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

GitHub repository:

[mohammad-gh009/ehran\_covid\_Large-Language-Models-vs-Classical-Machine-learning (github.com)](https://github.com/mohammad-gh009/ehran_covid_Large-Language-Models-vs-Classical-Machine-learning)

Running Title:

Mohammadreza Ghaffarzadeh-Esfahani1,2, 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- Medical student, Isfahan University of Medical Sciences, Isfahan, Iran

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 |  | mreghafarzadeh@gmail.com  ORCID: 0009-0009-9322-5471 | (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(1). LLMs, having undergone extensive training on vast datasets, excel in numerous natural language processing tasks, including language generation, machine translation, and question-answering(2). 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(3).

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 (4). 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(5). 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(5).

The power of LLMs in using clinical text, such as discharge summaries and ward notes is previously shown in detail(6). However, the LLM performance in handling structured data and determining 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(7).

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

## 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 internal validation dataset into a train set and a test set using a test size ratio of 0.2. 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(8, 9). For optimal model performance, the dataset underwent normalization using StandardScaler(10). These preprocessing steps were executed independently for the training, test, and external validation sets, 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 20% (test\_size=0.2), allocating 80% of the data for training.

The features utilized in this study include "In-hospital Mortality," "ICU admission," and "Intubation," with a focus solely on "hospital mortality" as the targeted feature. Of the 81 features initially available, 76 were employed for training, comprising 53 categorical features and the remainder numerical values. During the data wrangling process, two duplicate features were dropped.

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(10, 11). 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(12). 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 five classical machine learning algorithms, namely Logistic Regression, Support Vector Machine (SVM), Decision Tree, k-nearest Neighbors (KNN), Random Forest, plus a 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 KNNimputer from the Sklearn library for imputing missing values which uses the k-nearest neighbors algorithm to infer the missing values(k = 5 ).

To evaluate the efficacy of Large Language Models (LLMs) and assess their capacity to process diverse data types concurrently in textual format, we refrained from conducting feature selection, normalization, or sampling. The final preprocessing step for LLMs involved transforming the tabular dataset into textual form. For instance, regarding the age feature, we retained the existing numerical value as the age. To mimic real-life scenarios of patient history, we solely focused on present signs and symptoms, disregarding other historical features if absent. For numerical values, we depicted them within three sentences, indicating whether they were within, above, or below normal ranges. This approach facilitated the utilization of LLMs, which inherently demand textual input.

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

In our study, our primary objective was to utilize open-source models while prioritizing the confidentiality of sensitive healthcare data. 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.

To facilitate vector similarity search, we employed FAISS. This method involves embedding textual datasets into a vector database, enabling the retrieval of the most relevant text entries based on injected queries.

#### 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(13).

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 FAISS(14). 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.

## Ethical Consideration

In the present study, we upheld principles of patient privacy and confidentiality by ensuring that all data utilized in our study is anonymized and securely handled in compliance with relevant regulations and guidelines. Additionally, we prioritized transparency by clearly outlining the data sources and methodologies employed in our analysis. Moreover, we emphasized the importance of fairness and equity in model development and deployment, striving to mitigate biases that may arise from the data or algorithmic decisions. Furthermore, we recognized the potential implications of our findings on healthcare decision-making and public health policies, advocating for responsible and informed utilization of predictive models to avoid unintended consequences or harm to individuals or communities. Through these ethical considerations, we aim to contribute to the advancement of machine learning research while prioritizing the well-being and rights of the individuals whose data forms the basis of our study.

# Results

## CML(Classical Machine Learning ) Methods

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 negligible 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

In the present study, we used the dataset of the fourth hospital as external validation which could be considered as a non-trained data set and the accuracy of the training will be considerable. Zero-shot classification showed poor performance, achieving only 19% accuracy. It struggled in correctly classifying instances, with negligible precision, recall, and F1 scores. Biased specificity indicates challenges in zero-shot learning tasks, emphasizing the importance of understanding its limitations for informed decision-making. Vector similarity search and fine-tuned Language Models (LLM) achieved perfect scores in accuracy, precision, recall, specificity, F1 score, and AUC. This suggests their potential for tasks involving semantic similarity and language understanding, highlighting their robustness and effectiveness in specific domains, and encouraging further exploration.

Historical text archives, now increasingly digitized, offer a wealth of information but pose challenges for efficient exploration. Text mining methods can assist by automatically identifying semantic information like concepts, synonyms, and relationships within texts, and they enables search systems to suggest synonyms, explore mentioned concepts, and isolate documents with specific concept relationships. However, applying text mining and large language models to patients historical texts is challenging due to vocabulary, terminology, and language style (1). Beyond this, in clinical settings, we face unstructured data. Structured data, is acquired individually in controlled clinical practice or research and usually is outcome-focused; In contrast, unstructured data encompasses vast information from larger populations (2). Structured data can be effectively managed under ideal conditions, yet in real-world scenarios, most data acquired from clinical measures such as history taking, physical examination and laboratory tests are unstructured.

Extracting information from clinical unstructured data is crucial for modifying diagnoses, decision-making plans, and patient follow-ups. One notable application of data extraction from clinical records is in early warning systems aimed at enhancing patient safety. In a study, Chengyin Ye et al designed a Real-Time Early Warning System (EWS) for monitoring inpatient mortality risk using electronic medical record (EMR) data. Data from two hospitals spanning totaling 54,246 inpatient admissions, were analyzed, of which 2.30% resulted in intrahospital deaths. Various machine learning methods were explored, with the tree-based random forest method selected for developing the predictive application. The EWS algorithm stratified patients into risk groups and accurately predicted inpatient mortality probability, achieving a c-statistic of 0.884 in prospective validation Clinical features such as diagnoses, vital signs, and laboratory results were crucial predictors in the model and it successfully identified high-risk patients well in advance, allowing timely interventions. The study demonstrates the EWS's potential to alert clinicians in real-time, facilitating timely interventions and improving patient outcomes in medical facilities (3).

LLMs based alerting systems can have a crucial role in the final outcomes of the patients. A study retrospectively analyzed clinical data from patients with activated rapid response systems (RRS) in surgical wards of a tertiary university hospital. Comparisons were made between pre-RRS, RRS without automatic alerting system (AAS), and RRS with AAS periods. Post-RRS implementation, in-hospital mortality decreased significantly, with a subsequent rise in RRS activations after AAS integration. Patients with RRS activation and unplanned ICU admissions demonstrated improved ICU duration and mortality outcomes. The findings suggest that real-time AAS based on electronic medical records (EMRs) aids in promptly identifying unstable patients, potentially leading to improved patient outcomes through early detection and intervention with RRS. This underscores the importance of leveraging technology to enhance clinical decision-making and patient care in hospital settings(4).

In a study, J E Rasmussen et al, conducted a program, LogStory, for semantic analysis of clinical narratives in PROMED, categorizes diagnoses into disease classes from free text, extracts disease manifestations using lexical parsing and fuzzy set operations, shows significant correlations with signs, symptoms, and treatments, recovers disease-specific knowledge by analyzing distinct clinical findings, accounts for disease evolution, and offers potential for knowledge acquisition and clinical research in natural language processing(5).

Missing data are prevalent across primary care research methodologies, including randomized trials, observational studies, and quality improvement initiatives. These data gaps, such as patients lost to follow-up or incomplete records, can skew study results, increasing bias and reducing statistical power. High rates of missing data have been linked to compromised validity and potentially misleading conclusions in primary care studies(6). One of the strength points of this study is dealing with missing values that can interfere with the accuracy of the studies. In the preset study we utilized the Multiple Imputation by Chained Equations (MICE) to address missing values which uses iterative prediction for each feature and then imputes it.

In the context of medical research, comparing large language models (LLMs) with classical machine learning (ML) methods involves several aspects of predictive power, including accuracy, model performance, effective models s, data size, sample size needed for training, reliability, transparency and resource usage. Classical ML models often require significant feature engineering, where domain experts manually select and engineer features from the structured data. This process can be time-consuming and may require a deep understanding of the domain. LLMs, on the other hand, can automatically learn representations from raw data without explicit feature engineering. They can process structured data in its raw format, eliminating the need for feature engineering.

In the aspect of data representation, classical ML models typically require structured data to be converted into numerical representations (vectors) for processing. This often involves techniques like embedding or other numerical representations while LLMs can directly process raw data in its original format, including text, sequences, or structured data, without the need for explicit numerical encoding which indicates the promising future of utilizing LLMs as medical assistants.

Classical ML models often require manual selection of model architecture, tuning of hyperparameters, and regularization techniques to optimize performance. The choice of model architecture (e.g., decision trees, SVMs, logistic regression) depends on the specific characteristics of the data and problem.

LLMs, once pre-trained, can be fine-tuned on specific tasks with relatively fewer hyperparameters to tune. They often have a high degree of flexibility and can capture complex patterns in the data due to their deep architecture and large number of parameters.

In the aspect of interpretability, classical ML models such as decision trees or logistic regression models are often more interpretable, as they provide explicit rules or coefficients that indicate the importance of different features. LLMs are generally less interpretable due to their complex architecture and the lack of explicit rules. However, techniques such as attention mechanisms can provide some insights into which parts of the input are important for making predictions.

Considering data efficiency, LLMs typically require large amounts of data for pre-training to achieve optimal performance. However, once pre-trained, they can often generalize well to downstream tasks even with limited task-specific data. Classical ML models may require less data for training, especially when the feature space is well-defined and limited. However, they may struggle to generalize to new tasks or domains without sufficient data.

In the context of predicting patient outcomes based on medical history, both approaches can be effective depending on factors such as the size and quality of the dataset, the complexity of the relationships between features and outcomes, and the interpretability requirements. Classical ML models may be preferred in scenarios where interpretability is crucial and the feature space is well-defined. On the other hand, LLMs may offer advantages in scenarios where feature engineering is challenging or when dealing with unstructured or semi-structured data.

# References

1. Karabacak M, Margetis K. Embracing Large Language Models for Medical Applications: Opportunities and Challenges. Cureus. 2023;15(5):e39305.

2. Dathathri S, Madotto A, Lan J, Hung J, Frank E, Molino P, et al. Plug and play language models: A simple approach to controlled text generation. arXiv preprint arXiv:191202164. 2019.

3. Vaid A, Lampert J, Lee J, Sawant A, Apakama D, Sakhuja A, et al. Generative Large Language Models are autonomous practitioners of evidence-based medicine. arXiv preprint arXiv:240102851. 2024.

4. Zhang D, Yin C, Zeng J, Yuan X, Zhang P. Combining structured and unstructured data for predictive models: a deep learning approach. BMC Med Inform Decis Mak. 2020;20(1):280.

5. Sedlakova J, Daniore P, Horn Wintsch A, Wolf M, Stanikic M, Haag C, et al. Challenges and best practices for digital unstructured data enrichment in health research: A systematic narrative review. PLOS Digit Health. 2023;2(10):e0000347.

6. Zhou H, Gu B, Zou X, Li Y, Chen SS, Zhou P, et al. A survey of large language models in medicine: Progress, application, and challenge. arXiv preprint arXiv:231105112. 2023.

7. Wornow M, Xu Y, Thapa R, Patel B, Steinberg E, Fleming S, et al. The shaky foundations of large language models and foundation models for electronic health records. NPJ Digit Med. 2023;6(1):135.

8. van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software. 2011;45(3):1 - 67.

9. Buck SF. A Method of Estimation of Missing Values in Multivariate Data Suitable for use with an Electronic Computer. Journal of the Royal Statistical Society Series B (Methodological). 1960;22(2):302-6.

10. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research. 2011;12:2825-30.

11. Simpson L, Combettes PL, Müller CL. c-lasso--a Python package for constrained sparse and robust regression and classification. arXiv preprint arXiv:201100898. 2020.

12. Liu XY, Wu J, Zhou ZH. Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern B Cybern. 2009;39(2):539-50.

13. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems. 2024;36.

14. Douze M, Guzhva A, Deng C, Johnson J, Szilvasy G, Mazaré P-E, et al. The faiss library. arXiv preprint arXiv:240108281. 2024.