairwise cosine similarity scores between all patients using the first 200 vocabulary words of each token at the patient-level, and this was not randomized for reproducibility. These pairwise similarity scores were saved into a pairwise cosine similarity matrix, and the mean value of the upper triangle of this matrix served as a single scalar score for how cohesively the entire embedding space represented the corpus of all notes across all patients. This allowed for a range in semantic cohesion assessment across different parameter combinations for Word2Vec, while providing rapid iterations for comparing these combinations without having to retrain the full set of predictive models on each Word2Vec candidate parameter configuration.

### 2.5.3 Optimal Embedding Parameters

Using an orchestrator function, `optimize_word2vec`, several helper functions were deployed to run a grid search on each Word2Vec hyperparameter combination. Each Word2Vec combination variant was trained using the entire radiology corpus, then the intrinsic similarity scores between the pairwise embeddings were determined from the vectors created by the trained model, and then the scores for each trained Word2Vec model were appended to a records list. The best combination was selected and saved to a JSON file and then retrained on the full radiology corpus once more for clarity. See 4.3.2 Selected Optimal Embedding Configuration for the selected configuration.

## 2.6 Feature Engineering and Model Training Preparation

With the baseline and optimal Word2Vec hyperparameters defined for training and final training respectively, the optimal feature space was thereby created to add 100 Word2Vec embeddings per patient.

### 2.6.1 Data Partitioning

For each separate variant, `baseline_original`, `baseline_w2v`, and `w2v_optimized`, samples were randomly assigned to independent training and testing datasets using a split of 80/20 train/test, stratified by `hospital_expire_flag` to maintain uniform class proportions between splits. The split was performed using `train_test_split` from the `sklearn.model_selection` sub-module. The original structured dataset was split first prior to standard scaling, and the `subject_id` partitions were used across the downstream Word2Vec variant partition schema.

The training sets were used solely for resampling, model fitting, and hyperparameter tuning, while the holdout test sets were untouched for final evaluation and determining model metrics. A random state seed was implemented for all splits to maintain reproducibility.

### 2.6.2 Embedding Aggregation

A helper function, `apply_embeddings_to_subjects` was used to generate embeddings for each patient. This used an additional helper `get_subject_embedding` by computing the Word2Vec embedding for each patient using the optimized Word2Vec model and the patient's radiology notes token. These embeddings were averaged and aggregated for each patient, and then each patient was subset into the training and testing splits defined by the train/test split used for structured data.

### 2.6.3 Normalization and Merging

Normalization is a necessary step for distance-based classifiers such as Support Vector Classifier (SVC), and it optimizes convergence for gradient-descent based classifiers such as Logistic Regression and Multilayer Perceptron (MLP). While it is not necessary to apply normalization techniques for many tree-based models like Random Forest, XGBoost, and Decision Trees, it does not change the decision boundaries created by them. Therefore, after performing the train/test split on the original, structured-only feature space, `StandardScalar` was fit on the training data, and transformed on the test data, as it is best practice to apply the fit to the training set and then to apply the transformation on the test set. This was performed prior to merging to maintain consistent numeric distributions for the original features across the three main variants. Then, after embeddings were aggregated per subject, `StandardScalar` was fit on training embeddings for each Word2Vec feature space (baseline and optimized), and then transformed on the test set embeddings.

The normalized structured dataset was merged to each separate normalized Word2Vec embedding feature set (baseline and optimized). The `subject_id` variable was used as the joining variable and the merges were performed using the `merge` command. Finally, the `subject_id` was removed from the feature space in each data set to reduce noise as it does not contribute to model fitting.

The next chapter, Methods, outlines the modeling framework devised using the preprocessed training and testing data, incorporating resampling schemes, model selection, and evaluation techniques.

# 3.0 Methods

## 3.1 Addressing Class Imbalance

Before starting model development, it was important to address the class imbalance of the target variable, as sepsis-induced mortality tends to lead to imbalanced distributions among those who survive and those who do not.

### 3.1.1 Target Variable Distributions

Figure 1 shows the class distributions for sepsis mortality. It suggests that the class distribution among sepsis mortality is imbalanced, with survival counts being three times as high as mortality counts. This imbalance continued into the train/test split, which can affect model discrimination of the outcome, so efforts were made to account for these imbalances by incorporating a resampling strategy for model performance assessment and for sensitivity analysis.


Figure 1: Class Distributions of Sepsis Mortality.

### 3.1.2 Synthetic Minority Over-Sampling Technique

Synthetic Minority Over-Sampling Technique (SMOTE) is a method designed by Chawla, Nitesh V. et al. [8] to address class imbalances in a target variable. This can improve performance by improving minority class representation [7], and more appropriate discriminative surfaces can be established by smoothing irregular boundaries with the inclusion of additional points. When applied on internal cross-validation procedures, significant increases in the averaged within-fold AUROC can be observed, as demonstrated in the exploratory descriptive internal cross-validation analyses contained within the model training loop (see Mean CV AUROC scores in Tables 7, 8, and 16). Due to the imbalanced nature of sepsis mortality, previous work has incorporated resampling into the model training framework, such as that in Gao et al. [18], and others [19], [36]. These works [18], [19], and this demonstrate the power of applying SMOTE resampling for sepsis mortality prediction, reporting mean cross validation AUROC scores for models incorporating SMOTE resampling to be $> 0.90$ prior to holdout testing.

An important aspect of applying SMOTE for resampling is to apply it only to the training dataset, and furthermore when performing cross-validation, to the training folds. [16] Allowing SMOTE to be applied to the holdout test set leads to information leakage, leading to inflated accuracy measurements and AUROC scores. [28] Additionally, information leakage can enter into validation folds if SMOTE is applied on the full training set prior to model training, as opposed to a subset of it.

### 3.1.3 Applying SMOTE

In this study, cross-validation for hyperparameter tuning was not performed on SMOTE data. (1) The best hyperparameter configuration was determined from non-SMOTE model training, (2) SMOTE was applied to the full training set *after* model training, generating synthetic training points to balance minority and majority class labels, (3) each model generated in hyperparameter tuning was then fit once on the full SMOTE training set and then (4) evaluated on the holdout test dataset. In this way, there was no information leakage because the test set data was fully held out and SMOTE models assumed hyperparameter

configurations from non-SMOTE models. There was no cross-validation information leakage across folds because refitting on SMOTE datasets did not involve cross validation or model training at all. In this way, efforts were made to minimize information leakage while allowing for proper assessments of the impact of SMOTE resampling on $F_2$ statistics, impacting calibration and sensitivity.

After merging the original structured features with Word2Vec embeddings throughout each train and test split, a custom resampling function was applied to each training split in the original_baseline and w2v_baseline datasets. This function applied SMOTE to the training sets *only* and returned all of the necessary SMOTE training sets for fit-transforms. To provide a robust analysis and to inspect the impact of incorporating class re-balancing on prediction accuracy, model training was performed both on the non-SMOTE training variants and on the SMOTE-resampled variants.

## 3.2 Model Development

With the training and test sets fully prepared without and with SMOTE resampling applied where appropriate, model development began. MLflow, an ML experimentation and management platform, was used for artifact and metric tracking, and a random seed of 42 was used everywhere a random seed was needed, to ensure reproducibility.

### 3.2.1 Model Families

To begin model selection, a set of classifiers was chosen for baseline model training and hyperparameter tuning. All models selected by Gao et al. were included, as well as three additional models for expanded selection scope:

1. **Logistic Regression (LR):** Special case of General Linear Model using the binomial conditional distribution and a logit link. [46]
2. **Decision Tree (DT):** Non-parametric supervised learning technique that learns simple decision rules from feature space. Deeper trees allow for more complex

decisions. [31]

3. **Random Forest (RF):** A perturb and combine technique used on trees. A diverse set of randomized trees are constructed as classifiers, then the ensemble makes a prediction based on the average of the individual classifiers. [33]
4. **Gradient Boosting Machine (GBM):** A classifier method based on boosting with gradient descent in a functional space giving predictions based on an ensemble of weak prediction models, typically decision-trees. [56]
5. **Extreme Gradient Boosting (XGB):** A newer boosting method that works as Newton-Raphson in functional space and provides alternative features to traditional gradient boosting. [54]
6. **LightGBM (LGBM):** A lightweight gradient boosting method that scales efficiently with increasing sample size and feature space by randomly dropping shallower gradient descents. [55]
7. **Support Vector Classifier (SVC):** A classifier that creates decision boundaries using support vectors, and works well in high-dimensional spaces. Different kernel methods such as radial basis functions and linear basis functions can be used for determining boundaries. [29]
8. **Multilayer Perceptron (MLP):** A supervised neural network technique that learns a linear or non-linear function based on a pre-determined set of hidden layers and nodes. Weights are determined at each layer, followed by a linear or non-linear activation function. Input features are passed into the MLP and transformed from layer to layer based on optimizing the loss function. It continues training using backpropagation and ends when the loss function ceases to be minimized. [32]
9. **Naïve Bayes (NB):** A baseline technique that works by assuming conditional independence between each pair of features given the outcome variable, and it uses maximum a posteriori (MAP) estimation to estimate $P(Y)$. It is a fast method compared to tree-based methods. [30]
10. **CatBoost:** A high-performance gradient boosting method that works on diverse data sources. [47]

Each model was implemented using the `scikit-learn`, `XGBoost`, `LightGBM`, and `CatBoost`

Python libraries, allowing for consistent model training, cross-validation, testing, and overall evaluation.

### 3.2.2 Tree-Based Methods

Feature spaces from each node in a tree based model are mutually exclusive and exhaustive, covering all of the decision-space. Leaves correspond to the full splitting of the feature space into several decision boundaries, where nodes set the structure of the tree creating decision pathways. The minimum number of observations in a node or leaf and setting longer tree depth allows for model stability. Larger trees tend to cover more complex decision boundaries, but trees can sometimes be too complex and overfit to training data.

Pruning is a way of regularizing a tree. When a model has lower bias but higher variation, one can use pruning to allow introducing some bias, but reducing variance - leading to better overall MSE. Pruning is introduced by growing a forest of tree-based models, then cutting out trees that do not contribute to reduced variance. Cross-validation and weakest link training / cost complexity pruning are used to improve tree performance.

Entropy tends to be more sensitive to predicting rare events than Gini, but Gini tends to be more computationally efficient. These methods weren't contrasted in this study, but would lead to more exhaustive hyperparameter tuning in future work. The Random forest uses a random selection of features to build each tree. It is geared towards reducing the amount of trees that might be bad for overall fit, and uses a majority voting scheme to determine the final class.

### 3.2.3 Boosting-Based Methods

Boosting based methods build off of previous models, so if a tree being built makes mistakes in decision boundaries, it is carried to subsequent trees. This error can be propagated into the final model. However, by learning from previous trees, it makes them fast and efficient compared to simple tree and forest based models. Random Forests is a hedge by randomizing across the forest of models built, ideally removing some noise from the forest. The boosting methods utilized in this work contrast the Logistic Regression, Multilayer

Perceptron, Support Vector Classifier, and tree-based methods, and offer alternative discriminative surfaces for comparison in the high-dimensional space caused by the dimensionality of the word-vectors.

### 3.2.4 Hyperparameter Optimization

A custom function for hyperparameter optimization and presenting evaluation metrics was designed, `repeated_cv_with_mixed_search`, which can be found in Appendix A: Codebase under src/models.py. This function incorporated many important techniques that will be described in this and the following sections. Each of the 10 selected models uses a set of hyperparameters that can be tuned and adjusted to improve decision boundaries. For example, the tree-based methods described earlier contain the leaf and depth parameters that can be adjusted to develop the root system for probability decision boundaries. Under certain tunings, a 0 (survival) could be predicted, while under other tunings a 1 (mortality) could be predicted.

The goal of the hyperparameter optimization in this study was to increase AUROC scores, for each classifier. These hyperparameters were tuned in `repeated_cv_with_mixed_search` using `GridSearchCV` for LR, DT, SVC, MLP, and NB, or `RandomizedSearchCV` for RF, GBM, XGB, LGBM, and CatBoost (collectively the "Randomized Models". `GridSearchCV` and `RandomizedSearchCV` come from the `sklearn.model_selection` python package. Since major full-grid expansions (on the order of several orders of magnitude) were outside the scope of this experiment, a small full-grid expansion was performed for LR, DT, SVC, MLP, and NB while a random selection of the full-grid expansion for the Randomized Models was used.

The list of tuning parameters for each classifier can be found in Appendix A: Codebase under src/models.py. In conjunction with the grid search for optimal parameter spaces, `RepeatedStratifiedKFold` was performed using 5-fold cross-validation with 10 repeats for a total of 50 folds per classifier, for each parameter combination.

Hyperparameters were specifically selected on the non-smote training data using mixed search. Training on SMOTE resampled data leads to data leakage into validation folds as

described earlier, resulting in inflated metrics. Bias-variance tradeoff was maintained to ensure validity in tuning and model selection. Each model was standardized with the random_state of 42 (similar to a random seed) and used standard ML optimization parameters for computational efficiency.

When fitting SVC(probability=True), the Platt scaler for SVC fits an extra 5 rounds of cross-validation per fit, which scaled immensely on 1600 fits. The choice was made to only refit the best model with probability=True to reduce overhead but maintain parameter-selection rigor.

### 3.2.5 Descriptive Internal Cross-Validation

Descriptive internal cross-validation is a method used to assess the uncertainty of holdout test set prediction AUROC scores. By performing additional cross-validation on the training set using cross-validation, computed AUROC scores per fold are averaged with a variance to quantify model robustness and stability. [51]

Once hyperparameters were saved and the best models were fit and transformed, each best model was passed to a descriptive internal cross-validation round. Model stability was ensured by quantifying averaged AUROC holdout scores across 5 folds, and the standard deviation of the computed AUROC score was computed as a metric comparison to the final holdout-test AUROC metric. It effectively served as a quality assurance check and balance to test for overfitting of the selected hyperparameters to the best model. This was performed independently of sampling and occurred for both non-SMOTE and SMOTE training data.

### 3.2.6 Retraining on SMOTE Balanced Training Sets

The best parameter space per classifier was then retrained on the earlier defined SMOTE training data, and tested on the non-resampled final holdout test data. This step allowed for robust interpretations of AUROC independent of sampling technique, as comparing horizontally across SMOTE vs non-SMOTE (holding variant type equal) and vertically across variant type (holding resampling type equal). The justification for using NLP to improve AUROC could then be made independent of sampling technique and robustly supported by

either approach. [3]

Parameter and model stability were later evaluated by computing the mean absolute coefficient for logistic regression and mean absolute predicted probability differences between baseline and the SMOTE-retrained baseline models to assess the effect of resampling on model calibration. See Appendix A: Codebase under notebooks/06_visualization.

## 3.3 Evaluation Framework

The evaluation metrics were saved and exported to summary dataframes for extraction and further interpretation. To assess performance, the area under the Receiver Operating Curve, or area under the ROC curve (AUROC), was the primary metric for this analysis. Gao et al. chose to use this metric of choice, and so it was a reasonable benchmark to compare against. [6]

However, as Gao et al. used an implementation of SMOTE that is different from the approach taken in this study, fair comparisons are impossible to be made. As such, the best tests for model performance gains lie in testing the improvement of the Word2Vec multimodal models over the original structured-only models.

### 3.3.1 Metrics

Together with AUROC, a suite of performance metrics was used to provide additional comparisons between variants:

1. **AUROC (Area Under the ROC Curve):** A metric that measures performance of a classification model measured by ability to distinguish between positive and negative labels.
2. **Accuracy:** Measures the proportion of correct predictions out of all possible predictions.
3. **Precision:** Or positive predictive value, is the ratio of the number of true positives predicted to all positives predicted, where the denominator is the sum of true pos-

itives and false positives predicted.

4. **Recall:** Or sensitivity, or true positive rate, is the ratio of true positives predicted to all positives true positives, where the denominator is the sum of true positives and false negatives.
5. **$F_1$-Score:** Calculates the harmonic mean of the precision and recall of a model into a single metric that serves as a balance between positive predictive ability and false negative predictive ability.
6. **$F_2$-Score:** Is similar to $F_2$ score, but it adds weight to recall, which is important for situations where minimizing false negatives is more important than minimizing false positives. It is a good metric for allowing models to be sensitive to detecting true events while maintaining stability to falsely predicting positive events through the incorporation of precision.
7. **Brier Score:** Evaluates the accuracy of probabilistic predictions. It is the mean squared difference of the predicted and actual probabilities over all events.

Each metric was computed on cross-validation folds, internal cross-validation folds, and the held-out test set for non-SMOTE and SMOTE-trained variants.

### 3.3.2 Validation and Testing Design

To ensure that class imbalance was adjusted between training folds, stratified splits were made on the target variable for both training and testing sets. As mentioned above, rigorous validation steps were taken to ensure optimal hyperparameter tuning. Each dataset was split 80/20 into training and test datasets, and furthermore each training set was split into validation folds, using 5-fold cross-validation. Therefore, in each training set, 5 separate 80/20 splits were made for testing, followed by aggregated AUROC computation to determine the best fitting model.

Using a train/validate/test framework is essential in machine learning pipelines [27] as it leaves the holdout test set completely unknown to model biases and data leakage. Hypothesis tests can then be performed with confidence that actions to minimize bias were taken at all steps leading up to the significance tests.

### 3.3.3 Metric Visualizations

To support metric analysis, grouped bar plots were created to compare AUROC scores. Receiver operating curves (ROC) for AUC were created to compare the discriminative power of models, and precision-recall (PR) curves were used to compare sensitivity between model classifiers.

## 3.4 Explainability and Statistical Testing

### 3.4.1 Feature Importance

Models were interpreted using the following feature importance interpretation strategies:

1. **SHAP (SHapley Additive exPlanations):** A unified framework for feature attribution based on game theory. SHAP values assign contribution values to features based on how much the feature increases or decreases an individual prediction relative to baseline expectations. [39]
2. **Logistic Regression Coefficients:** Feature importance was assessed using coefficients for the logistic regression models. These coefficients represent the log-odds change in the outcome associated with a one-unit increase in each predictor, holding other variables constant. They provide information on the direction and magnitude of the contribution by each feature to predicting class labels.

After computing SHAP values, and creating SHAP plots and dependency plots for all models, aggregated SHAP plots were created to assess feature importances. Importantly, if Word2Vec embeddings were to gain significant ground in improving AUROC, it meant that those features likely needed to show up in the top 15 feature importance rankings.

Then, using per-patient SHAP values vs most important feature, plots were established to visualize that relationship. Of note, no Word2Vec feature was number one and so were not included in these most important feature plots, so alternative feature relationship figures were created to assess the SHAP analysis of the top Word2Vec feature for each classifier.

### 3.4.2 Calibration

Predicted probabilities were evaluated for reliability using calibration curves. Brier scores were computed for probability forecasts comparing model predicted probabilities with actual outcomes. Well calibrated models tend to predict probabilities that correctly reflect outcome probability distributions at a given predicted probability.

### 3.4.3 Clustering Embeddings

PCA semantic clusters were then performed using KMeans to demonstrate embedding structures and how they map together in 2 dimensional space. This process helps to validate the embedding selection process — embeddings that contain semantic structure are appropriate for the prediction models as they produce signal for sepsis mortality, and those that lack semantic cohesion provide noise or are unable to help predict sepsis mortality prediction or survival.

### 3.4.4 Comparative Significance Testing

Model discrimination differences were statistically assessed via:

1. **DeLong's Test:** To compare the performance of AUROC scores and test for statistical significance.
2. **Bootstrap Confidence Intervals:** Used to generate (95%) confidence intervals for AUROC scores generated using bootstrap methods.
3. **Multiple Comparisons Testing:** To compare pairwise AUROC scores, incorporating Bonferroni corrections.
4. **$F_2$ Significance:** McNemar's test was incorporated to compare $F_2$ across resampling schemes.

The next chapter shows the main results of the study including significance testing, model interpretability, calibration, sensitivity analysis, and clustering analysis of embeddings.

# 4.0 Results

## 4.1 Overview of Analytical Workflow

The analyses described previously in the Methods section were designed to test the hypotheses outlined in 1.4 Objectives and Hypotheses. The results primarily evaluate whether the integration of Word2Vec embeddings with structured clinical features improves discrimination power compared to the use of structured features alone. Secondly, the SMOTE resampling scheme is assessed for its effect on discrimination and sensitivity. Finally, the analyses observe whether multi-modal models maintain interpretability and calibration compared to the structured-only models.

To evaluate these hypotheses, a series of Jupyter notebooks (See Appendix A Codebase under notebooks/) were constructed to preprocess the data and perform embedding generation, and handle model training, evaluation, and statistical testing. The sequence begins with structured-only baselines, proceeds to add Word2Vec features, applies supplemental SMOTE resampling for class balancing, optimizes the Word2Vec embedding space, and concludes with significance testing of model performance using DeLong statistical tests and bootstrapped confidence intervals for measuring uncertainty.

This chapter closely follows the organization of the coding workflow. Tables and figures are presented in the order of analysis to support the logical flow of initial models to final optimized models for testing the core hypotheses. Supplemental tables and figures are provided in Appendix B and Appendix C, respectively.

## 4.2 Baseline Model Development and Performance

### 4.2.1 Structured-Only Baseline Models

To establish baseline model performance metrics, such models were trained using only structured clinical features. The ten algorithms described in 3.2.1 Model Families were implemented to find the best parameter configuration using restricted grid searches. Performance was evaluated across folds, using additional descriptive internal cross-validation, and using the best parameter combination from repeated k-fold cross-validation, the models were transformed on the holdout test set for classifier comparison. Table 4 summarizes holdout test set scores as AUROC and mean cross validation scores as Mean CV AUROC.

Table 4: Structured-Only Baseline Model Performance (Non-SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| CatBoost | 0.734 | 0.718 | 0.020 | 0.156 | 0.773 | 0.545 | 0.124 | 0.203 | 0.147 |
| XGBoost | 0.729 | 0.717 | 0.019 | 0.158 | 0.773 | 0.536 | 0.124 | 0.202 | 0.147 |
| Gradient Boosting | 0.727 | 0.717 | 0.020 | 0.158 | 0.773 | 0.551 | 0.112 | 0.186 | 0.133 |
| Random Forest | 0.726 | 0.720 | 0.021 | 0.159 | 0.778 | 0.917 | 0.046 | 0.087 | 0.056 |
| LightGBM | 0.726 | 0.706 | 0.017 | 0.160 | 0.766 | 0.483 | 0.174 | 0.256 | 0.200 |
| Logistic Regression | 0.723 | 0.706 | 0.018 | 0.160 | 0.767 | 0.483 | 0.120 | 0.193 | 0.142 |
| Naïve Bayes | 0.701 | 0.680 | 0.020 | 0.213 | 0.722 | 0.407 | 0.444 | 0.425 | 0.436 |
| Neural Network (MLP) | 0.695 | 0.670 | 0.022 | 0.164 | 0.761 | 0.433 | 0.108 | 0.173 | 0.127 |
| Support Vector Machine | 0.688 | 0.664 | 0.019 | 0.164 | 0.774 | 0.579 | 0.091 | 0.158 | 0.110 |
| Decision Tree | 0.635 | 0.643 | 0.023 | 0.173 | 0.768 | 0.492 | 0.129 | 0.204 | 0.151 |

### 4.2.2 Structured + Word2Vec Baseline Models

To learn the initial impact of integrating unstructured clinical radiology text data on model performance, the same classifiers were retrained using the baseline structured features combined with the baseline Word2Vec embeddings derived from the text data. These embeddings were created from the default configuration outlined in 2.4.3 Preliminary Embedding Parameters, so they were not yet optimized for window size, architecture type, or dimensionality. Table 5 shows performance for the structured + Word2Vec baseline models.

Table 5: Structured + Word2Vec Baseline Model Performance (Non-SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| CatBoost | 0.757 | 0.744 | 0.018 | 0.153 | 0.775 | 0.561 | 0.133 | 0.215 | 0.157 |
| Gradient Boosting | 0.753 | 0.738 | 0.018 | 0.154 | 0.773 | 0.551 | 0.112 | 0.186 | 0.133 |
| Logistic Regression | 0.752 | 0.736 | 0.017 | 0.153 | 0.781 | 0.587 | 0.183 | 0.279 | 0.212 |
| XGBoost | 0.752 | 0.744 | 0.017 | 0.161 | 0.773 | 0.532 | 0.137 | 0.218 | 0.161 |
| LightGBM | 0.750 | 0.743 | 0.018 | 0.155 | 0.777 | 0.565 | 0.162 | 0.252 | 0.189 |
| Neural Network (MLP) | 0.735 | 0.714 | 0.026 | 0.157 | 0.773 | 0.523 | 0.191 | 0.280 | 0.219 |
| Random Forest | 0.733 | 0.728 | 0.020 | 0.159 | 0.774 | 0.875 | 0.029 | 0.056 | 0.036 |
| Support Vector Machine | 0.731 | 0.711 | 0.017 | 0.157 | 0.773 | 0.564 | 0.091 | 0.157 | 0.110 |
| Naïve Bayes | 0.673 | 0.664 | 0.018 | 0.338 | 0.623 | 0.333 | 0.631 | 0.436 | 0.535 |
| Decision Tree | 0.598 | 0.630 | 0.024 | 0.182 | 0.757 | 0.357 | 0.062 | 0.106 | 0.075 |

### 4.2.3 Comparison of Structured and Multimodal Baseline Models

Across classifiers, the integration of Word2Vec text embeddings generally led to small but consistent improvements in AUROC score, as seen in the delta-performance table, Table 6, verified through consistent increases in Mean CV scores averaged over validation folds. SVC and MLP showed the greatest increased discrimination power with near 4% increases, and only Naïve Bayes and Decision Tree showed drops in AUROC. This was not without sacrifice in precision or recall as most classifiers have stable increases in $\Delta F_2$ or modest decreases.

Table 6: Performance Changes ($\Delta$) Between Structured-Only and Structured + Word2Vec Baseline Models (Non-SMOTE).

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Support Vector Machine | 0.043 | 0.047 | -0.002 | -0.007 | -0.001 | -0.015 | 0 | -0.001 | 0 |
| Neural Network (MLP) | 0.04 | 0.044 | 0.004 | -0.007 | 0.012 | 0.09 | 0.083 | 0.107 | 0.092 |
| Logistic Regression | 0.029 | 0.03 | -0.001 | -0.007 | 0.014 | 0.104 | 0.063 | 0.086 | 0.07 |
| Gradient Boosting | 0.026 | 0.021 | -0.002 | -0.004 | 0 | 0 | 0 | 0 | 0 |
| LightGBM | 0.024 | 0.037 | 0.001 | -0.005 | 0.011 | 0.082 | -0.012 | -0.004 | -0.011 |
| CatBoost | 0.023 | 0.026 | -0.002 | -0.003 | 0.002 | 0.016 | 0.009 | 0.012 | 0.01 |
| XGBoost | 0.023 | 0.027 | -0.002 | 0.003 | 0 | -0.004 | 0.013 | 0.016 | 0.014 |
| Random Forest | 0.007 | 0.008 | -0.001 | 0 | -0.004 | -0.042 | -0.017 | -0.031 | -0.02 |
| Naïve Bayes | -0.028 | -0.016 | -0.002 | 0.125 | -0.099 | -0.074 | 0.187 | 0.011 | 0.099 |
| Decision Tree | -0.037 | -0.013 | 0.001 | 0.009 | -0.011 | -0.135 | -0.067 | -0.098 | -0.076 |

### 4.2.4 Baseline Model Performance with SMOTE Resampling

Resampling with SMOTE was applied to both the baseline structured-only and structured + Word2Vec full training datasets to address the class imbalance nature of sepsis mortality data. After the best hyperparameter configurations were established for each non-SMOTE baseline model, these configurations were used to refit a model once on these SMOTE training sets, and then transformed on the holdout data.

Table 7 and Table 8 show model performances using the SMOTE retrained models. Mean CV AUROC scores tend to have much higher scores than in the non-SMOTE model performance tables, while the holdout test scores shrink slightly.

It is important to highlight that the Mean CV AUROC scores were evaluated across hundreds of fits per classifier, using 5-fold CV repeated 10 times on 30-50 epochs per classifier or large range grid searches. Consistent scores in the high 0.90s for boosting models indicate discriminatory power at a possible cost of overfitting the training data, with significant drops in AUROC from the CV mean score to the holdout test score.

Table 7: Structured-Only Baseline Model Performance (SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| CatBoost | 0.725 | 0.908 | 0.009 | 0.167 | 0.748 | 0.434 | 0.299 | 0.354 | 0.319 |
| Logistic Regression | 0.716 | 0.724 | 0.016 | 0.222 | 0.640 | 0.357 | 0.697 | 0.473 | 0.586 |
| XGBoost | 0.713 | 0.911 | 0.008 | 0.176 | 0.726 | 0.406 | 0.403 | 0.404 | 0.403 |
| LightGBM | 0.710 | 0.929 | 0.005 | 0.168 | 0.757 | 0.462 | 0.299 | 0.363 | 0.321 |
| Gradient Boosting | 0.703 | 0.849 | 0.015 | 0.198 | 0.682 | 0.377 | 0.573 | 0.455 | 0.519 |
| Random Forest | 0.702 | 0.890 | 0.013 | 0.193 | 0.700 | 0.385 | 0.498 | 0.434 | 0.470 |
| Support Vector Machine | 0.687 | 0.868 | 0.012 | 0.204 | 0.693 | 0.368 | 0.456 | 0.407 | 0.435 |
| Naïve Bayes | 0.682 | 0.703 | 0.018 | 0.312 | 0.586 | 0.323 | 0.718 | 0.445 | 0.577 |
| Neural Network (MLP) | 0.670 | 0.809 | 0.041 | 0.220 | 0.667 | 0.347 | 0.498 | 0.409 | 0.458 |
| Decision Tree | 0.630 | 0.727 | 0.013 | 0.220 | 0.663 | 0.335 | 0.465 | 0.390 | 0.431 |

Table 8: Structured + Word2Vec Baseline Model Performance (SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.748 | 0.779 | 0.013 | 0.206 | 0.665 | 0.374 | 0.664 | 0.478 | 0.575 |
| XGBoost | 0.742 | 0.953 | 0.005 | 0.164 | 0.763 | 0.475 | 0.232 | 0.312 | 0.259 |
| CatBoost | 0.741 | 0.941 | 0.008 | 0.159 | 0.759 | 0.465 | 0.274 | 0.345 | 0.298 |
| LightGBM | 0.737 | 0.946 | 0.006 | 0.161 | 0.758 | 0.455 | 0.232 | 0.308 | 0.258 |
| Support Vector Machine | 0.734 | 0.775 | 0.013 | 0.209 | 0.662 | 0.371 | 0.664 | 0.476 | 0.574 |
| Gradient Boosting | 0.721 | 0.917 | 0.010 | 0.177 | 0.713 | 0.389 | 0.419 | 0.403 | 0.413 |
| Random Forest | 0.713 | 0.931 | 0.011 | 0.182 | 0.716 | 0.387 | 0.390 | 0.388 | 0.389 |
| Neural Network (MLP) | 0.706 | 0.903 | 0.009 | 0.219 | 0.719 | 0.394 | 0.403 | 0.398 | 0.401 |
| Naïve Bayes | 0.649 | 0.722 | 0.013 | 0.395 | 0.578 | 0.305 | 0.647 | 0.415 | 0.529 |
| Decision Tree | 0.636 | 0.778 | 0.010 | 0.206 | 0.649 | 0.334 | 0.523 | 0.408 | 0.470 |

### 4.2.5 Comparison of Structured and Multimodal SMOTE Baseline Models

Table 9 shows stable increases in AUROC across all classifiers except for Naïve Bayes, with score improvements in similar ranges to that of the non-SMOTE baseline models. Accuracy and precision tended to increase from baseline structured + Word2Vec SMOTE models compared to baseline structured-only SMOTE. Recall decreased, leading to decreased $\Delta F_2$ across most classifiers. This indicates that including the Word2Vec features led to improvement in reducing false positive predictions at the expense of a drop in sensitivity.

Table 9: Performance Changes ($\Delta$) Between Structured-Only and Structured + Word2Vec Baseline Models (SMOTE), Ordered by $\Delta$AUROC.

| Model | $\Delta$AUROC | $\Delta$Mean CV AUROC | $\Delta$SD CV AUROC | $\Delta$Brier Score | $\Delta$Accuracy | $\Delta$Precision | $\Delta$Recall | $\Delta$F1 | $\Delta$F2 |
|---|---|---|---|---|---|---|---|---|---|
| Support Vector Machine | 0.047 | -0.093 | 0.001 | 0.005 | -0.031 | 0.003 | 0.208 | 0.069 | 0.139 |
| Neural Network (MLP) | 0.036 | 0.094 | -0.032 | -0.001 | 0.052 | 0.047 | -0.095 | -0.011 | -0.057 |
| Logistic Regression | 0.032 | 0.055 | -0.003 | -0.016 | 0.025 | 0.017 | -0.033 | 0.005 | -0.011 |
| XGBoost | 0.029 | 0.042 | -0.003 | -0.012 | 0.037 | 0.069 | -0.171 | -0.092 | -0.144 |
| LightGBM | 0.027 | 0.017 | 0.001 | -0.007 | 0.001 | -0.007 | -0.067 | -0.055 | -0.063 |
| Gradient Boosting | 0.018 | 0.068 | -0.005 | -0.021 | 0.031 | 0.012 | -0.154 | -0.052 | -0.106 |
| CatBoost | 0.016 | 0.033 | -0.001 | -0.008 | 0.011 | 0.031 | -0.025 | -0.009 | -0.021 |
| Random Forest | 0.011 | 0.041 | -0.002 | -0.011 | 0.016 | 0.002 | -0.108 | -0.046 | -0.081 |
| Decision Tree | 0.006 | 0.051 | -0.003 | -0.014 | -0.014 | -0.001 | 0.058 | 0.018 | 0.039 |
| Naïve Bayes | -0.033 | 0.019 | -0.005 | 0.083 | -0.008 | -0.018 | -0.071 | -0.03 | -0.048 |

### 4.2.6 Comparison of Resampling Strategies

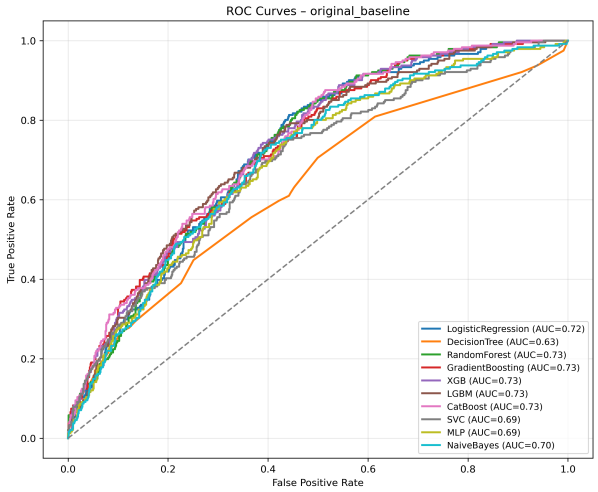
To assess the overall impact of SMOTE rebalancing on baseline models, Table 10 and Table 11 are provided. With increases in $F_1$ and $F_2$ scores across both structured-only and structured + Word2Vec model variants, the results suggest that SMOTE models provide significantly more stability than their non-SMOTE counterparts. This comes at the sacrifice of a few percentage points in AUROC and slight drops in accuracy and precision, with substantial gains in recall. Baseline ROC and Precision-Recall curves are provided next to show differences in AUROC and AUPRC between the four baseline variants.

Table 10: Performance Changes (Δ) Between Structured-Only Baseline Models with and without SMOTE, Ordered by ΔAUROC.

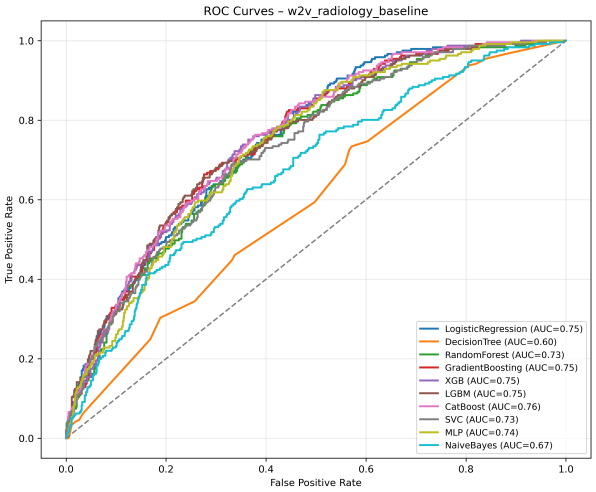
| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Support Vector Machine | -0.001 | 0.204 | -0.007 | 0.04 | -0.081 | -0.211 | 0.365 | 0.249 | 0.325 |
| Decision Tree | -0.005 | 0.084 | -0.01 | 0.047 | -0.105 | -0.157 | 0.336 | 0.186 | 0.28 |
| Logistic Regression | -0.007 | 0.018 | -0.002 | 0.062 | -0.127 | -0.126 | 0.577 | 0.28 | 0.444 |
| CatBoost | -0.009 | 0.19 | -0.011 | 0.011 | -0.025 | -0.111 | 0.175 | 0.151 | 0.172 |
| LightGBM | -0.016 | 0.223 | -0.012 | 0.008 | -0.009 | -0.021 | 0.125 | 0.107 | 0.121 |
| XGBoost | -0.016 | 0.194 | -0.011 | 0.018 | -0.047 | -0.13 | 0.279 | 0.202 | 0.256 |
| Naïve Bayes | -0.019 | 0.023 | -0.002 | 0.099 | -0.136 | -0.084 | 0.274 | 0.02 | 0.141 |
| Gradient Boosting | -0.024 | 0.132 | -0.005 | 0.04 | -0.091 | -0.174 | 0.461 | 0.269 | 0.386 |
| Random Forest | -0.024 | 0.17 | -0.008 | 0.034 | -0.078 | -0.532 | 0.452 | 0.347 | 0.414 |
| Neural Network (MLP) | -0.025 | 0.139 | 0.019 | 0.056 | -0.094 | -0.086 | 0.39 | 0.236 | 0.331 |

Table 11: Performance Changes (Δ) Between Structured + Word2Vec Baseline Models with and without SMOTE, Ordered by ΔAUROC.

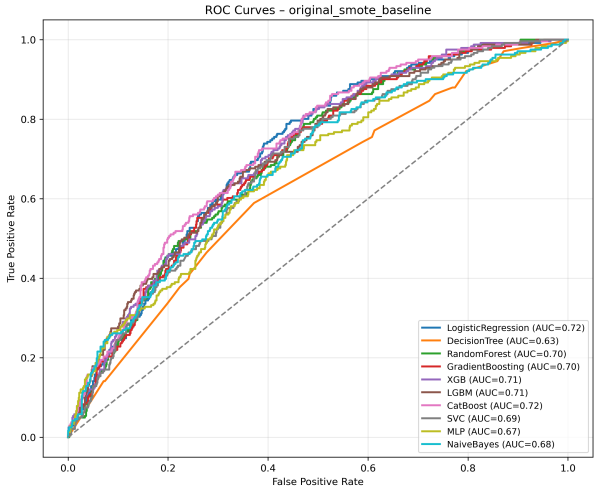
| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Decision Tree | 0.038 | 0.148 | -0.014 | 0.024 | -0.108 | -0.023 | 0.461 | 0.302 | 0.395 |
| Support Vector Machine | 0.003 | 0.064 | -0.004 | 0.052 | -0.111 | -0.193 | 0.573 | 0.319 | 0.464 |
| Logistic Regression | -0.004 | 0.043 | -0.004 | 0.053 | -0.116 | -0.213 | 0.481 | 0.199 | 0.363 |
| XGBoost | -0.01 | 0.209 | -0.012 | 0.003 | -0.01 | -0.057 | 0.095 | 0.094 | 0.098 |
| LightGBM | -0.013 | 0.203 | -0.012 | 0.006 | -0.019 | -0.11 | 0.07 | 0.056 | 0.069 |
| CatBoost | -0.016 | 0.197 | -0.01 | 0.006 | -0.016 | -0.096 | 0.141 | 0.13 | 0.141 |
| Random Forest | -0.02 | 0.203 | -0.009 | 0.023 | -0.058 | -0.488 | 0.361 | 0.332 | 0.353 |
| Naïve Bayes | -0.024 | 0.058 | -0.005 | 0.057 | -0.045 | -0.028 | 0.016 | -0.021 | -0.006 |
| Neural Network (MLP) | -0.029 | 0.189 | -0.017 | 0.062 | -0.054 | -0.129 | 0.212 | 0.118 | 0.182 |
| Gradient Boosting | -0.032 | 0.179 | -0.008 | 0.023 | -0.06 | -0.162 | 0.307 | 0.217 | 0.28 |



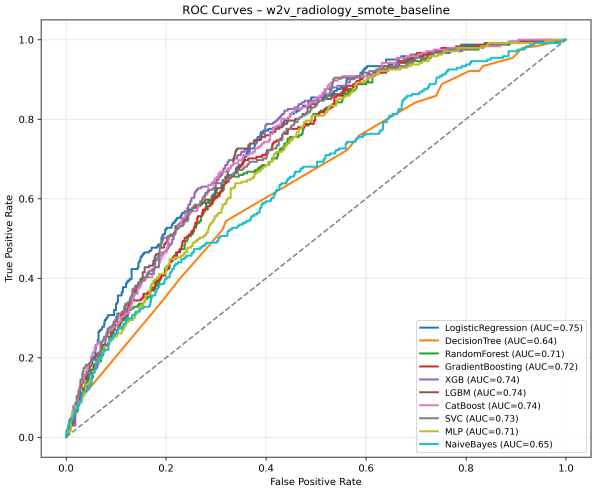

(a) Structured-Only (Non-SMOTE)




(b) Structured + Word2Vec (Non-SMOTE)

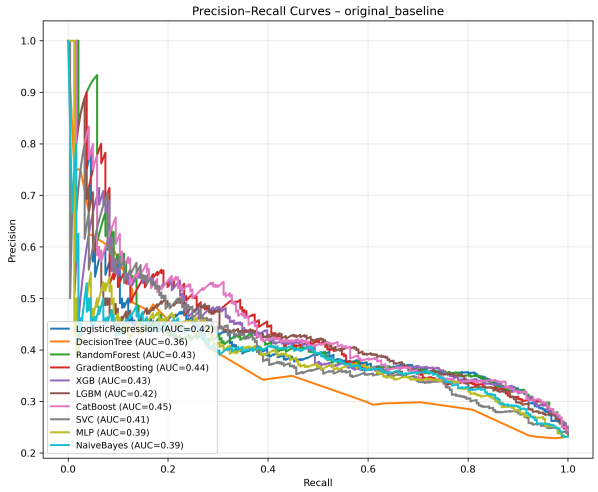



(c) Structured-Only (SMOTE)

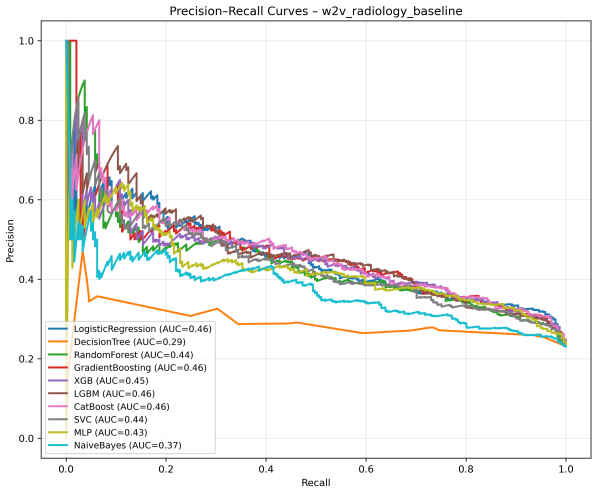



(d) Structured + Word2Vec (SMOTE)

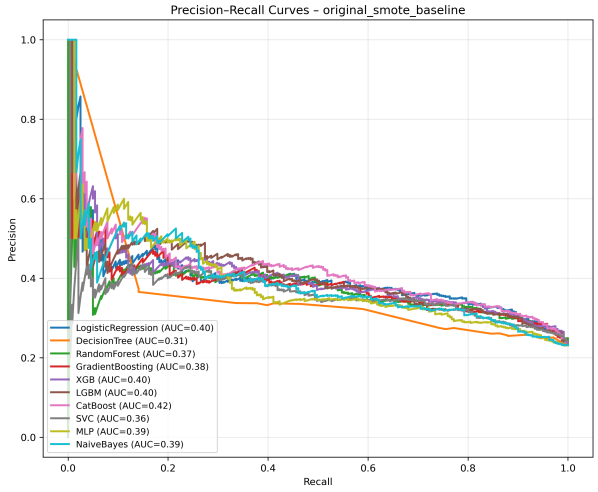
Figure 2: ROC Curves for Structured-Only and Structured + Word2Vec Baseline Models, with and without SMOTE Resampling.



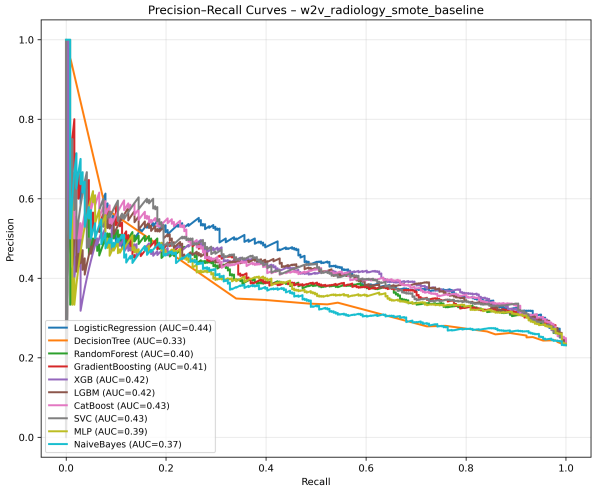

(a) Structured-Only (Non-SMOTE)




(b) Structured + Word2Vec (Non-SMOTE)




(c) Structured-Only (SMOTE)




(d) Structured + Word2Vec (SMOTE)

Figure 3: Precision–Recall Curves for Structured and Structured + Word2Vec Baseline Models, with and without SMOTE Resampling.

ROC and PR curves will be discussed in more detail in 4.5.2 ROC and PR Curves. See Appendix A Codebase under results/figures for additional baseline figures such as SHAP analysis, feature importance, and calibration plots

### 4.2.7 Top Selected Classifiers for Optimized Training

The top 5 baseline classifiers based on non-SMOTE and SMOTE variants for the structured + Word2Vec models were selected for additional, optimized training to search across larger grid spaces for fine-tuned parameter searches. A subset was chosen to reduce training time, and SVC was left out due to computational complexity associated with Platt scaling and convergence, as it did not provide immense increases in AUROC. These top 5 classifiers were CatBoost, Logistic Regression, LGBM, Gradient Boosting, and XGB. Random Forest was also provided as a robust comparison model, as most of these selected classifiers use boosting methods. In total, these 6 classifiers were re-tuned and trained on the embedding space discussed next in 4.3 Word2Vec Embedding Optimization.

## 4.3 Word2Vec Embedding Optimization

This section discusses how the Word2Vec embeddings were further optimized according to cosine similarity scores, to provide better word-vectors that capture semantic meaning across text tokens, per patient, in a formalized way compared to the preliminary selected tuning parameters.

### 4.3.1 Intrinsic Evaluation of Embeddings

The intrinsic plot, Figure 4, highlights that certain Word2Vec tuning combinations provided higher mean cosine similarity scores than others. For example, sg = 0 corresponded to the CBOW architecture while sg = 1 corresponded to the skip-gram architecture. The skip-gram architecture produced significantly higher mean cosine similarity scores than the CBOW architecture. This signifies that this step was extremely important for optimizing the embedding space according to mean cosine similarity, especially since the original baseline models used the CBOW architecture. The vector size of 100 had better performance than vector size of 200, and the larger window size of 10 tended to give better scores than the size 5 counterpart. Since the notes were collapsed into one note per patient, this makes a lot of

sense as the tokens were extremely long, and so having a larger window to search through allowed for retaining semantic meaning for each patient. The subtle remaining variations are fluctuations based on the remaining minimum count, negative, and epochs variables, each contributing to minor changes in the cosine similarity score.

This evaluation considered vocabulary coverage by incorporating minimum counts of vocabulary as a hedge for rare words. This allowed consideration of unique words, but implemented a minimum frequency requirement inside the sentence token. A minimum frequency of 2 or 5 was considered, with minimum counts of 2 tending to lead to higher cosine similarity scores compared to the more restrictive minimum of 5, indicating that certain low-usage vocabulary words played important roles in semantic cohesion across embeddings. Radiology notes tend to be relatively short and technical, and so these low-usage lexical terms can be important to providing insight into the prognosis of a patient, thereby improving cohesion as opposed to adding noise.

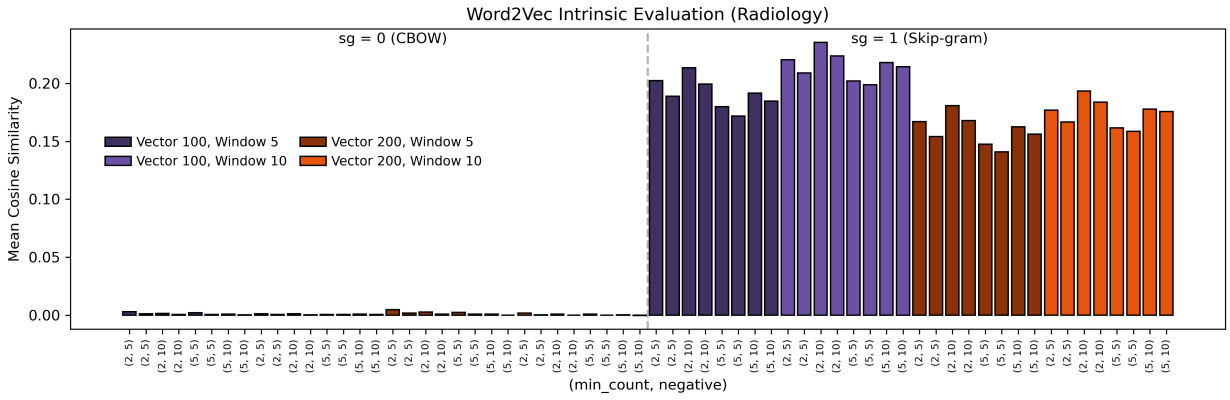



Figure 4: Intrinsic Evaluation of Cosine Similarity Scores for Word2Vec Embedding Optimization. Each Bar Represents a Word2Vec Parameter Configuration Using a Subset of the Word2Vec Parameters Used in Hyperparameter Tuning.

### 4.3.2 Selected Optimal Embedding Configuration

The optimal parameter combination provided the highest cosine similarity, with a score of 0.2353. This parameter combination is described in Table 12. The final optimized configuration balances each of these parameter effects, with a smaller embedding space of 100, a larger window to search through, and allowing for rare vocabulary words to impact semantic cohesion. Thus creating the best balance for semantic similarity across vocabulary, and ensuring comprehensive lingual representation in each tokenized sentence.

Table 12: Final Optimized Word2Vec Parameter Configuration for Radiology Note Embeddings.

| **Parameter** | **Selected Value** |
|---|---|
| `vector_size` | 100 |
| `window` | 10 |
| `min_count` | 2 |
| `sg` | 1 (skip-gram model) |
| `negative` | 10 |
| `epochs` | 15 |
| `workers` | 12 |

## 4.4 Optimized Word2Vec Multimodal Performance and Evaluation

### 4.4.1 Classifier Re-Tuning on Optimized Embeddings

To address the core hypotheses of the study, the 6 classifiers outlined in 4.2.7 Top Selected Classifiers for Optimized Training were re-tuned and retrained on the optimized Word2Vec embedding feature space since a different embedding space was generated by the optimized Word2Vec training scheme. Expanded grid search parameter spaces were provided to further fine-tune the best selected models from the baseline assessment. After reconfiguring the

hyperparameters for each of the classifiers over the optimized embedding space, the final Word2Vec-optimized models produced AUROC scores using the same coding logic from baseline. The best parameter combinations for all classifiers are found in Table 13, and the full hyperparameter grid search spaces for each classifier can be found in Appendix A Codebase under src/models.py in `get_param_distributions`. The same validation and scoring metrics strategy was implemented as described in 3.3 Evaluation Framework.

Table 13: Optimized Hyperparameters for Each Classifier for Optimized Word2Vec Multimodal Model Training.

**CatBoost**

| Parameter | Value |
|---|---|
| Depth | 8 |
| Iterations | 858 |
| L2 leaf reg | 4 |
| Learning rate | 0.017 |

**Logistic Regression**

| Parameter | Value |
|---|---|
| C | 0.1 |
| L1 ratio | 0 |
| Max iter | 1000 |
| Penalty | l1 |
| Solver | saga |

**Random Forest**

| Parameter | Value |
|---|---|
| Bootstrap | 0 |
| Max depth | 20 |
| Max features | $\lfloor\sqrt{143}\rfloor$ |
| Min samples leaf | 4 |
| Min samples split | 5 |
| N estimators | 933 |

**XGBoost**

| Parameter | Value |
|---|---|
| Colsample bytree | 0.945 |
| Gamma | 0.43 |
| Learning rate | 0.012 |
| Max depth | 10 |
| Min child weight | 3 |
| N estimators | 848 |
| Subsample | 0.845 |

**LGBM**

| Parameter | Value |
|---|---|
| Colsample bytree | 0.788 |
| Learning rate | 0.014 |
| Max depth | 20 |
| Min child samples | 90 |
| N estimators | 591 |
| Num leaves | 193 |
| Subsample | 0.937 |

**Gradient Boosting**

| Parameter | Value |
|---|---|
| Learning rate | 0.029 |
| Max depth | 5 |
| Max features | log2 |
| Min samples leaf | 3 |
| Min samples split | 3 |
| N estimators | 486 |
| Subsample | 0.79 |

These highly tuned models provided very low uncertainty values in cross-validated mean AUROC scores, giving high confidence that the selected parameter combinations were not

unduly influenced by noise. The following sections demonstrate the robustness of the analysis with confidence in evaluation metrics.

### 4.4.2 Optimized Multimodal Performance (Non-SMOTE)

Table 14 demonstrates a jump in AUROC scores with Logistic Regression and the boosting methods, with Random Forest demonstrating a modest score as well. These performance metrics are compared to baseline through Δscores in Table 15. Both ΔAUROC and ΔMean CV AUROC scores increase in range from 1% to 3.9% over each optimized classifier. Precision and recall generally increase, demonstrating relative stability of the models over the baseline structured-only models. Brier scores tend to drop slightly, while accuracy slightly increases.

Table 14: Optimized Word2Vec Multimodal Model Performance (Non-SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.757 | 0.745 | 0.018 | 0.153 | 0.776 | 0.550 | 0.183 | 0.274 | 0.211 |
| CatBoost | 0.756 | 0.748 | 0.018 | 0.154 | 0.783 | 0.703 | 0.108 | 0.187 | 0.130 |
| Gradient Boosting | 0.752 | 0.746 | 0.019 | 0.154 | 0.775 | 0.549 | 0.162 | 0.250 | 0.188 |
| XGBoost | 0.751 | 0.748 | 0.018 | 0.161 | 0.776 | 0.561 | 0.153 | 0.241 | 0.180 |
| LightGBM | 0.743 | 0.745 | 0.019 | 0.157 | 0.777 | 0.557 | 0.183 | 0.275 | 0.211 |
| Random Forest | 0.736 | 0.733 | 0.021 | 0.158 | 0.777 | 0.909 | 0.042 | 0.079 | 0.051 |

Table 15: Performance Changes (Δ) Between Optimized Word2Vec Multimodal and Structured-Only Models (Non-SMOTE), Ordered by ΔAUROC.

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.034 | 0.039 | 0 | -0.007 | 0.009 | 0.067 | 0.063 | 0.081 | 0.069 |
| Gradient Boosting | 0.025 | 0.029 | -0.001 | -0.004 | 0.002 | -0.002 | 0.05 | 0.064 | 0.055 |
| CatBoost | 0.022 | 0.03 | -0.002 | -0.002 | 0.01 | 0.158 | -0.016 | -0.016 | -0.017 |
| XGBoost | 0.022 | 0.031 | -0.001 | 0.003 | 0.003 | 0.025 | 0.029 | 0.039 | 0.033 |
| LightGBM | 0.017 | 0.039 | 0.002 | -0.003 | 0.011 | 0.074 | 0.009 | 0.019 | 0.011 |
| Random Forest | 0.01 | 0.013 | 0 | -0.001 | -0.001 | -0.008 | -0.004 | -0.008 | -0.005 |

### 4.4.3 Optimized Multimodal Performance (SMOTE)

Next, non-SMOTE optimized Word2Vec multimodal models were retrained on SMOTE training data and transformed on the holdout test set. Each of the SMOTE models demonstrated in Table 16 and Table 17 used the same hyperparameter configurations as outlined in Table 13. The corresponding Δperformance metrics table is provided in Table 17. There is a slight drop in AUROC scores, but other metrics tend to display much higher scores, as will be demonstrated in Table 20 over non-SMOTE counterparts.

Table 16: Optimized Word2Vec Multimodal Model Performance (SMOTE).

| Model | AUROC | Mean CV AUROC | SD CV AUROC | Brier Score | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.753 | 0.779 | 0.011 | 0.203 | 0.681 | 0.393 | 0.697 | 0.503 | 0.604 |
| XGBoost | 0.752 | 0.945 | 0.008 | 0.161 | 0.771 | 0.510 | 0.220 | 0.307 | 0.248 |
| LightGBM | 0.751 | 0.939 | 0.007 | 0.156 | 0.781 | 0.555 | 0.274 | 0.367 | 0.305 |
| Random Forest | 0.737 | 0.948 | 0.008 | 0.164 | 0.778 | 0.547 | 0.241 | 0.334 | 0.271 |
| CatBoost | 0.736 | 0.945 | 0.006 | 0.157 | 0.779 | 0.551 | 0.245 | 0.339 | 0.275 |
| Gradient Boosting | 0.734 | 0.935 | 0.008 | 0.161 | 0.757 | 0.460 | 0.286 | 0.353 | 0.310 |

Table 17: Performance Changes (Δ) Between Optimized Word2Vec Multimodal and Structured-Only Models (SMOTE), Ordered by ΔAUROC.

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| LightGBM | 0.041 | 0.01 | 0.002 | -0.012 | 0.024 | 0.093 | -0.025 | 0.004 | -0.016 |
| XGBoost | 0.039 | 0.034 | 0 | -0.015 | 0.045 | 0.104 | -0.183 | -0.097 | -0.155 |
| Logistic Regression | 0.037 | 0.055 | -0.005 | -0.019 | 0.041 | 0.036 | 0 | 0.03 | 0.018 |
| Random Forest | 0.035 | 0.058 | -0.005 | -0.029 | 0.078 | 0.162 | -0.257 | -0.1 | -0.199 |
| Gradient Boosting | 0.031 | 0.086 | -0.007 | -0.037 | 0.075 | 0.083 | -0.287 | -0.102 | -0.209 |
| CatBoost | 0.011 | 0.037 | -0.003 | -0.01 | 0.031 | 0.117 | -0.054 | -0.015 | -0.044 |

### 4.4.4 Comparison of Optimized and Baseline Multimodal Models

As demonstrated below, there is not significant improvements in the optimized Word2Vec multimodal models over baseline Word2Vec multimodal models, and changes in metrics are different from classifier to classifier. Models tended to be about as good or slightly better in holdout ΔAUROC, while ΔMean CV AUROC was mixed. Brier scores had a tendency to drop off while accuracy and precision increased. $\Delta F_1$ and $\Delta F_2$ scores were mixed. This lack of consistency across the board provides more indication that using alternative text batch preprocessing methods could be useful in the future to better diagnose which architecture leads to more consistent Δ score changes.

Table 18: Performance Changes (Δ) Between Baseline Word2Vec and Optimized Word2Vec Multimodal Models (Non-SMOTE), Ordered by ΔAUROC.

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.005 | 0.009 | 0.001 | 0 | -0.005 | -0.037 | 0 | -0.005 | -0.001 |
| Random Forest | 0.003 | 0.005 | 0.001 | -0.001 | 0.003 | 0.034 | 0.013 | 0.023 | 0.015 |
| CatBoost | -0.001 | 0.004 | 0 | 0.001 | 0.008 | 0.142 | -0.025 | -0.028 | -0.027 |
| Gradient Boosting | -0.001 | 0.008 | 0.001 | 0 | 0.002 | -0.002 | 0.05 | 0.064 | 0.055 |
| XGBoost | -0.001 | 0.004 | 0.001 | 0 | 0.003 | 0.029 | 0.016 | 0.023 | 0.019 |
| LightGBM | -0.007 | 0.002 | 0.001 | 0.002 | 0 | -0.008 | 0.021 | 0.023 | 0.022 |

Table 19: Performance Changes (Δ) Between Baseline Word2Vec and Optimized Word2Vec Multimodal Models (SMOTE), Ordered by ΔAUROC.

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| Random Forest | 0.024 | 0.017 | -0.003 | -0.018 | 0.062 | 0.16 | -0.149 | -0.054 | -0.118 |
| LightGBM | 0.014 | -0.007 | 0.001 | -0.005 | 0.023 | 0.1 | 0.042 | 0.059 | 0.047 |
| Gradient Boosting | 0.013 | 0.018 | -0.002 | -0.016 | 0.044 | 0.071 | -0.133 | -0.05 | -0.103 |
| XGBoost | 0.01 | -0.008 | 0.003 | -0.003 | 0.008 | 0.035 | -0.012 | -0.005 | -0.011 |
| Logistic Regression | 0.005 | 0 | -0.002 | -0.003 | 0.016 | 0.019 | 0.033 | 0.025 | 0.029 |
| CatBoost | -0.005 | 0.004 | -0.002 | -0.002 | 0.02 | 0.086 | -0.029 | -0.006 | -0.023 |

### 4.4.5 Comparison of Resampling Strategy for Optimized Multimodal Models

As touched on earlier, while the SMOTE models sacrifice slightly in ΔAUROC they have much higher $F_1$, $F_2$, and recall scores. Additionally, they have significantly higher ΔMean CV AUROC due to their greater ability to discriminate, with access to an equal number of minority samples to majority samples. This shows the power of oversampling for a models ability to discriminate, but also shows the effect of overfitting on precision scores and through drops in ΔAUROC. More on overfitting will be demonstrated in 4.5.4 Calibration Plots and Brier Scores.

The next section, 4.5 Visual Summaries provides visual demonstrations of some of the key results presented in the tables from this section, including ROC and PR curves, SHAP summaries and dependency plots, and calibration curves for each of the top 6 models.

Table 20: Performance Changes (Δ) Between Optimized Word2Vec Multimodal Models with and without SMOTE, Ordered by ΔAUROC.

| Model | ΔAUROC | ΔMean CV AUROC | ΔSD CV AUROC | ΔBrier Score | ΔAccuracy | ΔPrecision | ΔRecall | ΔF1 | ΔF2 |
|---|---|---|---|---|---|---|---|---|---|
| LightGBM | 0.008 | 0.194 | -0.012 | -0.001 | 0.004 | -0.002 | 0.091 | 0.092 | 0.094 |
| Random Forest | 0.001 | 0.215 | -0.013 | 0.006 | 0.001 | -0.362 | 0.199 | 0.255 | 0.22 |
| XGBoost | 0.001 | 0.197 | -0.01 | 0 | -0.005 | -0.051 | 0.067 | 0.066 | 0.068 |
| Logistic Regression | -0.004 | 0.034 | -0.007 | 0.05 | -0.095 | -0.157 | 0.514 | 0.229 | 0.393 |
| Gradient Boosting | -0.018 | 0.189 | -0.011 | 0.007 | -0.018 | -0.089 | 0.124 | 0.103 | 0.122 |
| CatBoost | -0.02 | 0.197 | -0.012 | 0.003 | -0.004 | -0.152 | 0.137 | 0.152 | 0.145 |

## 4.5 Visual Summaries

### 4.5.1 AUROC Comparison Across Variants

Figure 5 is provided to demonstrate the improvements of using Word2Vec embeddings over structured-only features without SMOTE resampling. The Word2Vec variants consistently post higher scores over structured-only models except for Naive Bayes and Decision Tree. The differences in AUROC score tend to be on the order of thousandths of a difference, with the optimized models tending to slightly outperform baseline Word2Vec models. See Figure 16 in Appendix C for the AUROC comparison chart for SMOTE models.

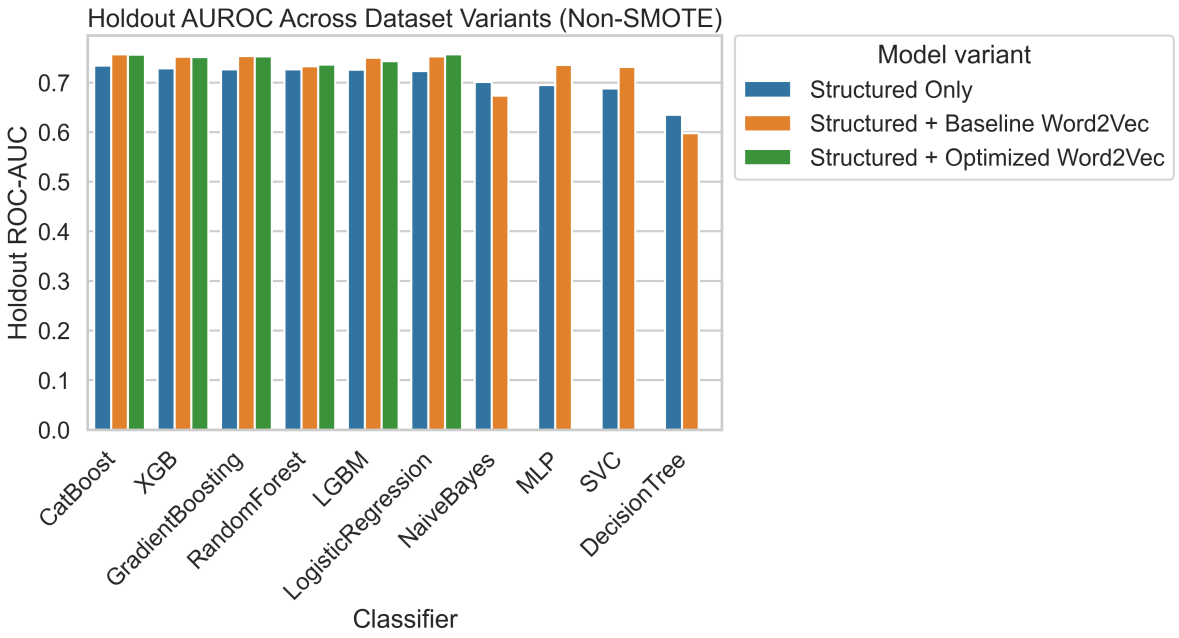
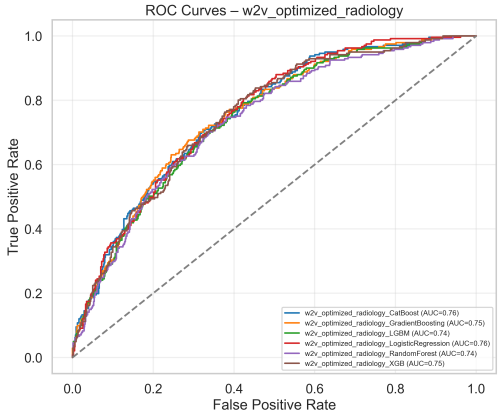


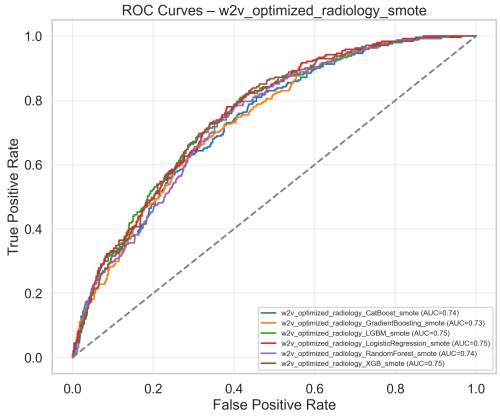

Figure 5: AUROC Comparison of Baseline and Optimized Models (Non-SMOTE).

### 4.5.2 ROC and PR Curves

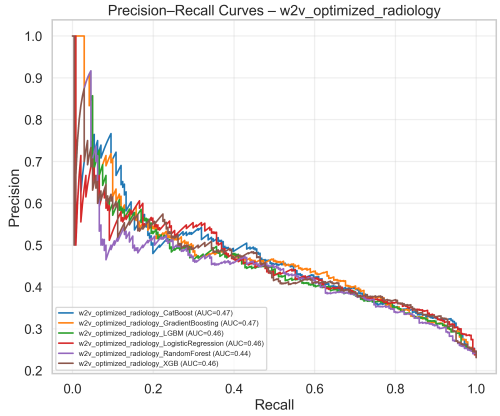
Side-by-side non-SMOTE and SMOTE results are visualized using ROC and PR curves to demonstrate that while non-SMOTE models provide higher AUROC scores, the SMOTE models tend to have higher AUPRC scores, indicating greater discrimination stability.



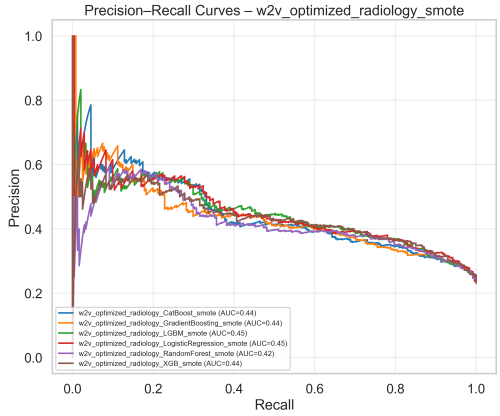

(a) ROC Optimized Word2Vec Multimodal (Non-SMOTE)




(b) ROC Optimized Word2Vec Multimodal (SMOTE)




(c) PR Optimized Word2Vec Multimodal (Non-SMOTE)




(d) PR Optimized Word2Vec Multimodal (SMOTE)

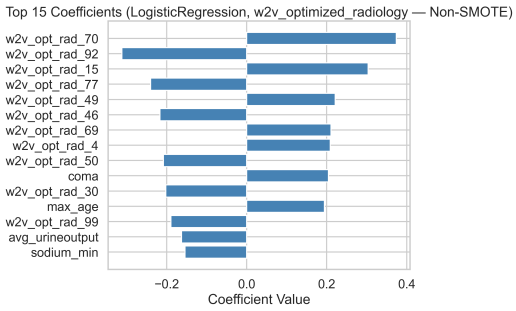
Figure 6: ROC and PR Curves for Word2Vec Multimodal Models, with and without SMOTE.

### 4.5.3 Dependency and SHAP Analysis

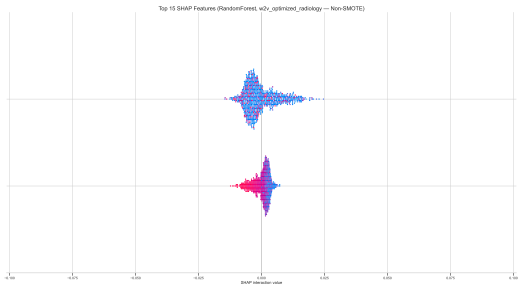
The sub-figures in Figure 7 show the top-ranked feature importances for Logistic Regression and the ensemble boosting algorithms. Many of the Word2Vec features dominate in the feature importance rankings, consistently appearing in the top 15. While there are many more semantic features than structured features, it is important to highlight that the introduction of these word-vectors caused a shift in model interpretability as new features became more important in setting discrimination boundaries compared to baseline.

When looking at the top features in each classifier, w2v_opt_rad_15 continued to play a prominent role. To explore the meanings behind these vectors, clustering techniques can be used to investigate the conceptual and clinically important semantic structures represented by the vectors. Dimensionality reduction methods such as PCA can help visualize how the word-vectors are arranged in reduced space, potentially identifying coherent clusters for further inferential analysis.

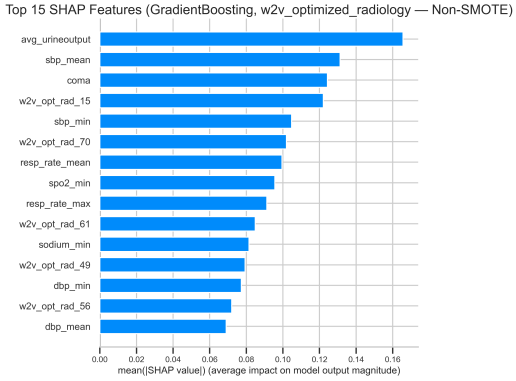
SHAP analysis shows high separation of w2v_opt_rad_15 and w2v_opt_rad_57, see Figure 8. Since feature 57 had the highest Pearson correlation coefficient to feature 15, it was used in these SHAP analyses. Performing additional SHAP analyses can lead to understanding how the embeddings relate to one another and to important features, such as average urine output, shown in Figure 19 in Appendix C. See also Appendix C for additional figures for SHAP and dependency analysis for SMOTE models, Figures 17, and 18.



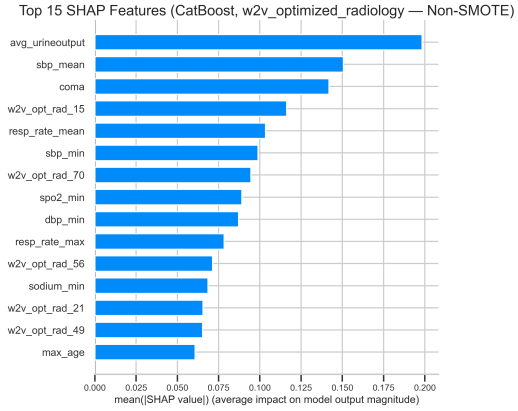

Logistic Regression



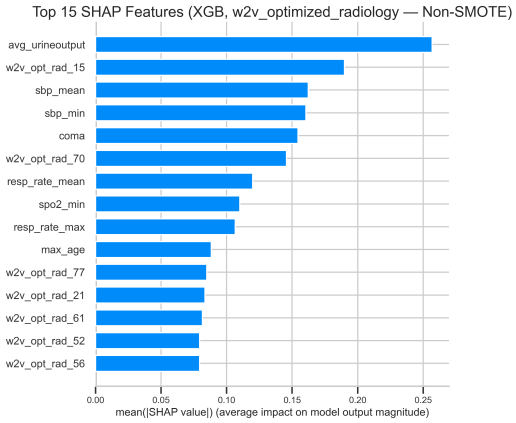
Random Forest



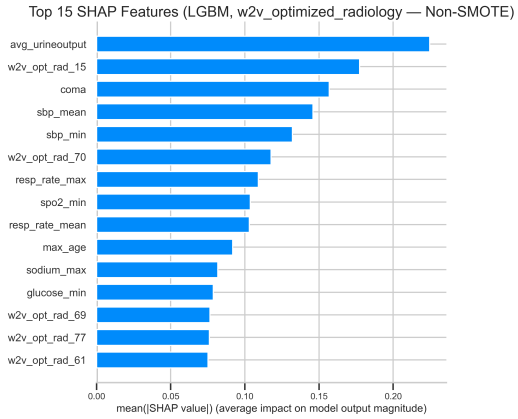

Gradient Boosting




CatBoost

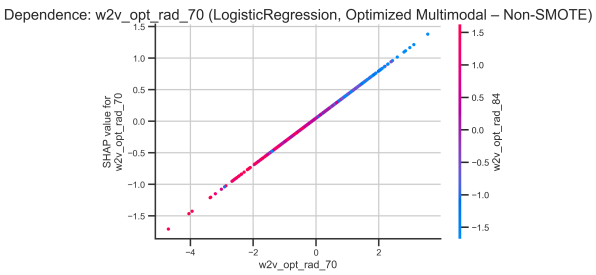



XGBoost

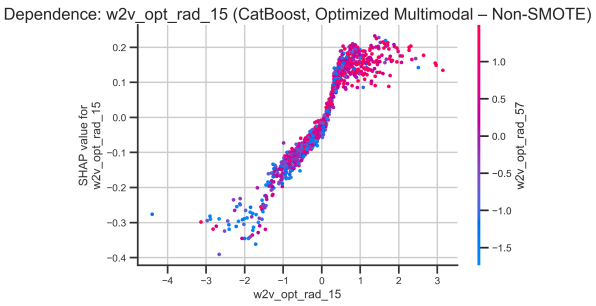



LightGBM

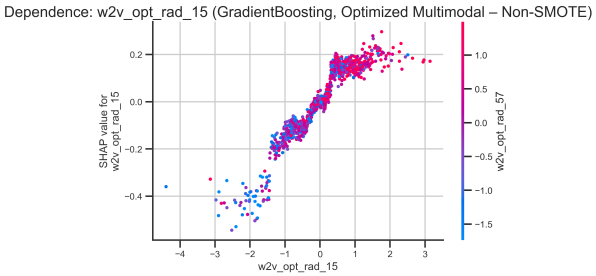
Figure 7: SHAP Summary Plots Illustrating Feature Contributions for Optimized Word2Vec Multimodal Models (Non-SMOTE).



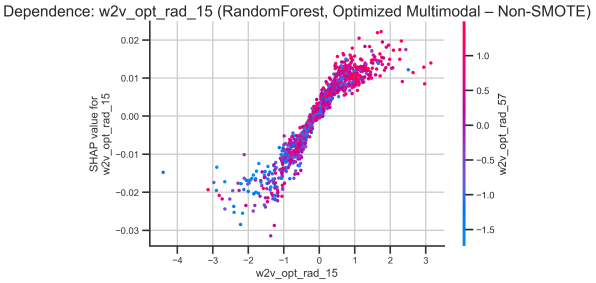

Logistic Regression



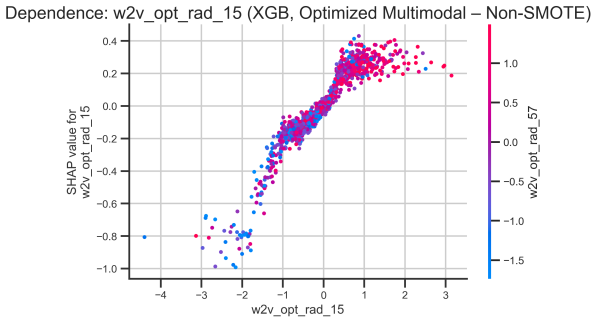

CatBoost



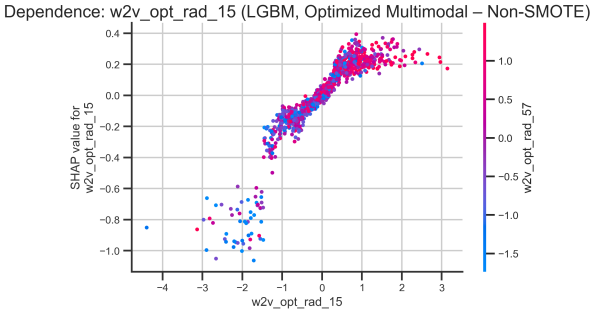

Gradient Boosting




Random Forest




XGBoost




LightGBM

Figure 8: SHAP Dependence Plots Showing the Top Word2Vec Feature Contributions for Optimized Multimodal Models (Non-SMOTE). Each Subplot Illustrates Feature Value Relationships for the Top Embedding-Based Predictors Across Classifiers.

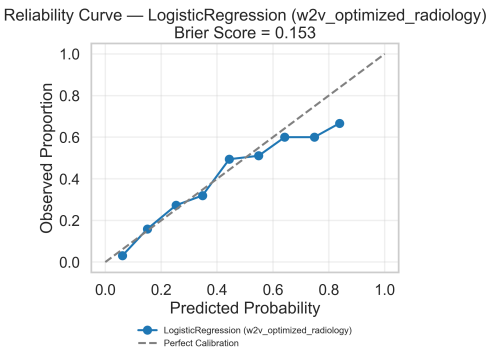
### 4.5.4 Calibration Plots and Brier Scores

Brier scores for each of the non-SMOTE models can be found in Table 14, and for each of the SMOTE models can be found in Table 16. These scores quantitatively assess the mean squared difference between predicted probabilities and observed outcomes. Lower Brier scores lead to higher calibration, meaning that the model fits correctly and the likelihood of overfitting or underfitting is reduced.

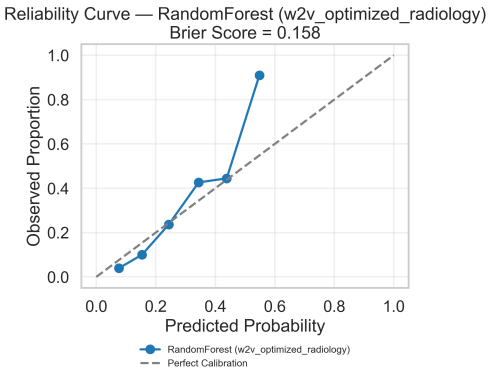
Looking at the non-SMOTE models as seen in Figure 9, at actual probabilities lower than 0.4, the models tend to under-fit the data, while at probabilities between 0.5 and 0.8 they overfit the data, predicting higher mortality outcomes than to what actually occurs. When the actual probability goes toward 1, the models start to under-fit again. The models that had the best calibration appear to be Logistic Regression, Gradient Boosting, and LightGBM with XGBoost and Random Forest providing the lowest well-calibrated Brier scores.

Overall, non-SMOTE models tend to be highly calibrated. On the other hand, the SMOTE models tend to give less ideal calibration curves, especially at highly predicted probabilities, where models tend to overfit the data more as observed probability increases. See Figure 10. This is not visibly reflected too much in the Brier scores, as they tend to be a few thousandths off of the non-SMOTE counterpart models.

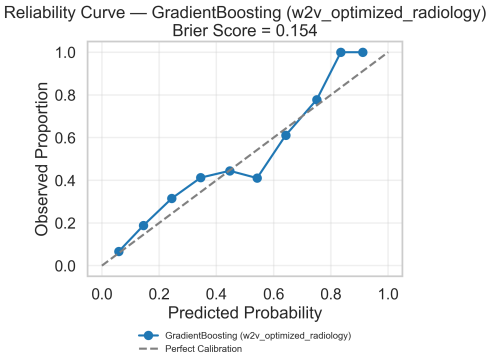
This lack of calibration and overfitting could be influenced by the introduction of the synthetic over-sampled minority data points, causing the models to predict mortality even though the observed probability of mortality is actually quite low. This suggests that while SMOTE models may have higher $F_1$, $F_2$, and recall scores, the predicted probabilities may not always match the actual outcomes, leading to potential model instability when it comes to generalizability.



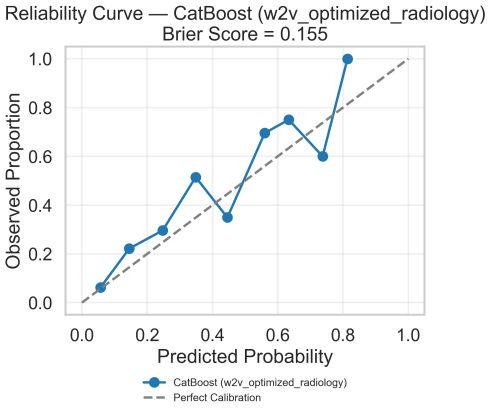

Logistic Regression



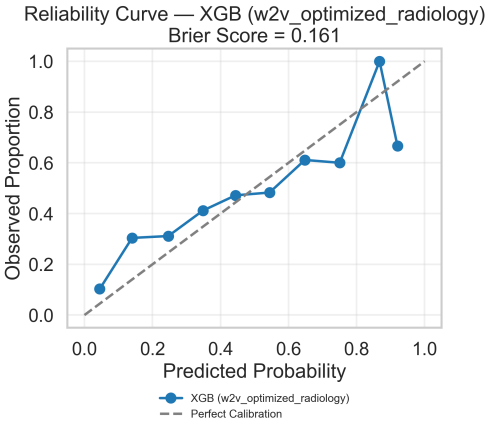

Random Forest



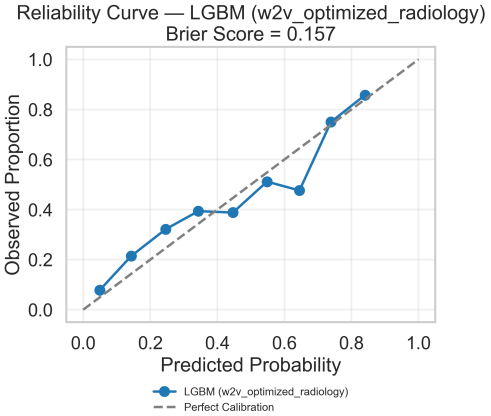

Gradient Boosting




CatBoost

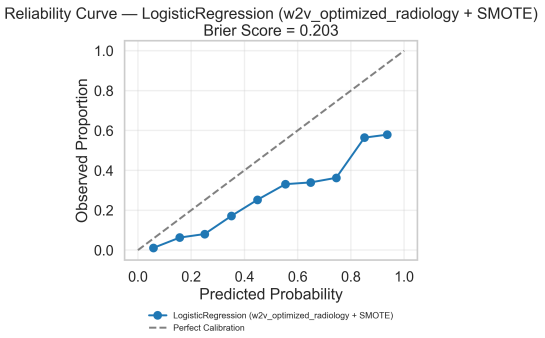



XGBoost

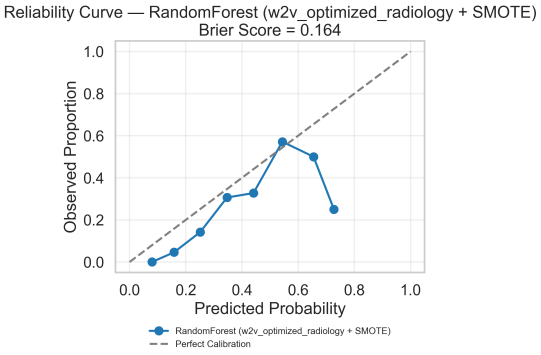



LightGBM

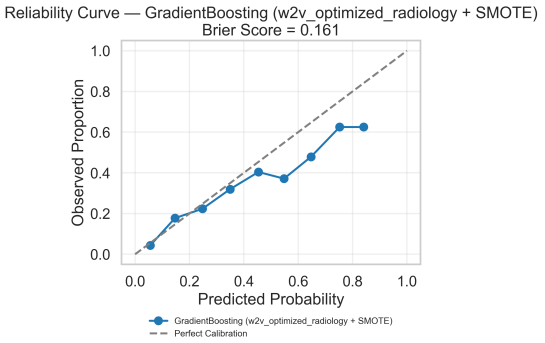
Figure 9: Calibration Plots for Optimized Word2Vec Multimodal Models (Non-SMOTE). Each Subplot Compares Predicted Probabilities with Observed Event Frequencies to Assess Model Calibration.



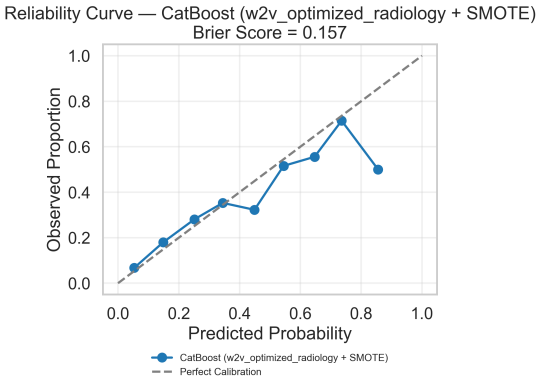

Logistic Regression



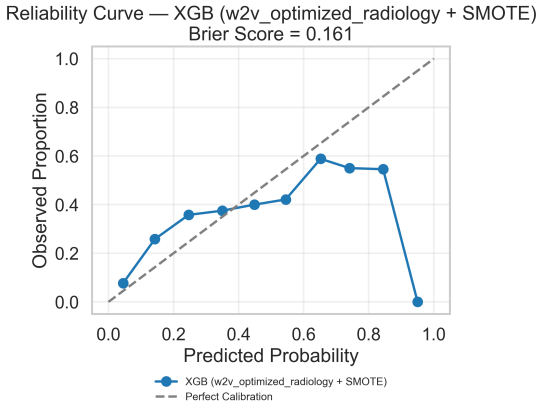

Random Forest



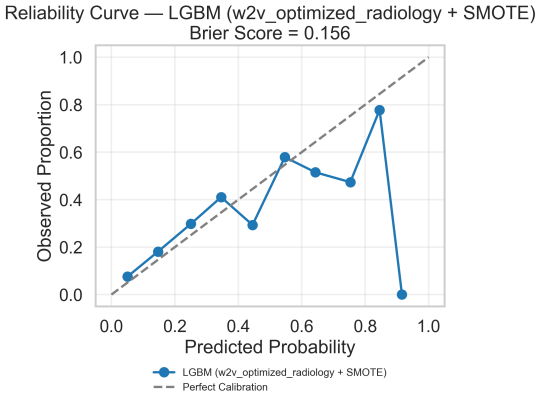

Gradient Boosting




CatBoost




XGBoost




LightGBM

Figure 10: Calibration Plots for Optimized Word2Vec Multimodal Models with SMOTE Resampling. Each Subplot Compares Predicted Probabilities with Observed Event Frequencies to Assess Model Calibration.

### 4.5.5 Clustering Analysis

To understand the word-vectors created by the optimized Word2Vec model, the following unsupervised clustering scheme was performed using K-means and performing two-dimensional principle component analysis projection of optimized Word2vec embedding space for the radiology notes. It answers the question: Do the Word2Vec embeddings meaningfully cluster semantically similar radiology terms?

Vector representations are extracted for every word in the radiology vocabulary accumulated across all tokens, and then grouped into k = 10 clusters based on embedding similarity. Since the embedding dimensionality is 100, it is reduced via PCA to 2 for visualization. Then, semantically related radiology concepts group together, color coded by hierarchical concept mapping.

The resulting graph, Figure 11, demonstrates distinct clusters that capture different coherent semantic groups such as imaging modalities (cluster 0), body-regions (cluster 2), procedural terms (clusters 4 and 9), and common radiologic findings (cluster 6). These clusterings demonstrate the model-learned semantic cohesion for radiologic domain structure.

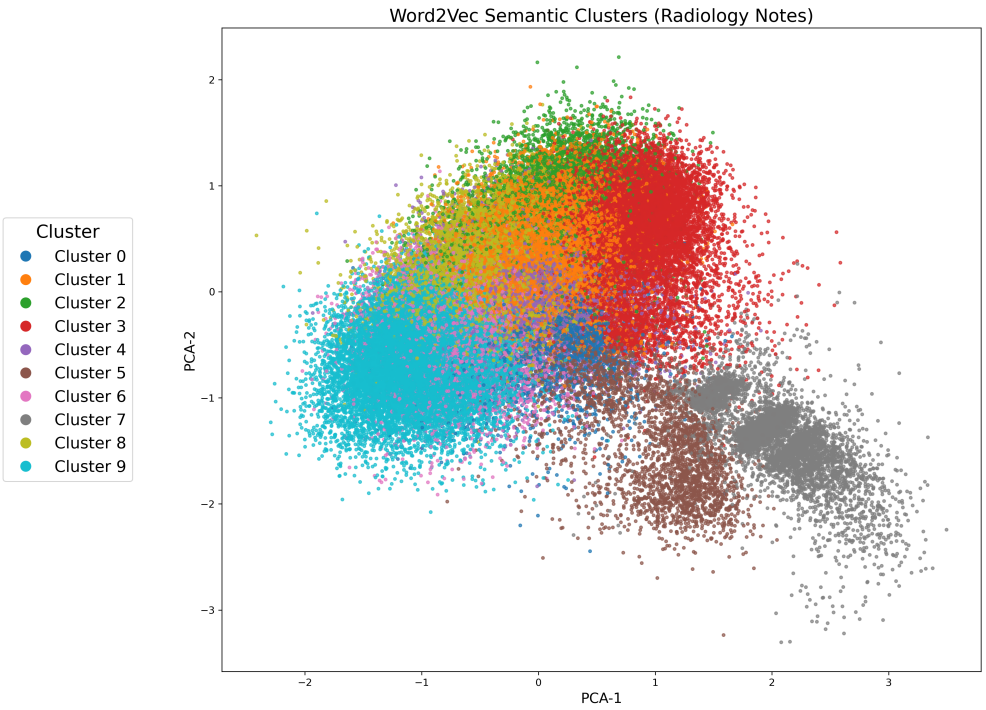



Figure 11: Two-Dimensional PCA Projection of the Optimized Word2Vec Embedding Space for Radiology Notes. Words Were Clustered Using K-Means (k=10).

```
Cluster 0: AND, OF, THE, WITH, RIGHT, LEFT, WITHOUT, OR, VIEWS, UPPER, INTRAVENOUS, BILATERAL, ULTRASOUND:, ULTRASOUND, DIGITAL
Cluster 1: the, of, is, and, with, in, The, There, are, to, right, no, a, left, No
Cluster 2: lateral, changes, posterior, soft, fracture, tissue, degenerative, head, femoral, frontal, vertebral, fracture., intracranial, signal, bone
Cluster 3: cm, lobe, x, mm, measuring, measures, lesion, lymph, cm., kidney, previously, image, up, 2, nodes
Cluster 4: was, catheter, through, into, over, biliary, procedure, then, advanced, wire, 5, placed, sheath, sterile, using
Cluster 5: mGy-cm., DLP, DOSE:, Total, CONTRAST:, CHEST:, thyroid, administered., (Head), LOWER, Oral, reviewed., PACS., VASCULAR:, DLP:
Cluster 6: ., FINDINGS:, COMPARISON:, at, chest, tube, CHEST, ,, tip, pneumothorax., HISTORY:, Dr., line, AP, study
Cluster 7: Acquisition, (Body), mGy, cm;, CTDIvol, s,, Spiral, 1), sequence:, 0.5, 2), Stationary, Sequenced, 3), 3.5
Cluster 8: for, on, were, CT, be, images, by, patient, performed, contrast, an, this, obtained, ultrasound, contrast.
Cluster 9: TECHNIQUE:, old, man, interval, woman, axial, sp, now, Doppler, evaluate, eval, Evaluate, color, transplant, assess
```

Figure 12: Cluster Labels for Figure 11 and Highlighted Vocabulary Words Used in Optimized Radiology Word2Vec Word-Vector Embeddings.

## 4.6 Final Evaluation and Statistical Testing

The AUROC scores obtained in 4.4 Optimized Word2Vec Multimodal Performance and Evaluation were compared to the baseline scores obtained in 4.2 Baseline Model Development and Performance using DeLong's Test to assess within resampling scheme performance improvement. Then, the AUROC scores were bootstrapped to obtain 95% bootstrapped confidence intervals. Finally, multiple testing adjustments were made using the Holm-Bonferroni correction to account for the pairwise tests used to assess $\Delta$AUROC scores. The following sections outline these tests and evaluation metrics.

### 4.6.1 Holdout Evaluation Design

Holdout test-set data was fully partitioned prior to model training to ensure independence from cross-validated folds and the training set. This ensured bias would not be introduced due to dependence nor data leakage that could skew test statistics, thereby violating the independence assumptions of classical statistical tests. Evaluation metrics for paired DeLong tests are consistently compared on the holdout test set metrics to ensure statistical validity across pairwise tests. The Delong test was implemented as outlined in 3.4.4 Comparative Significance testing. Pairwise tests were used within classifier and within resampling scheme to compare optimized Word2Vec multimodal models to complementary baseline models, structured-only baseline or structured + Word2Vec baseline.

### 4.6.2 DeLong Tests for Non-SMOTE Models

Paired AUROC comparisons for the non-SMOTE resampling scheme is shown in Table 21. $\Delta$AUROC scores are listed with 8 out of the 12 comparisons having increases over baseline, with low standard error scores. Gradient Boosting and Logistic Regression demonstrated statistical significance at the raw comparison level, with CatBoost and XGB nearly demonstrating significance as well. None of the classifiers using the non-SMOTE resampling scheme demonstrated statistically significant changes in $\Delta$AUROC after incorporating the Holm-Bonferroni correction.

Table 21: Paired DeLong Tests Comparing Optimized Multimodal Non-SMOTE Models Against Baseline (Original or W2V) with Holm–Bonferroni Correction.

| Classifier | Comparison | AUROC(opt) | AUROC(base) | ΔAUROC | SE(Δ) | z | p | $p_{adj}$ |
|---|---|---|---|---|---|---|---|---|
| CatBoost | opt vs original | 0.756 | 0.734 | +0.022 | 0.012 | 1.855 | 0.0636 | 0.636 |
| | opt vs w2v | 0.756 | 0.757 | -0.001 | 0.008 | -0.072 | 0.943 | 1 |
| GradientBoosting | opt vs original | 0.752 | 0.727 | +0.026 | 0.012 | 2.100 | 0.0357 | 0.393 |
| | opt vs w2v | 0.752 | 0.753 | -0.001 | 0.009 | -0.110 | 0.912 | 1 |
| LGBM | opt vs original | 0.743 | 0.726 | +0.017 | 0.014 | 1.246 | 0.213 | 1 |
| | opt vs w2v | 0.743 | 0.750 | -0.007 | 0.010 | -0.699 | 0.485 | 1 |
| LogisticRegression | opt vs original | 0.757 | 0.723 | +0.034 | 0.012 | 2.818 | 0.00483 | 0.058 |
| | opt vs w2v | 0.757 | 0.752 | +0.005 | 0.007 | 0.682 | 0.495 | 1 |
| RandomForest | opt vs original | 0.736 | 0.726 | +0.010 | 0.012 | 0.782 | 0.434 | 1 |
| | opt vs w2v | 0.736 | 0.733 | +0.004 | 0.007 | 0.482 | 0.63 | 1 |
| XGB | opt vs original | 0.751 | 0.729 | +0.022 | 0.013 | 1.767 | 0.0772 | 0.695 |
| | opt vs w2v | 0.751 | 0.752 | -0.001 | 0.009 | -0.134 | 0.893 | 1 |

### 4.6.3 DeLong Tests for SMOTE Models

Paired AUROC comparisons for the SMOTE resampling scheme is shown in Table 22. ΔAUROC scores are listed with 11 out of the 12 comparisons having increases over baseline, with slightly larger standard error scores than in the non-SMOTE complementary comparisons. Gradient Boosting, LGBM, Logistic Regression, Random Forest, and XGB, 5 out of the 6 classifiers, demonstrated statistical significance at the raw comparison level. While overfitting is a slight concern due to the sampling technique, these results still demonstrate improved discrimination power on externally validated holdout test data.

Specifically, after incorporating the Holm-Bonferroni correction, LGBM and XGB demonstrated statistically significant improvements over structured-only baseline. To note, the optimized Word2Vec multimodal Random Forest SMOTE model significantly outperformed its baseline Word2Vec counterpart. Even after incorporating a conservative correction, the results from the DeLong test demonstrated that including Word2Vec embeddings in the feature space led to significant performance gains over structured-only features using the LGBM and XGB classifiers.

Table 22: Paired DeLong Tests Comparing Optimized Multimodal SMOTE Models Against Baseline (Original or W2V) with Holm–Bonferroni Correction.

| Classifier | Comparison | AUROC(opt) | AUROC(base) | ΔAUROC | SE(Δ) | z | p | $p_{adj}$ |
|---|---|---|---|---|---|---|---|---|
| CatBoost | opt vs original (SMOTE) | 0.736 | 0.725 | +0.012 | 0.012 | 0.940 | 0.347 | 1 |
| | opt vs w2v (SMOTE) | 0.736 | 0.741 | -0.005 | 0.008 | -0.606 | 0.544 | 1 |
| GradientBoosting | opt vs original (SMOTE) | 0.734 | 0.703 | +0.031 | 0.013 | 2.318 | 0.0204 | 0.143 |
| | opt vs w2v (SMOTE) | 0.734 | 0.721 | +0.013 | 0.009 | 1.408 | 0.159 | 0.955 |
| LGBM | opt vs original (SMOTE) | 0.751 | 0.710 | +0.041 | 0.014 | 2.933 | 0.00335 | 0.0369 |
| | opt vs w2v (SMOTE) | 0.751 | 0.737 | +0.014 | 0.010 | 1.371 | 0.17 | 0.955 |
| LogisticRegression | opt vs original (SMOTE) | 0.753 | 0.716 | +0.037 | 0.013 | 2.763 | 0.00573 | 0.0516 |
| | opt vs w2v (SMOTE) | 0.753 | 0.748 | +0.005 | 0.008 | 0.644 | 0.52 | 1 |
| RandomForest | opt vs original (SMOTE) | 0.737 | 0.702 | +0.035 | 0.014 | 2.579 | 0.00991 | 0.0793 |
| | opt vs w2v (SMOTE) | 0.737 | 0.713 | +0.024 | 0.008 | 2.844 | 0.00445 | 0.0445 |
| XGB | opt vs original (SMOTE) | 0.752 | 0.713 | +0.039 | 0.013 | 2.968 | 0.003 | 0.0359 |
| | opt vs w2v (SMOTE) | 0.752 | 0.742 | +0.010 | 0.008 | 1.145 | 0.252 | 1 |

### 4.6.4 DeLong Tests for Best Models

The DeLong test was then used to determine whether the best optimized Word2Vec multimodal model provided a significantly higher AUROC score compared to the best baseline model. Table 23 demonstrates that there was no significant change in AUROC score when resampling scheme was held constant after incorporating the Holm-Bonferroni multiple comparisons correction. The raw p-values obtained were borderline non-statistically significant comparing the best optimized Word2Vec multimodal model to baseline in both sampling schemes, but the naïve model did show statistical significance over baseline. While they are non-statistically significant at the 95% confidence level, since they are close to being significant at the 90% significance level, the intuition is that NLP can meaningfully increase ΔAUROC scores, and more rigorous transformer models could exploit these differences even further. A bootstrap comparison shows the 95% bootstrapped confidence intervals for each of the best models across resampling schemes, the full tables are shown in the next section, 4.6.5 Bootstrapped Confidence Intervals.

Table 23: Paired DeLong Tests for Best-Performing Models Under Each Resampling Scheme (Non-SMOTE and SMOTE), with Holm–Bonferroni Adjusted p-values. Each Row Compares Two Best Models (e.g., Optimized Multimodal vs Structured-Only, or Optimized Multimodal vs Baseline Multimodal) On the Shared Holdout Set.

| Resampling | Comparison (A::B) | AUROC(A) | AUROC(B) | ΔAUROC | SE(Δ) | z | p | $p_{adj}$ |
|---|---|---|---|---|---|---|---|---|
| Non-SMOTE | w2v_opt::LogisticRegression vs structured-only::CatBoost | 0.757 | 0.734 | +0.023 | 0.014 | 1.637 | 0.102 | 0.203 |
| | w2v_opt::LogisticRegression vs w2v_base::CatBoost | 0.757 | 0.757 | +0.000 | 0.012 | 0.015 | 0.988 | 0.988 |
| | w2v_base::CatBoost vs structured-only::CatBoost | 0.757 | 0.734 | +0.023 | 0.011 | 2.052 | 0.0402 | 0.121 |
| SMOTE | w2v_opt::LogisticRegression vs structured-only::CatBoost | 0.753 | 0.725 | +0.028 | 0.015 | 1.911 | 0.0561 | 0.168 |
| | w2v_opt::LogisticRegression vs w2v_base::LogisticRegression | 0.753 | 0.748 | +0.005 | 0.008 | 0.644 | 0.52 | 0.52 |
| | w2v_base::LogisticRegression vs structured-only::CatBoost | 0.748 | 0.725 | +0.023 | 0.014 | 1.598 | 0.11 | 0.22 |

Table 24: Best-Model AUROC Estimates with 95% Bootstrap Percentile Confidence Intervals (2,000 Resamples) Across Resampling Schemes and Variants. Each Row Corresponds to the Single Best Classifier Selected per Variant and Resampling Scheme.

| Resampling | Variant | Classifier | AUROC | SE | 95% $CI_{low}$ | 95% $CI_{high}$ | AUROC ± SE (95% CI) |
|---|---|---|---|---|---|---|---|
| Non-SMOTE | original | CatBoost | 0.734 | 0.017 | 0.700 | 0.768 | 0.734 ± 0.017 [0.700–0.768] |
| | w2v_optimized_radiology | LogisticRegression | 0.757 | 0.016 | 0.725 | 0.789 | 0.757 ± 0.016 [0.725–0.789] |
| | w2v_radiology | CatBoost | 0.757 | 0.016 | 0.724 | 0.788 | 0.757 ± 0.016 [0.724–0.788] |
| SMOTE | original | CatBoost | 0.725 | 0.017 | 0.692 | 0.757 | 0.725 ± 0.017 [0.692–0.757] |
| | w2v_optimized_radiology | LogisticRegression | 0.753 | 0.016 | 0.719 | 0.785 | 0.753 ± 0.016 [0.719–0.785] |
| | w2v_radiology | LogisticRegression | 0.748 | 0.017 | 0.714 | 0.779 | 0.748 ± 0.017 [0.714–0.779] |

After performing all pairwise tests for best models across resampling schemes, there was no pairs that were statistically significant after incorporating the multiple comparisons adjustment; see Figure 28 in Appendix B for the table showing full pairwise comparison tests.

Although the best Word2Vec model did not perform statistically significantly better than the best structured-only model, based on the evidence from the within resampling scheme comparisons and the within classifier comparisons, it is still seen that using NLP can meaningfully improve ΔAUROC scores.

### 4.6.5 Bootstrapped Confidence Intervals

Non-SMOTE AUROC confidence intervals are available in Table 25 and SMOTE AUROC confidence intervals are shown in Table 26). These bootstrapped intervals are performed on the final holout AUROC. These bootstrapped confidence intervals tend to have about a 6% range, indicating that the bootstrapped estimates for the holdout AUROC have wide intervals, indicating decently large uncertainty. This further suggests that more fine tuning in the embedding space for NLP-derived features is necessary to get more certain estimates for the AUROC scores generated by the model classifiers.

Table 25: Optimized Model AUROC Estimates with 95% Bootstrap Percentile Confidence Intervals (2,000 Resamples) for Each Classifier and Comparison Variant.

| Classifier | AUROC | SE | 95% $CI_{low}$ | 95% $CI_{high}$ | AUROC ± SE (95% CI) |
|---|---|---|---|---|---|
| CatBoost | 0.756 | 0.016 | 0.725 | 0.787 | 0.756 ± 0.016 [0.725–0.787] |
| GradientBoosting | 0.752 | 0.017 | 0.719 | 0.785 | 0.752 ± 0.017 [0.719–0.785] |
| LGBM | 0.743 | 0.017 | 0.710 | 0.776 | 0.743 ± 0.017 [0.710–0.776] |
| LogisticRegression | 0.757 | 0.016 | 0.725 | 0.789 | 0.757 ± 0.016 [0.725–0.789] |
| RandomForest | 0.736 | 0.017 | 0.702 | 0.769 | 0.736 ± 0.017 [0.702–0.769] |
| XGB | 0.751 | 0.016 | 0.718 | 0.783 | 0.751 ± 0.016 [0.718–0.783] |

Table 26: Optimized SMOTE Model AUROC Estimates with 95% Bootstrap Percentile Confidence Intervals (2,000 Resamples) for Each Classifier and Comparison Variant.

| Classifier | AUROC | SE | 95% $CI_{low}$ | 95% $CI_{high}$ | AUROC ± SE (95% CI) |
|---|---|---|---|---|---|
| CatBoost | 0.736 | 0.017 | 0.704 | 0.770 | 0.736 ± 0.017 [0.704–0.770] |
| GradientBoosting | 0.734 | 0.017 | 0.702 | 0.767 | 0.734 ± 0.017 [0.702–0.767] |
| LGBM | 0.751 | 0.016 | 0.719 | 0.782 | 0.751 ± 0.016 [0.719–0.782] |
| LogisticRegression | 0.753 | 0.016 | 0.719 | 0.785 | 0.753 ± 0.016 [0.719–0.785] |
| RandomForest | 0.737 | 0.016 | 0.705 | 0.770 | 0.737 ± 0.016 [0.705–0.770] |
| XGB | 0.752 | 0.016 | 0.721 | 0.782 | 0.752 ± 0.016 [0.721–0.782] |

### 4.6.6 Multiple Comparison Adjustment

The Holm–Bonferroni correction was used as a conservative hedge to significance testing as multiple pairwise comparisons were made while performing DeLong's test. Without the adjustment, there were several significant hits demonstrating that the addition of Word2Vec embeddings consistently improved model discrimination ability. However, once implementing the adjustment, none of the non-SMOTE models gave statistically significant $\Delta$AUROC values over structured-only baseline.

Yet, three of the models gave statistically significant improvements, with LGBM and XGB being more important for the analysis. The optimized Word2Vec multimodal LGBM was significant over structured-only baseline and the optimized Word2Vec multimodal XGB was significant over structured-only baseline, demonstrating the power of incorporating NLP embeddings to improve discrimination power over detecting sepsis mortality. The optimized Word2Vec multimodal Random Forest was significant over Word2Vec multimodal baseline, which may be due to the improved embedding space and utilizing the skip-gram architecture over CBOW. Logistic Regression was barely insignificant in both the non-SMOTE and SMOTE models, demonstrating that it was also an important discriminator, albeit not technically significant at the 95% confidence level. Figures 13 and 14 provide visual representations comparing the optimized Word2Vec multimodal models against baseline, contrasting raw p-values to adjusted p-values.

### 4.6.7 Cross-Model Interpretation

The results suggest that each of these top 6 classifiers give stable results with a range of 0.736 to 0.756 (2%) across non-SMOTE models and 0.734 to 0.753 (1.9%) across SMOTE models, indicating that each of the models do an okay job in overall discrimination power, but not touching 0.90 to 0.99 scores that excellent discriminators provide. While these scores are stable, measures can be taken to improve discrimination power as will be described later in 5.4 Future Work.

Figure 13 shows significance markers that reflect raw p-values from paired DeLong tests performed on the holdout set. Figure 14 shows adjusted significance testing per classifier. It

identifies which performance improvements remain statistically robust after controlling for multiple comparisons.

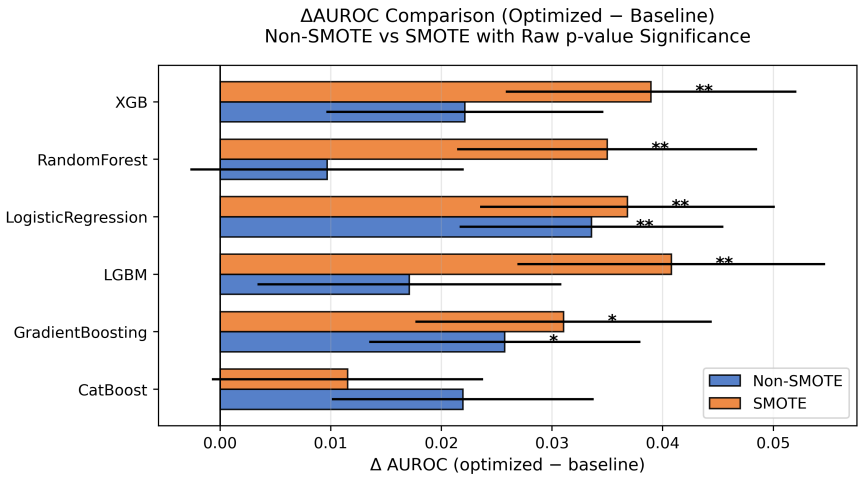



Figure 13: Comparison of ΔAUROC (Optimized minus Baseline) for the Six Optimized Classifiers Under the Non-SMOTE and SMOTE Training Schemes. Bars Represent the Change in Discrimination After Word2Vec Embedding Optimization.

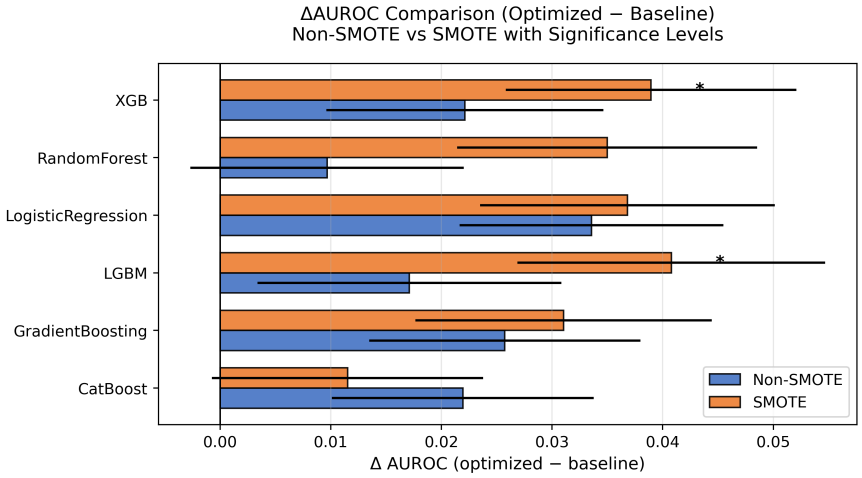


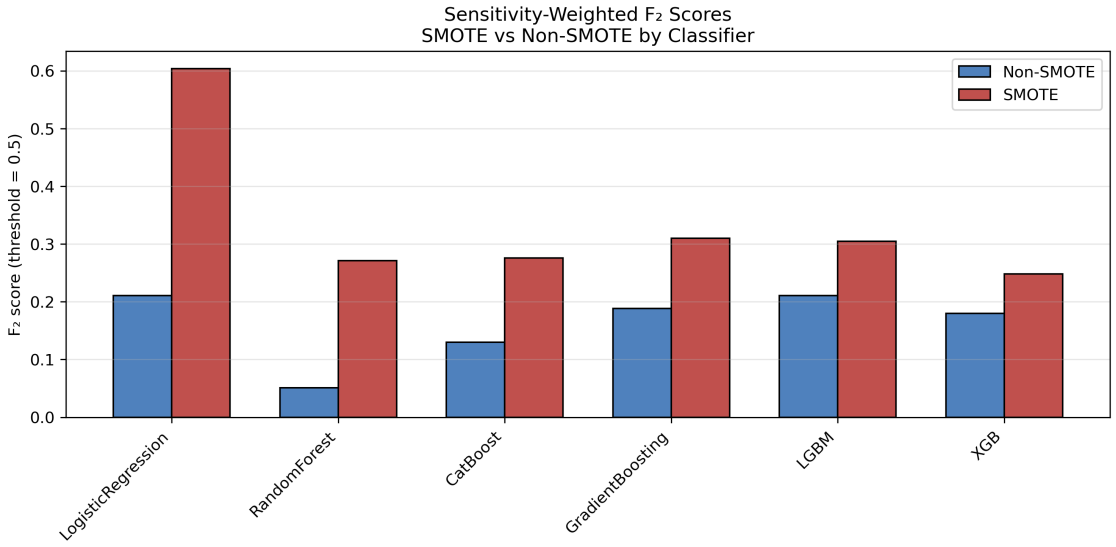

Figure 14: Comparison of ΔAUROC (Optimized minus Baseline) for the Six Optimized Classifiers Under the Non-SMOTE and SMOTE Training Schemes, with Significance Levels Adjusted Using the Holm–Bonferroni Correction.

### 4.6.8 McNemar's Test for $\Delta F_2$ Significance Testing

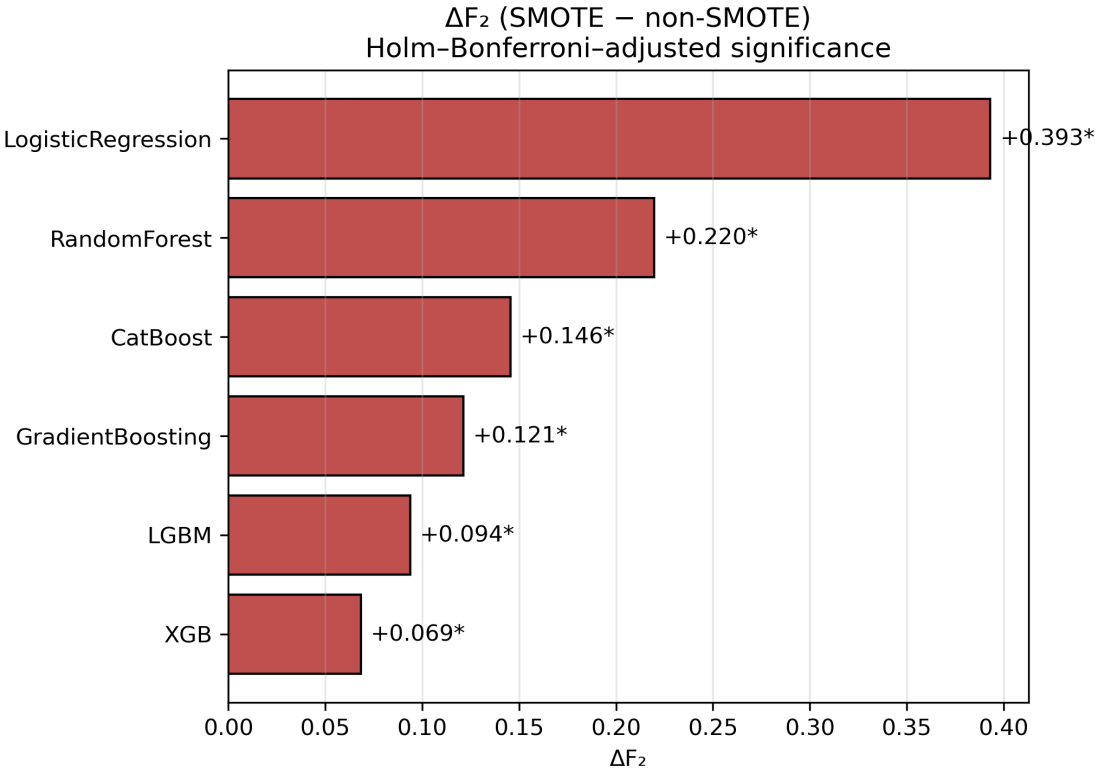
To formally assess the impact of SMOTE resampling on sensitivity-weighted performance, McNemar's test was incorporated with the Holm-Bonferroni correction to test core Hypothesis 2 for a significant difference in $\Delta F_2$ scores in SMOTE models compared to non-SMOTE models. Table 27 demonstrates that all 6 of the classifiers demonstrated extremely significant results demonstrating a huge increase in sensitivity when applying SMOTE models for sepsis mortality prediction.

Table 27: $\Delta F_2$ Performance Comparison for SMOTE vs Non-SMOTE Optimized Word2Vec Multimodal Models On the Holdout Set, Including Bootstrapped Confidence Intervals and Holm-Adjusted McNemar p-values.

| Classifier | $F_2$ (non-SMOTE) | $F_2$ (SMOTE) | $\Delta F_2$ | $\Delta F_2$ Mean | CI Low | CI High | McNemar $\chi^2$ | p | $p_{adj}$ |
|---|---|---|---|---|---|---|---|---|---|
| LogisticRegression | 0.211 | 0.604 | 0.393 | 0.393 | 0.331 | 0.454 | 122.01 | 2.3e-28 | 1.38e-27 |
| RandomForest | 0.051 | 0.271 | 0.220 | 0.220 | 0.161 | 0.281 | 39.45 | 3.37e-10 | 1.69e-09 |
| CatBoost | 0.130 | 0.275 | 0.146 | 0.146 | 0.096 | 0.197 | 27.68 | 1.43e-07 | 5.74e-07 |
| GradientBoosting | 0.188 | 0.310 | 0.121 | 0.122 | 0.064 | 0.179 | 17.52 | 2.84e-05 | 8.53e-05 |
| LGBM | 0.211 | 0.305 | 0.094 | 0.093 | 0.046 | 0.143 | 12.97 | 0.000316 | 0.000633 |
| XGB | 0.180 | 0.248 | 0.069 | 0.068 | 0.025 | 0.111 | 9.38 | 0.0022 | 0.0022 |




(a) $F_2$ Comparison Between SMOTE and Non-SMOTE Models




(b) Adjusted $\Delta F_2$ Scores after Holm–Bonferroni Correction

Figure 15: Comparison of $F_2$ Performance Metrics Across SMOTE and Non-SMOTE Models and Corresponding Holm–Bonferroni Adjusted Differences.

## 4.7 Results Summary

1. **Multimodal integration** of clinical text data with structured features improved AUROC and AUPRC across nearly all models, in baseline and optimized embedding environments.
2. **Resampling with SMOTE** led to higher $F_2$ scores, indicating greater sensitivity to mortality events.
3. **Feature attribution** SHAP analyses demonstrated that physiologic and text-based features joitly contributed to making predictions for sepsis mortality.
4. **Calibration performance** confirmed probabilistic reliability across models, demonstrating non-SMOTE models are slightly better calibrated over using SMOTE resampling schemes.

In the next chapter, Discussion, the results reported here will be discussed along with public health significance of these findings. As well, important limitations of this study and directions for future work will be addressed.

# 5.0 Discussion

## 5.1 Principal Findings

### 5.1.1 Hypothesis 1 – Word2Vec Improves Discrimination

This was demonstrated through statistically significant improvements, according to DeLong's test after incorporating the Holm-Bonferroni correction, in $\Delta$AUROC for the LGBM and XGB classifiers using optimized Word2Vec feature integration over a structured-only feature space. While the best Word2Vec model did not improve statistically significantly over the structured-only model, the results suggest that further refinement of NLP methods could improve this difference in discrimination further.

### 5.1.2 Hypothesis 2 – SMOTE Enhances Sensitivity

This was demonstrated through statistically significant improvements, according to McNemar's test after incorporating the HOlm-Bonferroni correction, in $\Delta F_2$ score for all SMOTE Word2Vec multimodal classifiers over non-SMOTE counterparts.

### 5.1.3 Hypothesis 3 – Interpretability Maintained Across Modalities

Clustering across word-vectors used for optimized Word2Vec model selection shows coherent semantic cohesion across concept-maps for sepsis mortality. Feature importance plots suggest Word2Vec embeddings carry important weight. SHAP analyses verified interpretability of the models after including Word2Vec features. Reliable, calibrated plots and summaries through calibration curves and Brier score comparison demonstrate stability across model families.

### 5.1.4 Summary of Statistical Testing Outcomes

With these findings, it is firmly suggested that incorporating NLP embeddings from radiology notes into static prediction software for multimodal sepsis mortality prediction can provide significant advances over baseline structured-only approaches when using a radiology background profile at the patient-level. Furthermore, incorporating SMOTE resampling into NLP embedding schemes can increase model sensitivity to detecting rare sepsis mortality events at the slight sacrifice of discriminative power in external validation. These findings demonstrate the discriminative power of NLP techniques and demand that they be refined and improved to further increase predictive and discriminative power.

## 5.2 Clinical Implications

### 5.2.1 Interpretability and Feature Insights

The clinical interpretation of these results is difficult to assess as each of the ensemble boosting methods provide complex decision boundaries that lead to less than optimal rules to provide to physicians and care team leaders working with a rapidly changing disease like sepsis. Since these results depend largely on medical text being largely available through radiology notes, these notes need to be present in the first place. Therefore, patients need to already be in the hospital or have a history of radiology tests that give clinical indicators that look like the embedding features that lead to predicting positive sepsis mortality in order for these models to actually be useful. Additionally, these results use a full radiology background as opposed to targeted windows of time, and so this fundamental aspect needs to be considered when assessing feature importances for word-vectors derived from the tokenized documents.

### 5.2.2 Public Health Significance

This work has major implications for public health as sepsis is a major cause of mortality in the ICU. With the breadth of available EHR data, natural language processing can incorporate decision models into everyday healthcare decisions to prolong life for individuals in acute medical care. While the primitive, static models presented in this study will not be enough to advance longevity, the clinical findings demonstrate that the integration of natural language processing with classical methods may advance discriminative power in dynamic settings in manners similar to what is demonstrated in this work.

Incorporating dynamic models that use real-time data, and incorporating reinforcement learning and federated learning models would provide access to dynamic discrimination as well as access to more patient data, leading to a more significant contribution to this field, as these static results for now only ascertain the utility of NLP in a static environment based on a given set of laboratory, vital, demographic, comorbidities, and radiology notes data.

## 5.3 Limitations

### 5.3.1 Data Level Limitations

This work required trust in the validity and accuracy of the preprocessed structured dataset curated by Gao et al. These authors state to have used a non-conventional assessment for coma score to be $\geq 8$ [18] and so this classification was carried into this work for consistency. To address limitations related to data sourcing, efforts were made to recreate the dataset from BigQuery, but the necessary derived data tables for reconstruction were unavailable. Additionally, previous works use different sets of structured clinical features, so using the reproducible pipeline methodology created in this study with those varying sets of clinical variables would increase verifiability of the impact of NLP features on model discrimination.

A major assumption this work also draws on is the independence of notes across patients. If a radiologist tended to use certain language features for one patient over the other, this

bias would impact the embedding space and thus skew embedding vectors across patients, potentially allowing for shifts in labeling distributions for prediction. This assumption is hard to regularize because it is not necessarily something that can be stratified over or adjusted for, as de-identified data is being used.

Independence of notes across radiologists is also assumed, such that the semantic language used for radiology documentation is treated as coming from a stationary distribution of language shared between providers. Under the assumption, linguistic variability reflects random stylistic differences or clinically driven terminology as opposed to differences in a radiologist's ability to convey complex information. Individuals that used consistent documentation or styles different from others could result in clustering patterns of embeddings unrelated to physiological markers related to sepsis mortality, affecting individual embeddings and the overall embedding space being trained and tested on. Stratifying by provider would be a possible hedge against this in follow up studies.

### 5.3.2 Clinical Note Preprocessing, Tokenization, and Temporality

The temporal ordering of radiology notes was not incorporated early on in the data processing stage because full modeling had already been completed before this issue was recognized. However, it is important to re-clarify that Word2Vec is time-independent: it learns semantic relationships from local co-occurrence patterns, not from clinical event chronology like radiology tests and visits. The lack of temporal ordering did not distort the global embedding space in a way that introduces biased AUROC estimates.

Since textual embeddings were not ordered by chart time, the temporal nature of the tokenized embeddings was lost. Additionally, as stated in 2.3, Unstructured Text Data, the resulting text documents for each patient may have contained visits outside of their time in the ICU. Although this would pose more issues in a time-based ICU isolated analysis, it still can induce adjustments in the word-vectors described for each patient if there are some radiology notes visits before ICU stay that contained information not associated with sepsis mortality, making the word-vectors less informative. This could introduce systematic biases related to diluted signal information and therefore alter feature importances of the

word-vectors, but it is more likely that it added noise to the system, resulting in deflated AUROC and other metric scores.

The way this study utilized notes was derived from a holistic approach to a patients radiology background and how it contributes to sepsis mortality in general, but it fails to address temporality and targeted time in ICU, and it cannot be used as an early detection mechanism due to the concatenation of all notes for at least the full ICU stay. In the procedure used, the significance testing says that Word2Vec embeddings improve discrimination using a full radiology notes background for each patient. This could introduce biased estimates from an early detection perspective, as information leakage from late term notes including near-death terminology could bias the AUROC metrics measured, but this study does not use that framework. As stated in 2.3, an improved methodology for note extraction could be incorporated in future works, for other types of NLP modeling for sepsis mortality prediction, which will be described.

Tokens were searched across using window size and negative search size, which scanned for semantic cohesion in 5 to 10 word chunks based on parameter tuning, according to the architecture of Word2Vec CBOW and skip-gram architectures. When transitioning from one note to the next, these temporal relations are lost due to the random nature of the parallel processing that originally grouped and combined a patient's text documents together. However, due to Word2Vec's training on the full corpora and the shuffling used by Gensim, these transitions do not lead to meaningfully different embedding spaces. So while the word-vectors created for each patient lacked temporality, it does not impact the aggregated embedding space nor patient-level embeddings. To this end, the shuffling of notes is more a hindrance for downstream modeling tasks like early detection and time-series analysis.

The temporal ordering of notes can be addressed for other types of NLP techniques that require temporality, or to incorporate early assessment and/or time-series analyses of the radiology notes. Batch-based processing could be used to continue parallel processing techniques, but with each batch ordered over event time, starting at ICU admission and ending at a pre-specified interval such as 24 or 48 hours post-admission for early detection purposes, or discharge or death to have a more targeted approach to utilizing ICU stay length as opposed to the full radiology background.

The amount of radiology visits and the length of notes are also potential confounders to the effect of discriminative power, as this study combined all notes into one token, thereby losing the number of stays and average note length as elements to stratify over or adjust for. But once again, Word2Vec trains on the entire notes corpora, so the length and amount of notes each patient has does not impact the vector magnitude or direction. The issue is that those who have more notes or longer notes have more semantic context structures that could possibly be important and related to predicting sepsis mortality.

### 5.3.3 Modeling Limitations

The machine learning methods introduced in this work are a small subset of the expansive modeling solutions that are available for prediction. The models presented in this work tend to be prediction-focused classifiers rather than inferential classifiers. To learn more about the actual language features associated with causal inference for sepsis mortality, linear, generalized linear, and generalized additive models could be used to provide a more robust analysis for clinical interpretability.

The models developed in this study excel in providing prediction-oriented metrics, but they heavily rely on the hyperparameters passed into them, as well as the training data from which they discover decision boundaries. The Gini metric was used for tree-based classifiers in this study, but incorporating the Entropy metric could lead to better results at the slight expense of computational efficiency. Additionally, many of the models used for tuning scale extensively by increasing sample size, and so relatively small search spaces were utilized as performing model training in a cloud service was not available. Using cloud services along with optimized CPU hardware would significantly speed up training for future reproductions of this work, allowing for expanded model testing and coverage.

Balancing computational efficiency with robustness was a constant tradeoff in this study. A BERT transformer-based architecture was briefly explored, but utilizing cloud-based software and iterating over greater than 100 million parameters was outside the scope of this work. Incorporating high performance computing and distributed systems can allow for more expansive grid searches as well as applying more complex transformer-based architectures

would reduce the computational demand while retaining robust analyses.

### 5.3.4 Interpretability and Bias

The models developed in this study are not meant to infer or establish causal relationships. There were no learned features that have direct causal associations with increased mortality from sepsis. Rather, models highlight associations that improve predictive performance when clinical text features are included. Although informative, the SHAP values and correlations analyses do not provide evidence of mechanistic pathways to sepsis mortality, they should be viewed from the lens of descriptive approximations of model behavior.

Several clinician-based biases involved in clinical text creation could also be at play that can propagate bias into embedding spaces. Some of these biases include availability bias, anchoring bias, confirmation bias, and systematic differential documentation across patient groups. These biases cause systematic differential construction of notes, shaping how language patterns are learned by embeddings, thus impacting fairness, calibration, and generalizability of models.

## 5.4 Future Work

In future work, advanced NLP models using transformer based architectures such as ClinicalBERT, a transformer based architecture for global learning across clinical language data can be used. Light-weight and computationally inexpensive alternatives can also be explored — advanced neural networks such as recurrent neural networks that can incorporate time-based components into learning, and convolutional neural networks that can look at 1D linguistic data similarly to Word2Vec. It would also be useful to compare the discriminative effects of linearized discriminant analysis, generalized additive models, and Bayesian models to the results of this study. These methods could exploit further discrimination differences over baseline structured-only approaches.

To address the temporal limitations of the static models presented here, dynamic mod-

eling can be introduced by incorporating a time component into modeling. Survival analysis could be performed by studying the association of language features with increased admission time to diagnosis, where discovering language-based features from clinical notes and EHR data could provide doctors more time to intervene on sepsis-related health events.

Finally, applying the methodology presented in this work on additional external validation datasets, and incorporating transfer learning into the methodology, would provide clear motivation to continue extensive work in the incorporation of textual embeddings in sepsis related research, and clinical care medicine at large.

The following conclusion will provide closing comments on this work and assess its findings and impact.

# 6.0 Conclusion

In conclusion, natural language processing is an important tool that is used in large language modeling, and is gaining traction in modern machine learning in healthcare. Utilizing EHR data at the patient level can be analyzed for prediction and inference. This study demonstrates the power of NLP in sepsis mortality prediction by showing statistically significant increases in AUROC across resampling methodologies. Additionally, although the models trained and used can be black boxes from a mathematical viewpoint, features derived using Shapley and feature importance analyses maintain the integrity of the advanced machine learning models. To further advance the discriminative power of identifying sepsis mortality events, dynamic models using more cutting edge transformers are necessary, as well as incorporating survival analyses to project patient outcomes and quantify survivability for patients with sepsis. To this end, this work proposes a modular and reproducible work to advance sepsis mortality prediction from varying modalities.

# Appendix A Codebase

The SQL, Python, R, and LaTeX code used for data analysis are presented in the following GitHub code repository:

`https://github.com/tylerkelly7/Masters-Thesis/blob/main/README.md`

## Appendix B Additional Tables

Table 28: Global Best-Versus-Best Paired DeLong Comparisons Across All Six Best-Performing Models, Spanning Structured-Only, Multimodal Word2Vec, and SMOTE Versus Non-SMOTE Resampling Schemes. Holm–Bonferroni Adjusted p-values ($p_{adj}$) Control the Family-Wise Error Rate Across All Pairwise Tests.

| Reference model (A) | Comparison model (B) | AUROC(A) | AUROC(B) | ΔAUROC | SE(Δ) | z | p | $p_{adj}$ |
|---|---|---|---|---|---|---|---|---|
| Non-SMOTE, Structured | Non-SMOTE, W2V-Optimized | 0.734 | 0.757 | -0.023 | 0.014 | -1.637 | 0.102 | 1 |
| | Non-SMOTE, W2V-Radiology | 0.734 | 0.757 | -0.023 | 0.011 | -2.052 | 0.0402 | 0.523 |
| | SMOTE, Structured | 0.734 | 0.725 | +0.009 | 0.008 | 1.144 | 0.252 | 1 |
| | SMOTE, W2V-Optimized | 0.734 | 0.753 | -0.019 | 0.014 | -1.293 | 0.196 | 1 |
| | SMOTE, W2V-Radiology | 0.734 | 0.748 | -0.013 | 0.015 | -0.899 | 0.368 | 1 |
| Non-SMOTE, W2V-Optimized | SMOTE, Structured | 0.757 | 0.725 | +0.032 | 0.014 | 2.282 | 0.0225 | 0.315 |
| | SMOTE, W2V-Optimized | 0.757 | 0.753 | +0.004 | 0.003 | 1.146 | 0.252 | 1 |
| | SMOTE, W2V-Radiology | 0.757 | 0.748 | +0.009 | 0.008 | 1.128 | 0.259 | 1 |
| Non-SMOTE, W2V-Radiology | Non-SMOTE, W2V-Optimized | 0.757 | 0.757 | -0.000 | 0.012 | -0.015 | 0.988 | 1 |
| | SMOTE, Structured | 0.757 | 0.725 | +0.032 | 0.013 | 2.507 | 0.0122 | 0.183 |
| | SMOTE, W2V-Optimized | 0.757 | 0.753 | +0.004 | 0.013 | 0.304 | 0.761 | 1 |
| | SMOTE, W2V-Radiology | 0.757 | 0.748 | +0.009 | 0.013 | 0.736 | 0.462 | 1 |
| SMOTE, Structured | SMOTE, W2V-Optimized | 0.725 | 0.753 | -0.028 | 0.015 | -1.911 | 0.0561 | 0.673 |
| | SMOTE, W2V-Radiology | 0.725 | 0.748 | -0.023 | 0.014 | -1.598 | 0.11 | 1 |
| SMOTE, W2V-Radiology | SMOTE, W2V-Optimized | 0.748 | 0.753 | -0.005 | 0.008 | -0.644 | 0.52 | 1 |

# Appendix C Additional Figures

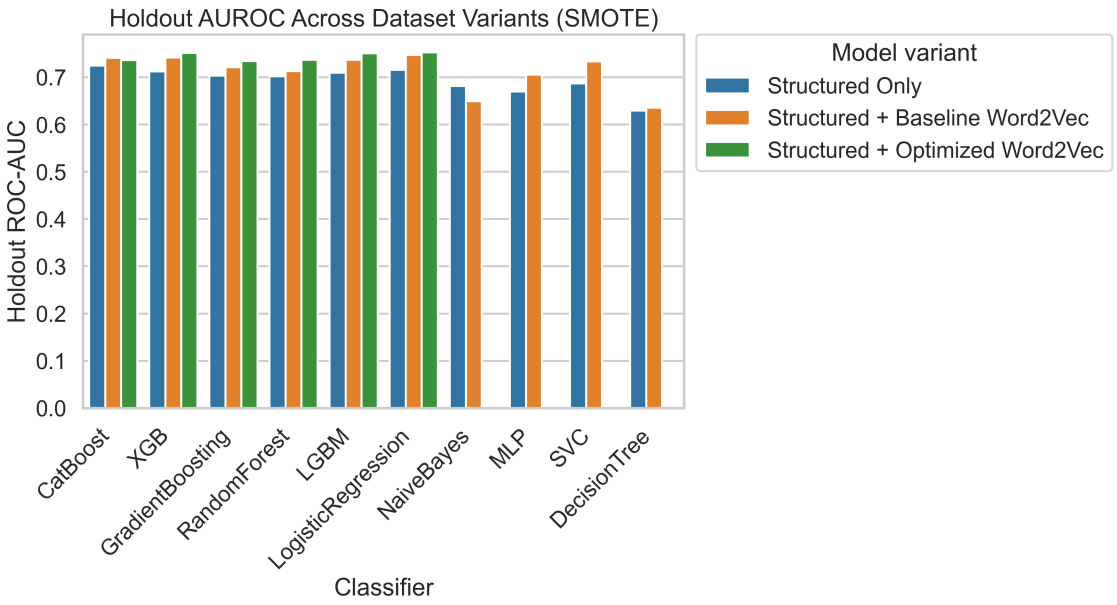
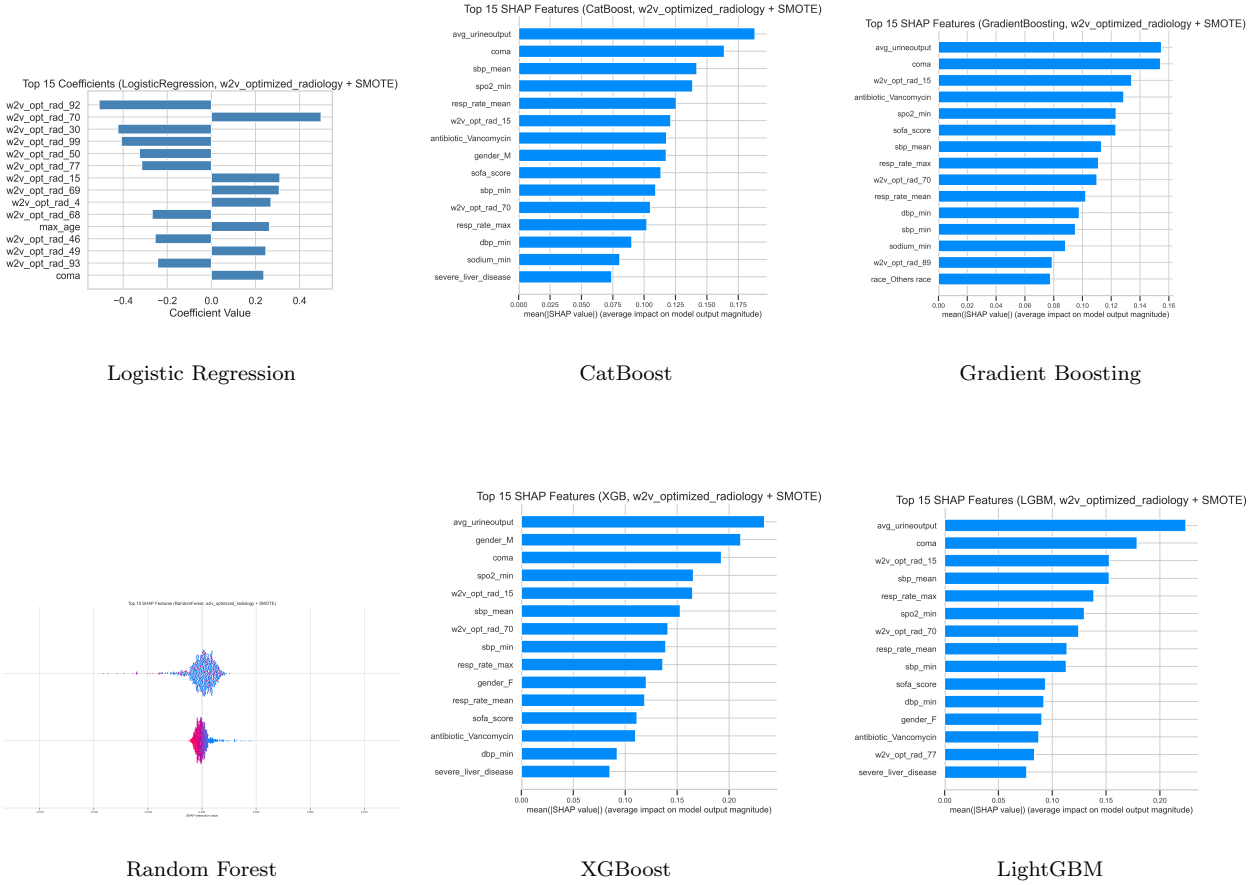


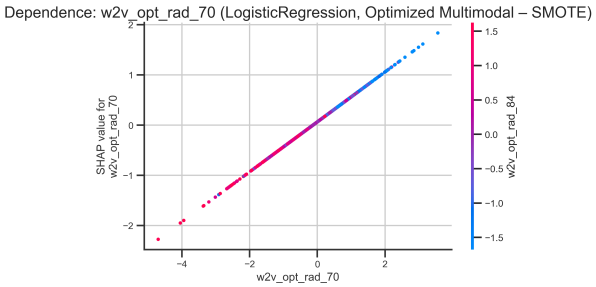

Figure 16: AUROC Comparison of Baseline and Optimized Models (SMOTE).



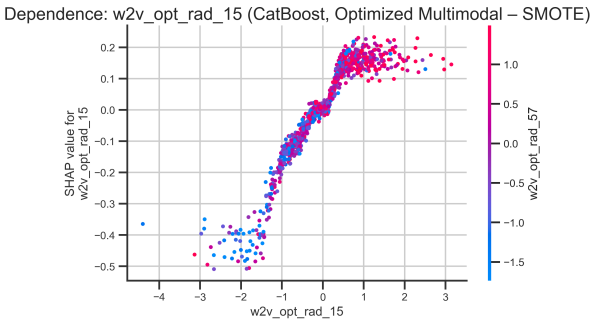

Logistic Regression CatBoost Gradient Boosting

Random Forest XGBoost LightGBM

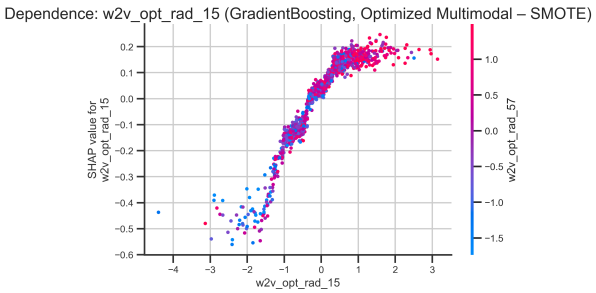
Figure 17: SHAP Summary Plots Illustrating Feature Contributions for Optimized Word2Vec Multimodal Models (SMOTE). Each Subplot Represents the Top Predictive Features for the Six Classifiers Evaluated After Resampling.



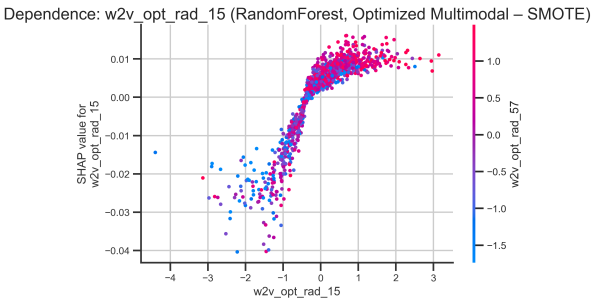

Logistic Regression



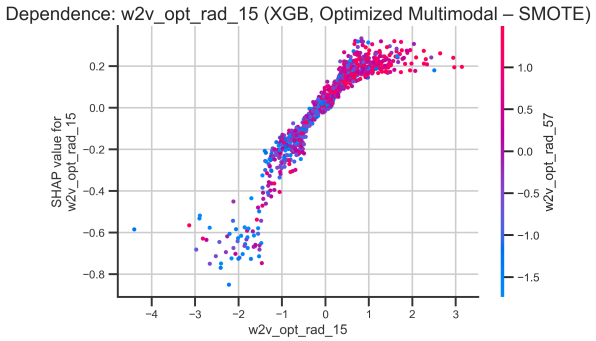

CatBoost



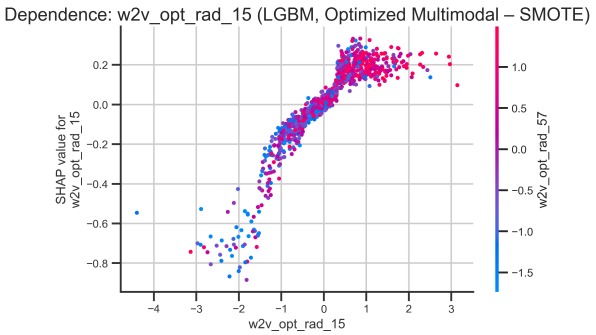

Gradient Boosting




Random Forest

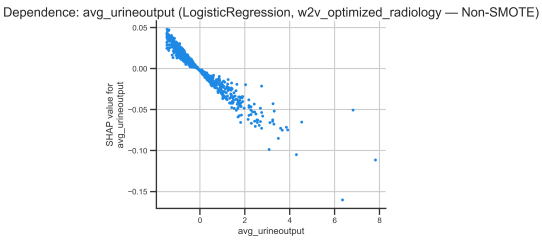



XGBoost

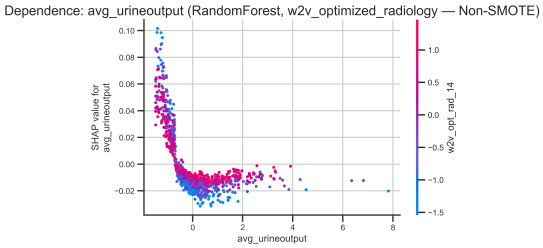



LightGBM

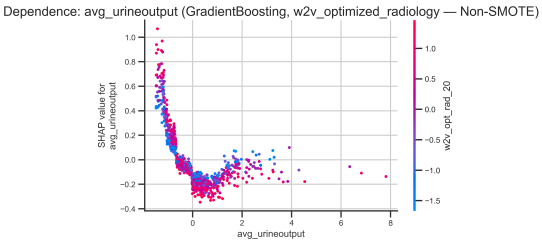
Figure 18: SHAP Dependence Plots Showing the Top Word2Vec Feature Contributions for Optimized Multimodal Models (SMOTE). Each Subplot Illustrates Feature Value Relationships for the Top Embedding-Based Predictors Across Classifiers.



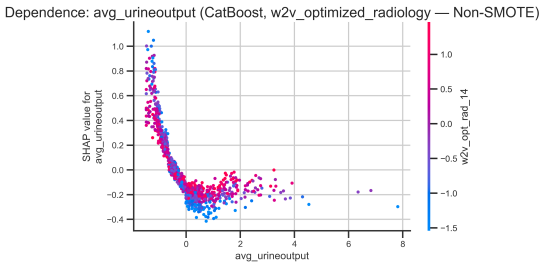

Logistic Regression



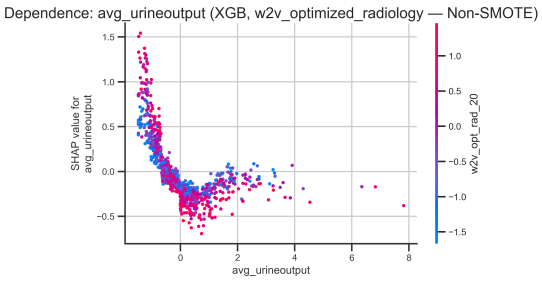

Random Forest



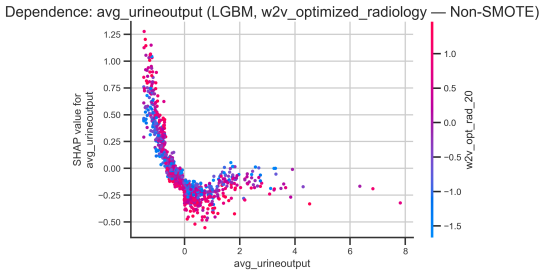

Gradient Boosting




CatBoost




XGBoost




LightGBM

Figure 19: SHAP Dependence Plots Showing Feature Contributions of `avg_urineoutput` for Optimized Word2Vec Multimodal Models (Non-SMOTE). Each Plot Illustrates how Predicted Mortality Risk Varies with Urine Output Across Classifiers.

## Bibliography


[1] Martín Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015.

[2] Fatemeh Amrollahi et al. Contextual embeddings from clinical notes improves prediction of sepsis. *NPJ Digital Medicine*, 4:101, 2021.

[3] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[4] Sean R. Bennett. Sepsis in the intensive care unit. *Surgery (Oxford)*, 33(11):565–571, 2015.

[5] Hongying Bi, Xiaofei Liu, Cheng Chen, Ling Chen, Xin Liu, Jiawei Zhong, Yiqiao Yang, and Jianpin Xie. The pao2/fio2 is independently associated with 28-day mortality in patients with sepsis: a retrospective analysis from the mimic-iv database. *BMC Pulmonary Medicine*, 23:187, 2023.

[6] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[7] Jason Brownlee. Smote for imbalanced classification with python, 2021.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[10] Jee-Woo Choi et al. Prognostic prediction of sepsis patients using transformer with skip-connected token for tabular data. *Artificial Intelligence in Medicine*, 146:102820, 2024.

[11] David R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B*, 20(2):215–242, 1958.

[12] Corentin da Costa-Luis. tqdm: A fast, extensible progress bar for python, 2023.

[13] Elizabeth R. DeLong et al. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44(3):837–845, 1988.

[14] Joblib Developers. Joblib: Running python functions as pipeline stages, 2024.

[15] Lea Draeger, Carolin Fleischmann-Struzek, Sabine Gehrke-Beck, Christoph Heintze, Daniel O. Thomas-Rueddel, and Konrad Schmidt. Barriers and facilitators to optimal sepsis care – a systematized review of healthcare professionals' perspectives. *BMC Health Services Research*, 25:591, 2025.

[16] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets*. Springer, 2018.

[17] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[18] Jiayi Gao, Yuying Lu, Negin Ashrafi, Ian Domingo, Kamiar Alaei, and Maryam Pishgar. Prediction of sepsis mortality in icu patients using machine learning methods. *BMC Medical Informatics and Decision Making*, 24:228, 2024.

[19] Kim Huat Goh et al. Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nature Communications*, 12:711, 2021.

[20] Matteo Guarino, Benedetta Perna, Alice Eleonora Cesaro, Martina Maritati, Michele Domenico Spampinato, Carlo Contini, and Roberto De Giorgio. 2023 update on sepsis and septic shock in adult patients: Management in the emergency department. *Journal of Clinical Medicine*, 12(9):3188, 2023.

[21] Charles R. Harris et al. Array programming with numpy. *Nature*, 585:357–362, 2020.

[22] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[23] Alistair E. W. Johnson et al. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

[24] Alistair E. W. Johnson et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 9:106, 2022.

[25] Guolin Ke et al. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[26] Thomas Kluyver et al. Jupyter notebooks – a publishing format for reproducible computational workflows. In *Proceedings of the 20th International Conference on Electronic Publishing*, pages 87–90, 2016.

[27] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[28] Bartosz Krawczyk. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[29] Scikit learn Developers. Scikit-learn api reference, 2024.

[30] Scikit learn Developers. Scikit-learn cross-validation documentation, 2024.

[31] Scikit learn Developers. Scikit-learn documentation: Model evaluation, 2024.

[32] Scikit learn Developers. Scikit-learn model selection guide, 2024.

[33] Scikit learn Developers. Scikit-learn user guide: Preprocessing, 2024.

[34] Ke Li et al. Predicting in-hospital mortality in icu patients with sepsis using gradient boosting decision tree. *Journal of Critical Care*, 63:89–96, 2021.

[35] Ran Liu et al. Natural language processing of clinical notes for improved early prediction of septic shock in the icu. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1344–1347, 2019.

[36] Zheng Liu et al. Interpretable machine learning for predicting sepsis risk in emergency triage patients. *Scientific Reports*, 15:17889, 2025.

[37] Yuying Lu. Sepsis mortality prediction repository, 2023.

[38] Scott M. Lundberg. Shap python package documentation, 2024.

[39] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[40] Wes McKinney. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, pages 51–56, 2010.

[41] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[42] Florentina Mușat, Dan Nicolae Păduraru, Alexandra Bolocan, Cosmin Alexandru Palcău, Andreea-Maria Copăceanu, Daniel Ion, Viorel Jinga, and Octavian Andronic. Machine learning models in sepsis outcome prediction for icu patients: Integrating routine laboratory tests—a systematic review. *Biomedicines*, 12(12):2892, 2024.

[43] National Institute of General Medical Sciences. Sepsis fact sheet, 2023.

[44] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632, 2005.

[45] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[46] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] Liudmila Prokhorenkova et al. Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[48] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[49] Yasser Sakr, Ulrich Jaschinski, Xavier Wittebole, Tamas Szakmany, Jeffrey Lipman, Silvio A. Ñamendys-Silva, Ignacio Martin-Loeches, Marc Leone, Mary-Nicoleta Lupu, Jean-Louis Vincent, and ICON Investigators. Sepsis in intensive care unit patients: Worldwide data from the intensive care over nations audit. *Open Forum Infectious Diseases*, 5(12):ofy313, 2018.

[50] Mervyn Singer et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801–810, 2016.

[51] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.

[52] Pauli Virtanen et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[53] Michael L. Waskom. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[54] Wikipedia contributors. Area under the curve, 2024. Accessed 2025.

[55] Wikipedia contributors. F-score, 2024. Accessed 2025.

[56] Wikipedia contributors. Receiver operating characteristic, 2024. Accessed 2025.

[57] Melissa Y. Yan et al. Sepsis prediction, early detection, and identification using clinical text for machine learning: A systematic review. *Journal of the American Medical Informatics Association*, 29(3):559–572, 2022.

[58] Xudong Zhu et al. Embedding, aligning and reconstructing clinical notes to explore sepsis. *BMC Research Notes*, 14(1):261, 2021.

[59] Muhammad Zubair et al. Revolutionizing sepsis diagnosis using machine learning and deep learning models: a systematic literature review. *BMC Infectious Diseases*, 25(1):1396, 2025.

[60] Radim Řehůřek. Gensim: Word2vec model documentation, 2023.

[61] Radim Řehůřek. Gensim: Word2vec tutorials, 2023.

[62] Radim Řehůřek and Petr Sojka. Gensim: Word2vec implementation details, 2024.