Measuring What LLMs Think They Do: SHAP Faithfulness and Deployability on Financial Tabular Classification

Saeed AlMarri[1, 2], **Mathieu Ravaut**[2], **Kristof Juhasz**[2], **Gautier Marti**[2], **Hamdan Al Ahbabi**[1, 2], **Ibrahim Elfadel**[1]

[1]Khalifa University, Abu Dhabi, United Arab Emirates
[2]Abu Dhabi Investment Authority (ADIA), Abu Dhabi, United Arab Emirates

Abstract

Large Language Models (LLMs) have attracted significant attention for classification tasks, offering a flexible alternative to trusted classical machine learning models like LightGBM through zero-shot prompting. However, their reliability for structured tabular data remains unclear, particularly in high-stakes applications like financial risk assessment. Our study systematically evaluates LLMs and generates their SHAP values on financial classification tasks. Our analysis shows a divergence between LLMs self-explanation of feature impact and their SHAP values, as well as notable differences between LLMs and LightGBM SHAP values. These findings highlight the limitations of LLMs as standalone classifiers for structured financial modeling, but also instill optimism that improved explainability mechanisms coupled with few-shot prompting will make LLMs usable in risk-sensitive domains.

1 Introduction

Chatbots powered by Large Language Models (LLMs) such as GPT-4 (Achiam et al. 2023), have demonstrated strong performance across a range of natural language processing (NLP) tasks, including classification and reasoning (Wei et al. 2022). Their ability to function as classifiers without explicit training pipelines, relying solely on few-shot or zero-shot prompting, has gained significant attention (Brown et al. 2020; Qin et al. 2023). This raises fundamental questions about the reliability and validity of LLM-based classification, particularly in comparison to classical machine learning models such as XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017).

Traditional classification tasks require structured pipelines involving feature engineering, model training, validation, and hyperparameter tuning. Fine-tuning models on tabular data, in particular, demands finesse and expertise in data preprocessing, GPU management, and balancing class distributions to prevent trivial solutions. In contrast, LLMs bypass fine-tuning entirely, requiring only natural language prompting. This reduces technical barriers, making them accessible to non-experts - a valuable boost to adoption of these tools. Nonetheless, before entrusting LLMs with critical decisions, a question remains: ***How to explain predictions from LLM classifiers?***

This question is particularly relevant in finance, a high-stakes domain where transparency and accountability are critical because algorithm outputs directly affect credit access, interest rates and regulatory compliance (Doshi-Velez and Kim 2017). Financial institutions operate under strict governance frameworks such as Basel III (Basel-III 2017) and GDPR (GDPR 2016), where opaque risk assessment models can lead to regulatory breaches, reputational damage, and unfair or discriminatory decisions, causing trust concerns. Unlike decision trees or gradient boosting models, LLMs are complex black-box models with billions of parameters, making interpretability a key challenge. This has led to increasing interest in Explainable AI (XAI) techniques to analyze LLMs' internal logic and assess their alignment with human-interpretable decision patterns.

In this study, we investigate LLMs' capacity to introspect and explain their own predictive mechanism. For LLM explainability, we employ Shapley Additive Explanations (SHAP) (Lundberg and Lee 2017), for which we provide an efficient LLM implementation. To explain the prediction mechanism, we prompt LLMs to self-explain on the impact of each feature on the classification task. From a deployability perspective, we go beyond accuracy to assess the faithfulness of explanations, their sensitivity to prompt and serialization variations, and the feasibility of post-hoc auditing under regulatory expectations for high-risk financial AI systems.

Across four open-source LLMs and three binary classification tasks with financial tabular data, our experiments show that overall, zero-shot LLMs are poorly aware of their predictive mechanism, as their self-explanations do not align with their SHAP values. LLMs SHAP values also highly differ from those of LightGBM. Additional work is needed to improve usability of these LLMs in the financial domain, such as model augmentation (Theuma and Shareghi 2024), or more elaborate inference pipelines involving few-shot prompting.

Template 1: Instance-Level Prompt

Predict `<Task Description>`. Use the features provided below to assess the likelihood of `<Positive Class Name>`.

`<Task Name>` Details:

```
<feature_1 name>: <feature_1 value>
...
<feature_N name>: <feature_N value>
```

Provide your estimated probability of the `<Positive Class Name>`. Do NOT perform coding or calculations, just provide the probability.
Your answer should only contain the probability estimate in JSON:

```
{
  "Estimated <Positive Class Name>":
  <float value between 0 and 1>
}
```

Template 2: Feature-Level Prompt

You are working on predicting <Task Description>. It is a binary classification task where the positive class corresponds to <Positive Class Name>.
One of the features is the following:

```
<feature name>
```

What impact do you think this feature will have on the classification task? Provide your answer among 3 possible strings: positive | neutral | negative. Do NOT output reasoning or explanations, just output the feature impact string. Write it in the following format in JSON:

```
{
  "Feature impact": <string among
     positive | negative | neutral>,
}
```

Template 3: Feature-Level Prompt with Self-Explanation

You are working on predicting <Task Description>. It is a binary classification task where the positive class corresponds to <Positive Class Name>.
One of the features is the following:

```
<feature name>
```

What impact do you think this feature will have on the classification task? Provide your answer among 3 possible strings: positive | neutral | negative. Also provide a brief explanation of this feature's impact. Just output the feature impact string and your explanation. Write it in the following format in JSON:

```
{
  "Feature impact": <string among
     positive | negative | neutral>,
  "Explanation": <string value>
}
```

2 Related Work

LLMs for Tabular Data

The application of LLMs to tabular data has emerged as a novel approach in regression tasks. Unlike traditional machine learning (ML) models, which require explicit training on labeled datasets, LLMs can be prompted with feature sets in a zero-shot manner, eliminating task-specific training. This method involves serializing tabular data into a natural language format and leveraging the LLM's pre-trained knowledge to make predictions.

Hegselmann et al. (2023) introduce TabLLM, a framework that utilizes LLMs for few-shot classification of tabular data by converting rows into natural language representations and providing a brief description of the classification problem. Their findings suggest that LLMs can outperform traditional deep learning models in certain tabular classification tasks (Hegselmann et al. 2023). Similarly, Shi et al. (2024) propose Zero-shot Encoding for Tabular data with LLMs (ZET-LLM), an approach that treats auto-regressive LLMs as feature embedding models for tabular prediction tasks. By implementing a feature-wise serialization and addressing challenges like limited token lengths and missing data, they demonstrated that LLMs could serve as effective zero-shot feature extractors without fine-tuning (Shi et al. 2024).

In this study, we leverage LLMs as zero-shot classifiers on three financial datasets, directly injecting feature names and feature values in the prompt.

LLMs Explainability

Feature attribution methods such as *SHAP* (Lundberg and Lee 2017) are widely used to assess feature importance in classical machine learning models like XGBoost (Chen and Guestrin 2016), LightGBM (Ke et al. 2017) or CatBoost. However, their role in LLM-based classification remains underexplored, largely due to the high computational cost: SHAP requires a high number of inference passes.

TokenSHAP (Goldshmidt and Horovicz 2024) combines cooperative game theory framework with efficient token attribution. Through Monte Carlo sampling, it estimates each token's SHAP contribution to the prediction. Mohammadi (2024) reduced the input space by using a fixed prompt template dissected into segments.

Our study is the first to compute SHAP-based feature importance on LLMs prompted to predict a probabilistic outcome on structured financial classification datasets.

LLMs Self-Explanations

Another line of research focuses on LLM-generated *rationales* or *self-explanations* of their predictions.

Huang et al. (2023) found that LLM rationales can be plausible but do not reflect internal reasoning. Deghanighobadi, Fischer, and Zafar (2025) analyzed counterfactual explanations, showing that LLMs struggle with causal dependencies. Sarkar (2024) argues that LLMs lack self-explanatory capabilities due to opaque training dynam-

	Bankruptcy		Loan Repayment		License Expiration		Average	
Model	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Gemma-2-9B	**0.641**	0.059	**0.669**	**0.878**	**0.601**	**0.029**	**0.637**	**0.322**
Llama-3.2-3B	0.524	0.041	0.616	0.851	0.439	0.019	0.526	0.304
Qwen-2.5-7B	0.630	0.054	0.591	0.839	0.433	0.018	0.551	0.304
Mistral-7B-v0.3	0.624	**0.060**	0.651	0.873	0.573	0.026	0.614	0.320

Table 1: LLMs classification performance summary. Bold numbers highlight best performance on each dataset.

LLM	Using rationale?	Bankruptcy	Loan Repayment	License Expiration	Average
Gemma-2-9B	✗	50.0% (10/20)	**66.7**% (8/12)	33.3% (5/15)	50.0%
	✓	**65.0**% (13/20)	**66.7**% (8/12)	40.0% (6/15)	**57.2**%
Llama-3.2-3B	✗	35.0% (7/20)	41.7% (5/12)	**60.0**% (9/15)	45.6%
	✓	30.0% (6/20)	58.3% (7/12)	53.3% (8/15)	47.2%
Qwen-2.5-7B	✗	15.0% (3/20)	33.3% (4/12)	26.7% (4/15)	25.0%
	✓	30.0% (6/20)	25.0% (3/12)	33.3% (5/15)	29.4%
Mistral-7B-v0.3	✗	35.0% (7/20)	25.0% (3/12)	26.7% (4/15)	28.9%
	✓	20.0% (4/20)	58.3% (7/12)	33.3% (5/15)	37.2%

Table 2: Percent features in agreement between LLMs self-explanation and LLMs SHAP values. Due to the three-class setup, the baseline accuracy is 33%. Bold numbers highlight best performance on each dataset. The rationale column shows whether a prompt asking for self-explanation was presented. In parenthesis are shown the fraction of features which each LLM correctly predicted.

ics, while Turpin et al. (2023) showed that CoT-generated explanations (Wei et al. 2022) can be misleading.

A key question arising from this endeavour, and which we explore through this paper, is whether LLMs' self-explanations *align* with actual feature contribution.

3 Methodology

Datasets

We experiment with three classification tasks (each framed as a binary classification) covering vastly different aspects of financial machine learning.

Bankruptcy We use the Polish Companies Bankruptcy dataset, explored in Zięba, Tomczak, and Tomczak (2016), keeping the subset with 1-year future bankruptcy prediction, totaling 7,027 companies including 271 going bankrupt (3.9% positive ratio). To keep the features size manageable, we only keep the top 20 features out of the dataset's initial 64, identified by computing the feature importance of a LightGBM model. All 20 features are numeric, see Table 4 in Section A.

Loan Repayment We use a very popular Kaggle dataset[1], with 79,206 loan applications, including 63,629 that are fully paid (80.3% positive ratio). Of the 21 features, 12 are numeric, see Table 5 in Section A.

License Expiration We leverage the recently introduced Hong Kong Securities and Futures Commission (SFC) dataset (AlKetbi et al. 2024). Due to the size, we only keep the last month (January 2024), where 23,001 employees are recorded and 478 see their SFC license not renewed within the next month (2.1% positive ratio). All 15 features used are numeric, see Table 6 in Section A.

Models & Inference

We use four recent open-source LLMs: `gemma-2-9b-instruct` (**Gemma-2-9B**, Team et al. 2024), `llama-3.2-3b-instruct` (**Llama-3.2-3B**, Dubey et al. 2024), `qwen-2.5-7b-instruct` (**Qwen-2.5-7B**, Yang et al. 2024) and `mistral-7b-instruct-v0.3` (**Mistral-7B-v0.3**, Jiang et al. 2023). Weights were downloaded from HuggingFace (Wolf et al. 2020), and inference was done locally through vLLM[2] on two NVIDIA A10G 24GB GPUs. The same instance-level prompt was used for each type of LLM, shown in Template 1. The probability of the positive class was generated in JSON format.

Explainability

LLMs SHAP Values We use SHAP for post hoc explanations (Lundberg and Lee 2017), specifically the model-agnostic `PermutationExplainer`. We adopt this efficient SHAP estimator because our prediction function is an LLM inference, which is costly. To balance accu-

[1] https://www.kaggle.com/datasets/sndpred/loan-data

[2] https://github.com/vllm-project/vllm

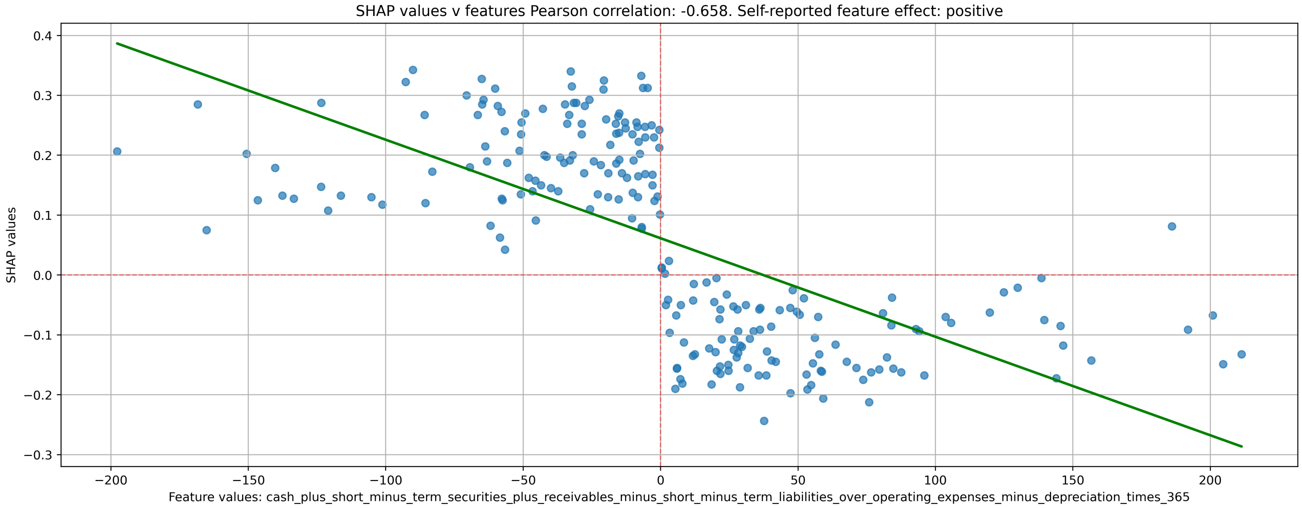

Figure 1: SHAP dependence plot for Qwen-2.5-7B highest importance feature on the Bankruptcy dataset.

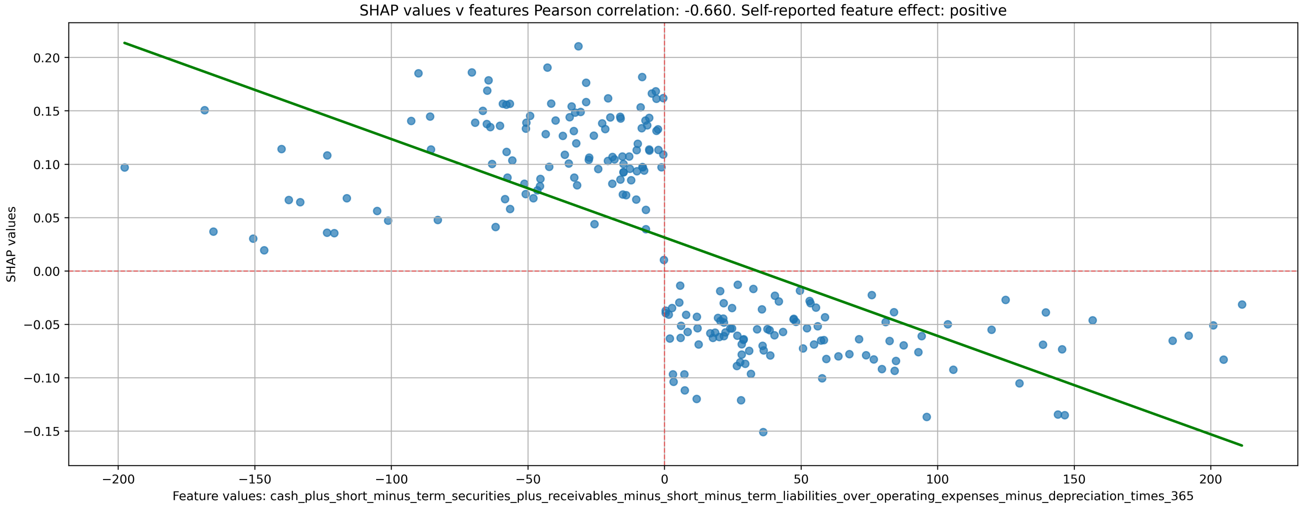

Figure 2: SHAP dependence plot for Mistral-7B-v0.3 highest importance feature on the Bankruptcy dataset.

racy and runtime, we sample 250 instances from each dataset for explanation. We construct the background (masker) via k-means clustering with $C = 5$ centroids using `shap.kmeans`, and set the `max_evals` budget, so the explainer executes exactly $T = 4$ permutations in our experiments.

Approximate cost (model calls). Let K be the number of instances explained, M the number of features (e.g. 20 on Bankruptcy), B the number of background draws per masked evaluation (here $B = C = 5$), and T the number of random permutations. The **PermutationExplainer** requires approximately:

$$\#\text{calls} \;\approx\; K \times T \times (M+1) \times B \;=\; \mathcal{O}(K\,T\,M\,B) \quad (1)$$

model evaluations. In SHAP's implementation, T is governed by `max_evals` via the practical rule:

$$T \;\approx\; \left\lfloor \frac{\texttt{max\_evals}}{2M} \right\rfloor \quad (2)$$

i.e., roughly $2M$ masked evaluations per permutation path. With `max_evals` $= 200$ and $M = 21$, this yields $T = \lfloor 200/(2 \times 21) \rfloor = 4$. Using $B = 5$, the per instance cost is therefore $\approx 4 \times (21{+}1) \times 5 = 440$ in the model calls (for the loan repayment dataset with M = 21; other datasets scale accordingly).

Why it is more efficient than `KernelExplainer`? With a summarized background of C centroids, the dominant model-call complexity of `KernelExplainer` scales as:

$$\#\text{calls} \;\approx\; K \times C \times M^2 \;=\; \mathcal{O}(K\,C\,M^2) \quad (3)$$

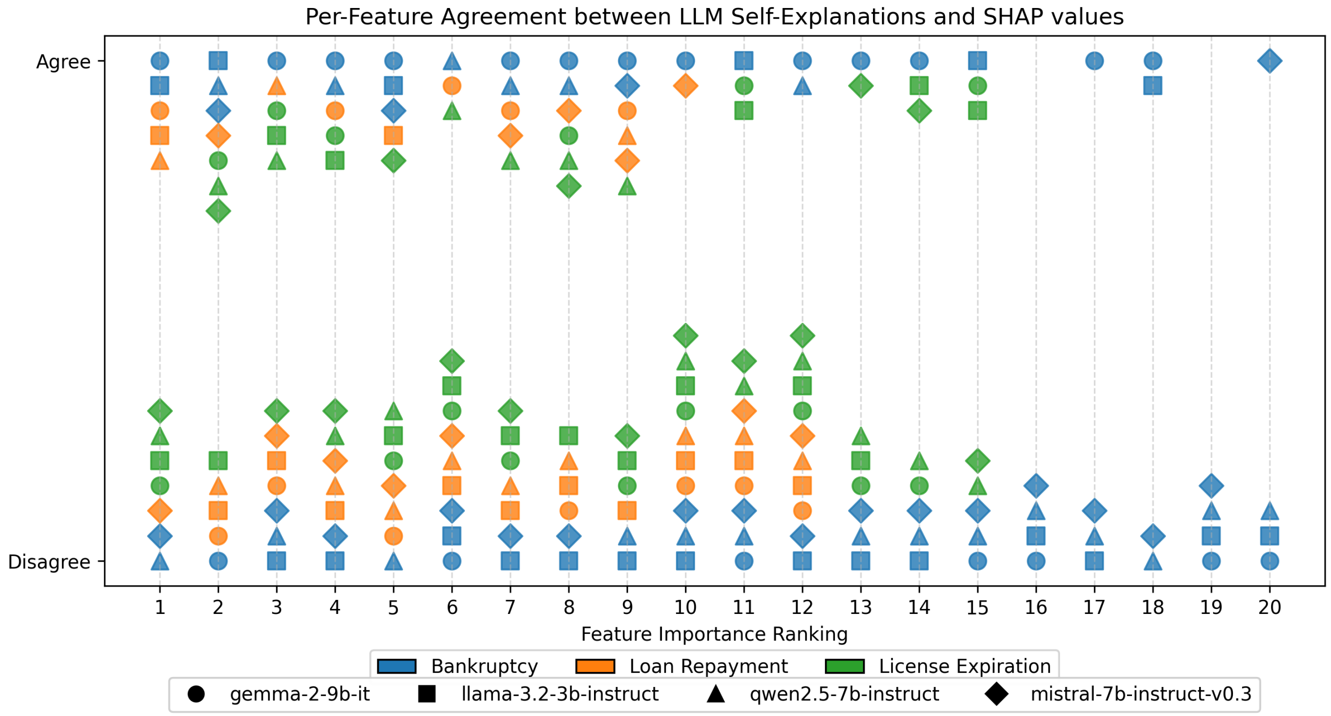

Figure 3: Per-feature agreement between LLMs self-explanations with rationale and LLMs SHAP values. Clear disagreements persist even among the top-k important features

due to sampling coalitions and fitting a kernel-weighted regression. In our setting ($C = 5$, $M = 21$) this is $\approx 5 \times 21^2 = 2205$ evaluations per instance. By contrast, `PermutationExplainer` scales linearly in M and avoids the regression solve, yielding an expected per-instance reduction of

$$\text{speedup} \approx \frac{C\,M^2}{T\,(M+1)\,B} \approx \frac{2205}{440} \approx 5\times \tag{4}$$

The fivefold decrease in model calls translates into substantially lower LLM inference time while maintaining faithful attributions, which is why we use `PermutationExplainer` with $T = 4$.

LLMs Self-Explanations Motivated by the new reasoning capabilities of LLMs (Wei et al. 2022; Huang and Chang 2022; Ke et al. 2025), we leverage the LLM as an explainability tool on its own (Huang et al. 2023). Specifically, for each feature, we prompt its description to the LLM and ask it to predict whether the feature will have a negative, neutral or positive impact on the classification ; with an option to provide a self-explanation (rationale) about its prediction. Template 2 and Template 3 show the two corresponding feature-level prompt templates.

4 Experiments

Overall, the classification performance results are consistently above random chance. Because the models are used zero-shot, performance is modest (Table 1); nevertheless, results indicate *detectable* signal on finance datasets. To us, this means that with some modification, LLMs have potential for being used in the financial domain. In the following analysis, we only consider numerical features when computing SHAP values.

LLMs SHAP Values and LLMs Self-Explanations Comparison

Figure 1 and Figure 2 demonstrate the SHAP dependence plots on the examples of the Qwen-2.5-7B and Mistral-7B-v0.3 models, showcasing the disparity between what LLMs think they do (positive or negative) vs. what they actually do (SHAP). In both cases, the LLM incorrectly predicts a positive feature impact whereas the SHAP values are strongly negatively correlated with the feature values.

To quantify this disparity, we compare SHAP values with the feature values through Pearson correlation coefficient. We classify the correlation into three feature impacts: negative (Pearson below -0.1), neutral (between -0.1 and 0.1) and positive (greater than 0.1). This feature impact is compared against the LLM self-explanation as a way to assess the LLM's own understanding of its classification process.

Results are shown in Table 2. The prompt asking for a rationale provides a moderate, yet consistent improvement in classification agreement with SHAP feature impact over the baseline prompt. Gemma-2-9B outperforms other LLMs both in terms of performance and self-explanation accuracy (Tables 1 and 2). However, even Gemma-2-9B scores only a

LLM	Bankruptcy		Loan		License	
	τ	Dir%	τ	Dir%	τ	Dir%
Gemma-2-9B	0.011	65.0%	0.000	50.0%	-0.352	66.7%
Llama-3.2-3B	0.084	50.0%	0.276	58.3%	-0.276	53.3%
Qwen-2.5-7B	-0.042	50.0%	0.190	75.0%	0.029	60.0%
Mistral-7B-v0.3	0.116	55.0%	0.124	58.3%	-0.143	53.3%
Average	0.042	55.0%	0.148	60.4%	-0.186	58.3%

Table 3: Alignment between LLMs and LightGBM SHAP values. τ is the Kendall rank correlation on full feature order, and **Dir%** = % of features with identical SHAP sign.

bit above 50% self-explanation accuracy on average across all datasets.

Figure 3 extends the analysis by showing the LLM self-explanation agreement for each individual, and sorting features by decreasing feature importance. At each feature importance rank, we split the agreement on each (dataset, model) pair in two buckets: Agree on top and Disagree on bottom. As seen, even for the top three most important features, which should be trivial to classify, there are many cases where LLMs cannot predict the correct feature impact on classification.

Thus, we conclude that **zero-shot LLMs are not able to identify a feature's impact on classification**. To take advantage of the potential demonstrated in Table 1, few-shot performance will need to be assessed.

LLMs and LightGBM SHAP Values Comparison

To further investigate LLMs SHAP Values, we compare them against the ones of LightGBM (Ke et al. 2017), a well-established, state-of-the-art gradient boosting decision tree model. Results displayed in Table 3 show that LLMs and LightGBM have low correlation in terms of SHAP values. Their agreement on the direction of a feature's impact is just a bit above random chance on average (50-60%). We conclude that LLMs' classification reasoning greatly differs from the classification process of LightGBM.

Why do LLMs mis-sign top features?

Figure 1 and Figure 2 illustrate cases where LLM self-reported impacts ("positive") contradict SHAP dependence trends (strongly *negative*). Table 2 shows that asking for rationales increases agreement only modestly (e.g., Gemma-2-9B improves from 50.0% to 57.2% on average), and Table 3 reports low Kendall's τ alignment with LightGBM. Together, these suggest that LLM self-explanations are shaped by *lexical priors* in feature names rather than the dataset-specific conditional relationships captured by SHAP.

Potential mitigation techniques (not requiring re-training):

- **Feature-name neutralization:** anonymize feature names (e.g., $f_1, \ldots, f_M$) in prompts to reduce bias from tokens such as *cash*, *profit*, or *liabilities*, then map back for human consumption.
- **Serialization robustness:** vary feature order, delimiters, and descriptions to assess prediction/explanation stability; LLMs on tables are known to be sensitive to serialization choices (Hegselmann et al. 2023).
- **Agreement reporting:** in addition to percent agreement, report beyond-chance measures (e.g., Cohen's κ or Matthews correlation) between SHAP signs and LLM labels.
- **Sanity checks for explanations:** apply label or feature randomization tests to ensure explanations collapse under appropriate perturbations (Adebayo et al. 2018).

We view the persistent disagreements at top importance ranks (Figure 3) as a deployability red flag whenever feature-level justifications are required by policy or governance.

Interpreting Performance Under Class Imbalance

Why PR-AUC matters here. All three tasks are substantially imbalanced: bankruptcy (3.9% positives), license expiration (2.1%), and loan repayment (80.3% positive class defined as fully paid). In such settings, PR-AUC is more informative than ROC-AUC because it directly reflects precision at given recalls and is sensitive to the positive rate, unlike ROC-AUC which can appear optimistic when negatives dominate (Saito and Rehmsmeier 2015; Davis and Goadrich 2006).

Lift over baseline. To contextualize Table 1, we report the *PR-AUC lift* defined as Lift $=$ PR-AUC$/$baseline. On Bankruptcy, PR-AUCs of respectively 0.059, 0.041, 0.054, 0.060 correspond to lifts of $\mathbf{1.51}\times$, $1.05\times$, $1.39\times$, and $\mathbf{1.54}\times$, respectively (average $1.37\times$) over the 3.9% baseline. On Loan Repayment, PR-AUCs of respectively 0.878, 0.851, 0.839, 0.873 imply lifts of $1.09\times$, $1.06\times$, $1.05\times$, and $1.09\times$ (average $1.07\times$) against the 80.3% baseline. On License Expiration, PR-AUCs of respectively 0.029, 0.019, 0.018, 0.026 yield lifts of $1.38\times$, $0.91\times$, $0.86\times$, and $1.24\times$ (average $1.10\times$) over the 2.1% baseline. These numbers (from Table 1) show weak but non-trivial signal on Bankruptcy and mixed results on the very sparse License task; Loan Repayment gains are small because the baseline (0.803) is already high.

Deployability implication. Even without any fine-tuning or feature engineering, zero-shot LLMs recover modest signal on some tabular finance tasks. However, the small deltas over baseline and the sub-baseline result in one License setting indicate that *few-shot prompting*, *ensembling*, or *hybridization* with tabular models are likely prerequisites for deployment in risk-sensitive contexts (Hegselmann et al. 2023; Shi et al. 2024).

5 Discussion

Limitations and Threats to Validity

Several limitations of this study warrant discussion. First, the interpretation of performance metrics under data imbalance presents challenges. We emphasized the use of PR-AUC and baseline-normalized lift, as relying solely on ROC-AUC can overstate apparent gains on highly skewed datasets (Saito and Rehmsmeier 2015). Second, our SHAP-based attribution analysis depends on the choice of background samples and estimator. For computational efficiency, we employed the *PermutationExplainer* with k-means maskers, but attributions may vary with both the background distribution and the `max_evals` parameter. A small ablation varying these choices would further strengthen interpretability claims.

Third, LLM outputs exhibit sensitivity to prompt serialization, that is, variations in feature order or phrasing can affect the model's reasoning trace. Future work should systematically quantify this prompt sensitivity through controlled perturbation benchmarks such as SUC or serialization robustness tests. Fourth, the computational cost of explainability presents a practical constraint. We explained $K = 250$ rows per dataset using SHAP, with $T = 4$ and $B = 5$, leading to approximately 110k model calls per dataset-model pair. Across three datasets and four LLMs, this results in roughly 1.32 million evaluations, which poses scalability challenges for deployment-grade auditing.

Finally, our analysis highlights that LLM self-explanations should not be equated with true causal mechanisms. The low agreement observed between SHAP attributions and LLM-generated rationales (see Tables 2–3) indicates that while LLM rationales may appear plausible, they often lack faithfulness to underlying model behavior. Consequently, explanation outputs must be independently audited, for instance through sanity checks or falsification tests, before being considered reliable in regulated decision-making contexts.

Calibration and Decision-Theoretic Concerns.

Since LLMs output *probabilities*, deployment must prioritize calibration. Classic calibration methods, such as Platt scaling or isotonic regression, trained on validation sets can enhance downstream decision quality (Niculescu-Mizil and Caruana 2005; Zadrozny and Elkan 2002). Future work should consider (i) reporting reliability diagrams and Brier scores; (ii) providing 95% CIs for AUC and PR-AUC; (iii) translating metrics into cost-sensitive operating points. These calibration-aware strategies are critical, especially in asymmetric cost domains such as financial services

Deployment Implications

When (Not) to Use Zero-Shot LLMs for Tabular Finance. Our findings align with standard deployability dimensions: *validity, explainability, calibration, robustness, governance*, and *cost*. While zero-shot LLMs show modest predictive validity (see Section 4), their self-explanatory faithfulness is lacking (Tables 2-3). This supports the case against direct deployment in regulated finance without safeguards. The results imply that any LLM-based tabular classifier must undergo rigorous validation (e.g. explanation checks, serialization tests), calibration, threshold governance, and human-in-the-loop review before deployment.

Go/No-Go Deployability Guidance

Green-lights (pilot only): Use in small-data scenarios where gradient-boosted trees underperform; low-stakes triage; auxiliary signals in a hybrid model (e.g., LLM + LightGBM) with independent tabular baselines and auditing.

Red-flags (no deployment without mitigation): Applications with regulatory or legal consequences (e.g., credit, AML, HR); those requiring faithful feature-level explanations; high prompt formatting sensitivity; failure to pass calibration or explanation sanity tests; absence of monitoring/-fallback plans.

6 Conclusion

The growing adoption of LLMs for structured classification, especially by non-expert users, raises foundational concerns about their reliability and interpretability. This study systematically evaluates zero-shot LLMs on a suite of financial tabular classification tasks. While results show potential, performance remains modest and explanation fidelity is limited.

Currently, zero-shot LLMs are best viewed as fallback options in small-data settings where fine-tuning is infeasible. Their outputs should not be trusted without rigorous auditing.

Future work should benchmark few-shot and many-shot LLMs using structured and unstructured data. Research into domain-specific fine-tuning and hybrid model integration will be key to making LLMs viable for deployment in high-stakes financial applications.

References

Achiam, J.; et al. 2023. GPT-4 Technical Report. *arXiv:2303.08774*.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems (NeurIPS)*.

AlKetbi, A.; Marti, G.; AlNuaimi, K.; Jaradat, R.; and Henschel, A. 2024. Mapping Hong Kong's Financial Ecosystem: A Network Analysis of the SFC's Licensed Professionals and Institutions. In *International Conference on Complex Networks and Their Applications*, 251–263. Springer.

Basel-III. 2017. Basel Committee on Banking Supervision and Bank for International Settlements. https://www.bis.org/bcbs/publ/d424.htm.

Brown, T. B.; et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 1877–1901.

Chen, T.; and Guestrin, C. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Davis, J.; and Goadrich, M. 2006. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 233–240.

Dehghanighobadi, Z.; Fischer, A.; and Zafar, M. B. 2025. Can LLMs Explain Themselves Counterfactually? *arXiv preprint arXiv:2502.18156*.

Doshi-Velez, F.; and Kim, B. 2017. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

GDPR. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). 88 pages. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

Goldshmidt, R.; and Horovicz, M. 2024. TokenSHAP: Interpreting Large Language Models with Monte Carlo Shapley Value Estimation. *arXiv preprint arXiv:2407.10114*.

Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; and Sontag, D. 2023. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Huang, J.; and Chang, K. C.-C. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Huang, S.; Mamidanna, S.; Jangam, S.; Zhou, Y.; and Gilpin, L. H. 2023. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Renard Lavaud, L.; Lachaux, M.; Stock, P.; Le Scao, T.; Lavril, T.; Wang, T.; Lacroix, T.; and El Sayed, W. 2023. Mistral-7B. *arXiv preprint arXiv:2310.06825*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Ke, Z.; Jiao, F.; Ming, Y.; Nguyen, X.-P.; Xu, A.; Long, D. X.; Li, M.; Qin, C.; Wang, P.; Savarese, S.; et al. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 4765–4774.

Mohammadi, B. 2024. Explaining large language models decisions using shapley values. *arXiv preprint arXiv:2404.01332*.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 625–632.

Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; and Yang, D. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Saito, T.; and Rehmsmeier, M. 2015. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3): e0118432.

Sarkar, A. 2024. Large Language Models Cannot Explain Themselves. *arXiv preprint arXiv:2405.04382*.

Shi, Z.; Kim, J.; Jeong, D.; and Pfister, H. 2024. Surprisingly Simple: Large Language Models are Zero-Shot Feature Extractors for Tabular and Text Data. *arXiv preprint arXiv:2409.00079*.

Team, G.; et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Theuma, A.; and Shareghi, E. 2024. Equipping language models with tool use capability for tabular data analysis in finance. *arXiv preprint arXiv:2401.15328*.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36: 74952–74965.

Wei, J.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wolf, T.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Yang, A.; et al. 2024. Qwen2.5 technical report. *arXiv:2412.15115*.

Zadrozny, B.; and Elkan, C. 2002. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 694–699.

Zięba, M.; Tomczak, S. K.; and Tomczak, J. M. 2016. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert systems with applications*, 58: 93–101.

Appendix

A Features Descriptions

Features for all datasets are presented on Tables 4 to 6.

Feature description	**Range**
((cash + short-term securities + AR - short-term liabilities) / (OPEX - depreciation)) x 365	[-3039.4, 2104.43]
retained earnings / total assets	[-0.72, 0.8]
sales / total assets	[0.46, 7.88]
(gross profit + extraordinary items + financial expenses) / total assets	[-0.24, 0.84]
(gross profit + depreciation) / sales	[-0.18, 0.71]
sales (n) / sales (n-1)	[0.53, 2.98]
profit on operating activities / total assets	[-0.21, 0.76]
gross profit (in 3 years) / total assets	[-0.42, 1.26]
(equity - share capital) / total assets	[-0.58, 0.95]
(net profit + depreciation) / total liabilities	[-0.32, 8.45]
profit on operating activities / financial expenses	[-11.97, 4116.67]
logarithm of total assets	[2.9, 6.01]
(total liabilities - cash) / sales	[-0.41, 2.22]
operating expenses / total liabilities	[-0.27, 27.02]
(current assets - inventory) / long-term liabilities	[0.15, 609.35]
constant capital / total assets	[-0.19, 0.97]
(current assets - inventory - receivables) / short-term liabilities	[0.0, 8.45]
net profit / inventory	[-2.68, 46.08]
(current assets - inventory) / short-term liabilities	[0.13, 13.6]
total costs / total sales	[0.16, 1.23]

Table 4: Bankruptcy dataset features. AR - account receivables, OPEX - operating expenses. The numerical value intervals are bounded by the 1st and the 99th percentiles for each variable.

Feature description	**Range**
Loan Amount	[1600.0, 35000.0]
Term	categorical: {’36 months’, ’60 months’}
Interest Rate	[6.03, 25.29]
Installment	[55.32, 1204.57]
Grade	categorical: {’A’, ’B’, ’C’, ’D’, ’E’, ’F’, ’G’}
Sub-grade	categorical: {’A1’, ’A2’, ... ’G4’, ’G5’}
Employment Length	categorical: {’<1 year’, ’1 year’, ’2 years’, ..., ’10+ years’}
Home Ownership	categorical: {’MORTGAGE’, ’NONE’, ’OTHER’, ’OWN’, ’RENT’}
Annual Income	[19000.0, 250000.0]
Verification Status	categorical: {’Not Verified’, ’Source Verified’, ’Verified’}
Purpose	categorical: 14 possible values (e.g ’wedding’)
Debt-to-Income (DTI) Ratio	[1.6, 36.41]
Open Credit Accounts	[3.0, 27.0]
Public Records	[0.0, 2.0]
Revolving Balance	[169.05, 83505.9]
Revolving Utilization Rate	[1.2, 98.0]
Total Accounts	[6.0, 60.0]
Initial Listing Status	categorical: {’f’, ’w’}
Application Type	categorical: {’DIRECT PAY’, ’INDIVIDUAL’, ’JOINT’}
Mortgage Accounts	[0.0, 9.0]
Public Record Bankruptcies	[0.0, 1.0]

Table 5: Loan Repayment dataset features. The numerical value intervals are bounded by the 1st and the 99th percentiles for each variable. For categorical features, the values space is shown.

Feature description	**Range**
Number of unique companies that the employee has worked at	[1.0, 7.0]
Number of unique companies that the employee has worked at per working day	[0.0, 0.01]
Tenure across all companies (days)	[142.0, 3631.0]
Tenure at the current company (days)	[23.0, 3584.0]
Longest tenure (days)	[10.0, 2897.69]
Average tenure (days)	[7.86, 2744.34]
Shortest tenure (days)	[1.0, 2739.07]
Gender	{0.0, 1.0}
Is the employee Hongkonger?	{0.0, 1.0}
Is the employee Chinese?	{0.0, 1.0}
Is the employee British?	{0.0, 1.0}
Number of employees in the company	[2.0, 1669.0]
Days of existence of the company	[0.0, 3647.0]
Cumulated tenure of all employees in the company	[0.0, 1094284.0]
Average tenure in the company	[85.0, 1432.4]

Table 6: License Expiration dataset features. The numerical value intervals are bounded by the 1st and the 99th percentiles for each variable.