An Efficient Retrieval-based Method for Tabular Prediction with LLM

Jie Wu and Mengshu Hou[*]
University of Electronic Science and Technology of China, Chengdu, China
jiewu@std.uestc.edu.cn, mshou@uestc.edu.cn

Abstract

Tabular prediction, a well-established problem in machine learning, has consistently garnered significant research attention within academia and industry. Recently, with the rapid development of large language models (LLMs), there has been increasing exploration of how to apply LLMs to tabular prediction tasks. Many existing methods, however, typically rely on extensive pre-training or fine-tuning of LLMs, which demands considerable computational resources. To avoid this, we propose a retrieval-based approach that utilizes the powerful capabilities of LLMs in representation, comprehension, and inference. Our approach eliminates the need for training any modules or performing data augmentation, depending solely on information from target dataset. Experimental results reveal that, even without specialized training for tabular data, our method exhibits strong predictive performance on tabular prediction task, affirming its practicality and effectiveness.

1 Introduction

Tables are one of the most prevalent data formats in real-world applications, with widespread use across diverse fields such as healthcare, e-commerce, and manufacturing. The analysis of tabular data facilitates solutions to various practical needs, including trend prediction, risk control, anomaly detection, personalized services, and so on. Ensemble tree models (Chen and Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018), based on gradient boosting decision trees (GBDT), are the leading approaches in tabular data analysis. Today, deep learning's remarkable breakthroughs have inspired exploration of its potential applications to tabular data (Kotelnikov et al., 2023; Chen et al., 2023b,a; Levin et al., 2023). However, in terms of experimental performance, computational efficiency and generalization ability, deep learning still fails to achieve the dominance of GBDT-like methods in tabular data (Gorishniy et al., 2021; Grinsztajn et al., 2022; Shwartz-Ziv and Armon, 2022).

Over the past two years, the rapid advancement of large language models (LLMs) (Brown et al., 2020; OpenAI, 2023; Dubey et al., 2024) has generated considerable interest in their application to tabular data. A typical way is to collect extensive tabular datasets from diverse domains, and then fine-tune LLMs to support a wide range of tabular prediction tasks (Wang et al., 2023a; Yang et al., 2024; Wen et al., 2024; Wang et al., 2023b). Additionally, some studies are dedicated to optimizing input formats or leveraging prompt-based learning, to enhance the adaptability of LLMs in tabular prediction tasks and improve their predictive performance in few-shot settings (Hegselmann et al., 2023; Jaitly et al., 2023; Dinh et al., 2022). While these methods have, in some scenarios, achieved performance comparable to XGBoost (Chen and Guestrin, 2016), fine-tuning LLMs remains an inevitable step in these approaches, leading to excessive computational costs. Given the relatively modest performance improvements, this level of resource consumption is often not cost-effective.

Our motivation is to minimize unnecessary resource costs: is it possible to leverage the powerful capabilities of LLMs for tabular tasks by relying solely on LLM APIs, without resorting to data augmentation or model fine-tuning? To this end, we propose a straightforward yet efficient method that fully exploits LLMs' strengths in representation, comprehension, and reasoning. The workflow of our proposed method is illustrated in Figure 1. Specifically, it utilizes the embedding capability of LLM to encode tabular instances and retrieve similar instances based on these representations. These similar instances, along with their labels, are then used as prompts to guide LLM in understanding the characteristics of different categories, enabling it

[*]Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 9917–9925
January 19–24, 2025. ©2025 Association for Computational Linguistics

to make predictions for target unlabeled instances. The entire process requires only API calls to LLM, with no fine-tuning of the model or training of additional modules, while also demonstrating high efficiency in data utilization. As a result, this method provides a resource-efficient solution that significantly reduces computational and annotation costs while maintaining strong performance.

2 Methodology

2.1 Instance Serialization and Representation

To enable the application of LLMs to table-related tasks, it is essential to first transform tables into a text sequence format. Previous studies have explored various strategies for this transformation, including template-based, model-based, and LLM-based methods. In this work, we opt a more transparent and efficient method by representing tabular data as key-value pairs. Our choice is motivated by the remarkable proficiency of LLMs in processing code. This approach not only reduces the input length and highlights distinctions between data samples but also avoids potential errors introduced during model-based transformations.

Specifically, a tabular dataset is formulated as containing records $\{(x_i, y_i)|i \leq N\}$ and features $\{f_j|j \leq M\}$, where x_i is i-th record's features with values $\{c_{ij}|i \leq N, j \leq M\}$ and y_i is i-th record's label. The feature's values can be a number, a word, or a phrase. Uncommon numerical values can be challenging for LLMs to comprehend, so we selectively convert certain overly complex numerical features by quantiles, describing them like "low", "medium", and "high". Each sample record is serialized to key-value pairs like:

$$s(x_i) = \{f_1 : c_{i1}, f_2 : c_{i2}, \ldots, f_M : c_{iM}\}. \quad (1)$$

Subsequently, we utilize the embedding function of LLM to encode the representation of each instance:

$$\tilde{x}_i = \mathrm{LLM}(s(x_i)). \quad (2)$$

These encoded representations are then collectively indexed and stored for further use.

2.2 Instance Retrieval

Although LLMs are capable of processing long inputs, directly inputting all tabular instances into an LLM can be counterproductive. Common challenges in tabular data, such as repeated values, outliers, and the long-distance forgetting problem in LLMs, can hinder their ability to effectively generalize and identify categorical features.Therefore, existing research often employs sampling techniques to reduce the number of tabular instances presented to LLMs. Building on this, we introduce the concept of retrieval and nearest neighbors. Retrieval-augmented methods have been widely adopted in tasks such as question-answering and recommendation systems. These techniques enable models to access more relevant information, thereby enhancing their ability to discern patterns, perform analogies, and reason effectively. However, the use of retrieval augmentation in tabular prediction tasks remains relatively underexplored.

In our approach, we intend to retrieve labeled instances from training set that exhibit similar features to the target unlabeled instance, to be used as prompt instances. These retrieved instances will serve as contextual information and input alongside the target instance into LLM. This will assist LLM in comprehending the discriminative characteristics of different categories, thereby enabling it to make more precise and coherent predictions. To compute the similarity between instances, we use the inner product, which is widely recognized as a reliable metric for assessing vector similarity. The function $sim(\cdot)$ is used to calculate the similarity between target instance x_i and candidate instance x_c, with $\odot$ symbolizing the inner product operation:

$$sim(x_i, x_c) = \tilde{x_i} \odot \tilde{x_c}. \quad (3)$$

2.3 LLM for Prediction

Given that LLMs have not undergone specialized training for tabular prediction tasks, they may struggle to directly grasp the target task. To enable LLM to perform tabular prediction tasks effectively, it is crucial to use appropriate prompts to consolidate all relevant information. These contents specifically include: (1) metadata of dataset (task descriptions, features, and classes); (2) retrieved instances and target unlabeled instance; and (3) an explicit initiation of prediction task. An example of the prompt composition is shown in Figure 2. A dataset's metadata typically includes its source, features, and task objectives. The retrieved instances are still presented as key-value pairs, with each instance tagged with label. The target instance is placed after the retrieved instances. Textual coherence is maintained in a plain and clear manner. Lastly, the prompt for prediction is appended at the end. The LLM will then generate the final prediction directly.

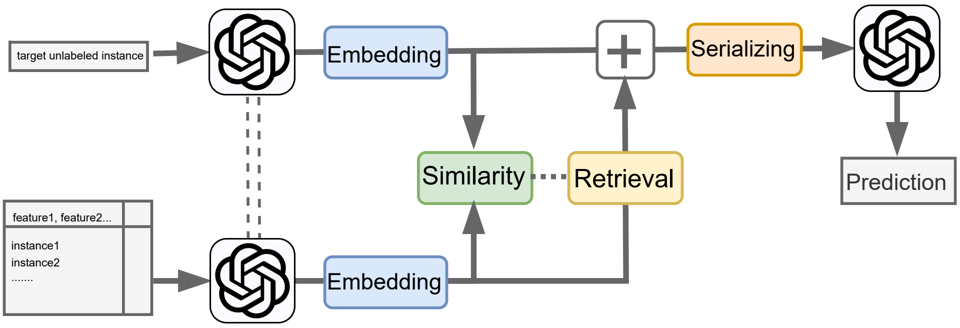

Figure 1: The workflow of the proposed method. The unlabeled instances and training set samples are encoded via LLM to obtain their vector representations. The similarity between instances is then computed to retrieve relevant prompt instances. Subsequently, the retrieved instances, target instance, and task prompt are serialized into text and fed into the LLM for prediction.

3 Experiment Setup

3.1 Dataset

We conduct experiments on 13 classic machine learning tabular datasets, sourced from UCI (Kelly et al., 2023) and OpenML (Vanschoren et al., 2014). These datasets span diverse fields including healthcare, biological sciences, social statistics, and so on, incorporating both numerical and categorical data types. The dataset sizes are generally of medium scale. Further specifics about datasets' size, data type, description and features are presented in Table 2 and Table 3.

3.2 Experimental Settings

The utilization of LLMs in our method are all based on OpenAI's APIs, utilizing `GPT-3.5` for embedding processing and `GPT-4.0-mini` for executing prediction tasks. All datasets are partitioned into training, validation, and test sets following a 5:1:4 split, and hyperparameter optimization is guided by performance on validation set. For instance retrieval, we utilize the FAISS (Facebook AI Similarity Search) tool (Johnson et al., 2021), which supports similarity searches for vectors. The number of instances retrieved is set to 5. The experimental results presented are based on the average of 5 random seeds.

4 Results

We conduct a comparative analysis of the experimental results with several typical baselines for tabular prediction, including KNN, XGBoost (Chen and Guestrin, 2016), TabLLM (Hegselmann et al., 2023) and TabPFN (Hollmann et al., 2023). KNN and XGBoost are two classical wide-used machine learning methods in data science field. TabLLM employs parameter-efficient technique (Liu et al., 2022) to fine-tune LLMs for tabular prediction tasks, and we use T0-3B (Sanh et al., 2022) for implementation here. TabPFN is a pre-trained model specifically for tabular prediction, which is trained on 9216000 synthetically generated datasets.

4.1 Overall Performance

Table 1 presents the prediction accuracy of all methods on test set. The experimental results demonstrate that, with the exception of the `diabetes` and `hamster` datasets, our method performs remarkably well across the majority of datasets. When compared to XGBoost, the most prevalent model for tabular data, our approach yields comparable results and even outperforms it on certain datasets. These findings underscore the practicality and effectiveness of our method. The performance of KNN is relatively weaker, showing notable advantages only on the `hamster` and `iris` datasets. Our method, which also draws inspiration from certain aspects of nearest neighbor algorithms, demonstrates improved predictive performance, further highlighting the powerful reasoning capabilities of LLM. Notably, TabLLM, as a fine-tuning approach for LLMs, exhibits only moderate performance on most datasets when trained with the full dataset. This suggests that fine-tuning LLMs on medium-size datasets may not lead to significant performance improvements and could potentially impair their reasoning capabilities. In contrast, our method, which focuses on selecting appropriate prompt examples, more effectively harnesses the capabilities of LLMs. Moreover, compared to tabular deep learning models that require substantial

Dataset	Ours	TabPFN	TabLLM	KNN	XGBoost
caesarian	0.6314 ± 0.05	0.5684 ± 0.11	0.6014 ± 0.04	0.6125 ± 0.09	0.5812 ± 0.07
chlamydia	0.8750 ± 0.03	0.7950 ± 0.07	0.8200 ± 0.02	0.8100 ± 0.01	0.8150 ± 0.11
diabetes	0.6656 ± 0.01	0.7621 ± 0.03	0.7367 ± 0.05	0.7247 ± 0.03	0.7571 ± 0.03
glass	0.8472 ± 0.04	0.8014 ± 0.10	0.8202 ± 0.05	0.7865 ± 0.05	0.8102 ± 0.09
haberman	0.7642 ± 0.03	0.6342 ± 0.07	0.6842 ± 0.07	0.7447 ± 0.03	0.7154 ± 0.07
tae	0.5574 ± 0.04	0.4720 ± 0.09	0.5216 ± 0.01	0.4983 ± 0.16	0.5082 ± 0.03
environment	0.6178 ± 0.02	0.5812 ± 0.04	0.5463 ± 0.06	0.6089 ± 0.04	0.5555 ± 0.07
hamster	0.5933 ± 0.02	0.6082 ± 0.09	0.6241 ± 0.04	0.7067 ± 0.08	0.6133 ± 0.08
blood_donation	0.9740 ± 0.01	0.9700 ± 0.01	0.9740 ± 0.01	0.7927 ± 0.02	0.9960 ± 0.01
breast_cancer	0.7225 ± 0.04	0.7177 ± 0.04	0.7171 ± 0.05	0.7286 ± 0.05	0.7297 ± 0.06
heart_statlog	0.7676 ± 0.02	0.7715 ± 0.04	0.7786 ± 0.03	0.6352 ± 0.07	0.7778 ± 0.03
iris	0.9198 ± 0.03	0.9264 ± 0.02	0.9314 ± 0.03	0.9767 ± 0.03	0.9333 ± 0.03
somerville	0.6034 ± 0.05	0.5721 ± 0.04	0.5860 ± 0.04	0.5103 ± 0.02	0.5690 ± 0.07

Table 1: The test accuracy of different models on each dataset.

training consumption, our method is highly competitive, as it eliminates the need for training and is resource-efficient.

4.2 Analysis

We perform experiments to analyze the effects of our design. Table 2 presents the results of LLM in zero-shot and few-shot settings.

Zero-shot performance. In zero-shot setting, predictions are made by directly inputting target instance, dataset metadata, and associated prompts into LLM without utilizing any additional supporting instances. This setup also serves as an ablation experiment, excluding retrieval-related instances to assess their impact. Experiments results reveal that for datasets such as caesarian, diabetes, haberman, and heart_statlog, LLM can achieve acceptable prediction outcomes even under zero-shot scenario, highlighting the value of its prior knowledge in certain tabular tasks. However, in other datasets, the zero-shot performance of LLM is notably poor, considerably underperforming compared to its zero-shot capabilities in textual tasks. These findings suggest that while LLM may comprehend the textual information within tables, its grasp of prior knowledge and pattern learning in tables remains limited, which makes LLMs impossible for direct application to most tabular prediction tasks.

Few-shot performance. In few-shot setting, a small subset of training set is sampled to serve as prompt instances. For each dataset, two samples per category are randomly selected. On datasets such as glass, environment, blood_donation, breast_cancer, and iris, introducing a few randomly chosen prompt instances leads to notable improvements in prediction accuracy. This highlights the strong analogical reasoning capabilities of LLM when processing prompt instances. However, in certain datasets, the performance of LLM declines under the few-shot scenario, emphasizing the critical role of selecting high-quality prompt instances. Overall, the consistent improvements shown by our method across both scenarios validate its ability to fully exploit the strengths of LLM while mitigating the limitations associated with suboptimal prompt selection.

Large-scale dataset and time consumption. In our method, FAISS is employed for fast vector search during instance retrieval process. For large-scale datasets, FAISS maintains high efficiency through memory optimization and vector partitioning. The retrieval time difference between datasets containing 100 samples and 10,000 samples is minimal (about 0.002s and 0.006s, respectively), with noticeable increases only in datasets with millions of instances, which is rarely the case for tabular prediction tasks. Thus, the impact of dataset size on retrieval efficiency is negligible. The primary time consumption lies in the LLM's response time, which averages approximately 1.223s—nearly 600 times of retrieval process.

5 Limitations

While our method demonstrates a certain level of innovation and practicality, certain limitations re-

Dataset	Data type	Size	Zero-shot	Few-shot	Ours
caesarian	1c4d	80	0.5938 ± 0.10	0.6125 ± 0.06	0.6314 ± 0.05
chlamydia	1c2d	100	0.3125 ± 0.01	0.2800 ± 0.05	0.8750 ± 0.03
diabetes	8c0d	768	0.6351 ± 0.01	0.6383 ± 0.06	0.6656 ± 0.01
glass	9c0d	214	0.4678 ± 0.01	0.5729 ± 0.18	0.8472 ± 0.04
haberman	3c0d	306	0.7285 ± 0.00	0.7252 ± 0.01	0.7642 ± 0.03
tae	3c2d	151	0.3443 ± 0.00	0.3279 ± 0.00	0.5574 ± 0.04
environment	3c0d	111	0.4178 ± 0.02	0.5556 ± 0.13	0.6178 ± 0.02
hamster	5c0d	73	0.5333 ± 0.00	0.5667 ± 0.01	0.5933 ± 0.02
blood_donation	4c0d	748	0.7673 ± 0.00	0.9807 ± 0.02	0.9740 ± 0.01
breast_cancer	3c6d	286	0.5189 ± 0.03	0.6324 ± 0.17	0.7225 ± 0.04
heart_statlog	6c7d	270	0.6722 ± 0.00	0.6833 ± 0.04	0.7676 ± 0.02
iris	4c0d	150	0.7133 ± 0.02	0.8866 ± 0.03	0.9198 ± 0.03
somerville	0c6d	143	0.5345 ± 0.00	0.5345 ± 0.00	0.6034 ± 0.05

Table 2: The test accuracy of different scenarios with LLM on each dataset. The data type and size are provided correspondingly (c means continuous variable, d means discrete variable).

main, offering potential directions for future improvement. Firstly, due to the inherent limitations of language models in numerical understanding, we employs a quantile-based discretization strategy to transform numerical features into textual descriptions. While this approach enhances the compatibility of numerical data with the language processing capabilities of LLMs, it may introduce a loss of numerical precision, especially in tasks requiring detailed numerical comparisons or trend analyses. Further studies could investigate approaches that better retain numerical information, such as utilizing numerical embeddings or implementing more effective discretization techniques. Secondly, the direct use of LLMs to generate results for tabular prediction presents interpretability challenges, as these models function as complex black-box systems, lacking transparency of the reasoning process. In future work, we might explore the incorporation of chain-of-thought reasoning, which decomposes inference into a series of explicit, step-by-step processes. This method could reveal how the LLM extracts key information from tabular data and reaches conclusions, thereby providing greater clarity into its decision-making mechanisms.

6 Conclusion

In this paper, we propose a resource-efficient solution for tabular prediction tasks based on LLM APIs. By encoding instances through an LLM to calculate instance similarity, we retrieve prompt instances for a target unlabeled instance based on their similarity, thereby enhancing LLM's inductive and interpretative ability for task-specific features and ultimately accomplishing effective analogical reasoning and accurate predictions for unlabeled instances. Unlike prior research, our approach requires no training or fine-tuning, nor does it rely on additional data annotation or synthesis; it operates solely using LLM APIs and the intrinsic information of dataset. From this perspective, our method is both highly competitive and innovative for tabular prediction in terms of resource efficiency.

Acknowledgments

This research was funded by the National Natural Science Foundation of China under Grant No. 62072075.

References

Sercan Ö. Arik and Tomas Pfister. 2021. TabNet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, et al. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165:1–75.

Shaofeng Cai, Kaiping Zheng, Gang Chen, H. V. Jagadish, Beng Chin Ooi, and Meihui Zhang. 2021. ARM-Net: Adaptive relation modeling network for structured data. In *Proceedings of the 2021 International Conference on Management of Data*, page 207–220.

Jintai Chen, KuanLun Liao, Yanwen Fang, Danny Chen, and Jian Wu. 2023a. TabCaps: A capsule neural network for tabular data classification with bow routing. In *The Eleventh International Conference on Learning Representations*.

Kuan-Yu Chen, Ping-Han Chiang, Hsin-Rung Chou, Ting-Wei Chen, and Darby Tien-Hao Chang. 2023b. Trompt: towards a better deep neural network for tabular data. In *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, USA.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, San Francisco, California, USA.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. 2022. LIFT: Language-interfaced fine-tuning for non-language machine learning tasks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, volume 35, pages 11763–11784, New Orleans, LA, USA.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *ArXiv*, abs/2407.21783:1–92.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pages 18932–18943.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, volume 35, pages 507–520, New Orleans, LA, USA.

Hussein Hazimeh, Natalia Ponomareva, Petros Mol, Zhenyu Tan, and Rahul Mazumder. 2020. The tree ensemble layer: differentiability meets conditional computation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4138–4148.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 5549–5581.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar S. Karnin. 2020. TabTransformer: Tabular data modeling using contextual embeddings. *ArXiv*, abs/2012.06678:1–17.

Sergei Ivanov and Liudmila Prokhorenkova. 2021. Boost then convolve: Gradient boosting meets graph neural networks. In *International Conference on Learning Representations*, pages 1–16.

Sukriti Jaitly, Tanay Shah, Ashish Shugani, and Razik Singh Grewal. 2023. Towards better serialization of tabular data for few-shot classification with large language models. *ArXiv*, abs/2312.12464:1–6.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Liran Katzir, Gal Elidan, and Ran El-Yaniv. 2021. Net-DNF: Effective deep modeling of tabular data. In *International Conference on Learning Representations*, pages 1–16.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 3149–3157.

Guolin Ke, Zhenhui Xu, Jia Zhang, Jiang Bian, and Tie-Yan Liu. 2019. DeepGBM: A deep learning framework distilled by gbdt for online prediction tasks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 384–394.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. 2023. The UCI machine learning repository.

Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Tom Rainforth, and Yarin Gal. 2021. Self-attention between datapoints: going beyond individual input-output pairs in deep learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pages 28742–28756.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. TabDDPM: modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, USA.

Roman Levin, Valeriia Cherepanova, Avi Schwarzschild, Arpit Bansal, C. Bayan Bruss, Tom Goldstein, Andrew Gordon Wilson, and Micah Goldblum. 2023. Transfer learning with deep tabular models. In *The Eleventh International Conference on Learning Representations*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 1950–1965.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774:1–100.

Sergei Popov, Stanislav Morozov, and Artem Babenko. 2020. Neural oblivious decision ensembles for deep learning on tabular data. In *International Conference on Learning Representations*, pages 1–12.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, page 6639–6649.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, pages 1–216.

Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90.

Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C. Bayan Bruss, and Tom Goldstein. 2022. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS 2022 First Table Representation Workshop*, pages 1–22.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

Ruiyu Wang, Zifeng Wang, and Jimeng Sun. 2023a. Unipredict: Large language models are universal tabular predictors. *ArXiv*, abs/2310.03266:1–24.

Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. 2023b. MediTab: Scaling medical tabular data predictors via data consolidation, enrichment, and refinement. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, pages 1–21.

Zifeng Wang and Jimeng Sun. 2022. TransTab: learning transferable tabular transformers across tables. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 2902–2915.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. 2024. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3323–3333, Barcelona, Spain.

Yazheng Yang, Yuqi Wang, Sankalok Sen, Lei Li, and Qi Liu. 2024. Unleashing the potential of large language models for predictive tabular tasks in data science. *ArXiv*, abs/2403.20208:1–16.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico.

A Related Work

The application of deep learning to tabular data prediction can be broadly categorized into two types: hybrid models and Transformer-based methods. Hybrid models aim to combine neural networks and tree-based models, leveraging their respective strengths. One strategy involves making tree models differentiable through smoothing techniques (Popov et al., 2020; Hazimeh et al., 2020; Katzir et al., 2021), while another incorporates tree structures directly into neural networks (Ke et al., 2019; Ivanov and Prokhorenkova, 2021), thereby integrating the structural priors of trees. Transformer-based methods primarily focus on using attention mechanisms for feature encoding (Wang and Sun, 2022; Huang et al., 2020), optimizing feature interaction selection and reasoning (Arik and Pfister, 2021; Cai et al., 2021), and promoting sample information sharing (Somepalli et al., 2022; Kossen et al., 2021). In addition, recent efforts have fine-tuned LLMs on extensive tabular datasets (Hegselmann et al., 2023; Wang et al., 2023a; Yang et al., 2024; Zhang et al., 2024), enhancing their capacity for understanding and reasoning with tabular data.

Despite these advancements, classical Gradient Boosting Decision Tree (GBDT) methods, including XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), continue to outperform deep learning models on most datasets. These methods are also recognized for their robust performance, training efficiency, ease of hyperparameter tuning, and high interpretability, making them the preferred choice for tabular data prediction tasks (Gorishniy et al., 2021; Grinsztajn et al., 2022;

Dataset	Source (id)	Task description	Features
caesarian	OpenML (42901)	This dataset contains information about caesarian section results of pregnant women, with the objective of predicting whether a woman will undergo normal or caesarian delivery.	age; delivery number; delivery time; blood pressure; heart problem
chlamydia	OpenML (535)	This dataset contains results of individuals that tested for chlamydia, with the objective of predicting whether a person will test positive or negative for chlamydia.	age; gender; race
diabetes	OpenML (37)	This dataset contains diagnostic measurements taken from the National Institute of Diabetes and Digestive and Kidney Diseases, with the objective of predicting whether a patient will test positive or negative for diabetes.	number of times pregnant; plasma glucose concentration; diastolic blood pressure; triceps skin fold thickness;2-hour serum insulin; body mass index; diabetes pedigree function; age
glass	OpenML (41)	This dataset contains different minerals in glass, with the objective of predicting whether the glass was a type of "float" glass or not.	refractive index; Sodium; Magnesium; Aluminum; Silicon; Potassium; Calcium; Barium,Iron
haberman	OpenML (43)	This dataset contains cases from a study about the survival of patients who had undergone surgery for breast cancer.	age; patient's year of operation; number of positive axillary nodes detected
tae	OpenML (48)	This dataset contains evaluations of teaching performance over teaching assistant assignments, with the objective of predicting their class scores: low, medium, or high.	being native English speaker; course instructor; course; semester; class size
environment	OpenML (678)	This dataset contains indicators for predicting whether the environment is positive or negative.	ozone; radiation; temperature
hamster	OpenML (708)	This dataset contains indicators for predicting whether the hamster is ill or healthy.	lung; heart; liver; spleen; spleen
blood_donation	UCI (176)	This donor dataset is taken from the Blood Transfusion Service Center, with the objective of predicting whether he/she will donate blood.	months since last donation; total number of donation; total blood donated; months since first donation
breast_cancer	UCI (14)	This dataset contains information collected from breast cancer patients, with the objective of predicting whether a patient will experience tumor recurrence.	age; menopause; tumor size; number of affected lymph nodes; nodular capsules; deg-malig; breast side; breast-quad; radiotherapy
heart_statlog	UCI (145)	This dataset contains diagnostic measurements about individuals, with the objective of predicting whether a person has heart disease or not.	age; sex; chest pain; resting blood pressure; serum cholesterol; fasting blood sugar; electrocardiographic; maximum heart rate; angina; oldpeak; slope; major-vessels; thal
iris	UCI (53)	This dataset contains the attributes of iris flowers, with the objective of predicting the type of iris plant: virginica, versicolor, and setosa.	sepal length; sepal width; petal length; petal width
somerville	UCI (479)	This dataset contains ratings collected from Somerville Happiness Survey, with the objective of predicting whether a resident is happy or unhappy about the place.	availability of information about the city services; cost of housing; overall quality of public schools; trust in the local police; maintenance of streets and sidewalks; availability of social community events

Table 3: The task description and features of each dataset.

Shwartz-Ziv and Armon, 2022). Current tabular deep learning models typically require significant computational resources, including large model parameters, substantial training datasets, and advanced hardware. Our study introduces an efficient, training-free approach, contributing to the exploration of more resource-efficient solutions in tabular deep learning research.

B Dataset Details

All the data used in this study comes from publicly available datasets on OpenML and UCI. Table 3 lists the details of all datasets, including their sources, task descriptions, and features.

C Prompt Details

Figure 2 shows the prompt template used for LLM in our method. <task description> and <classes> are provided by dataset. <serialized labeled instance> are prompt instances retrieved from the dataset based on target instance. The serialization method for instance follows the format described in §2.1.

> You are a helpful data analyst. I'll give you a tabular dataset's task description, features, label classes, and some labeled instances in json format, from which you will make classification prediction for new instance. No analyzing, directly give the prediction answer, there can only be one category of prediction.
>
> Task description: <task description>
> Features: <features>
> Target label classes: <classes>
> Labeled instances:
> <serialized labeled instance>
>
> Now use the provided metadata and
> instances to infer by analogy about the
> label of this new instance:
> <serialized unlabeled instance>

Figure 2: The prompt template used for tabular prediction.