LightAutoDS-Tab: Multi-AutoML Agentic System for Tabular Data

Demonstration Track

Alexey Lapin
AI Talent Hub, ITMO University
AI Institute, ITMO University
Saint-Petersburg, Russian Federation
a.lapin@itmo.ru

Igor Hromov
Sber AI Lab
Moscow, Russian Federation
hromov.igor@gmail.com

Stanislav Chumakov
AI Institute, ITMO University
Saint-Petersburg, Russian Federation
svchumakov@itmo.ru

Mile Mitrovic
Sber AI Lab
Moscow, Russian Federation
milemitrovic888@gmail.com

Dmitry Simakov
Sber AI Lab
Moscow, Russian Federation
dmitryevsimakov@gmail.com

Nikolay O. Nikitin
AI Institute, ITMO University
Saint-Petersburg, Russian Federation
nnikitin@itmo.ru

Andrey V. Savchenko
Sber AI Lab
Moscow, Russian Federation
avsavchenko@hse.ru

ABSTRACT

AutoML has advanced in handling complex tasks using the integration of LLMs, yet its efficiency remains limited by dependence on specific underlying tools. In this paper, we introduce LightAutoDS-Tab, a multi-AutoML agentic system for tasks with tabular data, which combines LLM-based code generation with several AutoML tools. Our approach enhances the flexibility and robustness of pipeline design, outperforming state-of-the-art open-source solutions on several Kaggle data science tasks. The source code of LightAutoDS-Tab: https://github.com/sb-ai-lab/LADS.

Demonstration video: https://youtu.be/sDUo_Ke2xs0.

KEYWORDS

Multi-Agent Systems; AutoML; Tabular Data

ACM Reference Format:
Alexey Lapin, Igor Hromov, Stanislav Chumakov, Mile Mitrovic, Dmitry Simakov, Nikolay O. Nikitin, and Andrey V. Savchenko. 2026. LightAutoDS-Tab: Multi-AutoML Agentic System for Tabular Data: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/QBKY1209

1 INTRODUCTION

Tabular AutoML tools, such as AutoGluon [1] and H2O [6], generally rely on predefined search spaces and routines, often focusing on hyperparameter optimization and model ensembling. Leveraging natural language understanding and code generation abilities of large language models (LLMs) has recently led to the emergence of LLM-based agents capable of automating parts of the ML workflow [3]. General strategies involve iterative planning [4], a tree-based search process with agentic trial-and-error code construction [5], and fully autonomous agents for machine learning (ML) [11].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/QBKY1209

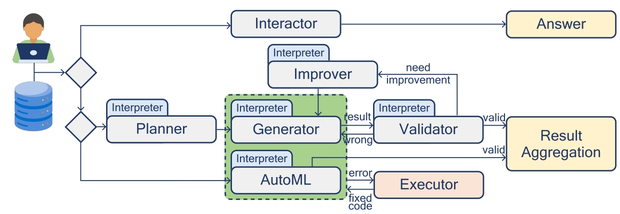

Figure 1: The proposed LightAutoDS-Tab multi-AutoML agentic system architecture.

While all of the mentioned strategies result in a robust search space across preliminary model training stages and broad flexibility of the pipeline design, they are prone to excessive branching before reaching an optimal solution. Each branch typically requires extensive context, including task descriptions, historical demonstrations, domain knowledge, intermediate reasoning, generated code, error logs, and feedback. Appending conversation history or providing detailed context in prompts can face limitations due to the limited LLM context window, which drastically increases cost while not guaranteeing model composition performance in the latter pipeline stages. For these reasons, classic AutoML frameworks often remain a more efficient solution [2]. LLM-based solutions for AutoML tasks are implemented in AutoKaggle [7], a multi-agentic approach for model selection, and AIDE (AI-Driven Exploration) [5], which performs a tree search over solutions. However, such techniques do not use the potential of various existing ML and AutoML tools.

To address these shortcomings, we introduce LightAutoDS-Tab, a novel multi-AutoML agentic system designed to bridge this gap. It combines LLMs for adaptive, data-aware code generation in early-stage tasks with the power of multiple AutoML tools for robust model configuration and training. This hybrid design improves pipeline construction, overcoming the limitations of traditional fixed-pipelines and purely LLM-driven AutoML techniques.

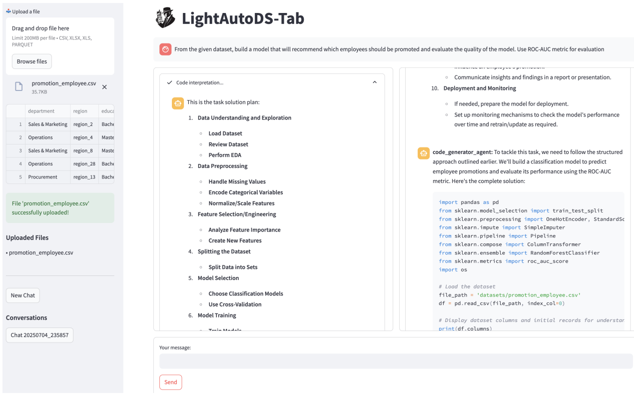

Figure 2: The LightAutoDS-Tab user interface.

2 PROPOSED SYSTEM

LightAutoDS-Tab combines LLM agents with existing ML and AutoML tools. The architecture of the system is shown in Figure 1. The system comprises specialized agents for tasks such as code generation, debugging, validation, providing technical and non-technical explanations, executing tools, and facilitating user interaction. Users provide a dataset and a query describing their goal.

If the query involves user interaction, the **Interactor** agent provides relevant explanations of the system processes to ensure user understanding. If the query requests building an ML pipeline, a router evaluates the input and chooses one of two routes:

(1) LLM-Driven Pipeline Generation: The Planner agent first formulates a plan based on the task description. The **Generator** agent, powered solely by LLMs, generates code using standard ML libraries such as Scikit-learn, CatBoost, TabPFN, and others. The generated code is subsequently evaluated by the **Validator** agent to ensure correctness and performance. If the validation criteria are not met, the **Improver** agent iteratively refines the code to enhance its performance metrics until satisfactory results are achieved.

(2) AutoML-Based Pipeline Configuration: Alternatively, the agent integrates with AutoML tools, utilizing an LLM to interpret the problem statement and data, and to generate the key configuration settings required to set up the frameworks.

We include several AutoML tools that are efficient in various tasks, e.g., LightAutoML [9], emerged as the winner of the Kaggle AutoML Grand Prix 2024[1] and FEDOT [8], which supports flexible pipelines for complex tasks. The router can select a route automatically from task characteristics or follow an explicit user instruction.

The **Executor** handles code execution and debugging. During execution, the **Interpreter** produces non-technical step summaries. These summaries are aggregated into a final report that documents pipeline decisions, generated code, and the trained model for transparency and traceability.

The LightAutoDS-Tab framework features a user-friendly interface (Figure 2), designed to streamline user interaction and the AutoML workflow. Users can upload and preview datasets in standard formats such as *csv*, *xlsx*, and *parquet*.

Table 1: The normalized performance score for LightAutoDS tools and existing solutions.

Dataset	Tools of LightAutoDS-Tab			Existing solutions	
	LAMA +LLM	Code Gen	FEDOT +LLM	Auto Kaggle	AIDE
Titanic	0.745	0.766	**0.780**	0.767	0.744
Sp. Titanic	**0.798**	0.788	0.790	0.771	0.793
House Prices	**0.886**	0.871	0.882	0.862	0.883
Monsters	**0.774**	**0.774**	0.733	0.723	0.721
Ac. Success	**0.836**	0.828	0.833	0.820	0.835
Bank Churn	0.883	**0.885**	0.881	0.856	0.786
Ob. Risk	**0.905**	0.888	0.904	0.896	0.896
Plate Defect	**0.886**	0.878	0.883	0.823	-

Once a dataset is uploaded, users define their task by entering a query. Expert users can provide detailed technical constraints, while non-experts can use high-level business descriptions.

The interface has two real-time panels. The *right panel* shows technical details of each pipeline step for expert users. The *left panel* shows a simplified non-technical summary, helping non-experts follow the process and understand results.

LightAutoDS-Tab can be easily extended to other tools, such as AutoGluon. It integrates with multiple LLM providers, and the system architecture enables the seamless addition of new LLM-based providers as needed.

3 EXPERIMENTS

In the experimental part, we follow the authors of AutoKaggle [7] in utilizing the normalized performance score and benchmarking our framework on 8 ML datasets from Kaggle Competitions, which comprise 7 classification tasks, 1 regression task, and 1 multi-target task. Each competition contains a train and test split, a sample submission, and a description downloaded from the corresponding Kaggle page. Competitions are split evenly between classic (up to 2024) and modern (after 2024), coinciding with the knowledge cutoff in GPT-4o and GPT-4o-mini, which we adopt as LLMs.

The results (Table 1) demonstrate that LightAutoDS-Tab achieves superior performance compared to AutoKaggle and AIDE. The comparative analysis of different tools within LightAutoDS-Tab confirms the validity of our multi-AutoML implementation, as no single tool consistently delivers the best results across all cases.

The influence of LLM selection on CodeGen performance was analyzed by comparing GPT-4o and GigaChat2Max [10]. The results[2] demonstrate that the CodeGen output is susceptible to the choice of the underlying LLM.

4 CONCLUSION

This paper introduces LightAutoDS-Tab, a multi-AutoML agentic system designed to significantly enhance the productivity of data scientists working with tabular data by automating the end-to-end creation of ML pipelines while maintaining full interpretability.

[1]https://www.kaggle.com/automl-grand-prix

[2]Detailed tables with experimental results are available in the GitHub repository.

ACKNOWLEDGMENTS

This work supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement No139-15-2025-010.

REFERENCES

[1] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. AutoGluon-Tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* (2020).

[2] Haoyang Fang, Boran Han, Nick Erickson, Xiyuan Zhang, Su Zhou, Anirudh Dagar, Jiani Zhang, Ali Caner Turkmen, Cuixiong Hu, Huzefa Rangwala, et al. 2025. MLZero: A multi-agent system for end-to-end machine learning automation. *arXiv preprint arXiv:2505.13941* (2025).

[3] Yang Gu, Hengyu You, Jian Cao, Muran Yu, Haoran Fan, and Shiyou Qian. 2024. Large language models for constructing and optimizing machine learning workflows: A survey. *ACM Transactions on Software Engineering and Methodology* (2024).

[4] Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. 2024. DS-Agent: automated data science by empowering large language models with case-based reasoning. In *Proceedings of the 41st International Conference on Machine Learning*. 16813–16848.

[5] Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. 2025. AIDE: AI-Driven Exploration in the Space of Code. *arXiv preprint arXiv:2502.13138* (2025).

[6] Erin LeDell and Sebastien Poirier. 2020. H2O AutoML: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML*, Vol. 2020. 24.

[7] Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tuney Zheng, Minghao Liu, Xinyao Niu, Yue Wang, Jian Yang, Jiaheng Liu, et al. 2024. AutoKaggle: A multi-agent framework for autonomous data science competitions. *arXiv preprint arXiv:2410.20424* (2024).

[8] Nikolay O Nikitin, Pavel Vychuzhanin, Mikhail Sarafanov, Iana S Polonskaia, Ilia Revin, Irina V Barabanova, Gleb Maximov, Anna V Kalyuzhnaya, and Alexander Boukhanovsky. 2022. Automated evolutionary approach for the design of composite machine learning pipelines. *Future Generation Computer Systems* 127 (2022), 109–125.

[9] Anton Vakhrushev, Alexander Ryzhkov, Maxim Savchenko, Dmitry Simakov, Rinchin Damdinov, and Alexander Tuzhilin. 2021. LightAutoML: Automl solution for a large financial services ecosystem. *arXiv preprint arXiv:2109.01528* (2021).

[10] Mamedov Valentin, Evgenii Kosarev, Gregory Leleytner, Ilya Shchuckin, Valeriy Berezovskiy, Daniil Smirnov, Dmitry Kozlov, Sergei Averkiev, Lukyanenko Ivan, Aleksandr Proshunin, et al. 2025. GigaChat Family: Efficient Russian Language Modeling Through Mixture of Experts Architecture. *arXiv preprint arXiv:2506.09440* (2025).

[11] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. 2025. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. In *The Thirteenth International Conference on Learning Representations (ICLR)*.