LLATAS: LARGE LANGUAGE MODELS AS TABULAR AUXILIARY FEATURE SYNTHESIZER

Yuzhen Mao, Martin Ester
Simon Fraser University, BC, Canada
{yuzhenm, ester}@sfu.ca

ABSTRACT

While classical models like Gradient Boosting remain state-of-the-art for tabular data, their performance is often bottlenecked by the limitations of heuristic feature engineering. To address this, we introduce **LLATAS**, a framework that leverages Large Language Models (LLMs) to synthesize semantic reasoning traces as auxiliary features. Grounded in the *Learning Using Privileged Information* (LUPI) paradigm, we use these generated signals to train a teacher model, which then guides a lightweight student model operating solely on original inputs. This distillation process allows the student to inherit complex reasoning capabilities without incurring the computational cost of LLMs at inference. Empirical evaluations on disease prediction tasks demonstrate that LLATAS significantly outperforms baselines, reducing test error rates by **17.6%** for XGBoost and **22.0%** for MLP models.

1 INTRODUCTION

While deep learning has achieved remarkable success in domains such as vision and language, traditional machine learning models, e.g., gradient boosting, random forests, and logistic regression, remain state-of-the-art in many tabular and structured data settings Grinsztajn et al. (2022); McElfresh et al. (2023), including finance Schmitt (2022), healthcare Christodoulou et al. (2019), and industrial analytics Carvalho et al. (2019). In these domains, predictive performance depends less on model architecture and more on the quality of input features. However, feature engineering, the process of constructing informative features from raw attributes, requires substantial domain knowledge and manual experimentation, making it both labour-intensive and non-scalable.

This challenge has motivated research into automated feature augmentation, which aims to algorithmically expand the feature space to uncover richer representations Horn et al. (2019); Zhang et al. (2023); Hollmann et al. (2023); Nam et al. (2024). Yet, existing methods suffer from key limitations. Most approaches rely on manually predefined search spaces (e.g., arithmetic transformations, aggregations, and pairwise combinations), which limit the information gain of the generated features and thus provide only marginal benefits for tabular data augmentation.

This limitation motivates our proposed framework, LLATAS, an auxiliary feature generation method for tabular data that leverages the reasoning capabilities of large language models (LLMs). Specifically, (1) LLATAS uses LLMs to move beyond the restricted space of handcrafted transformations, generating richer features from the original inputs; and (2) instead of repeatedly retraining and validating models during feature search, LLATAS trains only one teacher model on the augmented data and one final student model on the original features, significantly reducing computational cost. Overall, LLATAS formulates feature engineering as an *iterative process of reasoning, extraction, and distillation*: **Stage 1 – Reasoning generation.** Given a small seed dataset randomly sampled from the entire training dataset, a strong LLM $\mathcal{A}$ generates

high-quality fake conversations between a fake user, which serves as the reasoning traces linking features to target labels. These traces are then validated by a smaller LLM $\mathcal{B}$ in a *closed feedback loop*. Failed validations, i.e. reasonings that lead to a wrong prediction, trigger regeneration with the negative examples, producing reasoning data that achieves 100% predictive accuracy under $\mathcal{B}$. **Stage 2 – Auxiliary feature extraction.** We prompt LLM $\mathcal{A}$ to extract structured *auxiliary features* (key–value pairs) from the generated reasoning of the seed dataset. The most frequent or semantically salient features are selected and then used for an additional round of reasoning generation (Stage 1) on the full training dataset. **Stage 3 – Knowledge distillation.** A *teacher model* (classical machine learning models such as MLP and XGBoost (Chen, 2016)) trained on the augmented features guides a *student model* (same model type as the teacher model) trained only on the original feature set, transferring knowledge via logit matching. This produces compact models that inherit reasoning-derived knowledge without relying on LLM inference at deployment.

In summary, LLATAS **bridges reasoning and representation**. Compared to other baseline methods, our approach introduces a **reasoning-based augmentation pipeline** that offers a new paradigm for automated feature engineering.

2 Related Works

Recent advancements in Auxiliary Feature Generation (AFG) have evolved from heuristic-based expansion strategies to semantic reasoning powered by LLMs. Traditional frameworks primarily rely on statistical transformations to uncover non-linear relationships within tabular data. **AutoFeat** (Horn et al., 2019) enhances interpretable linear models through a mathematical "expansion-reduction" strategy, generating non-linear candidates via transformations like logarithms and trigonometry before filtering for robustness. Similarly, **OpenFE** (Zhang et al., 2023) targets expert-level predictive performance by optimizing this paradigm; it employs the "FeatureBoost" algorithm for efficient incremental performance estimation and a rigorous two-stage pruning mechanism to navigate massive search spaces.

In contrast, recent methodologies integrate LLMs to inject domain knowledge and iterative reasoning into the feature generation process. **CAAFE** (Context-Aware Automated Feature Engineering) (Hollmann et al., 2023) shifts the focus from statistical properties to semantic context, using LLMs to hypothesize meaningful features based on natural language dataset descriptions. Building on this, **OCTree** (Nam et al., 2024) advances automation by eliminating pre-defined search spaces, utilizing a feedback loop where an LLM refines its hypotheses based on the structural logic of decision trees trained on prior iterations.

Collectively, these works illustrate a methodological progression in AFG from brute-force mathematical enumeration to sophisticated, context-aware reasoning systems that mimic the iterative workflows of human data scientists.

3 Problem Definition

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote the original labeled tabular dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \mathcal{Y}$ is the corresponding label. Our goal is to improve predictive performance on $\mathcal{D}$ by automatically generating informative auxiliary features using LLM reasoning, while ensuring that the final deployed model relies only on the original feature set.

LLATAS achieves this objective through a three-stage pipeline with the following inputs and outputs:

Stage 1: Generation. Input: The original dataset $\mathcal{D}$, the prediction task specification, and the LLM $\mathcal{A}$. Output: A reasoning-augmented dataset $\mathcal{D}^{(1)} = \{(\mathbf{x}_i, y_i, r_i)\}_{i=1}^N$, where r_i denotes the reasoning trace (e.g., scenario-based conversation) generated by $\mathcal{A}$ for instance $\mathbf{x}_i$ that is aligned with label y_i.

Stage 2: Extraction. Input: The reasoning-augmented dataset $\mathcal{D}^{(1)}$ and an LLM $\mathcal{B}$ for structured extraction. Output: An auxiliary feature set $\mathcal{F} = \{f_1, \ldots, f_k\}$ and an augmented dataset $\mathcal{D}^{(2)} = \{(\mathbf{x}_i, \mathbf{z}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{z}_i \in \mathbb{R}^k$ contains the values of the extracted auxiliary features for instance i inferred from r_i.

Stage 3: Distillation. Input: The augmented dataset $\mathcal{D}^{(2)}$. Output: A teacher model T trained on $(\mathbf{x}_i, \mathbf{z}_i)$ and a student model S trained only on $\mathbf{x}_i$, where S approximates the predictive behavior of T via logit alignment.

Objective. At inference time, LLATAS deploys only the student model S, which takes the original features $\mathbf{x}$ as input and outputs predictions $\hat{y} = S(\mathbf{x})$.

4 METHOD

In this section, we introduce LLATAS, a three-stage framework for auxiliary feature generation and distillation for tabular learning, which leverages the reasoning capabilities of LLMs to enrich feature representations while preserving efficient inference with classical models. In general, LLATAS decomposes the feature engineering process into **reasoning generation**, **auxiliary feature extraction**, and **knowledge distillation**. We describe each step in detail below.

4.1 STAGE 1: SEED DATASET GENERATION

The goal of the first stage is to generate a set of high-quality reasoning outputs using the LLM $\mathcal{A}$, given the original tabular training dataset, the prediction task, and the ground-truth labels. Since it is not straightforward for LLMs to directly construct meaningful reasoning trajectories from raw tabular features, we introduce a scenario-mimicking strategy to facilitate vivid and coherent reasoning.

Specifically, we enrich each row in the sampled seed dataset by mapping it to a realistic scenario and prompt the LLM to elaborate it as a conversation between a user and an assistant. The roles in the conversation are adapted to the task semantics. For example, in a patient diagnosis dataset, the user can act as the patient while the LLM plays the role of a doctor whose goal is to determine whether the patient has a specific disease. The original tabular features are embedded into the dialogue as patient attributes or responses. Through this conversational process, the LLM naturally expresses its reasoning, similar to how a doctor gathers information by asking questions and synthesizing evidence to reach a diagnosis. This procedure enables the LLM to generate new, informative signals grounded in its domain knowledge. The resulting auxiliary features are highly relevant to label prediction and are not constrained by predefined transformation spaces (e.g., arithmetic formulas over original features), thereby offering richer and more flexible feature augmentation.

To ensure the quality of the generated reasoning, we implement a feedback loop using a smaller model, LLM $\mathcal{B}$. Given the original features and the reasoning produced by $\mathcal{A}$, $\mathcal{B}$ attempts to predict the target label. If $\mathcal{B}$ fails to produce the correct prediction, we prompt $\mathcal{A}$ to regenerate the reasoning, providing the failed reasoning as a negative example for guidance. This iterative refinement continues until $\mathcal{B}$ can correctly predict the label. Through this process, we construct a reasoning-augmented dataset that achieves 100% prediction accuracy under $\mathcal{B}$, ensuring that the generated reasoning is both informative and aligned with the target task.

4.2 STAGE 2: AUXILIARY FEATURE EXTRACTION

Given the reasoning artifacts generated in Stage 1 (e.g., doctor–patient dialogues in the diagnosis example), we further prompt LLM $\mathcal{A}$ to extract structured auxiliary features from each reasoning transcript in the form of name–value pairs. These features summarize salient attributes, conditions, or signals implied by

the reasoning process. We then aggregate all extracted features across the dataset and rank them by their frequency of occurrence. The top-k most frequent features are selected as the auxiliary feature set, as they capture the most common and informative patterns revealed by the LLM's reasoning.

We then generate reasoning transcripts for the whole tabular training set by augmenting the prompt with an additional instruction: "The conversation must include the following information: {aux-feat-1}, {aux-feat-2}, ...". This constrained step encourages the LLM to incorporate the chosen auxiliary features into the reasoning, resulting in a uniform, feature-aligned reasoning-augmented dataset for subsequent modeling.

4.3 Stage 3: Knowledge distillation

We follow the Learning Using Privileged Information (LUPI) framework Vapnik et al. (2015) by training a teacher on a reasoning-augmented dataset that includes auxiliary features available only during training, and distilling its knowledge into a student that operates solely on the original tabular features. Each training sample is represented as (x_i, x_i^*, y_i), where x_i denotes the original features and x_i^* denotes privileged features derived from LLM-generated reasoning. At inference time, only x is available, and predictions are made exclusively by the student, ensuring that no auxiliary features are required at deployment. This process is not merely data augmentation but a form of *knowledge transfer* where the student inherits the "explanations" captured by the teacher, effectively correcting its own concepts of similarity and decision margins without requiring the heavy reasoning features at inference time Vapnik et al. (2015).

Formally, this distillation-based approach aligns with the structural risk minimization principle for privileged information. While modern implementations often utilize logit matching, the canonical mathematical formulation of this paradigm models the student's slack variables (errors) as a function of the privileged information. The teacher estimates a correcting function in $\mathcal{X}^*$ that identifies "hard" examples, allowing the student to relax constraints where the privileged data suggests ambiguity. The objective is to minimize a combined functional of the decision rule in space $\mathcal{X}$ and the correcting function in space $\mathcal{X}^*$. The loss function for this knowledge transfer is given by minimizing the functional $\mathcal{T}$ subject to the constraints that the student's margin is lower-bounded by the teacher's privileged correction (Vapnik et al., 2015):

$$\min_{w,w^*,b,b^*} \quad \frac{1}{2}\left(\|w\|^2 + \gamma\|w^*\|^2\right) + C\sum_{i=1}^{l}[y_i((w^*, z_i^*) + b^*) + \zeta_i] + \Delta C\sum_{i=1}^{l}\zeta_i \tag{1}$$

where w and w^* parameterize the student and teacher (correcting) functions respectively, z_i^* represents the mapped privileged features, and ζ_i are additional slack variables ensuring the correcting function is valid. This mechanism enables lightweight models, such as Random Forests or SGBoost, to approximate the performance of reasoning-heavy systems with minimal computational overhead.

5 Experiments

We use a disease prediction dataset[1] for the evaluation. This dataset is designed for a classification task to predict disease diagnoses using patient attributes that include demographic details (age, gender), health metrics (blood pressure, cholesterol levels), and binary symptom indicators (fever, fatigue).

Generation and Selection. We pick Claude3.7 (Claude37) as the transcript generator (LLM $\mathcal{A}$) and Qwen2.5-32B-Instruct (Qwen32) as the response generator (LLM $\mathcal{B}$). The reason is that Claude37 is one of the most powerful LLMs available, while Qwen32 is a widely used open-source model with a relatively

[1] https://www.kaggle.com/datasets/uom190346a/disease-symptoms-and-patient-profile-dataset/data

small size. We therefore employ Qwen3.2 as the judge in the initial loop to ensure reliable prediction accuracy, and subsequently evaluate the results using additional open-source LLMs to verify that the conclusions are not biased toward any single judge model.

We generated the reasoning transcripts for the whole patient dataset (384 samples in total).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Qwen2.5-32b-instruct	100.00	100.00	100.00	100.00
Llama-3.1-70b-instruct	95.42	93.01	98.30	95.58
Gemini-2.0-flash-001	96.85	95.16	98.88	96.99
Claude-3.7-sonnet	96.85	98.39	95.81	97.08
Deepseek-v3	97.42	96.77	98.36	97.56

Table 1: Classification Accuracies of Different LLMs

From the table, since Qwen32 is used as the judge in the Stage 1 loop, all its metrics are 100%. For the other LLMs, their accuracy is consistently above 95%, providing further validation of the quality of the generated reasoning.

Prediction model	Baseline	AutoFeat	OpenFE	OCTree	**LLATAS (Ours)**
XGBoost	$28.09_{\pm 7.9}$	$27.91_{\pm 3.7}$ (0.6%)	$27.03_{\pm 4.9}$ (3.8%)	$25.72_{\pm 6.6}$ (8.4%)	$\mathbf{23.15}_{\pm 7.2}$ **(17.6%)**
MLP	$38.10_{\pm 3.6}$	$36.62_{\pm 3.5}$ (3.9%)	$33.71_{\pm 3.7}$ (11.5%)	$30.95_{\pm 5.8}$ (18.8%)	$\mathbf{29.72}_{\pm 5.1}$ **(22.0%)**

Table 2: Test error rates (%) for classification. The best results are highlighted in **bold**; values in parentheses represent the relative error reduction compared to the baseline.

Knowledge Distillation. We then assess the effectiveness of the generated auxiliary features via knowledge distillation. As presented in Table 2, our method (LLATAS) achieves substantial improvements over the baseline and all competing automated feature engineering frameworks. Specifically, LLATAS reduces the test error rate of the XGBoost model by **17.6%** (from 28.09% to 23.15%) and the MLP model by **22.0%** (from 38.10% to 29.72%). Notably, LLATAS consistently outperforms the strongest baseline, OCTree, widening the performance gap by nearly 9% on XGBoost (17.6% vs. 8.4%) and over 3% on MLP (22.0% vs. 18.8%). These findings validate the superior efficacy of our approach, demonstrating that the synergy between generated auxiliary features and knowledge distillation significantly outperforms established baselines.

REFERENCES

Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering, 137:106024, 2019.

Tianqi Chen. Xgboost: A scalable tree boosting system. Cornell University, 2016.

Evangelia Christodoulou, Jie Ma, Gary S Collins, Ewout W Steyerberg, Jan Y Verbakel, and Ben Van Calster. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. Journal of clinical epidemiology, 110:12–22, 2019.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? Advances in neural information processing systems, 35:507–520, 2022.

Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. Advances in Neural Information Processing Systems, 36:44753–44775, 2023.

Franziska Horn, Robert Pack, and Michael Rieger. The autofeat python library for automated feature engineering and selection. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 111–120. Springer, 2019.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? Advances in Neural Information Processing Systems, 36:76336–76369, 2023.

Jaehyun Nam, Kyuyoung Kim, Seunghyuk Oh, Jihoon Tack, Jaehyung Kim, and Jinwoo Shin. Optimized feature generation for tabular data via llms with decision tree reasoning. arXiv preprint arXiv:2406.08527, 2024.

Marc Schmitt. Deep learning vs. gradient boosting: Benchmarking state-of-the-art machine learning algorithms for credit scoring. arXiv preprint arXiv:2205.10535, 2022.

Vladimir Vapnik, Rauf Izmailov, et al. Learning using privileged information: Similarity control and knowledge transfer. J. Mach. Learn. Res., 16(1):2023–2049, 2015.

Tianping Zhang, Zheyu Aqa Zhang, Zhiyuan Fan, Haoyan Luo, Fengyuan Liu, Qian Liu, Wei Cao, and Li Jian. Openfe: Automated feature generation with expert-level performance. In International Conference on Machine Learning, pp. 41880–41901. PMLR, 2023.