# Predicting Hospital Readmission from Clinical Discharge Summaries

Stanford CS229 Project

**Alina Gavrilov**
Computation and Mathematical Engineering
Stanford University
alinagav@stanford.edu

**Kalyani Limaye**
Computation and Mathematical Engineering
Stanford University
limayk@stanford.edu

**Nils Kuhn**
Department of Electrical Engineering
Stanford University
nfkuhn@stanford.edu

## Abstract

Predicting hospital readmissions from electronic health records (EHR) can improve patient outcomes and reduce healthcare costs. This study explores whether free-text discharge summaries alone can predict readmissions using LLM-driven embeddings from ClinicalBERT (pre-trained and fine-tuned) and OpenAI's 4o-mini API. We compare these against keyword-based embeddings across five classifiers using MIMIC-IV data. Random forests perform best (F1: 79%, Accuracy: 74%) with OpenAI summaries, though dataset limitations pose challenges. Despite a 30% error rate, our findings suggest LLMs can extract meaningful clinical signals. Future work focuses on scaling, refining embeddings, and optimizing models.

## 1 Introduction

Hospital readmissions represent a significant challenge in healthcare, imposing substantial financial burdens on institutions while indicating potential gaps in patient care quality. Developing reliable, interpretable methods to predict these readmissions can enable targeted interventions, optimize resource allocation, and ultimately improve patient outcomes through data-driven decision-making. Recent advances in machine learning (ML) and natural language processing (NLP) have created new opportunities to extract actionable insights from the rich, unstructured text data contained in electronic health records (EHRs). While previous studies have explored various ML approaches using structured clinical records and combinations of numerical and textual information, our research focuses on the value of free-text discharge summaries alone for readmission prediction. This project leverages the MIMIC-IV Note dataset of clinical discharge summaries, providing narrative descriptions of patient's hospital stays, including admission reasons, hospital course, and discharge instructions. We employ LLMs to identify and extract the most relevant predictive indicators from these clinical narratives. Our results will be compared to a baseline logistic regression model that uses predefined keywords from existing literature [Chiu et al. (2024)] and multi-hot encoding for feature representation. Further, we explore three approaches to leveraging contextual embeddings for improved prediction performance: (1) using pre-trained ClinicalBERT to generate contextual hierarchical embeddings from full-text discharge summaries, (2) fine-tuning ClinicalBERT on MIMIC-IV discharge summaries to better capture dataset-specific representations before running prediction models, and (3) using OpenAI's 4o-mini API to generate summary-based embeddings by first summarizing the discharge notes and then embedding these condensed representations using OpenAI's text-embedding-ada-002 model. By evaluating these approaches, we aim to determine the viability of LLM-driven feature extraction in clinical readmission prediction and compare its performance against traditional keyword-based models.

## 2 Related Work

Predictive modeling for hospital readmission using NLP has gained traction [Orangi-Fard et al. (2022); Sheetrit et al. (2023); Li and Liu (2024)]. Early work applied traditional ML methods like bag-of-words [Orangi-Fard et al. (2022)], while deep learning approaches, such as ClinicalBERT, have since leveraged richer semantic information [unk (2024b); Li and Liu (2024)]. Hybrid models combining topic modeling with LSTMs capture thematic and temporal aspects [Chiu et al. (2024); Lin et al. (2019)]. Embedding-based NLP techniques, including self-distillation, further enhance semantic representation learning [Chen and Xiao (2024)]. However, direct comparisons of LLM strategies for readmission prediction remain limited. Existing benchmarks focus on general medical knowledge or single-note tasks, often overlooking the complexities of feature extraction across multiple patient admissions [Kweon et al. (2024); Loshchilov and Hutter (2017)].

## 3 Dataset and Features

Our study uses discharge summaries from the MIMIC-IV Note dataset, containing 331,794 summaries from 145,915 patients. Each summary includes details on admission reasons, hospital course, and discharge instructions. Key fields are the full-text summary, note_id, subject_id, chart_time, and store_time. To create binary labels for readmission, we sorted records by subject ID and chart time, setting $Y = 1$ if a future hospital visit was recorded and $Y = 0$ otherwise.

For baseline feature extraction, we adapted medically relevant keywords from Chiu et al. (2024), which used BertTopic, HDBSCAN clustering, and class-based TF-IDF. Using these keywords, we vectorized 10,000 sampled summaries into multi-hot and one-hot embeddings—capturing keyword frequency and presence per document, respectively. These structured embeddings serve as our baseline model's input. To balance computational efficiency and model performance within GPU constraints, we experimented with varying dataset sizes for finetuning and summarization, ensuring $Y$ labels remained between 45-55%. We then created five different embeddings using three LLMs.

For pre-trained ClinicalBERT, we used a hierarchical sentence-wise approach to prevent mid-sentence chunk splitting. Text was split into sentences, then further divided into non-overlapping 512-token chunks, each embedded with ClinicalBERT. These embeddings were averaged at both sentence and document levels to generate a final representation. Since ClinicalBERT required hierarchical methods to fit within its context window, we sought a way to generate concise summaries of clinical notes. An initial attempt using FalconsAI medical_summarization produced low-quality outputs: *"We report a case of a recurrence in a patient's sex with a male, female, and male male. He was a female female. The male male was accompanied by a woman with hepatitis III."* Given these limitations, we pivoted to OpenAI's 4o-mini API, summarizing 500 notes before embedding them with text-embedding-ada-002 for comparison.

We implemented two fine-tuning methods. The first trained Bio_ClinicalBERT (emilyalsentzer/Bio_ClinicalBERT) for readmission prediction by adding an untrained linear layer with cross-entropy loss for binary classification. A major issue with finetuning Bert was the amount of computation. BERT processes 512-token chunks. In our first finetuning method we decided to use only the first chunk of each text. That allowed the model to train three epochs ( 11 hours) on 179,836 samples (44,959 for validation), excluding the last two months to avoid misleading labels. We used a learning rate of $5 \times 10^{-6}$, batch size of 4, and 16 gradient accumulation steps. The second method trained semantic embeddings using cosine similarity loss, penalizing embeddings with small angular distances between different labels. Given GPU memory constraints, we incorporated six chunks per input (split at sentence boundaries) while limiting batch size to 2. Instead of a shifted window approach, we averaged embeddings across chunks to reduce computation. This method completed one epoch in 12 hours, with a learning rate of $1 \times 10^{-5}$, 16 gradient accumulation steps, and a cosine similarity margin of 0.5 (i.e., embeddings were considered different if their angular distance exceeded $60°$).

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

## 4 Methods

For our classification task, we used five different methods. The first method is a standard logistic regression model with optional regularization terms. A bias term is added at index zero, which is not regularized. The model's weights are updated using gradient ascent until convergence or after one million iterations. The prediction is obtained using the sigmoid function, where the output is classified as 1 if $\sigma(x) \geq 0.5$ and 0 otherwise. For the final model, we used a learning rate of 0.01 and a regularization term of 1.

The second method is a boosting algorithm, implemented using the XGBoost library, which also provides functionalities for random forests and decision trees, allowing for hyperparameter tuning. The boosting model constructs up to 5000 small trees, each with a maximum depth of 3. After each iteration, the algorithm increases the weight of misclassified data points while decreasing the weight of correctly classified ones. These adjusted weights guide the construction of subsequent trees. Each tree is evaluated based on classification performance, and the model is validated on a separate dataset. The model with the lowest validation loss is used for testing. To prevent overfitting, we applied L2 regularization with a coefficient of 1, along with both sub-sampling and column sampling at a rate of 80%. Sub-sampling means that only 80% of the training data is used for tree construction, while column sampling ensures that only 80% of the features are selected for each tree. The loss function used is the logarithmic loss, defined as:

$$LogLoss = -1/N \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

The random forest algorithm constructs multiple trees in parallel, rather than sequentially as in boosting. In our case, we generated up to 100 trees, each with a maximum depth of 8. To reduce overfitting, we again applied 80% sub-sampling and 80% column sampling per tree. Additionally, we introduced column sampling per node at 80%, meaning that for each node, a feature is randomly selected 20% of the time. Unlike boosting, random forests use only a single iteration, after which the model predicts both the validation and test sets. We observed that random forests are sensitive to imbalanced data, often favoring the majority class. To address this, we downsampled the majority class in both the training and validation sets to enhance model performance.

The decision tree model relies on a deep, single-tree structure to classify the dataset. It generates only one tree without using techniques such as sub-sampling or column sampling to mitigate overfitting. Due to its lack of regularization and reliance on a single tree, it performed the worst across most datasets.

Lastly, we implemented a feedforward neural network that uses two hidden layers with 50 neurons each, and a softmax output layer for binary classification. The model is configured with a learning rate of 0.01 and L2 regularization set to 0.0025. These hyperparameters were determined through manual experimentation, where we evaluated model learning performance under different configurations and ultimately selected a consistent set of hyperparameters to ensure uniformity across all our datasets. During forward propagation, the network applies the sigmoid activation in the hidden layers. The loss function is calculated as the average negative log-likelihood, with a regularization term to control overfitting. Gradients are derived through backpropagation and weight updates are executed via batch gradient descent with a learning rate that is dynamically reduced over epochs. A 5-fold cross-validation strategy is implemented to partition the dataset and compute average performance metrics across folds, with both training and validation losses and error rates being recorded.

## 5 Experiments / Results / Discussion

The results presented in the figure below compare each classification model's performance on each different embedding strategy. F1-score and accuracy were specifically selected as evaluation metrics to provide a balanced view of model effectiveness, ensuring that performance on both minority and majority classes is adequately assessed. Specifically, F1-score combines precision and recall, making it a strong indicator of how well the model identifies the minority class without over- or under-classifying it. We seek to minimize false positives to lessen unnecessary psychological duress potentially caused by attaching a high readmission predictor to a patient and minimize false negatives because patients and their doctors should know if there is extra risk of readmission. Accuracy

measures the proportion of correct predictions, offering a straightforward understanding of the model's overall performance across all classes.

| Model \ Dataset | Fine-tuned Bert CE | Fine-tuned Bert CS | OpenAI Full Text | OpenAI Summarized | Pre-trained Bert |
|---|---|---|---|---|---|
| **Decision Tree** | F1: 0.63<br>Acc: 0.63 | F1: 0.58<br>Acc: 0.58 | F1: 0.61<br>Acc: 0.52 | F1: 0.61<br>Acc: 0.54 | F1: 0.66<br>Acc: 0.61 |
| **Logistic Regression** | F1: 0.67<br>Acc: 0.67 | F1: 0.62<br>Acc: 0.63 | F1: 0.61<br>Acc: 0.54 | F1: 0.59<br>Acc: 0.56 | F1: 0.70<br>Acc: 0.65 |
| **NN** | F1: 0.67<br>Acc: 0.68 | F1: 0.62<br>Acc: 0.61 | F1: 0.55<br>Acc: 0.49 | F1: 0.63<br>Acc: 0.55 | F1: 0.71<br>Acc: 0.63 |
| **Random Forest** | F1: 0.69<br>Acc: 0.68 | F1: 0.66<br>Acc: 0.66 | F1: 0.73<br>Acc: 0.66 | F1: 0.79<br>Acc: 0.74 | F1: 0.68<br>Acc: 0.66 |
| **XGBoost** | F1: 0.65<br>Acc: 0.65 | F1: 0.65<br>Acc: 0.65 | F1: 0.70<br>Acc: 0.60 | F1: 0.67<br>Acc: 0.60 | F1: 0.73<br>Acc: 0.68 |

Figure 1: F1/Accuracy Matrix

We observe that Random Forest consistently demonstrates strong performance, especially notable with OpenAI summaries (F1: 0.79, Accuracy: 0.74). However, we must note that the dataset used for OpenAI embeddings was considerably smaller (500 vs. 10,000 for the other strategies) which may introduce bias and skew results. Another interesting observation is that the neural network achieves a relatively high F1-score (0.71) but moderate accuracy (0.63) with pre-trained ClinicalBERT embeddings. This discrepancy may indicate that the NN effectively captures certain patterns relevant to one class but misclassifies other data points, potentially due to noisy or ambiguous data.

The plots below show the ROC curves first for each embedding strategy using our best-performing model, Random Forest, as determined by accuracy metric [Figure 1], then for each model using our best-performing embedding strategy, fine-tuned Bert with CE loss [Figure 2].

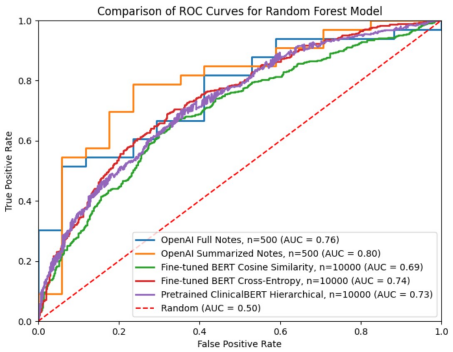


Figure 2: ROC Random Forest Models

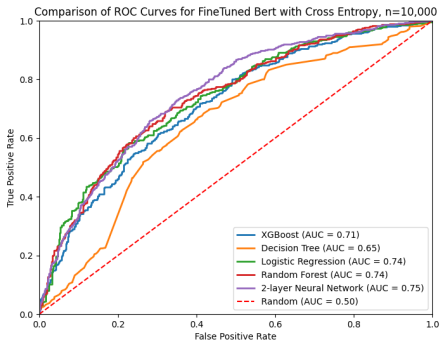


Figure 3: ROC Fine-Tuned Bert Cross-Entropy

The classification methods yielded an error rate of approximately 0.3, which remains relatively high. Several factors contribute to this elevated error rate. First, the clinical notes describe only the patient's current condition. Rehospitalization can be significantly influenced by unpredictable factors such as accidents, social circumstances, or unforeseen complications that are unrelated to the patient's immediate health status. Second, there is potential misclassification due to hospital transfers. Some patients may have been rehospitalized at a different institution, making it appear as though they were not readmitted when, in reality, they were. Third, since the dataset consists primarily of patients with severe health conditions, there is a high likelihood of mortality even after discharge. This can lead to cases where rehospitalization does not occur, not because the patient was healthy, but because they passed away before returning to the hospital. Beyond these structural limitations, the quality, length,

completeness and consistency of clinical notes vary significantly. Some patients may withhold crucial health information, leading to incomplete or misleading records. Finally, personal choice plays a role. Some individuals may choose to avoid rehospitalization, even when medically advisable. Since our labels are based solely on recorded hospital admissions, such decisions are not captured in the dataset.

## 6 Conclusion / Future Work

In conclusion, the error rate stayed around approximately 0.3 across embedding strategies and models. This suggests that some meaningful signal relevant to readmission can be extracted from these semantic features, yet further refinement is necessary for a more reliable predictive model. Depending on the approach, we obtained different indications of overfitting. Overall, performance on the validation set and the test set differed by only about 0.02, suggesting that the model does not suffer from severe overfitting. Although the error rate remains relatively high, which may point to underfitting, it is also possible that other factors contribute to this elevated error.

In the future, with more time and increased computational resources, we would first aim to establish a parallel processing setup, such as running jobs on a dedicated computing cluster, to take advantage of our full set of 330,000+ clinical notes. This would capture a much broader range of linguistic and patient-condition variability than our current smaller samples allow. Such comprehensive coverage would likely reduce sampling bias and lead to more robust, generalizable embeddings. When fine-tuning ClinicalBert by training on our data, we were able to process only a subset of each clinical note which still took multiple hours for a single epoch. In future work, incorporating more epochs over the full text could yield more comprehensive embeddings. Additionally, incorporating hierarchical embedding or retrieval-augmented generation (RAG) methods may further improve representational quality and downstream performance. We would also replace our manual hyperparameter tuning process with more advanced methods such as Bayesian optimization or grid search, which systematically explore a wider search space and potentially discover more optimal settings for training. Finally, although we researched various clinically oriented language models, we only had the opportunity to test a subset; going forward, we intend to experiment with additional LLMs such as BlueBERT, PubMedGPT, and MedAlpaca. Evaluating these models could yield embeddings and downstream classification results more specifically tailored to our readmission task.

Lastly, since our small subset of ChatGPT 4o-mini summaries tended to perform slightly better than several other embeddings, we think fine-tuning ChatGPT 4o-mini on medical notes can create a valuable tool for estimating patient hospital readmission risk when combined with more robust chain-of-thought prompting on discharge summaries.

## 7 Contributions

The teamwork on this project was highly efficient, collaborative, and well-coordinated. Alina spearheaded obtaining the datasets, generating both full-text and summary embeddings through the OpenAI API, and conducting literature review. Kalyani handled hierarchical chunking, creating embeddings using pre-trained ClinicalBERT, and developed helper functions for classifiers and error-metric calculations. Nils built the classifiers from scratch, worked extensively on fine-tuning ClinicalBERT, and interpreted model outputs to guide parameter tweaks. Throughout the project, all team members contributed to debugging, running experiments, synthesizing conclusions, and shaping the overall vision.

## References


2023. Ehrshot: An ehr benchmark for few-shot evaluation of foundation models (neurips 2023). *arXiv preprint*. Available at: `https://arxiv.org/pdf/2307.02028`.

2024a. Do we still need clinical language models? *Proceedings of Machine Learning Research (PMLR)*. Available at: `https://proceedings.mlr.press/v209/eric23a/eric23a.pdf`.

2024b. Emerge: Enhancing multimodal electronic health records predictive modeling with retrieval augmented generation. *arXiv preprint*. Available at: `https://arxiv.org/pdf/2406.00036`.

Rasha Assaf and Rashid Jayousi. 2020. 30-day hospital readmission prediction using mimic data. In *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–6.

Jianlv Chen and Shitao Xiao. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. Synthical. Available at: `https://synthical.com/article/9ffce599-0640-457c-bd1c-502cab06e8af`.

C.-C. Chiu, C.-M. Wu, T.-N. Chien, L.-J. Kao, and C. Li. 2024. Predicting icu readmission from electronic health records via bertopic with long short-term memory network approach. *Journal of Clinical Medicine*, 13:5503.

William A. Falcon. 2019. Pytorch lightning. GitHub.

Sunjun Kweon, Jiyoun Kim, Heeyoung Kwak, Dongchul Cha, Hangyul Yoon, Kwanghyun Kim, Jeewon Yang, Seunghyun Won, and Edward Choi. 2024. Ehrnoteqa: An llm benchmark for real-world clinical practice using discharge summaries. *arXiv preprint*.

Xintao Li and Sibei Liu. 2024. Predicting 30-day hospital readmission in medicare patients: Insights from an lstm deep learning model. *medRxiv*.

Y-W Lin, Y Zhou, F Faghri, MJ Shaw, and RH Campbell. 2019. Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long short-term memory. *PLoS ONE*, 14(7):e0218942.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint*.

Liantao Ma, Chaohe Zhang, Junyi Gao, Xianfeng Jiao, Zhihao Yu, Yinghao Zhu, Tianlong Wang, Xinyu Ma, Yasha Wang, Wen Tang, Xinju Zhao, Wenjie Ruan, and Tao Wang. 2023. Mortality prediction with adaptive feature importance recalibration for peritoneal dialysis patients. *Patterns*, 4(12).

N. Orangi-Fard, A. Akhbardeh, and H. Sagreiya. 2022. Predictive model for icu readmission based on discharge summaries using machine learning and natural language processing. *Informatics*, 9:10.

M. Pishgar and J. Theis. 2022. Prediction of unplanned 30-day readmission for icu patients with heart failure. *BMC Medical Informatics and Decision Making*, 22:117.

Eitam Sheetrit, Menachem Brief, and Oren Elisha. 2023. Predicting unplanned readmissions in the intensive care unit: A multimodality evaluation. *Scientific Reports*, 13:18407.

Assaf and Jayousi (2020) Lin et al. (2019) Li and Liu (2024) Chiu et al. (2024) Pishgar and Theis (2022) Kweon et al. (2024) Orangi-Fard et al. (2022) Sheetrit et al. (2023) Falcon (2019) Loshchilov and Hutter (2017) Ma et al. (2023) Chen and Xiao (2024) unk (2024b) unk (2024a) unk (2023)