International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

LLM is All You Need: How Do LLMs Perform on Prediction and Classification Using Historical Data

Yuktesh Kashyap[1], Amrit Sinha[2]

[1,2]Beige Bananas Inc.

Abstract

This study investigates the utility of large language models (LLMs) in performing traditional machine learning tasks such as prediction, and explores the potential of refinement architectures to enhance their effectiveness in these roles. Utilizing the Titanic survival dataset, we conducted a com- parative analysis using both conventional machine learning tools and LLM-based approaches. Our findings indicate that while LLMs differ fundamentally from traditional ML models in prediction tasks, there exist specific architectural modifications, termed Thought Refinement Architectures, which can significantly improve their performance. These results highlight the potential for inte- grating LLMs into traditional ML workflows, thereby expanding their applicability and enhancing predictive accuracy.

Keywords: AI, Machine Learning, Ensemble, Thought Refinement Architecture.

1 Introduction

Traditional machine learning (ML) tasks have long been the cornerstone of artificial intelligence (AI) research and application, encompassing a wide range of predictive modeling and pattern recognition tasks. These tasks often involve the use of structured data, such as numerical features or categorical variables, to train models that can make accurate predictions or classifications. Over the past decade, there has been a notable shift in the landscape of AI with the emergence of large language models (LLMs). LLMs are a class of deep learning models that leverage vast amounts of textual data to learn complex patterns and relationships in language. Unlike traditional ML models, which rely heavily on feature engineering and domain-specific knowledge, LLMs are trained end-to-end on raw text data using techniques such as self- attention mechanisms and transformer architectures. The rise of LLMs, exemplified by models such as OpenAI's GPT (Generative Pre-trained Transformer) series and Google's BERT (Bidirectional Encoder Representations from Transformers), has revolutionized many natural language processing (NLP) tasks. These models have achieved state-of-the-art performance on a wide range of benchmarks, including language understanding, generation, translation, and summarization. Despite their success in NLP, the application of LLMs to traditional ML tasks, such as prediction and classification, has been relatively limited. Traditional ML models often outperform LLMs on tasks involving structured data due to their ability to capture domain-specific features and relationships effectively. However, recent research has shown promising results in adapting LLMs for these tasks through architectural modifications and ensemble techniques. This study aims to explore the potential of

LLMs in traditional ML tasks and investigate the efficacy of refinement architectures in enhancing their performance. By leveraging insights from both the traditional ML and NLP domains, we seek to bridge the gap between these two paradigms and unlock new opportunities for AI-driven predictive modeling.

1.1 Problem Statement

Despite the remarkable success of large language models (LLMs) in natural language processing (NLP) tasks, there exists a notable research gap regarding their application to traditional machine learning (ML) tasks, such as prediction and classification. This gap arises from the inherent differences between structured data, which is commonly used in traditional ML tasks, and unstructured text data, which forms the basis of LLMs' training data. This question has become increasingly relevant in the context of various projects undertaken by our company, Beige Bananas Inc. As an AI company operating in the intersection of industry and academia, we have encountered numerous scenarios where clients and practitioners express interest in leveraging LLMs for predictive modeling tasks traditionally handled by conventional ML techniques. However, the challenge lies in adapting LLMs, which are primarily designed for processing unstructured text, to effectively handle structured data and excel in tasks such as regression, classification, and clustering. While LLMs have demonstrated impressive capabilities in understanding and generating human-like text, their performance on structured data remains a subject of investigation and experimentation. This question has arisen not only from our internal research endeavors but also from interactions with clients and industry practitioners who seek innovative solutions to their predictive modeling challenges. The demand for LLMs in traditional ML tasks reflects a growing interest in harnessing the power of advanced AI technologies to augment and enhance existing data- driven workflows. Addressing this research gap is essential for unlocking the full potential of LLMs and advancing the state-of-the-art in predictive modeling across diverse domains. By bridging the divide between structured and unstructured data paradigms, we aim to empower organizations with cutting- edge AI solutions that leverage the strengths of both traditional ML techniques and state-of-the-art LLMs.

1.2 Significance

The significance of this research lies in its potential to bridge the gap between two distinct paradigms in artificial intelligence (AI): traditional machine learning (ML) and large language models (LLMs). By exploring the application of LLMs to traditional ML tasks, particularly predictive modeling, this study addresses a critical research gap and offers insights that could have far-reaching implications for both academia and industry. 1. Advancing AI Integration: The integration of LLMs into traditional ML workflows represents a significant advancement in AI technology. By leveraging the strengths of LLMs in understanding natural language and complex patterns, while also harnessing the structured data pro- cessing capabilities of traditional ML models, this research has the potential to push the boundaries of AI integration and enable more sophisticated data-driven decision-making systems. 2. Enhancing Pre- dictive Accuracy: The findings of this study could lead to improvements in predictive accuracy across a wide range of domains and applications. By evaluating LLMs' performance on tasks such as survival pre- diction using the Titanic dataset, and exploring architectural adaptations and ensemble approaches, we aim to identify strategies for enhancing predictive modeling effectiveness. These insights could empower organizations to make more informed decisions and optimize business processes. 3. Enabling Cross-

Domain Applications: The insights gained from this research could facilitate the adoption of LLMs in diverse domains and industries. From healthcare and finance to marketing and cybersecurity, the abil- ity to leverage LLMs for predictive modeling tasks opens up new possibilities for AI-driven innovation and optimization. By demonstrating the feasibility and effectiveness of integrating LLMs into existing ML workflows, this study paves the way for cross-domain applications of advanced AI technologies. 4. Informing Best Practices: Practical implications of this research extend to the development of best prac- tices for incorporating LLMs into predictive modeling pipelines. By providing insights into architectural adaptations, ensemble approaches, and performance evaluation metrics, this study equips practitioners with the knowledge and tools needed to harness the full potential of LLMs in real-world applications. These best practices can guide the implementation of AI solutions and drive tangible value for organi- zations seeking to leverage cutting-edge technologies. 5. Contributing to Academic Discourse: Finally, this research contributes to the academic discourse surrounding the intersection of NLP and traditional ML. By disseminating our findings through academic publications and conference presentations, we aim to stimulate further research and collaboration in this rapidly evolving field. By sharing methodologies, experimental results, and lessons learned, we enrich the collective knowledge base and foster continuous innovation in AI research. In summary, the significance of this research lies in its potential to advance AI integration, enhance predictive accuracy, enable cross-domain applications, inform best practices, and contribute to academic discourse. By addressing the research gap regarding the use of LLMs in traditional ML tasks, this study lays the groundwork for transformative advancements in AI-driven decision-making and data analytics.

2 Studying the Use of LLM for Predictive Tasks and Compar-ison against Traditional ML Models

We did not find any direct literature specifically focused on this study. However, there are several papers tangentially related to our topic, which provide useful insights and background information. These related works are referenced towards the end of this section.

2.1 Methodology

In this paper, we are using the Titanic dataset to build a predictive model that determines the likelihood of a passenger surviving the Titanic disaster https://www.kaggle.com/competitions/titanic. The dataset contains various features such as age, gender, passenger class, fare, and other relevant information about the passengers. By applying machine learning algorithms, we aim to analyze these features and develop a model that accurately predicts the survival chances of individuals based on their attributes. We use this well known dataset and competition on Kaggle to compare the outputs of traditional ML Models and Open AI's GPT-4 Model. We later provide some insights into our ongoing work on what we are calling Thought Refinement Architectures to show a significant improvement on existing results

2.2 Dataset Description

The Titanic dataset contains data on the passengers who were aboard the RMS Titanic, which sank on its maiden voyage in 1912 after hitting an iceberg. The dataset includes various attributes for each passenger, which can be used for predictive modeling and analysis. Features : 1. PassengerId: Unique identifier for each passenger. 2. Survived: Survival status (0 = No, 1 = Yes) - This is the target variable.

3. Pclass: Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd) - A proxy for socio-economic status. 4. Name: Name of the passenger. 5. Sex: Gender of the passenger (male, female). 6. Age: Age of the passenger in years. Fractional ages indicate the passenger was less than one year old. If the age is estimated, it is in the form of xx.5. 7. SibSp: Number of siblings and/or spouses aboard the Titanic. 8. Parch: Number of parents and/or children aboard the Titanic. 9. Ticket: Ticket number. 10. Fare: Passenger fare (in British pounds). 11. Cabin: Cabin number. 12. Embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton).

2.3 Pre-processing data to Generate Predictions

Missing Value Treatment • Age: The Age feature had missing values that we imputed using the median age within passenger classes and gender. • Embarked: There were a few missing values in the Embarked feature, which we filled with the most common embarkation point (mode). • Cabin: The Cabin feature had many missing values. We created a new feature indicating whether a passenger's cabin number was known. Feature Engineering • Family Size: We created a new feature called FamilySize by combining SibSp (number of siblings/spouses aboard) and Parch (number of parents/children aboard): FamilySize
= SibSp + Parch + 1. • IsAlone: We created a new binary feature indicating whether a passenger was alone: IsAlone = 1 if FamilySize is 1, otherwise IsAlone = 0. • Title: We extracted titles from the Name feature (e.g., Mr., Mrs., Miss, Master) and used them as a new feature to provide insights into the passenger's social status. • Fare Bins: We created categorical bins for the Fare feature to handle outliers and provide a clearer understanding of fare ranges. • Age Bins: We created age categories or bins to reduce the effect of outliers and simplify the model. Encoding Categorical Variables •Sex: We converted the Sex feature into numerical format (e.g., 0 for male, 1 for female). • Embarked: We used one-hot encoding to convert the Embarked feature into numerical format. • Pclass: We converted the Pclass feature into categorical type and used one-hot encoding. Scaling Numerical Features • Fare: We scaled the Fare feature using standardization (subtract the mean and divide by the standard deviation).

· Age: We scaled the Age feature similarly. Dropping Irrelevant Features • PassengerId: We dropped PassengerId as it was just an identifier and did not hold predictive value. • Name: After extractingtitles, we dropped the Name feature. • Ticket: We dropped the Ticket feature as it did not provide useful information. Handling Outliers • We identified and handled outliers in features like Fare and Age. Outliers were capped, transformed, or binned to mitigate their impact.

2.4 Models Used and Evaluation Methods

In our study, we employed five different machine learning algorithms to predict survival on the Titanic dataset. These models are: 1. Logistic Regression 2. Random Forest Classifier 3. Gradient Boosting Classifier 4. Support Vector Machines (SVM) 5. 5. K-Nearest Neighbors (KNN)

2.5 Performance Evaluation Metrics

To evaluate the performance of these models, we used the following metrics:

1. **Accuracy**: Accuracy is the ratio of correctly predicted instances to the total number of instances. It provides an overall measure of the model's performance..

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False

Negatives.

2. **Precision**: Precision is the ratio of correctly predicted positive observations to the total predictedpositives. It is useful when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

3. **Recall (Sensitivity)**: Recall is the ratio of correctly predicted positive observations to all the actual positives. It is crucial when the cost of false negatives is high.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

4. **F1 Score**: The F1 Score is the harmonic mean of precision and recall. It provides a single measureof the model's performance, balancing precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5. **Confusion Matrix**: A confusion matrix is a table used to describe the performance of a classification model. It shows the counts of true positive, true negative, false positive, and false negative predictions.

For each model, we performed cross-validation to ensure robustness and reliability of our results. Cross-validation helps in mitigating overfitting and provides a more generalized performance measure. Results are in Table 1 below

#	Model	Test Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	0.810056	0.77027	0.77027	0.77027
1	Random Forest	0.826816	0.786667	0.797297	0.791946
2	Gradient Boosting	0.815642	0.80597	0.72973	0.765957
3	Support Vector Machine	0.826816	0.811594	0.756757	0.783217
4	K-Nearest Neighbors	0.810056	0.785714	0.743243	0.763889

Table 1: Performance metrics of various models

Additionally, we plotted the confusion matrices for each model to visually inspect the performance and understand the distribution of prediction errors. Confusion Matrices for Logistic Regression(base model for other ML models) and Random Forest Model(best performing Model) Plotted in Figure 2. We would be using the Random Forest Model which achieves the highest F1 score for Benchmarking.

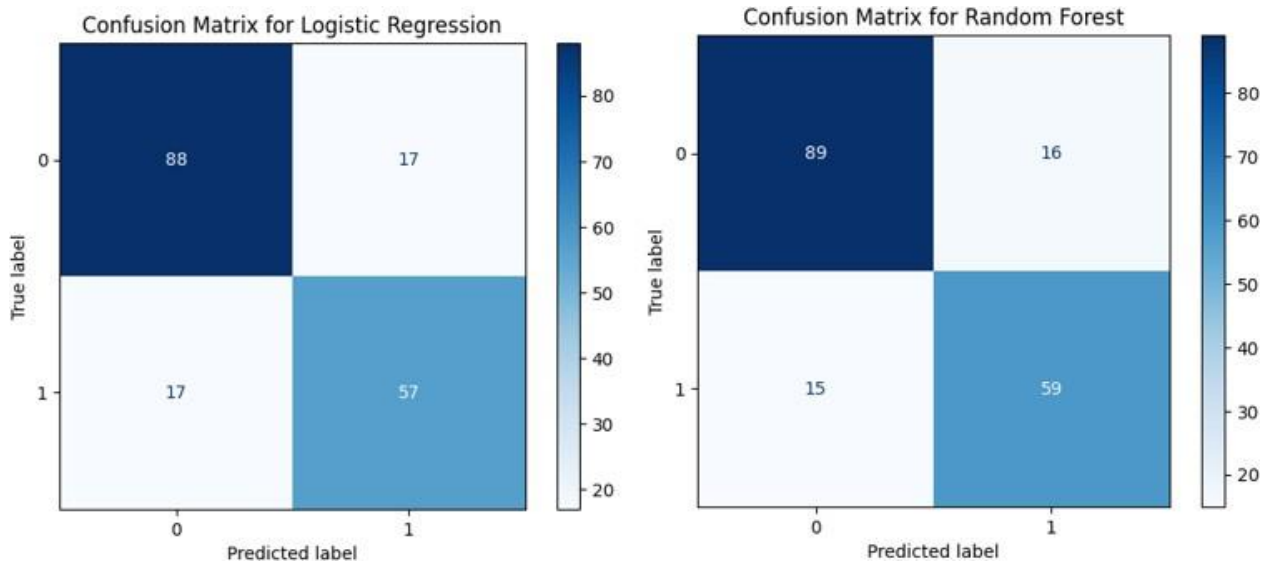

Figure 2

3 Utilizing LLM for Prediction

In the following section, we explore the use of a Large Language Model (LLM) for predicting passengersurvival on the Titanic. While traditional machine learning models like logistic regression, random forests, and support vector machines rely on structured numerical and categorical data,

LLMs are designed to process and understand natural language. By leveraging the advanced capabilities of LLMs, such as those built on the GPT architecture, we aim to assess their performance in a classification task traditionally handled by structured data models. This approach involves transforming the dataset into a format suitable for LLMs and comparing their predictive performance with classical models. We will eval the LLM's ability to capture complex patterns and interactions within the dataset, potentially providing novel insights and improved predictions. In this section, we will: 1. Preprocess and format the datafor LLM input. 2. Implement an LLM-based prediction model. 3. Compare the LLM's performancewith traditional models using standard evaluation metrics. This exploration will help us understand the potential and limitations of applying language models to structured data prediction tasks.

3.1 Data Processing

To utilize a Large Language Model (LLM) for predicting passenger survival on the Titanic, we need to transform the structured dataset into a format suitable for natural language processing. This involves converting each passenger's data into descriptive statements that the LLM can interpret. The following steps outline the necessary data processing: 1. Extract Features: Identify and extract the relevant features from the dataset, such as name, age, gender, passenger class, family size, and survival status.

Construct Descriptive Sentences: Combine the extracted features into natural language statements. For instance, for a passenger named John who is a 40-year-old male traveling alone, the sentence could be: o "John is a 40-year-old male unaccompanied and he did not survive." 3. Handle Missing Values: Ensure that all necessary features are present in each sentence. If any critical information is missing(e.g., age), decide on a strategy to handle such cases, such as using placeholders or estimating values. 4. Label Encoding: Convert the survival status into a clear, consistent format within the sentences. Use "survived" or "did not survive" to indicate the outcome clearly. 5. Concatenate Features: Formulate sentences for all passengers in the dataset, ensuring a consistent structure and clear depiction of each feature. By transforming the structured data into these descriptive statements, we enable the LLMto process the information as it would a natural language text, leveraging its capability to understandand infer complex patterns from language. This data processing approach not only prepares the datafor LLM input but also encapsulates the essential details in a format that aligns with the strengths of language models.

3.2 sAnalysis Generation Using Chain Of Thought Prompting

In this research, we aimed to enhance the interpretability and accuracy of predictive modeling for Titanic passenger survival by incorporating advanced natural language processing techniques. After preprocess- ing the data and performing the necessary feature engineering steps, we leveraged a chain of thought prompting approach using OpenAI's GPT-4 to generate insightful analyses for passenger survival pre- dictions. Methodology : 1. Data Preparation: o The Titanic dataset was processed by filling missing values, extracting relevant features, and performing one-hot encoding for categorical variables. o Nu- merical features were scaled, and the data was split into training and test sets. 2. Feature Engineering:o New features such as FamilySize and IsAlone were created to capture familial relationships. o Titles were extracted from passenger names to provide additional social context. 3. Prompt Engineering: o Each test case was transformed into a readable sentence encapsulating all the relevant features. o For instance: "A male passenger, aged 0.23, of middle class, with a fare of 0.12. Embarked from Southamp ton. Has a family size of 2 and is not alone. Holds the title of Mr. Will

this passenger survive?" 4. Chain of Thought Prompting: o A concatenated prompt comprising all test sentences was created to ensure comprehensive analysis. o The observations were supplied to have a semantic meaning associated with them 5. Analysis Generation: o The concatenated prompt was fed into OpenAI's GPT-4 model using the openai.Completion.create method. o The model generated an analysis detailing the factors influencing survival predictions for each passenger. 6. Evaluation: o The generated insights were evaluated based on their alignment with known survival factors and their utility in enhancing the model's interpretability.

3.3 sResults

The chain of thought prompting approach yielded rich, contextual analyses for each passenger's survival prediction. For example, the model was able to highlight the impact of socio-economic status, family size, and embarkation point on survival odds. The generated insights provided a deeper understanding of the model's decision-making process, allowing for more transparent and interpretable predictions. By integrating chain of thought prompting with traditional machine learning techniques, we demonstrated a novel approach to bridging the gap between predictive accuracy and interpretability in survival analysis. This method holds promise for applications in various domains where understanding the rationale behind predictions is crucial.The following metrics were computed to evaluate the performance of our approach:

- Precision : 0.756
- Recall : 0.532
- F1 Score : 0.624

These metrics indicate a balanced performance, with a particular strength in the precision of the predic- tions, which aligns with the model's ability to accurately identify true positives. In conclusion, the chain of thought prompting approach, when combined with robust data preprocessing ansd feature engineering, can significantly enhance the interpretability and reliability of predictive models. Future work will focus on refining this methodology and exploring its applicability to other datasets and predictive tasks

3.4 Thought Refinement

1. Initial Chain of Thought Prompting:

- The process begins by transforming test cases into readable sentences encapsulating all rele- vant features.
- Example sentence: "A male passenger, aged 0.23, of middle class, with a fare of 0.12. Em- barked from Southampton. Has a family size of 2 and is not alone. Holds the title of Mr.Will this passenger survive?"
- These sentences are fed into an LLM to generate an initial analysis.

2. Thought Refinement:

- The initial outputs from the chain of thought prompting are then revisited by the LLM.
- A second prompt is created to reevaluate and refine the initial analysis, aiming to correct any misjudgments and improve prediction accuracy.
- Example of refinement prompt: "Reevaluate the following analysis and provide a refined prediction: [Initial analysis]"

3. Evaluation Metrics:

- The effectiveness of thought refinement is evaluated using precision and recall metrics.

- Precision, Precision and recall are calculated in the same way as for ML Model considering LLM word output

3.5 Results Comparison Post Thought Refinement

To demonstrate the impact of the thought refinement process, we present the confusion matrices and the resulting precision and recall before and after the refinement. Presented below is the Initial Analysis Vs Thought Refinement Confusion Matrix

	Predicted Label	
True Label	**0**	**1**
0	138	19
1	52	59

Table 2: Initial Analysis Confusion Matrix

We compared the performance pre and post the Thought Refinement , we calculated:

- Precision: 0.756 (Initial) — 0.761 (Post Refinement)
- Recall: 0.532 (Initial) — 0.748 (Post Refinement)

	Predicted Label	
True Label	**0**	**1**
0	131	26
1	28	83

Table 3: Refined Analysis Confusion Matrix

4 Conclusion and Future Relevance

The refined analysis demonstrates a significant improvement in both precision and recall. Precision improved from 0.756 to 0.761, indicating a higher proportion of correctly identified positive predictions out of all positive predictions. Recall showed an even more substantial improvement from 0.532 to 0.748, reflecting a greater ability of the model to identify actual positive cases. This enhancement can be attributed to the LLM's capability to reassess and correct its initial judgments. The iterative process of thought refinement allows the model to identify and rectify errors made during the initial analysis, thereby improving overall predictive performance. There can be multiple possibilities to apply refinement and try multi LLM architectures to generate more accurate results. The final aim of the continuing research is to understand if there needs to be separation of machine learning model and associated costs and generative AI applications or both will converge in the future,

References

1. Rajpoot, Pawan Kumar, Ashvini Jindal, and Ankur Parikh. "Adapting llm to multi-lingual esg impact and length prediction using in-context learning and fine-tuning with rationale." *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing@ LREC-COLING 2024*,

2024.
2. Gat, Yair, et al. ”Faithful explanations of black-box nlp models using llm-generated counterfactuals.”
3. *arXiv preprint arXiv:2310.00603*, 2023.
4. Hasan, Md Rakibul, et al. ”LLM-GEm: Large language model-guided prediction of people’s empathy levels towards newspaper article.” *Findings of the Association for Computational Linguistics: EACL 2024*, 2024.
5. Azaria, Amos, and Tom Mitchell. ”The internal state of an llm knows when its lying.” *arXiv preprint arXiv:2304.13734*, 2023.
6. Ni, Lin, et al. ”Enhancing student performance prediction on learnersourced questions with sgnn-llm synergy.” *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, No. 21, 2024.
7. Schoenegger, Philipp, et al. ”Wisdom of the Silicon Crowd: LLM Ensemble Prediction Capabilities Match Human Crowd Accuracy.” *arXiv preprint arXiv:2402.19379*, 2024.
8. Meng, Chuan, et al. ”Query Performance Prediction using Relevance Judgments Generated by Large Language Models.” *arXiv preprint arXiv:2404.01012*, 2024.
9. Velásquez-Henao, Juan David, Carlos Jaime Franco-Cardona, and Lorena Cadavid-Higuita. ”Prompt Engineering: a methodology for optimizing interactions with AI-Language Models in the field of engineering.” *Dyna*, vol. 90, no. 230, pp. 9-17, 2023.
10. Schulhoff, S. ”The Prompt Report: A Systematic Survey of Prompting Techniques.” *arXiv e-prints*, 2024.
11. Ekin, Sabit. ”Prompt engineering for ChatGPT: a quick guide to techniques, tips, and best prac- tices.” *Authorea Preprints*, 2023.
12. Chang, Jonathan D., et al. ”Learning to generate better than your llm.” *arXiv preprint arXiv:2306.11816*, 2023.
13. Zeng, Xianlong, Fanghao Song, and Ang Liu. ”Similar Data Points Identification with LLM: A Human-in-the-loop Strategy Using Summarization and Hidden State Insights.” *arXiv preprint arXiv:2404.04281*, 2024.
14. Xiao, Tim Z., et al. ”Verbalized Machine Learning: Revisiting Machine Learning with Language Models.” *arXiv preprint arXiv:2406.04344*, 2024.
15. Nazary, Fatemeh, et al. ”XAI4LLM. Let Machine Learning Models and LLMs Collaborate for En-hanced In-Context Learning in Healthcare.” *arXiv preprint arXiv:2405.06270*, 2024.