Large Language Models vs Classical Machine Learning: Predictive Power on Structured Data

Running Title:

Mohammadreza Ghaffarzadeh-Esfahani1, XX, XX, Ali Soroush, Hamidreza Hatamabadi, Ilad Alavi Darazam, Seyed Amir Ahmad Safavi-Naini \*1,2,3 , Mohamad Amin Pourhoseingholi\*1

1- Research Institute for Gastroenterology and Liver Diseases, Shahid Beheshti University of Medical Sciences, Tehran, Iran

2-

3- Division of Data Driven and Digital Health (D3M), The Charles Bronfman Institute for Personalized Medicine, Icahn School of Medicine at Mount Sinai, New York, NY, USA

4-

5-

X- Department of Emergency Medicine, School of Medicine, Safety Promotion and Injury Prevention Research Center, Imam Hossein Hospital, Shahid Beheshti University of Medical Sciences, Tehran, Iran

X- Infectious Diseases and Tropical Medicine Research Center, Shahid Beheshti University of Medical Sciences, Tehran, Iran

X- Department of Infectious Diseases, Loghman Hakim Hospital, Shahid Beheshti University of Medical Sciences, Tehran, Iran

\* Corresponding to: Seyed Amir Ahmad Safavi-Naini ([sdamirsa@ymail.com](mailto:sdamirsa@ymail.com)) and Mohamad Amin Pourhoseingholi (xxx)

Authors’ Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Full Name | Position | Email; ORCID | Affiliation | Contribution |
| Mohammadreza Ghaffarzadeh-Esfahani |  |  | (a) Research Institute for Gastroenterology and Liver Diseases, Shahid Beheshti University of Medical Sciences, Tehran, Iran  (b) xxxx | Conceptualization, Methodology, Programming, Investigation, Writing Original Draft |
|  |  |  |  |  |
| Ali Soroush | MD, Assistant Professor of Gastroenterology | [ali.soroush.2012@gmail.com](mailto:ali.soroush.2012@gmail.com); 0000-0001-6900-5596 | Division of Data Driven and Digital Health (D3M), The Charles Bronfman Institute for Personalized Medicine, Icahn School of Medicine at Mount Sinai, New York, NY, USA | Methodology, Validation |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Hamidreza Hatamabadi |  |  | Department of Emergency Medicine, School of Medicine, Safety Promotion and Injury Prevention Research Center, Imam Hossein Hospital, Shahid Beheshti University of Medical Sciences, Tehran, Iran |  |
| Ilad Alavi Darazam |  | [ilad13@yahoo.com](mailto:ilad13@yahoo.com); 0000-0002-4440-335X | (a) Infectious Diseases and Tropical Medicine Research Center, Shahid Beheshti University of Medical Sciences, Tehran, Iran  (b) Department of Infectious Diseases, Loghman Hakim Hospital, Shahid Beheshti University of Medical Sciences, Tehran, Iran |  |
| Seyed Amir Ahmad Safavi-Naini | MD, Research Fellow | [sdamirsa@ymail.com](mailto:sdamirsa@ymail.com); 0000-0001-9295-9283 | (a) Research Institute for Gastroenterology and Liver Diseases, Shahid Beheshti University of Medical Sciences, Tehran, Iran  (b) Division of Data Driven and Digital Health (D3M), The Charles Bronfman Institute for Personalized Medicine, Icahn School of Medicine at Mount Sinai, New York, NY, USA | Conceptualization, Methodology, Data Curation, Editing the Original Draft, Project Administration |
| Mohamad Amin Pourhoseingholi |  |  |  |  |

# Introduction

Recent advancements in large language models (LLMs) have led to a surge in their practical applications, notably in the medical field. These models, capable of rapidly assimilating specialized knowledge across various medical domains, offer versatility in language and context adaptation, thereby broadening global access to medical expertise ([ref](https://pubmed.ncbi.nlm.nih.gov/37378099/)). LLMs, having undergone extensive training on vast datasets, excel in numerous natural language processing tasks, including language generation, machine translation, and question-answering ([2](https://arxiv.org/abs/1912.02164), [3](https://arxiv.org/abs/2110.05448), [4](https://arxiv.org/abs/1909.01066)). While their primary training focuses on predicting subsequent words, LLMs can function akin to an evidence-based knowledge assistant for practitioners, offering valuable insights and support (ref).

In the realm of medical and clinical practice, machine learning models are increasingly used for predicting patient outcomes, prognoses, and mortality rates. These models typically include supervised (e.g., classification models) and unsupervised learning (e.g., clustering algorithms) methods, primarily utilizing structured data ([5](https://pubmed.ncbi.nlm.nih.gov/33121479/)). Clinical datasets often contain a mix of structured and unstructured data, with clinical notes being a prime example of the latter.

In the domain of patient information management, traditional machine learning paradigms often adopt a bifurcated procedural approach. The initial phase involves the transformation of textual information, inherently unstructured, into a structured tabular format. The subsequent phase entails the utilization of these structured datasets for the training of various machine learning models (ref). However, this dual-phase process is frequently associated with potential pitfalls, including the loss of critical information and the introduction of complexities in model deployment, thereby posing significant challenges in the practical application of these technologies in clinical settings (ref).

The power of LLMs on using clincal text, such as discharge summaries and ward notes are previously showed in detail (ref). However, the LLM performance in handling structured data and to determine whether they outperform classic machine learning models is an area for study. This is important since much of the previous medical data is stored in a structured format. Although these structured data originate from unstructured sources, such as clinical notes, the predictive power of LLMs on tabular data is an unanswered debate ([ref](https://pubmed.ncbi.nlm.nih.gov/37516790/)).

This study seeks to address this knowledge gap by conducting a rigorous experiment on LLMs' predictive capabilities in the context of clinical outcomes. Specifically, we intend to evaluate their effectiveness in utilizing tabular structured data, which has been manually extracted from clinical notes recorded by healthcare professionals. Our second experiment is focused on evaluating the effect of missing values, and the use of LLMs as imputer. This pilot study can guide the future use of LLMs on mixed or structured-only datasets.

# Method

## Ethical Consideration

## Dataset

In this study we tested our experiments on the previously collected dataset of X patients

### Dataset: Data Collection and Outcomes

## Seven CML Predictive Performance

In the first experiment, we employed five classical machine learning methods: Logistic Regression, Support Vector Machine (SVM), Decision Tree, k-Nearest Neighbors (KNN), Random Forest plus  Neural Networks, and XGBoost. These models were implemented on our dataset to predict mortality outcomes.

To enhance the robustness of our analysis, we initially partitioned our dataset into two categories: internal validation and external validation. Subsequently, we divided the dataset into a train set and a test set using a test size ratio of 0.3. This process was undertaken to ensure a rigorous evaluation of the models and to establish a clear distinction between internal and external validation for comprehensive performance assessment.

### Imputing and normalizing

To address missing values in the dataset we used iterativeimputer from the skitlearn library. this method uses iterative prediction for each feature and then imputes it considering the Multiple Imputation by Chained Equations (MICE) method[[1](https://www.jstatsoft.org/article/view/v045i03),[2](https://www.jstor.org/stable/2984099)]. for optimal model performance, the dataset underwent normalization using StandardScaler[[3](https://arxiv.org/abs/1201.0490)]. These preprocessing steps were executed independently for the training set, the test set, and the external validation set, ensuring a consistent approach in handling missing values across the experimental sets.

### Feature selection and sampling

the dataset has 81 features as on-admission features. we separated the dataset into external and internal validation by using patient hospitals. There are four hospitals in our dataset we separated patients from hospital4 for external validation and the rest for internal validation. The dataset was partitioned into two categories: external validation and internal validation the internal validation size to external validation size ratio is 6.8. for each external and internal validation The splitting was performed with a test size of 30% (test\_size=0.3), allocating 70% of the data for training. we separated four features as targeted features. these features are as follows: In-hospital Mortality, ICU admission Intubation, and Dialysis in this study we only used hospital mortality as a targeted feature.76 features were used for training features 53 of which are categorical and the rest are numerical values. in the process of data wrangling, we dropped two repeated features. we strategically employed the Lasso method for feature selection due to its effectiveness in handling high-dimensional data. The Lasso method introduces regularization by adding a penalty term to the linear regression objective function, encouraging sparsity in feature coefficients[[3](https://arxiv.org/abs/1201.0490),[4](https://arxiv.org/abs/2011.00898)]. This approach proved superior to alternative methods, facilitating notable enhancements in our results. Through the application of Lasso, we derived a refined dataset that highlighted the most impactful features based on their importance, aiding in dimensionality reduction. Subsequently, we ranked and selected the top 40 features for further analysis.

To address imbalanced data in the dataset., characterized by uneven class distribution, we applied the Random undersampling technique from the Sklearn Python library[[6](https://pubmed.ncbi.nlm.nih.gov/19095540/)]. The effective use of Random Undersampling demonstrated its ability to address imbalances in our dataset. This, in turn, enhanced the reliability of our analysis for predicting mortality.

### Model implementation

we employed seven classical machine learning algorithms, namely Logistic Regression, Support Vector Machine (SVM), Decision Tree, k-nearest Neighbors (KNN), Random Forest, Neural Network, and XGBoost.

In the logistic regression model employed for this study, L2 regularization (penalty='l2') was applied with a convergence tolerance set to 0.0001, a regularization strength of 1.0, and the use of the Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm for optimization. The model was configured to fit the intercept term, with intercept scaling set to 1, and no specific class weights assigned. The logistic regression utilized a maximum of 100 iterations for convergence, automatic detection of the multi-class scenario,5-fold cross-validation was used and verbosity level set to 0 for minimal output during training. The random state of 42 was set for reproducibility. An SVC (Support Vector Classification) model was employed in this study with a regularization parameter C set to 1.0, utilizing the radial basis function kernel ('rbf') with a default degree of 3. The gamma parameter was set to 'scale', the coefficient of the kernel function to 0.0, and shrinking during optimization was enabled. The model was configured without probability estimates and a tolerance for convergence set to 0.001. A cache size of 200 was allocated for optimization (cache\_size=200), with no specific class weights assigned (class\_weight=None). The model had no specified maximum iteration limit (max\_iter=-1) and employed the 'ovr' (one-vs-rest) strategy for decision function shape. Break ties were not considered in the decision function (break\_ties=False),5-fold cross-validation was used and The random state of 42 was set for reproducibility.

A Decision Tree Classifier was employed in this study with the Gini impurity criterion ('gini'), utilizing the 'best' strategy for splitting nodes. The tree had no specified maximum depth (max\_depth=None) and a minimum of 2 samples required to split an internal node (min\_samples\_split=2). The minimum number of samples required to be in a leaf node was set to 1 (min\_samples\_leaf=1), with no specified minimum weight fraction for a leaf node (min\_weight\_fraction\_leaf=0.0). The maximum number of features considered for splitting was not restricted (max\_features=None),5-fold cross-validation was used and The random state of 42 was set for reproducibility.. There were no limitations on the maximum number of leaf nodes (max\_leaf\_nodes=None), and the minimum impurity decrease for a split was set to 0.0 (min\_impurity\_decrease=0.0). No specific class weights were assigned (class\_weight=None), and the cost complexity pruning alpha (ccp\_alpha) parameter was set to 0.0. Additionally, monotonic constraints on features were not specified (monotonic\_cst=None). A K-Neighbors Classifier was utilized in this study with the number of neighbors set to 5 (n\_neighbors=5), employing uniform weights for neighbor contributions ('weights='uniform''). The algorithm used for computing nearest neighbors was automatically determined ('algorithm='auto''), with a leaf size of 30 (leaf\_size=30). The Minkowski distance metric with a power parameter of 2 (p=2) was employed ('metric='minkowski''). Additional metric parameters were not specified (metric\_params=None),5-fold cross-validation was used and parallel processing was not utilized (n\_jobs=None).A RandomForestClassifier was employed in this study with 100 estimators (n\_estimators=100), utilizing the Gini impurity criterion ('criterion='gini''). The decision trees in the random forest had no specified maximum depth (max\_depth=None), and a minimum of 2 samples was required to split an internal node (min\_samples\_split=2). The minimum number of samples required to be in a leaf node was set to 1 (min\_samples\_leaf=1), with no specified minimum weight fraction for a leaf node (min\_weight\_fraction\_leaf=0.0). The square root of the total number of features was considered for splitting at each node ('max\_features='sqrt''). The maximum number of leaf nodes was unrestricted (max\_leaf\_nodes=None), and the minimum impurity decrease for a split was set to 0.0 (min\_impurity\_decrease=0.0). Bootstrap sampling was enabled (bootstrap=True), out-of-bag scoring was not utilized (oob\_score=False),5-fold cross-validation was used and parallel processing was not employed (n\_jobs=None). The random state of 42 was set for reproducibility during the model-building process (random\_state=42). The model verbosity level was set to 0 (verbose=0), and warm starting was disabled (warm\_start=False). No specific class weights were assigned (class\_weight=None), and the cost complexity pruning alpha (ccp\_alpha) parameter was set to 0.0. The maximum number of samples for bootstrapping was not specified (max\_samples=None), and monotonic constraints on features were not specified (monotonic\_cst=None).The XGBClassifier model was configured with default hyperparameters, including a base score of 0.5, utilizing a gradient boosting tree-based approach ('booster='gbtree''). The maximum depth of each tree was set to 3, and the learning rate was 0.1. The subsample ratio of training instances was set to 1, indicating the use of all training data. The objective function employed for binary classification was 'binary:logistic', and the number of boosting rounds (n\_estimators) was set to 100. The model was configured to use a single CPU core for parallelism (n\_jobs=1),5-fold cross-validation was used and the random seed was set to 42 for reproducibility. Default values were applied for parameters such as gamma, min\_child\_weight, colsample\_bytree, reg\_alpha, reg\_lambda, scale\_pos\_weight, and verbosity, among others.In this study, a neural network model was employed for classification using the MLPClassifier from Scikit-learn. Hyperparameter tuning was conducted through grid search optimization to identify the most effective configuration. The explored hyperparameters included variations in the hidden layer sizes, with options for architectures consisting of a single layer with 100 neurons or two layers with 50 neurons each. The rectified linear unit (ReLU) activation function was chosen, and the Adam solver was employed for optimization. The regularization term (alpha) was set to 0.0001 to control overfitting. The model's training was limited to a maximum of 200 iterations, and a random state of 42 was specified for reproducibility. Early stopping was enabled, with a validation fraction of 0.1 and a criterion of 10 consecutive iterations with no improvement (n\_iter\_no\_change) to halt the training process. The grid search was conducted using 5-fold cross-validation, and the best-performing neural network classifier was identified as the one with the optimal combination of hyperparameters. The resulting best classifier, determined through the grid search, was then further evaluated on the training data, and its performance metrics were used for subsequent analysis and interpretation.

The models were evaluated using six key metrics: Specificity, Recall, Accuracy, Precision, F1 Score, and area under the curve(AUC). These metrics provide a comprehensive assessment of the model's performance across different aspects, ensuring a thorough examination of their effectiveness in capturing specific nuances within the dataset. Using grid search and cross-validation enhances the reliability of our results and underscores the robustness of our model selection process.

## LLMs and Vector Similarity Search Predictive Performances

### Imputing, normalizing, Feature selection, and sampling

The primary objective of this experiment is to transform our dataset into a format compatible with large language models and subsequently compare their performance with classical machine learning models.

to prepare data for use in LLMs we used KKNimputer from the Sklearn library for imputing missing values which uses the k-nearest neighbors algorithm to infer the missing values(k = 5 ).

To estimate the power of LLMs and consider their ability to process different types of data simultaneously in the text format, we didn’t do feature selection, normalizing, and sampling.

the final step of data pre-processing for LLMs was converting a tabular dataset into text. for the age feature, we used the existing number as age. To simulate the real-life situation of conventional patient history, we just consider the present signs and symptoms of patients. For example, if a patient came with just 2 signs of cough and dyspnea, we didn’t consider the other 74 features in his history as negative results. for numerical values, we used their normal ranges and described them with 3 sentences. if the numerical feature was higher than the normal range we wrote the feature was higher than the normal range, if it was lower than the normal range we wrote it was lower than the normal range and if it was in the normal range we wrote it is in the normal range.

For the utilization of large language models, which inherently require text as input, a conversion process was imperative. Our original dataset, organized in tabular form, needed to be transformed into textual data to create meaningful narratives for patients. Given the inherent limitations of large language models in processing numerical data accurately, we strategically divided numerical features into categories of higher-than-normal and lower-than-normal. These categorical representations were then converted into text, facilitating a more effective integration of numerical information into the language model. This comprehensive transformation of our dataset sets the stage for a nuanced comparison between the performance of large language models and classical machine learning approaches.

### Model implementation

our goal in the study was to use an open-source model considering sensitive healthcare information. In this study, we used zephyr-7b-beta which is a GPT-like large language model with 7 billion parameters.it is Trained on a mixture of publicly available and synthetic data and can be used for NLP tasks. it is also a decoder-only model that is used for text-generation tasks.

another LLm that is used in this paper is bart-large-mnli that is developed by Facebook. this model is a Bart-based model that can be used for zero-shot classification. By zero-shot classification, we mean that the model can predict whether a patient will survive or die based on only one prompt.

vector similarity search is done by using FAISS. By vector similarity search we mean that when there is a dataset in the form of text and it is embedded in a vector database, it is possible to find the most relevant text in the database to the injected query.

#### Fine-tuning LLMs

fine-tuning an LLM is usually considered time-consuming and expensive, recently some methods have been introduced to lower costs. We aimed to improve the fine-tuning process for large language models by using recent efficient methods.

QLoRA is a novel and efficient fine-tuning approach designed to reduce memory usage significantly.QLoRA introduces innovations like a new 4-bit data type, double quantization to reduce memory footprint, and paged optimizers to manage memory spikes. The approach demonstrates superior performance across various instruction datasets, model types (LLaMA, T5), and scales (e.g., 33B and 65B parameter models), showcasing state-of-the-art results through fine-tuning on a small high-quality dataset[[ref](https://arxiv.org/abs/2305.14314)].

We enabled 4-bit loading, employed double quantization, utilized a quantization type of "nf4," and specified the compute data type as torch.bfloat16 for our model, zephyr-7b-beta. We then created a tokenizer using the pre-trained model and generated a causal language model, leveraging quantization configurations from the specified BitsAndBytesConfig ("bnb\_config"). Additionally, we assigned the model to the device "cuda:0" for processing. we enabled gradient checkpointing for our model and prepared it for knowledge bit (KBit) training using the prepare\_model\_for\_kbit\_training function from the PEFT (Post-training Energy-Aware Fine-Tuning) library.

We utilized the PEFT library to create a LoraConfig object with specified parameters, including an 8-layer model with Lora attention, targeted projection modules, a dropout rate of 0.05, no bias, and a task type of 'CAUSAL\_LM'. Subsequently, we generated a PEFT model based on this configuration.

For setting up a training pipeline for our language model, we used the Transformers library. We initialized a trainer with our specified model, training dataset, and training arguments. The training configuration included a per-device batch size of 1, gradient accumulation steps of 4, 2 warm-up steps, a maximum of 10 training steps, a learning rate of 2e-4, and enabled mixed-precision training with fp16. The chosen optimizer was "paged\_adamw\_8bit". Our data collator was configured for language modeling without masked language modeling (mlm=False). We carried out the training process, and we disabled the model's caching during training in the "outputs" directory to suppress warnings.

We specifically explored QLORA, combined with the bitsanbdbytes library, to enhance language models while requiring fewer resources.

#### Vector similarity search

we transformed the training dataset into embeddings utilizing the Hugging Face Embeddings from the Langchain library. Subsequently, we established a vector database for the training set utilizing FAIS[[ref](https://arxiv.org/abs/2401.08281)]S. we used embedings for each patient in the test set too. Our approach involved conducting a similarity search for each patient within the test set, allowing us to pinpoint the most pertinent result as the patient's predicted outcome.

#### Zero-shot classification

Zero-shot classification is the type of approach in prompt engineering in which the prompt is given to the model without any training. this approach is used in transfer learning in that the model that is used for different purposes would be used instead of fine-tuning a new model to reduce the cost of training a new model. to do Zero-shot classification we used Bart-large-mnli model from huggingface which at the date of writing this paper is the most downloaded model in the zero-shot classification category. we give the model of each patient's history as input to predict if the patient will die or survive and then store the results.

# Results

## Seven CML

In the initial set of experiments, various machine learning models were employed to evaluate their performance in internal and external validation scenarios. Logistic regression, a widely used linear classification algorithm, demonstrated an accuracy of 70% in internal validation, showcasing balanced precision and recall values at 0.70 and 0.71, respectively. Support Vector Machines (SVM) outperformed other models with an accuracy of 72%, exhibiting high precision at 0.74 and moderate recall at 0.68. Decision tree, k-nearest Neighbor (KNN), random forest, neural networks, and boosting algorithms also contributed to the analysis, each presenting distinct trade-offs in terms of accuracy, precision, recall, specificity, F1 score, and AUC. These diverse model performances highlight the nuanced nature of the dataset and the need for a comprehensive evaluation approach.

## Fine-tuning LLMs and vector similarity search

In contrast, the section on vector similarity search and fine-tuned Language Model (LLM) demonstrated exceptional outcomes, achieving perfect scores across all metrics. Vector similarity search and fine-tuned LLM displayed 100% accuracy, precision, recall, specificity, F1 score, and AUC, suggesting these methods' potential for tasks involving semantic similarity and language understanding. These results underscore the robustness and effectiveness of these specialized techniques in certain applications, potentially opening avenues for further research and exploration in specific domains where these methods excel.

## Zero-shot classification

On a different note, zero-shot classification exhibited significantly lower performance, with an accuracy of only 19%. This model struggled to correctly classify instances, resulting in low precision, recall, and F1 score values. The model's reliance on specificity suggests a biased classification approach, emphasizing the challenges associated with zero-shot learning tasks. Understanding the limitations and constraints of zero-shot classification is crucial for researchers and practitioners to make informed decisions regarding its applicability in various contexts.

In conclusion, the paper presents a comprehensive analysis of diverse machine learning models, showcasing their strengths and weaknesses in different validation scenarios. While traditional models such as logistic regression, SVM, and decision trees provide solid performance, specialized techniques like vector similarity search and fine-tuned LLM exhibit remarkable capabilities in specific tasks. However, the challenges associated with zero-shot classification emphasize the importance of selecting appropriate models based on the specific characteristics and requirements of the data at hand. Researchers and practitioners should carefully consider these findings when designing and implementing machine learning solutions for real-world applications, recognizing the nuanced landscape of model performance.

# Discussion

Topics I like to discuss:

یه بررسی کلی از مقاله

ظهور مدل های زبانی بزرگ و چت جی پی تی

اهمیت کوید و احتمال امرج یه همچنین ویروس هایی دوباره

این که ال ال ام ها به دیتا پریپراسسینگی مثل مدل های کلاسیک نیازی ندارند و از این لحاظ وقت کمتری می گیرند

نیاز به فاین تیون کردن کامل یه مدل نیست بلکه به صورت peft هم میشه انجام داد.

این که این مدل ها چجوری می توننن کمک کنن

زیروشات

فاین تیونینگ

مدل های کلاسیک

وکتور دیتابیس و ضعفش

مقایسه ی این ها

تاثیر این مدل ها توی کلینیک و اینکه با دیتای ان ادمیشن میشه میزان بستری و مورتالیتی رو کاهش داد.

# References

# Pictures



