

Summarize-Exemplify-Reflect: Data-driven Insight Distillation Empowers LLMs for Few-shot Tabular Classification

Yifei Yuan^{1,2}, Jiatong Li³, Weijia Zhang⁴, Mohammad Aliannejadi⁴,
Evangelos Kanoulas⁴, Renjun Hu^{5*}

¹ETH Zürich, ²University of Copenhagen, ³University of Science and Technology of China,
⁴ University of Amsterdam, ⁵ East China Normal University
yuanyif@ethz.ch, rjhu@dase.ecnu.edu.cn

Abstract

Recent studies show the promise of large language models (LLMs) for few-shot tabular classification but highlight challenges due to the variability in structured data. To address this, we propose distilling data into actionable insights to enable robust and effective tabular classification by LLMs. Inspired from human learning processes, we introduce InsightTab, an insight distillation framework guided by principles of divide-and-conquer, easy-first, and reflective learning. It integrates rule summarization, strategic exemplification, and insight reflection through deep collaboration between LLMs and data modeling techniques. The obtained insights enable LLMs to better align their general knowledge and capabilities with the particular requirements of specific tabular tasks. We extensively evaluate InsightTab on nine datasets. The results demonstrate consistent improvement over state-of-the-art methods. Ablation studies further validate the principle-guided distillation process, and in-depth analyses emphasize InsightTab’s effectiveness in leveraging labeled data and managing biases.

1 Introduction

Tabular data is widely used across applications due to the prevalence of relational databases (Shwartz-Ziv and Armon, 2021), inspiring tasks like tabular question answering (Vakulenko and Savenkov, 2017), retrieval (Zheng et al., 2023), etc. Among these tasks, tabular classification plays a key role by categorizing tabular data with numerical and categorical features into predefined classes.

Traditional tabular classification tasks leverage ensemble techniques with decision trees such as XGBoost to handle mixed numerical and categorical data (Shwartz-Ziv and Armon, 2021). The high cost of labeling large datasets and feature engineering has driven interest in few-shot learning,

where models make accurate predictions using only a few examples (Hegselmann et al., 2022). These approaches often leverage large language models (LLMs), which enhance prediction accuracy by transforming structured data into natural language sequences, enabling LLMs to process and make informed predictions. They show that combining proper serialization with LLMs achieves strong performance, with later studies refining this through expert and algorithmic prompting (Slack and Singh, 2023; Manikandan et al., 2023).

However, several challenges persist for effective LLM-based tabular classification in the few-shot setting. These include: (1) *Filling knowledge gap*. Tabular data often includes task-specific details (e.g., task description), failing to incorporate them can hinder the performance of general-purpose LLMs. (2) *Unlocking LLMs’ full potential*. Effective tabular classification requires complex processing. While methods like in-context learning help, they fall short of fully utilizing LLMs’ capabilities, making this an ongoing challenge. (3) *Balancing performance and cost*. Tabular classification tasks are critical for real-time services, demanding fast response times and cost efficiency. Traditional methods, while effective, are often time-intensive to train. Thus, LLM-based solutions must be both efficient and economical to meet these demands.

To address these challenges, we focus on the task of **LLM-based tabular classification in the few-shot setting**. We pose a fundamental research question: *Can actionable insights be distilled from the training data to enhance LLMs’ performance in this task?* In response, we introduce InsightTab, a novel insight distillation framework that integrates traditional machine learning techniques with LLMs, enabling them to acquire task-specific knowledge from a data modeling perspective. In this analogy, LLMs can be seen as students with broad general knowledge but limited task-specific expertise. Inspired by human learning processes,

*corresponding author

InsightTab employs three key principles: *divide-and-conquer*, *easy-first*, and *reflective learning*. The *divide-and-conquer* strategy, inspired by curriculum learning (Bengio et al., 2009), partitions training data into subsets, enabling LLMs to derive rules for each, *i.e.*, a cognitive process akin to **summarization**. This could simultaneously avoid context window overflow and yield more targeted rules than directly feeding all data into an LLM. The *easy-first* principle mimics the human cognitive procedure of **exemplifying**, prioritizing the selection of representative and simpler samples for initial learning (Sun et al., 2024). Finally, the *reflective learning* strategy allows LLMs to engage in **reflection**, *i.e.*, learning from misclassified cases to enhance their distilled insights.

We then use three operators for the insight distillation process: group for clustering similar samples, rank for example ordering with prediction difficulty, and summarize for extracting natural language rules. This process fosters collaboration between data modeling methods and LLMs, leveraging LLMs’ strength in summarization while addressing their limitation in data analysis.

We validate InsightTab through experiments on nine diverse datasets, demonstrating its superiority over state-of-the-art methods in average performance. Additionally, InsightTab outperforms other approaches using the same base LLMs in 19 out of 20 comparisons. This highlights the importance of insight distillation for few-shot tabular classification. Ablation and case studies further confirm the effectiveness of each module, and tests on biased data show InsightTab’s priority in robustness, reducing overfitting and improving generalization. In summary, our contributions are threefold:

- Conceptually, we highlight the importance of insight distillation through a deep integration of data modeling methods and LLMs prompting for few-shot tabular classification.
- Methodologically, we present InsightTab, an insight distillation framework inspired by human learning principles of divide-and-conquer, easy-first, and reflective learning.
- Experimentally, we validate the enhanced effectiveness of InsightTab through extensive tests, as well as its rationales and superiority in sample efficiency and robustness¹.

¹Code and dataset available at <https://github.com/yfyan01/InsightTab/>.

2 Related Work

Tabular classification. Tabular classification assigns labels to table rows based on column attributes (Fang et al., 2024). Traditional methods use gradient-boosted decision trees (Chen and Guestrin, 2016; Ke et al., 2017) for structured data (Shwartz-Ziv and Armon, 2021; Gorishniy et al., 2021), while others design specialized architectures (Chen et al., 2023; Shah et al., 2022; Du et al., 2021), such as TabularNet (Du et al., 2021), which encodes spatial and relational information. Transformer-based approaches have also been explored (Hollmann et al., 2022; Nassar et al., 2022), including models specifically optimized for small tabular datasets like TabPFN (Hollmann et al., 2022). Recently, LLMs have gained attention for their few-shot learning abilities (Hu et al., 2022), enabling classification with minimal samples (Hegselmann et al., 2022; Nam et al., 2023; Han et al., 2024). For instance, Nam et al. (2023) and Qu et al. (2025) incorporate meta-learning and curriculum learning into LLM prompting, drawing inspiration from traditional data mining techniques. Based on that, our approach focuses on LLM-based few-shot tabular classification and propose to distill actionable insights from tabular data.

LLMs for tabular data. Language models have demonstrated effective capabilities in handling structured data (Lu et al., 2024). Extensive studies have explored pretraining language models to effectively learn structured table representations, ranging from smaller-scale pretrained models (*e.g.*, BERT, GPT-2) (Yin et al., 2020; Liu et al., 2021; Iida et al., 2021) to state-of-the-art LLMs (Zha et al., 2023; Zhang et al., 2023, 2024). For example, Zhang et al. (2023) fine-tune Llama 2 to develop open-source LLMs capable of handling a variety of table-based tasks. Since LLMs are designed for natural language, converting tabular data into text enables them to analyze and generate insights effectively (Nan et al., 2023). This process, known as data serialization, typically involves linearizing tables row by row with column separators (Sui et al., 2023; Singha et al., 2023). For example, Gao et al. (2023) find that creating in-context learning prompts which include instruction helps improve Text-to-SQL performance. Manikandan et al. (2023) are the first to generate summaries and shows that incorporating task-specific knowledge during serialization can improve tabular classification. Building on their findings, our work

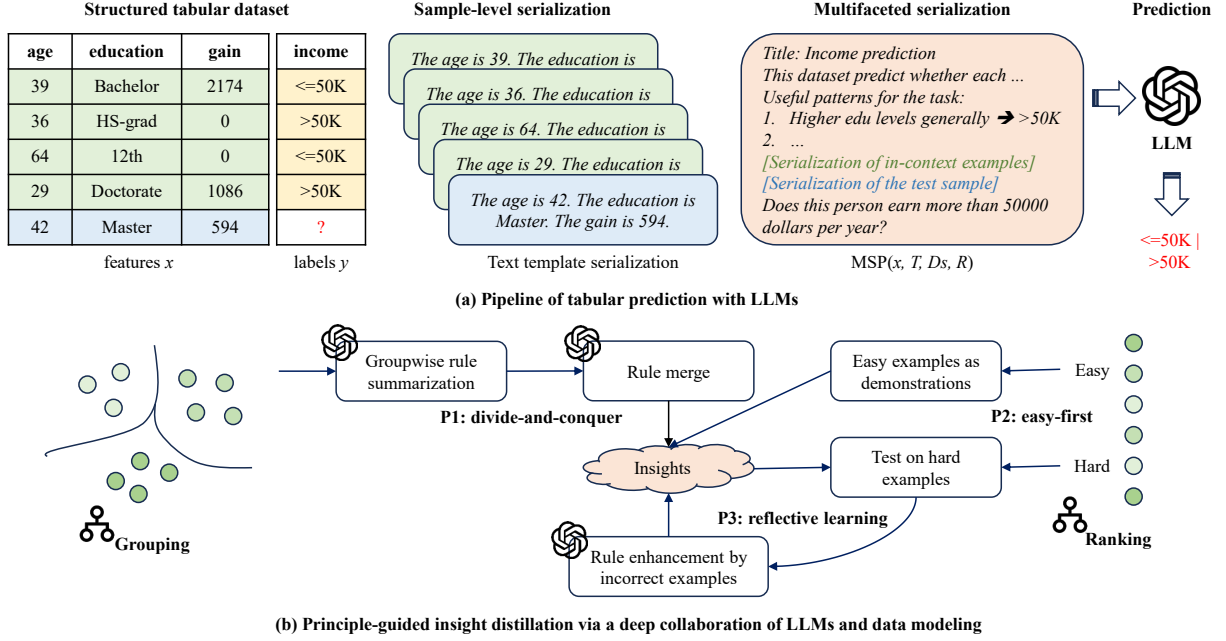


Figure 1: Framework overview of InsightTab. (a) We present an income prediction dataset with $n = 4$ samples and $d = 3$ features. The text template serialization converts column features into natural language sequences, while the multifaceted serialization further integrates them with task-specific knowledge into a prompt, for LLMs to predict test data. (b) Insight distillation involves both data modeling techniques and LLMs for sample grouping/ranking and rule summarization, respectively. The process is guided by three principles inspired from human learning.

aims to develop a multifaceted serialization prompt that combines examples and rules from the training samples in natural language format, providing a specialized form of serialization. This approach provides strong guidance for LLMs, helping them understand the data structure.

3 Methodology

In this section, we describe the structure of InsightTab, which integrates data modeling techniques with LLMs for better tabular classification. An overview of our approach is presented in Figure 1.

3.1 Preliminaries

Tabular Data. A labeled tabular dataset with n rows (*i.e.*, samples) and d columns (*i.e.*, features) could be denoted as $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$, where each $x^{(i)}$ is a d -dimensional feature vector and $y^{(i)} \in \mathcal{C}$ is the corresponding class label of $x^{(i)}$. $\mathcal{C}$ denotes the set of labels for a tabular classification task T . Each column of $\mathcal{D}$ is associated with a semantically meaningful feature name such as age, education, or marital status. These feature names are formally denoted as $\mathcal{F} = \{f_1, \dots, f_d\}$.

Multifaceted Serialization. A prerequisite for using LLMs in tabular classification is data serialization, which involves converting raw tabular data into natural language to enhance LLM understand-

ing. Hegselmann et al. (2022) empirically examined nine different serialization formats for tabular data and found that the *Text Template* approach performs well across their experiments. Specifically, this approach converts a feature vector x to a textual enumeration of all features as “The f_1 is x_1 The f_d is x_d .”, where x_j represents the j -th value of x (see Figure 1). We adopt it as the default sample-level serialization method. Additionally, concurrent studies (Slack and Singh, 2023; Manikandan et al., 2023) have verified that LLMs benefit from task-specific knowledge for tabular classification. The knowledge includes but not limited to task background, labeled few-shot examples, and rules mined from data. We hence extend the scope of serialization and define a multifaceted serialization prompt $\text{MSP}(x, T, \mathcal{D}_s, R)$ which converts not only test sample x but also the task T , few-shot in-context learning examples $\mathcal{D}_s \subset \mathcal{D}$, and mined rules R to natural language prompt. Figure 1 also demonstrates a brief example of it (prompts are listed in Appendix E).

Task Formalization. With the aforementioned concepts, we formalize the LLM-based tabular classification task, particularly in few-shot settings. Given the tabular classification task T , an associated labeled dataset $\mathcal{D}$, and an unlabeled test sample $\tilde{x}$, the task goal is to predict a class label

$\tilde{y} \in \mathcal{C}$ for $\tilde{x}$ with an LLM. This can be expressed as $\tilde{y} = \text{LLM}_p(\text{MSP}(\tilde{x}, T, \mathcal{D}_s, R))$, where R represents the rules mined by the LLM from the training samples. These few-shot settings often occur in realistic scenarios where the labeled data is limited and has complex patterns. In these cases, LLMs can effectively leverage their rich language knowledge to make accurate predictions, whereas traditional methods struggle to fit an effective decision function with limited data.

3.2 Collaborative Insight Distillation

We propose a method InsightTab that enables LLMs to approach the task from a data modeling perspective and distill training data into actionable *insights*. These insights are injected into the multifaceted serialization function of InsightTab, enabling LLMs to better adapt their general natural language understanding capabilities to different tabular prediction tasks.

In this context, the role of LLMs could be an analogy to students with extensive general knowledge but limited expertise in the specific task. As illustrated in Figure 1, drawing from human learning strategies, we employ three key principles, namely *divide-and-conquer*, *easy-first*, and *reflective learning*, to optimize the learning process for LLMs. Specifically, we adopt a divide-and-conquer strategy for rule summarization: training samples are firstly grouped into subsets of ‘similar’ examples. Summarizing rules for each subset individually then becomes more manageable than for the entire dataset. Next, following the easy-first principle, we choose the most confident samples as in-context demonstrations (Dong et al., 2024). This allows LLMs to quickly acquire skills when confronting a new tabular classification task. Finally, we design a reflective learning mechanism for LLMs to refine their formed knowledge. In this mechanism, a subset of challenging examples is selected from the training data. LLMs are required to make predictions based on the existing rules and demonstrations. Incorrectly predicted samples are retained for further rule summarization, resulting in new rules that augment the existing ones.

Overall, principle-guided insight distillation allows LLMs to draw task-specific knowledge from data, which ultimately enhances the overall tabular classification. To facilitate this insight distillation process, we employ three key operators:

- group divides samples into subsets with high

Algorithm 1 InsightTab

Require: classification task T , training data $\mathcal{D}$, test sample $\tilde{x}$, number n_e/n_h of easy/hard samples, LLM_s for rule summarization, LLM_p for prediction

Ensure: classification label $\tilde{y} \in \mathcal{C}$ of $\tilde{x}$

```

1:  $M \leftarrow \text{XGB}(\mathcal{D})$ ,  $M_1 \leftarrow$  the first tree in  $M$ 
2:  $F \leftarrow$  the number of leaves in  $M_1$ 
3: for  $i = 1$  to  $F$  do
4:   group:  $\mathcal{D}_i \leftarrow$  samples in the  $i$ -th leaf of  $M_1$ 
5:   summarize:  $R_i \leftarrow \text{LLM}_s(\mathcal{D}_i)$ 
6: end for
7: Merge rules:  $R \leftarrow \text{LLM}_s(R_1, \dots, R_F)$ 
8: Compute:  $h^{(i)} \leftarrow \text{entropy}(M(x^{(i)}))$ ,  $i \in \{1, \dots, n\}$ 
9: rank:  $\mathcal{D}_e \leftarrow$  subset of  $\mathcal{D}$  with the top- $n_e$  lowest  $h^{(i)}$ 
10: rank:  $\mathcal{D}_h \leftarrow$  subset of  $\mathcal{D}$  with the top- $n_h$  highest  $h^{(i)}$ 
11:  $\mathcal{D}'_h \subset \mathcal{D}_h = \{(x, y)\}$  s.t.  $\text{LLM}_p(\text{MSP}(x, T, \mathcal{D}_e, R)) \neq y$ 
12: summarize:  $R_h \leftarrow \text{LLM}_s(\mathcal{D}'_h)$ 
13: Enhance rule:  $R^+ \leftarrow R \oplus R_h$ 
14: Predict:  $\tilde{y} \leftarrow \text{LLM}_p(\text{MSP}(\tilde{x}, T, \mathcal{D}_e, R^+))$ 

```

intra-group similarity;

- rank orders samples based on the difficulty for prediction;
- summarize derives a concise set of natural language rules from the samples.

In summary, while LLMs excel at summarize, they may fall short in the group and rank operations which involve complex mathematical calculations, reasoning, and iterative refinement—tasks that are difficult to articulate clearly with natural language instructions. Conversely, group and rank correspond to well-defined data modeling tasks that traditional machine learning methods handle effectively. Therefore, InsightTab fosters a collaboration between LLMs and data modeling, enabling effective insight distillation for LLMs. Details are introduced in the next section.

3.3 Algorithm

We introduce an algorithmic implementation of our method in Algorithm 1. In detail, we adopt gradient-boosting decision trees, *e.g.*, XGBoost (Chen and Guestrin, 2016), as the data modeling technique with the flexibility to adopt other approaches as long as the requirements of group and rank are satisfied (we make a performance comparison in Appendix C.3). Our algorithm takes the classification task description T , the training dataset $\mathcal{D}$, a test sample $\tilde{x}$, a summarizer LLM, and a predictor LLM as input, and outputs the corresponding label $\tilde{y}$ for $\tilde{x}$. It first fits a XGBoost model M on $\mathcal{D}$ to support group and rank (lines 1–2). Specifically, we use the first tree of M for

Method	LLM _p	Bank	Blood	Calhou.	Car	Creditg	Diabe.	Heart	Income	Jungle	Avg
XGBoost	/	20.9	32.9	70.6	40.2	78.4	50.1	77.9	33.9	<u>67.9</u>	52.5
TabPFN	/	14.3	17.6	77.8	30.5	81.3	46.7	81.8	20.2	<u>65.3</u>	48.4
Original	mistral-7b	17.0	23.9	37.2	20.2	70.9	36.7	30.6	13.9	23.8	30.5
Tablet	mistral-7b	10.8	29.8	54.9	25.4	66.7	39.8	42.3	24.0	39.0	37.0
TabLLM	mistral-7b	21.7	38.0	69.2	35.3	80.9	59.2	74.1	58.5	63.8	55.6
SumBoost	mistral-7b	17.1	32.8	23.2	22.8	64.1	38.0	62.6	24.9	37.4	35.9
InsightTab	mistral-7b	41.1*	51.8*	<u>77.2*</u>	<u>48.6*</u>	<u>81.9*</u>	69.6*	76.7*	66.5*	69.4*	64.8*
Original	gpt-3.5	28.9	45.4	68.2	32.9	78.2	54.1	58.7	44.9	54.6	51.8
Tablet	gpt-3.5	31.2	43.7	63.1	34.5	74.2	61.1	76.6	55.5	62.8	55.9
TabLLM	gpt-3.5	34.6	44.8	69.1	46.3	82.0*	61.3	79.5	59.9	62.5	60.0
SumBoost	gpt-3.5	29.9	43.6	50.1	45.6	64.7	61.6	63.7	58.3	59.2	53.0
InsightTab	gpt-3.5	<u>38.8*</u>	<u>50.7*</u>	72.3*	50.8*	81.4	<u>65.1*</u>	<u>80.1*</u>	<u>61.8*</u>	64.4*	<u>62.8*</u>

Table 1: Overall performance for few-shot tabular classification. We use 16/32/64/128 samples for training and 16 shots for in-context demonstration. The averaged F1 scores (%) are reported, *i.e.*, each number (except for the Avg column) is averaged from $5 \times 4 = 20$ tests. The best and second-best results are **bolded** and underlined. * indicates the best performance among all methods using the same base LLM. Detailed results see Appendix C.1 and C.2.

sample grouping: samples within each leaf are grouped together and fed into a strong LLM LLM_s for rule summarization (lines 3–6). Since the group-wise rules could contain redundancy, we then perform a rule merging process, resulting in a concise set of task-level rules (line 7). The method then computes the entropy of the predicted class probabilities $M(x^{(i)}) \in \mathbb{R}^{|C|}$ of each training sample $x^{(i)} \in \mathcal{D}$ by the obtained model M (line 8). These entropy values are used as the ranking scores for selecting easy and hard samples (lines 9–10). For each chosen hard sample, we let the predictor LLM LLM_p (could be LLMs with a smaller size to save budget) make an intermediate prediction using easy examples as few-shot demonstrations and current rules as insights. Those incorrectly predicted hard samples are retained and sent to the summarizer LLM for rule enhancement (lines 11–13). Finally, we use the easy examples and updated rules as insights and derive the class prediction for $\tilde{y}$. An illustration of the mined rules see Appendix A.

4 Experiments

4.1 Experimental Setup

Datasets. Following previous works (Fang et al., 2024), we evaluate our approach with nine datasets, all collected by (Hegselmann et al., 2022). These datasets, with their diverse sizes and feature sets, serve as an ideal testbed for LLM-based tabular classification methods (details in Appendix B.1).

Baselines. We select a broad spectrum of methods for comparison, with an emphasis on few-shot approaches. These include classic tree-ensemble model **XGBoost** (Chen and Guestrin, 2016), pre-trained Transformer **TabPFN** (Hollmann et al.,

2022), a base LLM classifier **Original**, and more advanced LLM methods including **Tablet** (Slack and Singh, 2023), **TabLLM** (Hegselmann et al., 2022), and **SumBoost** (Manikandan et al., 2023).

Implementation. We employ both open- and close-sourced LLMs for prediction, *i.e.*, *mistral-7b* (Jin and Lu, 2023) and *gpt-3.5-turbo* (Brown et al., 2020). For rule summarization in the preprocessing step, we use *gpt-4-turbo*. All base prompts are borrowed from TabLLM. We conduct five-fold cross-validation by randomly sampling some instances from the training split as the training set, and reporting the average F1 score for comparison.

Detailed experimental setup information can be found in Appendix B (datasets in B.1, baselines in B.2, and the parameter used in B.3), with all prompt templates provided in Appendix E.

4.2 Overall Performance

We present the overall performance in Table 1. Our findings include: traditional tabular classification methods like XGBoost rely heavily on large datasets for optimal performance, making them notably ineffective in few-shot scenarios, as seen with the Bank, Blood, and Income datasets. TabPFN tackles the few-shot setting using prior-data modeling. However, it faces difficulties with class imbalance in certain datasets (*e.g.*, Bank and Blood).

The results affirm the potential of LLM-based methods for this task (*e.g.*, TabLLM). Building on this, InsightTab integrates data-driven insights with multifaceted serialization, leveraging summarization and reflection to outperform other LLM-based methods on **eight of nine** datasets. For the two datasets where InsightTab doesn’t reach the

LLM _p	Variant	Bank	Blood	Calhou.	Car	Creditg	Diabe.	Heart	Income	Jungle	Avg
mistral-7b	- demonstr.	17.6	<u>48.1</u>	13.9	20.6 [#]	11.5 [#]	51.8	11.4 [#]	31.0 [#]	46.3	28.0 [#]
	- grouping	10.1 [#]	<u>19.4[#]</u>	<u>12.3[#]</u>	44.3	12.9	51.3 [#]	55.6	44.5	37.8 [#]	32.0
	- reflection	<u>21.5</u>	44.8	<u>62.1</u>	47.4	<u>36.6</u>	<u>68.6</u>	<u>56.2</u>	47.8	48.9	<u>48.2</u>
	InsightTab	35.3	50.6	78.2	55.0	82.9	70.8	74.2	63.4	66.3	64.1
gpt-3.5-turbo	- demonstr.	21.5 [#]	40.9 [#]	47.2 [#]	36.4 [#]	47.3	55.3 [#]	15.8 [#]	33.2 [#]	62.4	40.0 [#]
	- grouping	25.1	<u>43.0</u>	<u>68.7</u>	38.6	57.1	59.6	64.0	59.5	<u>58.6[#]</u>	<u>52.7</u>
	- reflection	<u>28.1</u>	42.2	<u>68.0</u>	46.6	39.7 [#]	<u>60.6</u>	<u>66.1</u>	57.6	60.2	52.1
	InsightTab	37.9	50.2	72.6	50.4	79.8	64.2	83.6	59.5	65.6	62.6

Table 2: Ablation study results with 128 training samples, 16 shots, and averaged F1 scores (%). The best, second-best and worst results within each group are in **bold**, underlined, and with [#], respectively.

best performance, it ranks the second and closely follows the non-LLM leaders. Overall, it demonstrates average improvements of 21%, 32%, 55%, 37%, 10%, and 44% over XGBoost, TabPFN, Original, Tablet, TabLLM, and SumBoost, respectively. Interestingly, we find mistral-7b outperforms gpt-3.5 in InsightTab. This is because its architecture aligns better with structured, iterative reasoning tasks, whereas gpt-3.5 may overcomplicate or struggle with such setups. These findings highlight the advantage of LLMs with data-driven insight distillation for robust few-shot tabular classification.

4.3 Ablation Study

We conduct an ablation study by selectively removing the easy example demonstration, grouping, and reflection strategies from InsightTab. Table 2 presents the ablated F1 results averaged from five-fold cross-validation. With gpt-3.5-turbo as the predictor LLM, the demonstration strategy is the most impactful. In contrast, both demonstration and grouping notably boost performance with mistral-7b, achieving the highest gains across four and five datasets (with [#]). Overall, the performance gains are more pronounced for the ‘weaker’ LLM_p mistral-7b compared with gpt-3.5-turbo. Interestingly, InsightTab with mistral-7b ultimately outperforms with gpt-3.5-turbo by 1.5%. This result also implies that open-sourced LLMs enhanced with insight distillation can be effective few-shot tabular classifiers, significantly reducing serving costs. A closer analysis of the results with mistral-7b provides additional insights. For instance, removing grouping in the Bank, Blood, and Calhousing datasets leads to significant performance drop, indicating that summarized rules help manage class imbalance (statistics in Table 4) and better cope with numerical features (examples in Figure 12).

In summary, the three strategies consistently improve performance across nine datasets: demonstra-

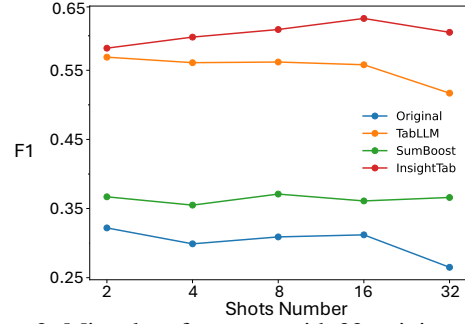


Figure 2: Mistral performance with 32 training samples across different number of shots (samples attached in the prompts). The results are averaged across 9 datasets.

tion aids in-context learning, grouping organizes data for rule summarization, and reflection refines results through iterative error correction. Together, these components ensure high accuracy and robustness. Appendix C.3 presents results of alternative strategies explored when developing InsightTab, further justifying the current model design.

5 In-depth Analysis

We conduct a series of in-depth analysis on the task, which could deepen our understanding for this emerging paradigm. Additionally, we put detailed cost analysis in Appendix D.

5.1 Demonstration Shot Analysis

We compare the performance of different models by adjusting the number of samples demonstrated in the prompts. Figure 2 shows the mistral performance (the results of gpt-3.5-turbo given in Appendix C.4). We observe a consistent trend across all four models as the number of shots increases. The performance initially improves but then gradually declines after 16 shots. In other words, while adding more examples in the prompt can enhance performance up to a point, an excessive number of examples may lead to negative impacts. This might be due to that including too many examples

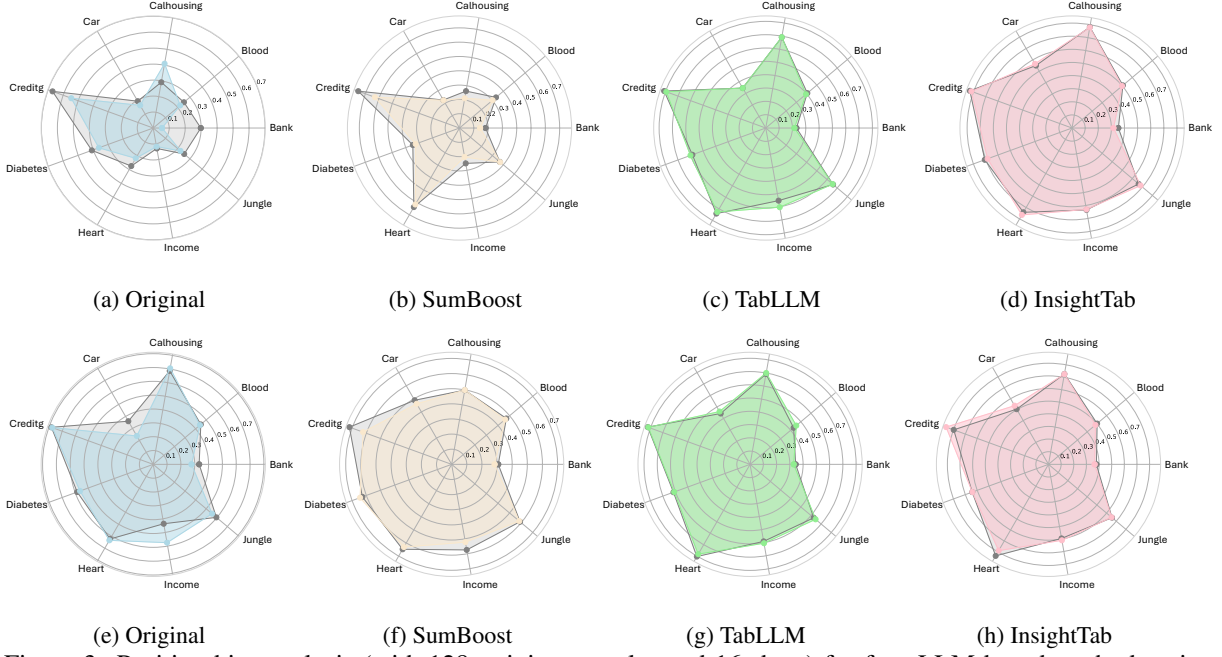


Figure 3: Position bias analysis (with 128 training samples and 16 shots) for four LLM-based methods using mistral-7b (upper) and gpt-3.5-turbo (lower). The grey/colored areas denote the F1 performance when the features of testing data are shuffled/non-shuffled, respectively.

causes the model to overfit to those specific instances, reducing its generalization to new, unseen data. However, methods incorporating rule generation (*e.g.*, SumBoost) alleviate this issue compared to others (*e.g.*, TabLLM). Specifically, our model shows advantage in handling more in-context samples. By integrating rule generation, our approach effectively mitigates the risk of overfitting, as it allows the model to extract and apply general principles rather than relying solely on individual examples, resulting in improved generalization and robustness in terms of large shot numbers.

5.2 Bias Analysis

We evaluate our model’s performance on position and class biases against baseline models.

Position bias analysis. We first investigate the model’s performance in addressing position bias. Specifically, we randomly shuffle the order of columns of every testing sample to assess whether the models can handle this discrepancy between the training and testing sets. We report the performance in Figure 3. We observe that both serialization (TabLLM vs. Original) and rule summarization (SumBoost vs. Original) improve the model’s ability to handle randomly shuffled testing cases, as shuffling has influenced the performance on many datasets (*e.g.*, Income) greatly in the Original method as shown in Figs. 3a & 3e. Since

the Original method may rely solely on memorizing specific feature order rules, lacking a meaningful semantic structure. In comparison, InsightTab demonstrates better robustness on feature order changes. The average performance change due to the shifting of the test data is significantly lower in our model compared to the SumBoost and Original baselines. After the shift, the performance of GPT-3.5-Turbo in our model reaches 64.2%, compared to 36.9% with SumBoost and 31.4% with the Original. Additionally, compared with TabLLM, InsightTab demonstrates better performance across different datasets. This demonstrates that our model effectively learns the semantic relationships between features via multifaceted serialization, rather than simply memorizing feature positions.

Unbalanced class analysis. To assess InsightTab’s performance on unbalanced datasets, Figure 5 compares F1 scores for True and False classes across datasets with increasing True/False ratios. We find the predictions of Original are heavily biased to class distributions. For example, the F1 score of True label on Bank with mistral-7b (see Figure 5a) is extremely low as 0.11 compared with 0.93 for the False label, because the True class constitutes only 11.7% of the dataset. However, compared with Original and TabLLM, this situation alleviates in InsightTab, showcasing more balanced results for both classes. In the Credit dataset, the

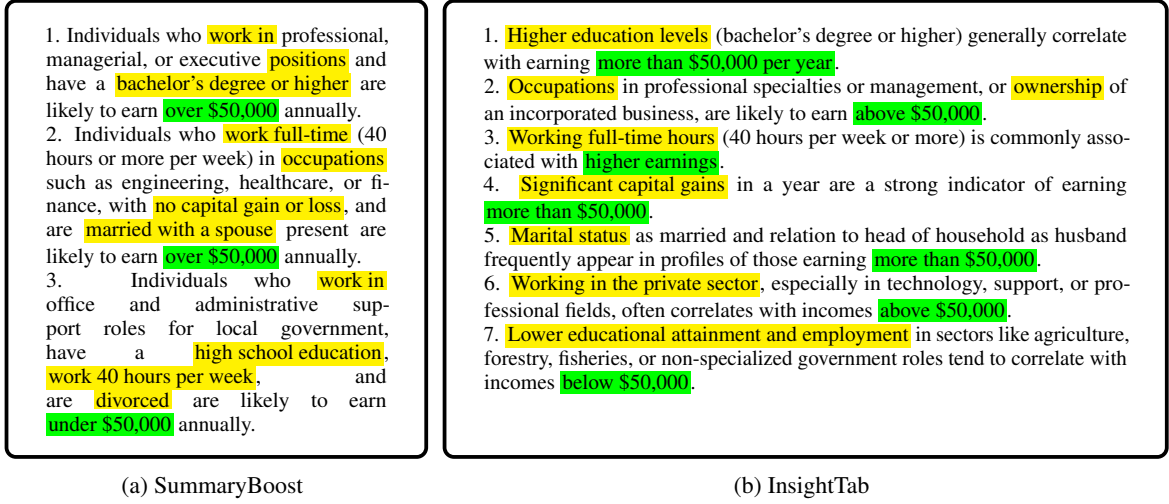


Figure 4: Rules summarized by SummaryBoost (baseline) and InsightTab (ours).

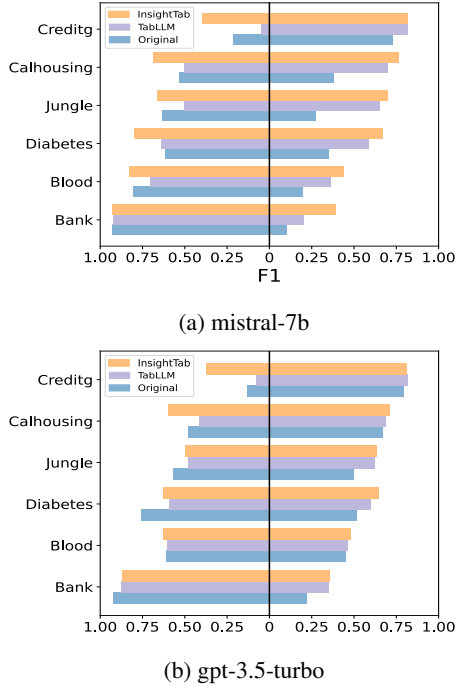


Figure 5: F1 scores for True and False classes across datasets (#32 training samples, 16 shots): False class on the left, True class on the right.

True/False F1 score of InsightTab is 0.82/0.41 with mistral-7b, compared with 0.81/0.06 by TabLLM. This confirms our model’s robustness on unbalanced datasets by ensuring representation of both classes via sampling.

5.3 Case Study

We showcase rules summarized by SummaryBoost and InsightTab to demonstrate the differences between the two methods in terms of data insights (see Figure 4). We select rules summarized in the Income dataset considering its scale and quality. We obtain several findings via the case study.

The rules offer a high-level interpretation of the feature-label relationship. We use yellow to highlight features and green to highlight labels, emphasizing the feature-label mapping in the summarized rules. A shared characteristic of rules generated by SummaryBoost and InsightTab is their essence as compressed representations of feature-label mappings. These rules precisely and clearly describe the relationships between specific features and labels, enabling LLMs (and humans) to efficiently interpret and perform tabular predictions.

Rules summarized by InsightTab are clearer and more understandable. A key characteristic of InsightTab’s rules is their one-to-one mapping, each focusing on a specific feature, while SummaryBoost generates many-to-one mappings, with each capturing multiple features for an individual. One-to-one mappings are simpler and more interpretable for weak table predictors, and the independence of the generated rules allows predictors to handle unseen feature combinations. In such cases, the predictor LLM is better equipped to prioritize important features and accurately predict labels based on the summarized rules.

6 Conclusion

We propose InsightTab, a novel method for accurate and robust LLM-based few-shot tabular classification. By distilling training data into actionable insights, InsightTab adapts LLMs to tabular tasks via the *divide-and-conquer*, *easy-first*, and *reflective learning* principles, implemented through group, rank, and summarize operators. They enable grouping samples to derive rules, selecting representative instances, and learning from errors

to enhance performance. We conduct experiments on nine datasets, comparing our method with state-of-the-art approaches. Our method consistently achieves superior average performance across all nine datasets. Extensive analysis shows that our model is more robust to positional and class biases.

Limitations

While InsightTab excels in few-shot tabular classification, its two-stage inference process incurs some cost. However, by leveraging off-the-shelf LLMs and few-shot demonstrations, it significantly reduces training time and expense compared to non-few-shot methods. It relies on a powerful language model LLM_s to summarize rules from tabular data, but this cost is minimal when amortized over many predictions and can be further reduced by reusing rules. Additionally, applying these rules to a more efficient open-source model LLM_p lowers serving costs without compromising performance.

Another limitation is our focus on the few-shot setting. When a sufficiently large training dataset is available, traditional methods like XGBoost are still more effective. Performance comparisons under full training data are discussed in Appendix C.2. Finally, evaluating LLM-based approaches faces challenges related to data contamination, which has been verified on tabular data (Bordt et al., 2024). This can lead to overestimated performance due to memorization of target values. We have made significant efforts to minimize this impact, such as employing five-fold cross-validation and studying feature position bias. Specifically, our position bias analysis suggests that both serialization and rule summarization, which can be viewed as perturbations to raw data (Li et al., 2024), enhance robustness of model behaviors. However, future work should follow this direction and explore whether our model remains effective when column names lack semantic meaning.

Ethics Statement

We utilize the full training and testing data from public databases (details provided in Appendix B.1) and ensure that no personal data is included in the datasets.

References

Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Naumovich Vapnik. 2002. [Support vector clustering](#). *J. Mach. Learn. Res.*, 2:125–137.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *International Conference on Machine Learning*.

Sebastian Bordt, Harsha Nori, Vanessa Cristiny Rodrigues Vasconcelos, Besmira Nushi, and Rich Caruana. 2024. [Elephants never forget: Memorization and learning of tabular data in large language models](#). In *First Conference on Language Modeling*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *ArXiv*, abs/2005.14165.

Jintai Chen, Kuan-Yu Liao, Yanwen Fang, Danny Ziyi Chen, and Jian Wu. 2023. [Tabcaps: A capsule neural network for tabular data classification with bow routing](#). In *International Conference on Learning Representations*.

Tianqi Chen and Carlos Guestrin. 2016. [Xgboost: A scalable tree boosting system](#). *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). *Preprint*, arXiv:2301.00234.

Lun Du, Fei Gao, Xu Chen, Ran Jia, Junshan Wang, Shi Han, and Dongmei Zhang. 2021. [Tabularnet: A neural network architecture for understanding semantic structures of tabular data](#). *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan H. Sengamedu, and Christos Faloutsos. 2024. [Large language models \(llms\) on tabular data: Prediction, generation, and understanding - a survey](#). *ArXiv*, abs/2402.17944.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. [Text-to-sql empowered by large language models: A benchmark evaluation](#). *ArXiv*, abs/2308.15363.

Yu. V. Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. [Revisiting deep learning models for tabular data](#). In *Neural Information Processing Systems*.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. [Why do tree-based models still outperform](#)

- deep learning on typical tabular data? In *Neural Information Processing Systems*.
- Sungwon Han, Jinsung Yoon, Sercan Ö. Arik, and Tomas Pfister. 2024. [Large language models can automatically engineer features for few-shot tabular learning](#). *ArXiv*, abs/2404.09491.
- Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David A. Sontag. 2022. [Tabllm: Few-shot classification of tabular data with large language models](#). *ArXiv*, abs/2210.10723.
- Noah Hollmann, Samuel G. Müller, Katharina Eggersperger, and Frank Hutter. 2022. [Tabpfn: A transformer that solves small tabular classification problems in a second](#). In *International Conference on Learning Representations*.
- Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. [In-context learning for few-shot dialogue state tracking](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Hiroshi Iida, Dung Ngoc Thai, Varun Manjunatha, and Mohit Iyyer. 2021. [Tabbie: Pretrained representations of tabular data](#). In *North American Chapter of the Association for Computational Linguistics*.
- Ziqi Jin and Wei Lu. 2023. [Tab-cot: Zero-shot tabular chain of thought](#). In *Annual Meeting of the Association for Computational Linguistics*.
- 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 *Neural Information Processing Systems*.
- Jiatong Li, Renjun Hu, Kunzhe Huang, Yan Zhuang, Qi Liu, Mengxiao Zhu, Xing Shi, and Wei Lin. 2024. [Perteval: Unveiling real knowledge capacity of LLMs with knowledge-invariant perturbations](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-Guang Lou. 2021. [Tapex: Table pre-training via learning a neural sql executor](#). *ArXiv*, abs/2107.07653.
- Weizheng Lu, Jiaming Zhang, Jing Zhang, and Yueguo Chen. 2024. [Large language model for table processing: A survey](#). *ArXiv*, abs/2402.05121.
- Hariharan Manikandan, Yiding Jiang, and J. Zico Kolter. 2023. [Language models are weak learners](#). *ArXiv*, abs/2306.14101.
- Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. 2023. [Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables](#). *ArXiv*, abs/2303.00918.
- Linyong Nan, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir R. Radev. 2023. [Enhancing few-shot text-to-sql capabilities of large language models: A study on prompt design strategies](#). *ArXiv*, abs/2305.12586.
- Ahmed Samy Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter W. J. Staar. 2022. [Tableformer: Table structure understanding with transformers](#). 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4604–4613.
- Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. [Tabicl: A tabular foundation model for in-context learning on large data](#). *arXiv preprint arXiv:2502.05564*.
- Chiranjibi Shah, Qian Du, and Yan Xu. 2022. [Enhanced tabnet: Attentive interpretable tabular learning for hyperspectral image classification](#). *Remote. Sens.*, 14:716.
- Ravid Shwartz-Ziv and Amitai Armon. 2021. [Tabular data: Deep learning is not all you need](#). *ArXiv*, abs/2106.03253.
- Ananya Singha, José Pablo Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023. [Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms](#). *ArXiv*, abs/2310.10358.
- Dylan Slack and Sameer Singh. 2023. [Tablet: Learning from instructions for tabular data](#). *ArXiv*, abs/2304.13188.
- Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. 2023. [Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning](#). *ArXiv*, abs/2312.09039.
- Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. [Easy-to-hard generalization: Scalable alignment beyond human supervision](#). *ArXiv*, abs/2403.09472.
- Svitlana Vakulenko and Vadim Savenkov. 2017. [Tableqa: Question answering on tabular data](#). *ArXiv*, abs/1705.06504.
- Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. [Tabert: Pretraining for joint understanding of textual and tabular data](#). *ArXiv*, abs/2005.08314.
- Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, Zhang Tao, Chengcheng Zhou, Kaizhe Shou, Miao Wang, Wufang Zhu, Guoshan Lu, Chaonan Ye, Yali Ye, Wen song Ye, Yiming Zhang, Xing yan Deng, J. Xu, Haobo Wang, Gang Chen, and Junbo Jake Zhao. 2023. [Tablegpt: Towards unifying tables, nature language and commands into one gpt](#). *ArXiv*, abs/2307.08674.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023. [Tablellama: Towards open large generalist models for tables](#). *ArXiv*, abs/2311.09206.

Xiaokang Zhang, Jing Zhang, Zeyao Ma, Yang Li, Bohan Zhang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-li, Jifan Yu, Shu Zhao, Juan-Zi Li, and Jie Tang. 2024. [Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios](#). *ArXiv*, abs/2403.19318.

Lei Zheng, Ning Li, Xianyu Chen, Quan Gan, and Weinan Zhang. 2023. [Dense representation learning and retrieval for tabular data prediction](#). *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

A Mined Rules

1. Higher education levels (bachelor’s degree or higher) generally correlate with earning more than \$50K per year.
2. Significant capital gains in a year are a strong indicator of earning more than \$50K.
3. Lower educational attainment and employment in sectors like agriculture, forestry, fisheries, or non-specialized government roles tend to correlate with incomes below \$50K.

Table 3: Illustration of mined rules for Income data.

Table 3 illustrates three mined rules related to the features presented in Fig. 1 for income prediction. As could be seen, these rules consist of both general knowledge from LLMs and task-specific details distilled from data, *e.g.*, bachelor’s degree or higher, capital gains, and employment. The combination of the two worlds enables the prediction LLM leveraging its capacity in a task-specific context for tabular prediction. The complete list of mined rules, multifaceted serialization prompts, and the prompts for rule summarization and merge could be found in Appendix E.

B Experimental Setup Details

B.1 Datasets

We adopt nine tabular datasets with varying sizes, diverse features, and different classification tasks for evaluation, first proposed by Hagselmann et al. (2022). Specifically, these datasets include **Bank** (predict whether a customer subscribed to a term deposit or not), **Blood** (predict whether a person will return for another blood donation), **California Housing / Calhousing** (predict the median house value in each district), **Car** (evaluate the safety state level of a car), **Creditg** (predict whether a person is at high or low probability for credit

Dataset	#Rows	#F	Classes	Cls Dist. (%)
Bank	45,211	16	binary	11.7/88.3
Blood	748	4	binary	23.8/76.2
Calhousing	20,640	8	binary	50.0/50.0
Car	1,728	8	0/1/2/3	70/22/4/4
Creditg	1,000	20	binary	70.0/30.0
Diabetes	768	8	binary	34.9/65.1
Heart	918	11	binary	55.3/44.7
Income	48,842	14	binary	23.9/76.1
Jungle	44,819	6	binary	48.5/51.5

Table 4: Dataset statistics, where ‘#F’ and ‘Cls Dist.’ denote the number of features and Class Distribution, respectively.

card approval), **Diabetes** (predict whether a person has diabetes), **Heart** (predict whether a person has coronary artery disease), **Income** (predict whether each person has an annual income over \$50,000) and **Jungle** (predict whether the white player will win in a jungle chess game). The statistics of each dataset are summarized in Table 4 and the raw data is available at <https://github.com/clinicalml/TabLLM>.

B.2 Baselines

We compare our method InsightTab with a broad spectrum of methods, with a focus on few-shot approaches:

- **XGBoost** (Chen and Guestrin, 2016) is a classic tree-ensemble model that remains one of the most effective methods for tabular prediction in general (Grinsztajn et al., 2022). However, it inherently relies on a substantial amount of data to identify optimal splitting features and values, which typically leads to its ineffectiveness in few-shot settings.
- **TabPFN** (Hollmann et al., 2022) is a trained Transformer that performs in-context learning to do supervised classification for small tabular datasets without requiring further parameter updates.
- **Original** is the most straightforward LLM baseline, which classifies tabular data by feeding the features as a textual prompt into LLMs. Each tabular data instance is serialized as {column_name}:{value} separated by a space between columns.
- **Tablet** (Slack and Singh, 2023) employs automatic generation of dataset-specific natural language instructions to boost the tabular classification performance with LLMs. We use

prototypes instructions as they are reported preforming better in the original paper.

- **TabLLM** (Hegselmann et al., 2022) is an LLM-based tabular classification model designed for few-shot settings. It explores extensively on the different serialization methods and finds that the *Text Template* approach performs well in general.
- **SumBoost** (Manikandan et al., 2023) is the most state-of-the-art LLM method for tabular classification. It leverages a boosting strategy to sample and summarize training data for LLM instruction optimization.

B.3 Implementation Details

We grid-search hyper-parameters for XGBoost (*i.e.*, `n_estimators` in {10, 50, 100, 200}, `max_depth` in {3, 4, 5, 6}, `learning_rate` in {0.01, 0.03, 0.1, 0.3}) and TabPFN (*i.e.*, `n_ensemble` in {2, 4, 8, 16, 32, 64}). We adopt the other parameters the same from the default settings in the XGBoost implementations in sklearn. We set the default value for the number of boosting rounds to 100, which is the same as that in `xgboost.XGBRFClassifier`. The rest of parameters are listed as follows:

```
XGBClassifier(
    base_score=0.5, booster='gbtree',
    colsample_bylevel=1,
    colsample_bynode=1,
    colsample_bytree=1, gamma=0,
    gpu_id=-1,
    importance_type='gain',
    interaction_constraints='',
    learning_rate=0.3,
    max_delta_step=0, max_depth=6,
    min_child_weight=1, missing=nan,
    monotone_constraints='()',
    n_estimators=100, n_jobs=12,
    num_parallel_tree=1,
    objective='multi:softprob',
    random_state=0, reg_alpha=0,
    reg_lambda=1,
    scale_pos_weight=None,
    subsample=1,
    tree_method='exact',
    use_label_encoder=False,
    validate_parameters=1,
    verbosity=None
)
```

Moreover, OrdinalEncoding is chosen for categorical feature encoding in TabPFN, which is recommended. For InsightTab, we set the hyper-parameter n_e of easy samples to the number of shots and n_h of hard samples equal to 50% of the training data. When LLM-based approaches produce invalid answers that do not fit proper formats,

we label them as ‘no answer’. We conduct five-fold cross validation for quantitative comparison. For each dataset and each run, we split the data into training and testing splits with an 80%/20% ratio. For our few-shot setting, we randomly sample 16, 32, 64, and 128 instances as the final training sets, along with the full training set for reference. Concerning the large scale of some datasets, we randomly sample 1,000 instances from any test set whose size exceeding 1,000 as the final testing set for resource concerns.

C Extra Experimental Results

C.1 Detailed Overall Performance Results

The results of detailed overall performance with $n = 16/32/64/128$ are presented in Table 5. As can be seen, in the very-few shot scenario (*i.e.*, $n = 16$), both XGBoost and TabPFN could occasionally fail completely. As n increases to 128, these two methods stand out on some datasets. However, LLM-based methods represented by InsightTab, are more effective in general. Moreover, InsightTab obtains the best results in most cases among LLM-based solutions, with only a few exceptions to TabLLM in $n = 16/32$. In other words, InsightTab are better in leveraging the few-shot examples than its competitor TabLLM. This few-shot scenario has great research value for reduced training overhead, as performance can be optimized primarily through effective prompting strategies, saving substantial time in the classification process. Additionally, it has great importance particularly by addressing bias and safety concerns without requiring full data visibility.

C.2 Performance Under Full Training Data

Table 5 further presents the results of tested models under full training data, *i.e.*, $n = \text{all}$. Note that TabPFN is designed for few-shot settings, which encounters runtime errors in this set of tests. With an abundant amount of training data, XGBoost yields the best performance on six of nine datasets, reclaiming its advantage for general tabular prediction. However, there are still three datasets where InsightTab outperforms XGBoost, indicating the unique values of LLMs for this task. Finally, InsightTab is the only LLM-based solution that outperforms XGBoost, which we consider a milestone for this topic of study.

Method	Bank	Blood	Calhou.	Car	Creditg	Diabe.	Heart	Income	Jungle
$n = 16$									
XGBoost	0.0	27.3	66.4	33.2	80.1	31.2	69.3	10.5	52.8
TabPFN	1.3	21.6	72.1	22.1	82.2	29.2	79.0	0.4	41.6
Original	5.6/19.6	29.0/45.3	44.7/67.0	20.1/22.9	72.9/73.8	35.3/43.1	27.4/55.5	13.2/44.9	12.4/43.8
Tablet	11.3/31.8	27.6/43.8	53.4/62.1	26.9/36.2	66.1/71.2	40.5/61.9	48.4/77.1	23.1/54.7	32.3/62.8
TabLLM	21.1/32.9	38.7/44.0	68.7/69.4	35.2/ <u>50.0</u>	80.2/ 82.2	60.0/62.1	73.7/ <u>79.9</u>	58.5/59.6	57.8/61.5
SumBoost	16.2/28.7	30.1/35.6	27.9/57.1	23.4/41.2	64.3/49.9	36.3/59.3	60.4/62.6	26.0/52.6	39.1/60.0
InsightTab	48.8/41.7	62.9/54.9	79.7/75.8	48.4/ 51.5	80.4/80.9	76.4/68.2	80.0/76.7	67.5/61.0	69.4/64.9
$n = 32$									
XGBoost	14.9	23.5	66.5	33.6	76.5	52.2	74.6	39.8	69.6
TabPFN	16.6	9.5	<u>76.0</u>	23.8	81.8	37.2	79.8	8.6	<u>70.1</u>
Original	10.4/31.5	19.9/45.5	38.3/67.1	19.7/35.1	72.7/80.2	35.3/57.9	34.9/60.7	13.9/44.7	27.5/58.1
Tablet	13.6/34.0	33.4/42.6	54.9/65.3	25.9/32.3	66.6/77.6	39.3/61.7	43.8/77.3	25.4/55.3	42.6/63.0
TabLLM	20.6/35.3	36.6/ <u>46.4</u>	70.1/69.1	36.8/ <u>45.5</u>	81.6/ 81.8	58.9/60.1	73.8/78.6	58.2/61.6	65.3/62.4
SumBoost	19.2/30.2	35.4/46.3	20.4/50.1	22.9/44.6	60.9/70.8	38.1/61.0	64.7/62.9	26.7/ <u>62.1</u>	36.8/58.8
InsightTab	39.5/35.8	44.1/ 48.3	76.4/71.2	43.9/ 49.1	81.8/81.4	66.8/64.6	<u>77.5/78.8</u>	62.0/ 64.7	70.2/63.6
$n = 64$									
XGBoost	32.3	39.2	72.6	43.3	77.3	58.5	<u>81.9</u>	36.2	<u>72.2</u>
TabPFN	21.4	11.6	80.0	28.5	80.4	57.7	83.7	35.0	72.3
Original	22.1/31.4	21.5/45.7	36.7/70.2	21.5/37.4	70.0/80.4	35.1/56.9	32.4/56.1	15.3/46.2	30.0/57.0
Tablet	5.9/27.4	28.0/42.0	55.3/63.8	24.3/34.4	66.6/76.0	37.0/61.4	37.4/74.8	22.4/56.4	40.7/62.1
TabLLM	22.1/35.6	36.1/45.9	68.9/68.8	34.6/45.2	80.7/81.8	59.1/61.5	74.7/79.3	58.9/59.3	66.2/63.5
SumBoost	14.8/29.8	32.3/45.2	18.2/43.3	22.4/ <u>47.8</u>	56.0/66.0	42.8/63.1	61.3/64.6	21.7/60.9	36.8/59.0
InsightTab	40.8/39.7	49.5/49.3	<u>74.6/69.7</u>	47.2/ 52.2	<u>82.6/83.5</u>	<u>64.4/63.3</u>	75.0/81.2	<u>72.9/62.0</u>	71.6/63.6
$n = 128$									
XGBoost	<u>36.2</u>	41.4	76.9	<u>50.7</u>	79.7	58.3	85.9	49.2	<u>76.8</u>
TabPFN	18.0	27.8	83.2	47.7	80.6	62.5	<u>84.8</u>	36.8	77.2
Original	29.8/33.2	25.1/44.9	29.1/68.6	19.4/36.0	68.1/78.3	40.9/58.5	27.5/62.3	13.0/43.6	25.3/59.6
Tablet	12.2/31.5	30.0/46.5	55.9/61.0	24.5/35.1	67.5/71.8	42.4/59.3	39.4/77.3	25.2/55.6	40.3/63.1
TabLLM	23.0/34.4	40.7/42.9	69.1/69.2	34.6/44.3	81.1/ <u>82.1</u>	58.6/61.5	74.1/80.3	58.3/59.0	65.9/62.7
SumBoost	18.2/30.8	33.3/47.2	26.2/49.8	22.6/48.9	75.3/71.9	34.7/62.8	63.8/64.8	25.1/57.4	36.8/59.0
InsightTab	35.3/ 37.9	50.6/50.2	<u>78.2/72.6</u>	55.0/50.4	82.9/79.8	<u>70.8/64.2</u>	74.2/83.6	63.4/59.5	66.3/65.6
$n = \text{all}$									
XGBoost	54.5	42.4	90.5	97.7	<u>82.8</u>	60.9	87.6	72.4	89.1
TabPFN					/				
Original	17.8/31.2	24.1/44.8	47.7/68.8	21.3/34.4	71.4/78.9	40.4/57.0	26.9/58.0	12.0/47.6	26.9/55.2
Tablet	10.1/33.7	29.3/46.0	57.9/66.1	24.2/33.5	60.2/70.5	39.1/61.5	36.8/77.1	28.5/55.8	25.3/63.4
TabLLM	24.1/34.4	40.1/45.0	69.0/69.1	33.1/43.4	80.1/82.2	57.9/61.5	73.5/78.9	55.9/59.2	65.7/62.3
SumBoost	20.5/32.0	34.2/45.3	48.9/67.7	26.7/46.1	79.9/78.7	41.0/58.4	65.4/79.1	30.2/52.9	28.6/61.1
InsightTab	36.8/ <u>36.9</u>	<u>47.6/48.8</u>	<u>71.6/69.4</u>	47.6/ <u>52.9</u>	83.7/79.8	65.5/62.7	<u>75.6/80.3</u>	<u>64.2/59.5</u>	<u>70.8/63.8</u>

Table 5: Overall F1 performance on the public tabular datasets with different number of training samples n . For each LLM-based model, we report the performance based on mistral-7b/gpt-3.5-turbo using 16 shots. The best and second-best results are **bolded** and underlined.

C.3 Comparative Strategy Details

During the development of InsightTab, we have explored various implementations of the sampling and grouping strategies. We believe reporting the results of these alternative strategies would also help better understanding our final InsightTab. We first summarize the details of these strategies:

- Random demonstration (R demos.) only adopts random examples as demonstration, *i.e.*, we only add random samples into prompts without any other rules.
- SV demonstration (SV demos.) is another demonstration-only strategy. Specifically,

we use the entropy given by an SVC algorithm (Ben-Hur et al., 2002) to rank samples and select those easiest as demonstration.

- LR rule is a rule-only method without demonstration. It leverages the linear regression algorithm to group training samples with fixed interval and use the rule summarization LLM to summarize rules.
- Cocktail combines SV demonstration and LR rule. To make a fair comparison, we also add the reflective learning strategy to this method.

We show the results in Table 6. From the table, we can see that the Cocktail approach achieves

LLM _p	Variant	Bank	Blood	Calhou.	Car	Creditg	Diabe.	Heart	Income	Jungle	Avg
mistral-7b	R demos.	10.4	12.4	17.3	39.2	35.1	42.2	56.0	32.4 [#]	13.3 [#]	28.7
	SV demos.	10.2 [#]	13.2	16.8	37.7 [#]	<u>35.8</u>	41.5 [#]	53.3 [#]	33.8	14.7	28.6 [#]
	LR rule	11.0	11.1 [#]	17.4	39.6	35.5	43.9	54.3	35.6	15.8	29.4
	Cocktail	<u>12.0</u>	<u>15.9</u>	14.1 [#]	38.4	34.8 [#]	<u>54.3</u>	<u>66.7</u>	<u>50.3</u>	<u>18.2</u>	<u>33.9</u>
	InsightTab	35.3	50.6	78.2	55.0	82.9	70.8	74.2	63.4	66.3	64.1
gpt-3.5-turbo	R demos.	26.6	42.8	73.9	41.6	<u>68.3</u>	62.2	68.3 [#]	52.2	65.8	<u>55.7</u>
	SV demos.	<u>27.4</u>	40.5 [#]	71.4	40.1	67.8	61.2	69.9	54.5	64.4	55.2
	LR rule	25.4 [#]	41.4	71.1	<u>42.3</u>	68.2	<u>62.5</u>	68.9	37.0 [#]	62.3	53.2 [#]
	Cocktail	26.1	<u>52.6</u>	70.5 [#]	39.7 [#]	47.5 [#]	59.6 [#]	<u>71.9</u>	<u>59.1</u>	59.2 [#]	54.0
	InsightTab	37.9	50.2	<u>72.6</u>	50.4	79.8	64.2	83.6	59.5	<u>65.6</u>	62.6

Table 6: Results of alternative strategies with 128 training samples, 16 shots, and averaged F1 scores (%). The best, second-best and worst results within each group are in **bold**, underlined, and with [#], respectively.

	Bank	Blood	Calhous.	Car	Creditg	Diabetes	Heart	Income	Jungle
Rule Generation (gpt-4o-mini)	0.02	0.03	0.01	0.02	0.01	0.01	0.02	0.02	0.01
Rule Generation (gpt-4-turbo)	1.58	1.66	1.87	1.97	1.80	1.67	1.70	1.80	1.72
Prediction (gpt-3.5-turbo)	0.69	0.28	0.42	0.26	0.29	0.22	0.30	0.73	0.84
SumBoost (gpt-4-turbo)	12.3	14.5	15.8	14.1	11.4	10.4	15.7	18.9	17.5

Table 7: Overall cost (USD) of InsightTab. Each number is reported using 5-fold validation on 128 training examples, 16 number of shots.

the best overall score compared to other variants, boasting the highest mistral F1 score across six datasets. This demonstrates the advantage of integrating rules and samples into the prompts, as it allows the LLM to comprehend the data structure at a high level through rules while also providing illustrative examples. The improvement is the most evident in the Income dataset, which contains the largest number of data records. We also notice that in the Car dataset, using rules-only (LR rule) achieves the best performance. This is likely because the dataset contains four classes, compared to other datasets which typically have only two classes. The presence of more classes might make it easier for the model to derive effective rules that capture the distinctions between categories without requiring additional examples. Overall, these findings highlight the importance of combining rules and samples to enhance the model’s performance. By leveraging the strengths of both rules and examples, we can significantly improve the model’s ability to understand and process diverse datasets effectively.

C.4 GPT-3.5 Training Shot Result

We show the GPT-3.5-based training shot result in Figure 6.

D InsightTab Training Cost

We attach the overall cost below in Tables 7 and 8. We see our method shows advantage against the

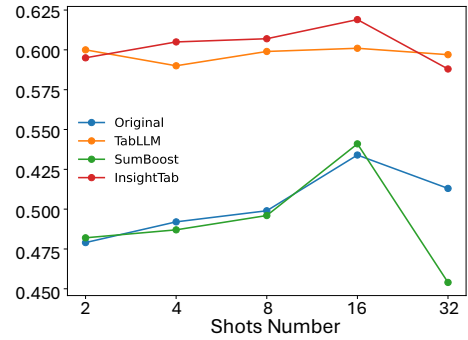


Figure 6: gpt-3.5-turbo performance with 32 training samples across different number of shots (samples attached in the prompts). The results are averaged across 9 datasets.

most state-of-the-art LLM prompt-based method SumBoost both in terms of time and budget cost.

E LLM Prompts

We finally present prompt templates and prompt examples used in InsightTab:

- **Prompt template for generating group-level rules** (Fig. 7) is used for group-wise rule summarization. Contents in braces are replaced with the meta data of datasets or sample information in application.
- **Prompt template for merging group-level rules** (Fig. 8) is used for merging group-wise rules.

	Time
Rule Generation (gpt-4o-mini)	2min 20s
Rule Generation (gpt-4-turbo)	2min 20s
Prediction (gpt-3.5-turbo)	5min per 10k testing samples
SumBoost (gpt-4-turbo)	20min per 10k testing samples

Table 8: Time cost of InsightTab and baselines.

- **Prompt template for tabular classification**

(Fig. 9) is used for tabular classification based on rules and few-shot samples.

- **Tabular classification prompt examples**

(Figs. 10–18) are examples of tabular classification prompts on the bank, blood, calhousing, car, creditg, diabetes, heart, income and jungle datasets, respectively. We reserve only one few-shot sample for each prompt due to page size limitation, which is different from the experiment setup. For creditg dataset (Figure 14), we reserve only the first, the second and the last rules due to page size limitation. As mentioned above, additional rules are adaptive in tabular classification prompts.

```

{Serialized features of sample 1}
{Question description}
Answer: {Label}
---
...
---
{Serialized features of sample  $n$ }
{Question description}
Answer: {Label}
---
Please distill the key trends that may assist an AI model in making future predictions. Output trends only without any further explanations.

```

Figure 7: Prompt template for generating group-level rules.

```

{Rules of group 1}
---
...
---
{Rules of group  $n$ }
---
Tl;dr / Summarize the rules into a small set of non-conflicting and complementary patterns for predicting whether a person earns more than 50000 dollars per year. Output patterns only without any further explanations.

```

Figure 8: Prompt template for merging group-level rules.

```

{Dataset title}
{Dataset description}

Useful patterns for the task at hand:
{Summarized rules}

Additional patterns for the task summarized from incorrectly classified examples with high entropy:
{Additional rules}

###

[FEW-SHOT EXAMPLES START]

{Few-shot examples}

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]
{Serialized features of the current question}
{Question description}
Answer:

```

Figure 9: Prompt template for tabular classification.

Title: Deposit subscription prediction

This data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. The classification goal is to predict if the client will subscribe a term deposit.

Useful patterns for the task at hand:

1. **Previous Campaign Outcome**: Clients with a successful outcome in the previous marketing campaign are more likely to subscribe to a term deposit.
2. **Contact Duration**: Longer last contact durations tend to correlate with a higher likelihood of subscription.
3. **Housing and Personal Loans**: Clients without housing or personal loans are more likely to subscribe to a term deposit.
4. **Number of Contacts in Current Campaign**: A higher number of contacts does not necessarily increase the likelihood of subscription; fewer, more effective contacts might be more beneficial.
5. **Communication Type**: While not a strong predictor on its own, effective communication types like cellular phones are commonly used.
6. **Yearly Balance**: Clients with higher average yearly balances tend to have a slightly higher likelihood of subscribing, though this is not a strong predictor.
7. **Previous Campaign Contact**: Clients not previously contacted in earlier campaigns show varied subscription outcomes, suggesting that this alone is not a strong predictor.
8. **Marital Status, Job Type, and Education Level**: These factors do not show a clear trend in influencing the decision to subscribe to a term deposit.
9. **Credit in Default**: Clients with no credit default are commonly targeted but this factor alone does not predict subscription.
10. **Last Contact Timing (Month and Day)**: No clear trend on how specific months or days impact the likelihood of subscription.

###

[FEW-SHOT EXAMPLES START]

The age is 41. The type of job is management. The marital status is married. The education is tertiary. The has credit in default? is no. The average yearly balance, in euros is 144. The has housing loan? is no. The has personal loan? is no. The contact communication type is cellular. The last contact day of the month is 27. The last contact month of year is feb. The last contact duration, in seconds is 123. The number of contacts performed during this campaign and for this client is 3. The number of days that passed by after the client was last contacted from a previous campaign is client was not previously contacted. The number of contacts performed before this campaign and for this client is 0. The outcome of the previous marketing campaign is unknown.

Does this client subscribe to a term deposit?

Answer: No

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The age is 36. The type of job is self-employed. The marital status is married. The education is secondary. The has credit in default? is no. The average yearly balance, in euros is 96. The has housing loan? is no. The has personal loan? is no. The contact communication type is cellular. The last contact day of the month is 27. The last contact month of year is aug. The last contact duration, in seconds is 125. The number of contacts performed during this campaign and for this client is 15. The number of days that passed by after the client was last contacted from a previous campaign is client was not previously contacted. The number of contacts performed before this campaign and for this client is 0. The outcome of the previous marketing campaign is unknown.

Does this client subscribe to a term deposit? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 10: A prompt example for Bank dataset.

Title: Blood donation prediction

This data is to predict whether a given individual will consent or avoid donating blood.

Useful patterns for the task at hand:

1. Lower recency values correlate with a higher likelihood of donating blood.
2. Higher frequency of donations increases the likelihood of donating again.
3. The total volume of blood donated does not consistently predict donation likelihood.
4. Shorter time since the first donation does not consistently predict donation likelihood.
5. A combination of low recency and high frequency often predicts a positive donation outcome.

###

[FEW-SHOT EXAMPLES START]

The Recency - months since last donation is 2. The Frequency - total number of donation is 13. The Monetary - total blood donated in c.c. is 3250. The Time - months since first donation is 53.

Will the person donate blood?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Recency - months since last donation is 3. The Frequency - total number of donation is 5. The Monetary - total blood donated in c.c. is 1250. The Time - months since first donation is 38.

Will the person donate blood? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 11: A prompt example for Blood dataset.

Title: House value prediction

This dataset is collected on the variables using all the block groups in California from the 1990 Census. It computes distances among the centroids of each block group as measured in latitude and longitude, aims to predict whether the house value of district is below or above the median.

Useful patterns for the task at hand:

1. Higher median income is associated with higher house block value.
2. Lower median age may indicate a more desirable area, potentially increasing property value.
3. A higher number of total rooms and bedrooms often correlates with more valuable properties.
4. Population size and number of households can suggest demand, but are not definitive indicators of value.
5. Geographic location (latitude and longitude) may influence property value based on regional trends.

###

[FEW-SHOT EXAMPLES START]

The median income is 3.9097. The median age is 52. The total rooms is 2684. The total bedrooms is 574. The population is 1395. The households is 549. The latitude is 37.76. The longitude is -122.48.

Is this house block valuable?

Answer: Yes

- - -

The median income is 3.2344. The median age is 36. The total rooms is 2433. The total bedrooms is 585. The population is 1565. The households is 563. The latitude is 34.08. The longitude is -118.12.

Is this house block valuable?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The median income is 3.4659. The median age is 31. The total rooms is 2567. The total bedrooms is 507. The population is 1198. The households is 499. The latitude is 32.78. The longitude is -117.02.

Is this house block valuable? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 12: A prompt example for Calhousing dataset.

Title: Car safety prediction

This dataset was derived from a simple hierarchical decision model originally developed for the demonstration of DEX. The goal is to evaluate the safety of cars.

Useful patterns for the task at hand:

1. Cars accommodating only two persons are generally rated as unacceptable.
2. High maintenance costs frequently contribute to an unacceptable rating.
3. Safety scores alone do not compensate for negative factors such as low person capacity or high maintenance costs.
4. High buying prices combined with very high maintenance costs lead to an unacceptable rating.
5. The number of doors and trunk size do not consistently influence the acceptability rating.

Additional patterns for the task summarized from incorrectly classified examples with high entropy:

1. Low buying price → Unacceptable decision.
2. High maintenance costs → Unacceptable decision.
3. Low safety score → Unacceptable decision.
4. Medium safety score + favorable factors → Acceptable decision.
5. More than four persons capacity + negative factors → Unacceptable decision.
6. Five or more doors → Favorable but not decisive.
7. Big trunk size → Positive factor but not decisive.

###

[FEW-SHOT EXAMPLES START]

The Buying price is medium. The Doors is five or more. The Maintenance costs is very high. The Persons is two. The Safety score is medium. The Trunk size is medium.

How would you rate the decision to buy this car?

Answer: Unacceptable

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Buying price is low. The Doors is five or more. The Maintenance costs is medium. The Persons is two. The Safety score is low. The Trunk size is medium.

How would you rate the decision to buy this car? Answer the question with either 'Unacceptable', 'Acceptable', 'Good', or 'Very Good' (without quotes).

Answer: <xxx, Unacceptable/Acceptable/Good/Very Good>

Figure 13: A prompt example for Car dataset.

Title: Credit risk prediction

This dataset originates from the UCI Machine Learning Repository. It is used to classify individuals as good or bad credit risks based on various attributes.

Useful patterns for the task at hand:

1. ****Status of Existing Checking Account****:
 - No checking account or a positive balance often leads to approval.
 - Higher balances (≥ 200 DM) tend to result in denial.
2. ****Credit History****:
 - A history of paying back credits duly often leads to approval.
- ...
16. ****Foreign Worker****:
 - Being a foreign worker generally does not prevent credit approval.

###

[FEW-SHOT EXAMPLES START]

The Status of existing checking account is < 0 DM. The Duration in month is 33. The Credit history is critical account/ other credits existing (not at this bank). The Purpose is furniture/equipment. The Credit amount is 4281. The Savings account/bonds is $500 \leq \dots < 1000$ DM. The Present employment since is $1 \leq \dots < 4$ years. The Installment rate in percentage of disposable income is 1. The Personal status and sex is female : divorced/separated/married. The Other debtors / guarantors is none. The Present residence since is 4. The Property is car or other, not in attribute 6. The Age in years is 23. The Other installment plans is none. The Housing is own. The Number of existing credits at this bank is 2. The Job is skilled employee / official. The Number of people being liable to provide maintenance for is 1.0. The Telephone is none. The foreign worker is yes.

Does this person receive a credit?
Answer: No

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Status of existing checking account is < 0 DM. The Duration in month is 9. The Credit history is existing credits paid back duly till now. The Purpose is car (new). The Credit amount is 654. The Savings account/bonds is $\dots < 100$ DM. The Present employment since is $1 \leq \dots < 4$ years. The Installment rate in percentage of disposable income is 4. The Personal status and sex is male : single. The Other debtors / guarantors is none. The Present residence since is 3. The Property is car or other, not in attribute 6. The Age in years is 28. The Other installment plans is none. The Housing is own. The Number of existing credits at this bank is 1. The Job is unskilled - resident. The Number of people being liable to provide maintenance for is 1.0. The Telephone is none. The foreign worker is yes.

Does this person receive a credit? Answer the question with either 'Yes' or 'No' (without quotes).
Answer: <xxx, No/Yes>

Figure 14: A prompt example for Creditg dataset.

Title: Diabetes risk prediction

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has high/low risk of developing diabetes.

Useful patterns for the task at hand:

1. **Plasma Glucose Concentration at 2 Hours in GTT**: Values above 140 mg/dL are indicative of diabetes.
2. **Body Mass Index (BMI)**: Values over 30 often correlate with diabetes diagnoses.
3. **Diabetes Pedigree Function**: Higher values generally correlate with a higher risk of diabetes.
4. **Age**: Middle-aged and older adults show a higher prevalence of diabetes.
5. **Number of Pregnancies**: Increased number of pregnancies tends to be associated with a higher likelihood of diabetes.
6. **2-Hour Serum Insulin**: Higher levels can be indicative of diabetes, especially when combined with other risk factors.
7. **Triceps Skin Fold Thickness**: Higher values are often observed in patients with diabetes.
8. **Diastolic Blood Pressure**: Readings above 80 mmHg frequently appear in patients diagnosed with diabetes.

###

[FEW-SHOT EXAMPLES START]

The Age is 52. The Number of times pregnant is 8. The Diastolic blood pressure is 76. The Triceps skin fold thickness is 24. The Plasma glucose concentration at 2 hours in an oral glucose tolerance test (GTT) is 124. The 2-hour serum insulin is 600. The Body mass index is 28.7. The Diabetes pedigree function is 0.687.

Does this patient have diabetes?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Age is 43. The Number of times pregnant is 7. The Diastolic blood pressure is 80. The Triceps skin fold thickness is 31. The Plasma glucose concentration at 2 hours in an oral glucose tolerance test (GTT) is 109. The 2-hour serum insulin is 0. The Body mass index is 35.9. The Diabetes pedigree function is 1.127.

Does this patient have diabetes? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 15: A prompt example for Diabetes dataset.

Title: Heart disease prediction

This dataset contains 11 features that can be used to predict a possible heart disease.

Useful patterns for the task at hand:

1. Presence of exercise-induced angina or asymptomatic chest pain with other risk factors suggests heart disease.
2. ST depression induced by exercise relative to rest, especially values of 2.0 or higher, indicates heart disease.
3. A flat or downsloping slope of the peak exercise ST segment is indicative of heart disease.
4. High fasting blood sugar levels (> 120 mg/dl) are associated with heart disease.
5. Lower maximum heart rate achieved during exercise, particularly when combined with other risk factors, suggests heart disease.
6. Abnormalities in resting electrocardiogram results (such as ST-T wave abnormality or left ventricular hypertrophy) indicate heart disease, even if chest pain is absent.
7. Elevated resting blood pressure and high serum cholesterol levels are risk factors that, when combined with other indicators, suggest heart disease.

Additional patterns for the task summarized from incorrectly classified examples with high entropy:

1. Older age increases the likelihood of heart disease.
2. Asymptomatic cases are more likely to indicate heart disease compared to non-anginal pain.
3. Elevated serum cholesterol levels are indicative of heart disease.
4. Elevated fasting blood sugar (> 120 mg/dl) may suggest heart disease.
5. Abnormal resting electrocardiogram results indicate a higher risk of heart disease.
6. Presence of exercise-induced angina is a strong indicator of heart disease.
7. Higher ST depression induced by exercise is associated with heart disease.
8. Flat slopes of the peak exercise ST segment may indicate heart disease.

###

[FEW-SHOT EXAMPLES START]

The Age of the patient is 46. The Sex of the patient is male. The Chest pain type is asymptomatic. The Resting blood pressure is 120. The Serum cholesterol is 231. The Fasting blood sugar > 120 mg/dl is no. The Resting electrocardiogram results is normal. The Maximum heart rate achieved is 115. The Exercise-induced angina is yes. The ST depression induced by exercise relative to rest is 0.0. The Slope of the peak exercise ST segment is flat.

Does the coronary angiography of this patient show a heart disease?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Age of the patient is 59. The Sex of the patient is male. The Chest pain type is asymptomatic. The Resting blood pressure is 170. The Serum cholesterol is 326. The Fasting blood sugar > 120 mg/dl is no. The Resting electrocardiogram results is probable or definite left ventricular hypertrophy. The Maximum heart rate achieved is 140. The Exercise-induced angina is yes. The ST depression induced by exercise relative to rest is 3.4. The Slope of the peak exercise ST segment is downsloping.

Does the coronary angiography of this patient show a heart disease? Answer the question with either 'Yes' or 'No' (without quotes). Answer: <xxx, No/Yes>

Figure 16: A prompt example for Heart dataset.

Title: Income prediction

This dataset predicts whether each person has an annual income over \$50,000 based on some information about the person.

Useful patterns for the task at hand:

1. Higher education levels, particularly master's and doctoral degrees, are associated with higher earnings.
2. Occupations in management, execution, professional specialties, and protective services (especially in local government) are linked to higher earnings.
3. Employment in government sectors (state and local) and ownership of incorporated businesses tend to correlate with higher earnings.
4. Working 40 hours per week or more is generally associated with higher earnings, particularly in professional, managerial, or government roles.
5. Marital status as married and relation as husband often correlates with higher earnings.
6. Capital gains can indicate higher earnings, but capital losses do not necessarily indicate lower earnings.
7. Native-born U.S. citizens do not show a clear trend affecting earnings compared to other countries based on the provided data.
8. Younger individuals, especially those under 25, are less likely to earn more than \$50,000, regardless of other factors.

Additional patterns for the task summarized from incorrectly classified examples with high entropy:

1. Higher education levels (master's degree or higher) are associated with earning more than \$50,000 per year.
2. Occupations in professional specialties are likely to earn more than \$50,000, while occupations in agriculture or with lower education levels typically do not.
3. Working higher hours (60 or more) in certain sectors can lead to earnings above \$50,000.
4. Local government positions' incomes vary based on occupation and education level, potentially leading to earnings above \$50,000.
5. Capital gains and losses do not directly affect annual earnings above or below \$50,000.

###

[FEW-SHOT EXAMPLES START]

The Age is 23. The Race is White. The Sex is Male. The Marital status is married. The Relation to head of the household is Husband. The Native country is United States. The Occupation is sales sector. The Work class is private sector employee. The Capital gain last year is 0. The Capital loss last year is 0. The Education is high school graduate. The Work hours per week is 40.

Does this person earn more than 50000 dollars per year?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The Age is 32. The Race is White. The Sex is Female. The Marital status is separated. The Relation to head of the household is Unmarried. The Native country is United States. The Occupation is sales sector. The Work class is private sector employee. The Capital gain last year is 0. The Capital loss last year is 0. The Education is finished 8th class. The Work hours per week is 35.

Does this person earn more than 50000 dollars per year? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 17: A prompt example for Income dataset.

Title: Game winning prediction

This dataset predicts whether the white player will win based on some game information.

Useful patterns for the task at hand:

1. White wins if the white piece has higher strength than the black piece, unless the positional difference is too great.
2. White does not win when both pieces have equal strength.
3. Proximity of the white piece to the opponent's den (lower ranks) increases the likelihood of a win for white.
4. The file position (horizontal alignment) of the pieces does not significantly influence the outcome.
5. White does not win when the white piece has the lowest strength (0), regardless of the black piece's position or strength.
6. Strategic positioning, such as cornering the opponent's piece, can increase the likelihood of winning for the stronger piece.

Additional patterns for the task summarized from incorrectly classified examples with high entropy:

1. Higher piece strength generally favors the player with the stronger piece.
2. Positioning on the board (file and rank) can influence the outcome, particularly in endgames.
3. The ability to control key squares or escape routes is crucial for determining the winner.
4. When piece strengths are equal, positioning becomes more significant in deciding the outcome.
5. A piece with greater strength positioned to attack is likely to lead to a win.

###

[FEW-SHOT EXAMPLES START]

The white piece strength is 4. The white piece file is 1. The white piece rank is 7. The black piece strength is 0. The black piece file is 4. The black piece rank is 8.

Does the white player win this two pieces endgame of Jungle Chess?

Answer: Yes

[FEW-SHOT EXAMPLES END]

###

[CURRENT QUESTION START]

The white piece strength is 7. The white piece file is 2. The white piece rank is 0. The black piece strength is 0. The black piece file is 1. The black piece rank is 4.

Does the white player win this two pieces endgame of Jungle Chess? Answer the question with either 'Yes' or 'No' (without quotes).

Answer: <xxx, No/Yes>

Figure 18: A prompt example for Jungle dataset.