Integrating Large Language Models into Automated Machine Learning: A Human-Centric Approach

Néstor Miguel-Morante[1][a], Iván Rivero[1][b], Diego García-Prieto[1][c], Rafael Duque[1][d], Camilo Palazuelos[1][e] and Abraham Casas[2][f]

[1]*Department of Mathematics, Statistics and Computer Science, University of Cantabria, Santander, Spain*

[2]*Centro Tecnológico CTC, Parque Científico y Tecnológico de Cantabria, Santander, Spain*

Keywords: Automated Machine Learning, Large Language Models, Human-Centered Artificial Intelligence.

Abstract: The growing complexity and volume of data in modern applications have amplified the need for efficient and accessible machine learning (ML) solutions. Automated Machine Learning (AutoML) addresses this challenge by automating key stages of the ML pipeline, such as data preprocessing, model selection and hyperparameter tuning. However, AutoML systems often remain limited in their ability to interpret user intent or adapt flexibly to domain-specific requirements. Recent advances in Large Language Models (LLMs), such as GPT-based models, offer a novel opportunity to enhance AutoML through natural language understanding and generation capabilities. This paper proposes a software system that integrates LLMs into AutoML workflows, enabling users to interact with ML pipelines through natural language prompts. The system leverages LLMs to translate textual descriptions into code, suggest model configurations and interpret ML tasks in a human-centric manner. Experimental evaluation across diverse public datasets demonstrates the system's ability to streamline model development while maintaining high performance and reproducibility. By bridging the gap between domain expertise and technical implementation, this integration fosters more intuitive, scalable and democratized ML development. The results highlight the potential of LLMs to transform AutoML into a truly interactive and accessible tool for a broader range of users.

1 INTRODUCTION

In recent years, Machine Learning (ML) has become a cornerstone of technological advancement across a wide range of domains, including healthcare, finance, manufacturing and education. Its ability to extract patterns from vast datasets and make data-driven decisions has enabled the development of intelligent systems that outperform traditional rule-based approaches. As data generation continues to accelerate, the demand for effective and scalable ML solutions has never been greater.

AutoML has emerged as a powerful tool to democratize access to ML by automating key steps in the model development pipeline, such as data preprocessing, feature selection, algorithm selection and hyperparameter tuning (Chang et al., 2024). By reducing the need for expert intervention, AutoML accelerates the deployment of ML models and enables non-experts to build high-performing solutions efficiently (Karmaker et al., 2021). This automation not only saves time but also enhances reproducibility and scalability.

Large Language Models (LLMs), such as GPT and its successors, have revolutionized the field of natural language processing through their ability to generate coherent text, understand context and perform complex language-related tasks with minimal supervision (Fan et al., 2024). Trained on massive corpora, LLMs demonstrate remarkable generalization capabilities and have been successfully applied to tasks ranging from summarization and translation to code generation and reasoning.

The integration of LLMs into AutoML workflows presents a promising frontier in ML research and

[a] https://orcid.org/0009-0004-9536-1374
[b] https://orcid.org/0009-0006-0466-3065
[c] https://orcid.org/0000-0002-7461-2961
[d] https://orcid.org/0000-0001-8636-3213
[e] https://orcid.org/0000-0003-4132-9550
[f] https://orcid.org/0000-0002-7060-9298

Miguel-Morante, N., Rivero, I., García-Prieto, D., Duque, R., Palazuelos, C. and Casas, A.
Integrating Large Language Models into Automated Machine Learning: A Human-Centric Approach.
DOI: 10.5220/0013819700004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 1: KDIR*, pages 465-472

application (Duque et al., 2025). LLMs can enhance AutoML systems by interpreting user intent expressed in natural language, generating code for ML pipelines and even suggesting model configurations based on textual descriptions of data or goals. This synergy opens the door to truly human-centric ML development, where domain experts can interact with ML systems in a more intuitive and accessible way, strengthening the role of human-computer interaction in machine learning workflows. This paper proposes a software system that leverages the capabilities of LLMs to streamline the AutoML process. The proposed approach is evaluated using a diverse set of publicly available datasets, ensuring transparency, reproducibility and practical relevance.

The remainder of this article is structured as follows. Section 2 provides a review of related work on AutoML and LLMs, highlighting recent advances and existing limitations. Section 3 describes the proposed approach for integrating LLMs into AutoML pipelines. Section 4 presents the experimental setup and results, followed by a discussion of key findings in Section 5. Finally, Section 6 concludes the article and outlines directions for future work.

2 RELATED WORK

AutoML tools have become essential for making AI more accessible to a broader audience. Among the leading commercial platforms, Google AutoML (Google AutoML,), Azure Automated ML (Azure Automated ML,) and Amazon SageMaker Autopilot (Das et al., 2020) are platforms that allow users without advanced expertise to create customized predictive models for images, text and tabular data, offering integration with their ecosystems, scalability, explainability and both visual and programmable tools. On the open-source side, projects like MindsDB (MindsDB, 2018), H2O AutoML (LeDell and Poirier, 2020) and Ludwig (Molino et al., 2019) provide powerful automation for tasks such as classification and regression. Ludwig, in particular, adopts a declarative approach using YAML configuration files, enabling users to build complex deep learning pipelines with minimal code. These platforms vary in complexity, ranging from low-code solutions for non-experts to fully customizable frameworks designed for developers and researchers.

Human-Centered Artificial Intelligence (HCAI) (Shneiderman, 2022) highlights the importance of integrating human domain expertise and values in the middle of the lifecycle of design, development and deployment of intelligent system. The result is a two-dimensional framework with high levels of both automation and human control simultaneously through thoughtful design rather than the unique dimension of automation like traditional AI presents. Human-Guided Machine Learning (HGML) (Gil et al., 2019) involves active human involvement throughout the ML process. The combination of high automation with human control enables users to influence data selection, model configuration and evaluation based on domain knowledge. By aligning AutoML systems with principles of usability, fairness and explainability, HCAI and HGML help ensure that ML solutions are not only technically effective but also ethically responsible and accessible to non-AI specialists.

With the rise of LLMs, their integration into AutoML systems offers a transformative opportunity to create more intuitive, context-aware interactions between human users and automated tools. Frameworks such as AutoM3L (Luo et al., 2024), Aliro (Choi et al., 2023), GizaML (Sayed et al., 2024) and JarviX (Liu et al., 2023) represent early efforts in this direction, although often face limitations related to flexibility, domain generalization, user interaction or reliance on specific technologies. LLMs help bridge the gap between technical complexity and user accessibility through their strengths in semantic understanding, natural language processing and code generation (Tornede et al., 2023). This enables users to define tasks, interpret outcomes and refine models using conversational language. Additionally, LLMs introduce a new dimension of language-driven reasoning and decision-making, allowing AutoML systems to better infer user intent, tailor solutions to domain-specific needs and automate more complex aspects of the ML pipeline.

3 HUMAN-CENTRIC APPROACH

To bridge the gap between user intent and the complexity of machine learning pipelines, we developed a human-centric software tool that integrates LLMs into the AutoML workflow. The system leverages the interpretability and flexibility of LLMs to support users in the design and configuration of machine learning models through natural language interactions.

3.1 Background on Tools and Technologies

To facilitate understanding of the technologies involved in our system, this subsection provides a brief overview of the key tools and frameworks that underpin the proposal:

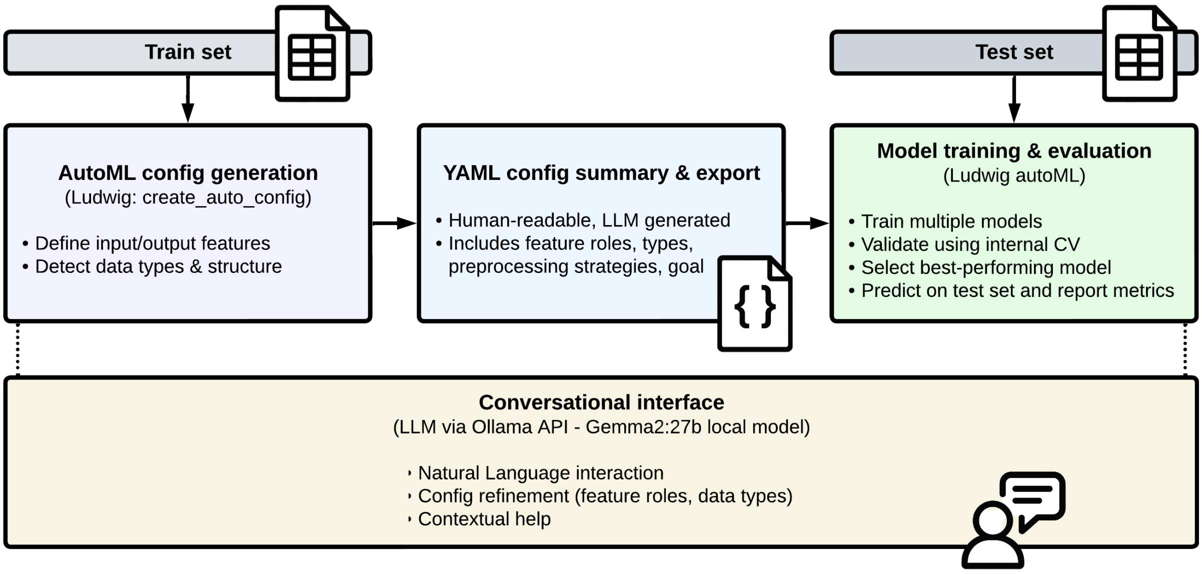

Figure 1: System architecture of the human-centric AutoML assistant.

- **Ludwig:** An open-source declarative machine learning framework developed by Uber that allows users to train deep learning models without writing code, using YAML configuration files (Molino et al., 2019).
- **YAML:** A human-readable data serialization format, well-suited for configuration files in machine learning pipelines due to its simplicity and hierarchical structure (Ben-Kiki et al., 2009).
- **Gemma:** A family of open-source large language models developed by Google DeepMind, designed for performance and efficient local deployment (DeepMind, 2024).
- **Ollama:** A lightweight server and runtime environment that enables local execution of large language models, offering RESTful APIs for seamless integration (Team, 2024).

3.2 System Architecture

The proposed system is implemented as a command-line assistant, structured around four main components (see Figure 1):

- **Data ingestion and preprocessing**, supporting multiple formats including `.csv`, `.xlsx` and `.arff`.
- **AutoML configuration generation**, using Ludwig's `create_auto_config` function to automatically define input/output features and model structure based on the target column.
- **Conversational interface**, powered by an LLM hosted via the Ollama API, which enables dynamic dialogue with the user for refining configurations, clarifying concepts and summarizing pipeline intent.
- **Model training and evaluation**, using the finalized configuration to automatically train and validate the best-performing machine learning model with Ludwig's AutoML pipeline.

The LLM used is a local instance of the `Gemma2:27b` open-source model accessed via the Ollama server. Communication is performed via RESTful API calls and responses are streamed to provide real-time feedback to the user.

3.3 Integration of LLM and AutoML

Once the dataset is loaded (e.g., the training set), the system begins an interactive configuration process driven by the conversational interface.

Users are guided through the setup process by responding to prompts or directly asking questions prefixed with `"help:"`. These are interpreted by the LLM to offer contextual help, enhancing the accessibility of the system for non-expert users.

The assistant also supports editing of automatically generated configurations by allowing users to select feature roles (input, output or ignored) and specify data types for each column, based on predefined Ludwig-compatible options.

This dialogic interaction model is designed to accommodate both novice and advanced users by offering a balance between automation and customization.

Once the configuration is finalized, the LLM generates a concise summary of the pipeline's intent.

This summary includes the role of each feature, data preprocessing strategies (such as separator type and missing value handling) and the goal of the model. This promotes transparency, facilitates documentation and aids in validating the configuration with domain experts.

All user choices are converted into a complete YAML configuration file that adheres to the schema expected by Ludwig's AutoML framework. This configuration includes input and output feature definitions, data preprocessing strategies and additional metadata.

The YAML file is then passed directly to Ludwig's `create_auto_config` utility, which uses it to automatically generate, train and evaluate candidate models (see Figure 2), selecting the best-performing configuration based on internal cross-validation and defined time constraints.

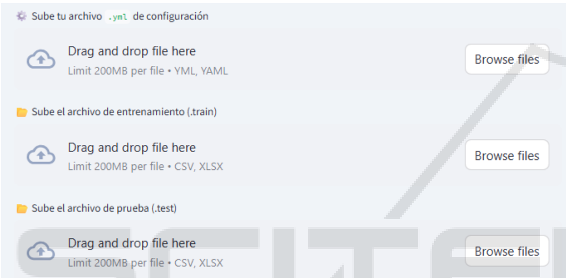

Figure 2: User interface to generate, train and predict on new data with candidate models.

The integration is deeply aligned with HCAI principles. Instead of merely replacing manual processes, the LLM acts as a collaborative assistant—allowing users to inject domain knowledge, understand design decisions and iteratively refine the ML pipeline.

This approach encourages meaningful interaction between human intuition and algorithmic automation, addressing common criticisms of AutoML systems as "black boxes".

4 EVALUATION

To assess the proposal, the following subsections present a two-fold evaluation strategy. First, a quantitative experimental evaluation is conducted by comparing the performance of the developed models with reference results generated by an expert user employing Ludwig on diverse public datasets covering both classification and regression tasks. This analysis aims to provide objective evidence of the models' predictive capabilities.

In addition, a separate subsection focuses on HCAI considerations. This complementary evaluation ensures that the models are not only technically sound but also aligned with human values and practical deployment requirements.

4.1 Performance Assessment on Public Benchmark Datasets

To evaluate the performance of the models, we selected several public datasets from OpenML and Kaggle, categorized into classification and regression tasks. For comparison, benchmark results were generated by an expert user with extensive experience in Ludwig, using the same training and test splits. To ensure fairness, training time was limited to 5 minutes per dataset for both our system and the Ludwig expert.

Classification problems involve predicting discrete class labels. The following datasets were used for classification tasks:

- **Pima Indians Diabetes.** This dataset was collected by the National Institute of Diabetes and Digestive and Kidney Diseases and is hosted on OpenML ((`Dataset ID 37, Task ID 267`). It includes medical measurements such as glucose level, BMI and age to predict the onset of diabetes in Pima Indian women.
- **Breast Cancer Wisconsin (Breast-w).** Provided by Dr. William H. Wolberg and hosted on OpenML (`Dataset ID: 15, Task ID: 245`), this dataset contains features derived from digitized images of fine needle aspirates (FNA) of breast masses to classify tumors as malignant or benign.
- **Contraceptive Method Choice (CMC).** Derived from the 1987 National Indonesia Contraceptive Prevalence Survey and available on OpenML (`Dataset ID: 23. Task ID: 253`), this dataset predicts the contraceptive method choice (no-use, long-term or short-term) among married women based on demographic data.
- **Hypothyroid.** A medical diagnosis dataset hosted on OpenML (`Dataset ID: 57, Task ID: 3044`), used to identify hypothyroidism using clinical and laboratory features. It contains both categorical and numerical data from patients.

Table 1 presents a summary of the classification datasets, including the number of instances and features, as well as their respective sources.

Regression tasks involve predicting continuous numeric values. The following datasets were selected:

Table 1: Summary of classification datasets.

Dataset name	Train size	Test size	Features	Classes
Pima Indians Diabetes	515	253	8	2
Breast-w	469	230	9	2
Contraceptive M.C.	987	486	9	3
Hypothyroid	2528	1245	29	3

Table 2: Summary of regression datasets.

Dataset name	Train size	Test size	Features
Liver-disorders	232	113	5
BrisT1D Blood Glucose	177024	3644	506
COVID-19 Death Prediction	129156	43052	18
COVID-19 Cases Prediction	2700	893	92

Table 3: Classification performance comparison.

Dataset	Metric	Our model	Ludwig expert
Pima Indians Diabetes	ROC-AUC	0.714	0.826
Breast-w	ROC-AUC	0.956	0.967
Contraceptive M.C.	Accuracy	0.484	0.495
Hypothyroid	Accuracy	0.740	0.798

Table 4: Regression performance comparison.

Dataset	Metric	Our model	Ludwig expert
Liver-disorders	RMSE	5.223	3.912
BrisT1D Blood Glucose	RMSE	4.395	3.300
COVID-19 Death Prediction	RMSE	2319	1743
COVID-19 Cases Prediction	RMSE	2.028	1.521

- **Liver Disorders.** Hosted on OpenML (`Dataset ID: 8, Task ID: 211690`)), this dataset includes biochemical test results and alcohol consumption indicators to predict liver disease.
- **BrisT1D Blood Glucose.** This Kaggle dataset contains data from Type 1 Diabetes patients, including glucose, insulin and physical activity readings. The goal is to predict glucose levels one hour ahead.
- **COVID-19 Deaths.** From Kaggle, this dataset contains historical COVID-19 death counts worldwide. It includes features like date, region and public health indicators to forecast future mortality.
- **COVID-19 Cases.** A classification dataset hosted on Kaggle (ML 2021 Spring - HW1 competition) with anonymized numerical features and binary labels related to COVID-19 infection cases.

Table 2 provides a summary of the regression datasets, including the size and source of each dataset.

To evaluate model performance, we selected specific metrics suited to the nature of each prediction task. For binary classification problems, we used the ROC-AUC score to assess the trade-off between true and false positive rates. For multiclass classification, accuracy was used as the primary performance indicator. In regression tasks, model performance was evaluated using the Root Mean Squared Error (RMSE), which measures the average magnitude of prediction errors.

To assess the relative performance of our models, we compared our classification accuracy with results obtained by the Ludwig expert. These results are shown in Table 3.

The proposed system performs competitively on classification tasks, with ROC-AUC and accuracy scores approaching the expert benchmark, particularly on the Breast-w and C.M.C datasets.

Finally, Table 4 presents the RMSE values for each regression dataset, comparing our model's performance against the Ludwig expert.

Regression performance is slightly below the benchmark, with higher RMSE values across all datasets. This indicates that while the system produces usable models, further refinement is needed for tasks involving high-dimensional data or datasets with a large number of samples, potentially due to the limited training time budget.

4.2 Analysis from HCAI Perspective

To complement the technical evaluation, an independent expert in Artificial Intelligence was consulted to assess various aspects related to the degree of automation and the level of human control in the proposed system, which integrates AutoML techniques with LLMs. The evaluation focused on the system's alignment with key principles of HCAI. To guide the assessment, the expert was asked to answer nine questions (see Appendix). Table 5 presents the ratings provided for each question in the survey, based on a scale from 1 (very poor) to 10 (excellent).

Table 5: Expert Rating Summary (1 = Very Poor, 10 = Excellent).

Question (ID + Descriptor)	Expert Rating
Q1 – Ease of Use	7
Q2 – Configurable Pipeline	7
Q3 – Interpretability	8
Q4 – Data Type Handling	9
Q5 – Data Cleaning	8
Q6 – Feature Engineering	6
Q7 – Model Training	8
Q8 – Hyperparameter Tuning	7
Q9 – Model Validation	7

From the expert's perspective, the system appears easy to use, even though it targets users with technical backgrounds. The AutoML configuration, in particular, is relatively straightforward. The application demonstrates a degree of flexibility, but there is room for improvement. Output responses are clear and understandable. The system seems capable of handling various data types effectively and the data cleaning module is integrated and works as expected. However, some components (e.g., feature engineering and validation) were not fully observable during the evaluation. The system heavily relies on AutoML for tasks like model training, hyperparameter tuning and validation. When properly configured, these components function adequately, though full evaluation was not always possible.

5 DISCUSSION

The experimental results across diverse datasets indicate that the proposed integration of LLMs into AutoML pipelines can maintain competitive performance while offering a significantly more accessible and interpretable interface for users. Particularly in classification tasks such as Breast-w and Hypothyroid, the models achieved high ROC-AUC and accuracy scores, demonstrating the effectiveness of Ludwig's AutoML pipeline when configured with the support of an LLM-based assistant.

Beyond raw performance, one of the most important contributions of the proposed system lies in its human-centric design. By embedding natural language interactions throughout the AutoML process, the system lowers the barrier to entry for users with limited technical backgrounds. The conversational interface—capable of handling questions, clarifying terminology and explaining configuration decisions—enables a more transparent and inclusive user experience. Users are no longer passive observers of automated decisions but active participants who can guide, question and refine the modeling process.

From the perspective of HCAI, the system exemplifies several core principles of human-centered design. It supports meaningful human control by allowing users to intervene at key decision points, such as model selection, data preprocessing and hyperparameter tuning. The integration of LLMs fosters increased interpretability by translating complex technical processes into comprehensible language, thus enhancing user trust and understanding. Moreover, the system promotes accountability, as its design encourages users to review and validate model configurations rather than relying blindly on automated outputs.

However, the evaluation also revealed areas for improvement. While the conversational interface is helpful, certain advanced functionalities—such as feature engineering and detailed validation workflows—remain less transparent or partially observable, particularly for users without prior knowledge of machine learning pipelines. Additionally, the reliance on AutoML implies some trade-offs in flexibility and fine-grained control, which could affect expert users seeking full customization.

Overall, the integration of LLMs into AutoML pipelines demonstrates promise not only for improving accessibility and performance but also for advancing the broader goals of HCAI: systems that are understandable, controllable and aligned with human values and expertise.

6 CONCLUSIONS

This work presents a novel human-centric software system that integrates LLMs into AutoML workflow. By leveraging natural language processing, the system enables users to design, configure and understand ML pipelines through a conversational interface. Experimental results demonstrate that this approach can achieve high model performance on both classification and regression tasks while enhancing user interaction, interpretability and accessibility.

The system contributes to the democratization of ML by allowing domain experts and non-specialists to meaningfully participate in the development of predictive models. It operationalizes key principles of HCAI by combining automation with user control, transparency and contextual support. Furthermore, its modular and open-source architecture provides a strong foundation for future enhancements.

Looking ahead, several directions for future work are identified. First, expanding the system's multilingual capabilities and fine-tuning LLMs on domain-specific corpora may improve accuracy in interpreting complex or specialized queries. Second, integrating additional AutoML frameworks beyond Ludwig could broaden compatibility and adoption. Third, introducing support for advanced data manipulation (e.g., time series decomposition, anomaly detection or unsupervised learning) would extend the system's versatility.

ACKNOWLEDGEMENTS

This work has been partially supported by grant PID2023-146243OB-I00 funded by MICIU/AEI/ 10.13039/501100011033 and by "ERDF/EU". This research was also partially developed in the project FUTCAN - 2023 / TCN / 018 that was co-financed from the European Regional Development Fund through the FEDER Operational Program 2021-2027 of Cantabria through the line of grants "Aid for research projects with high industrial potential of excellent technological agents for industrial competitiveness TCNIC".

REFERENCES

Azure Automated ML. Azure Automated ML. https://azure.microsoft.com/en-us/solutions/automated-machine-learning. Access: 2 January 2025.

Ben-Kiki, O., Evans, C., and Net, I. d. (2009). Yaml ain't markup language (yaml™) version 1.2. https://yaml.org/spec/1.2/spec.html.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Choi, H., Moran, J., Matsumoto, N., Hernandez, M. E., and Moore, J. H. (2023). Aliro: an automated machine learning tool leveraging large language models. *Bioinformatics*, 39(10):btad606.

Das, P., Ivkin, N., Bansal, T., Rouesnel, L., Gautier, P., Karnin, Z., Dirac, L., Ramakrishnan, L., Perunicic, A., Shcherbatyi, I., Wu, W., Zolic, A., Shen, H., Ahmed, A., Winkelmolen, F., Miladinovic, M., Archembeau, C., Tang, A., Dutt, B., Grao, P., and Venkateswar, K. (2020). Amazon sagemaker autopilot: a white box automl solution at scale. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, DEEM '20, New York, NY, USA. Association for Computing Machinery.

DeepMind, G. (2024). Gemma: Open models by google deepmind. https://deepmind.google/technologies/gemma.

Duque, R., Tîrnăucă, C., Palazuelos, C., Casas, A., López, A., and Pérez, A. (2025). Bridging automl and llms: Towards a framework for accessible and adaptive machine learning. In Filipe, J., Smialek, M., Brodsky, A., and Hammoudi, S., editors, *Proceedings of the 27th International Conference on Enterprise Information Systems, ICEIS 2025, Porto, Portugal, April 4-6, 2025, Volume 1*, pages 959–964. SCITEPRESS.

Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., and Hemphill, L. (2024). A bibliometric review of large language models research from 2017 to 2023. *ACM Transactions on Intelligent Systems and Technology*, 15(5):1–25.

Gil, Y., Honaker, J., Gupta, S., Ma, Y., D'Orazio, V., Garijo, D., Gadewar, S., Yang, Q., and Jahanshad, N. (2019). Towards human-guided machine learning. In *Proceedings of the 24th international conference on intelligent user interfaces*, pages 614–624.

Google AutoML. Google AutoML. https://cloud.google.com/automl. Access: 2 January 2025.

Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., and Veeramachaneni, K. (2021). Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8):1–36.

LeDell, E. and Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*.

Liu, S.-C., Wang, S., Lin, W., Hsiung, C.-W., Hsieh, Y.-C., Cheng, Y.-P., Luo, S.-H., Chang, T., and Zhang, J. (2023). Jarvix: A llm no code platform for tabular data analysis and optimization. *arXiv preprint arXiv:2312.02213*.

Luo, D., Feng, C., Nong, Y., and Shen, Y. (2024). Autom3l: An automated multimodal machine learning framework with large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8586–8594.

MindsDB (2018). MindsDB. https://mindsdb.com/. Access: 6 December 2024.

Molino, P., Dudin, Y., and Miryala, S. S. (2019). Ludwig: a type-based declarative deep learning toolbox.

Sayed, E., Maher, M., Sedeek, O., Eldamaty, A., Kamel, A., and El Shawi, R. (2024). Gizaml: A collaborative meta-learning based framework using llm for automated time-series forecasting. In *EDBT*, pages 830–833.

Shneiderman, B. (2022). *Human-centered AI*. Oxford University Press.

Team, O. (2024). Ollama: Run large language models locally. https://ollama.com.

Tornede, A., Deng, D., Eimer, T., Giovanelli, J., Mohan, A., Ruhkopf, T., Segel, S., Theodorakopoulos, D., Tornede, T., Wachsmuth, H., et al. (2023). Automl in the age of large language models: Current challenges, future opportunities and risks. *arXiv preprint arXiv:2306.08107*.

APPENDIX

The following questionnaire is intended to evaluate whether our AutoML application aligns with the principles of Human-Centered AI (HCAI), based on two key dimensions: Human Control and High Automation.

Human Control (Rate from 1 = Very Poor to 10 = Excellent)

Please rate the following aspects of the application on a scale from 1 (very poor) to 10 (excellent) and briefly justify your answer.

1. **Ease of Use**
   The application is easy to use and allows smooth interaction without requiring extensive technical expertise.
   Rating (1–10): ______
   Justification:
   ____

2. **Configurable Pipeline**
   The application provides sufficient flexibility to configure or customize different stages of the ML pipeline.
   Rating (1–10): ______
   Justification:
   ____

3. **Interpretable Information**
   The application offers clear, interpretable information that supports understanding and oversight of the models it generates.
   Rating (1–10): ______
   Justification:

High Automation (Rate from 1 = Very Poor to 10 = Excellent)

Please rate the level of automation for each function below using a scale from 1 (very poor) to 10 (excellent) and justify your response.

4. **Data Type Handling**
   The application can automatically detect and appropriately handle various types of data (e.g., numerical, categorical, text).
   Rating (1–10): ______
   Justification:
   ____

5. **Data Cleaning**
   The application performs necessary data cleaning operations (e.g., missing values, duplicates) automatically without requiring manual intervention.
   Rating (1–10): ______
   Justification:
   ____

6. **Feature Engineering**
   The application is capable of automatically generating or selecting relevant features to improve model performance.
   Rating (1–10): ______
   Justification:
   ____

7. **Model Training**
   The application can automatically select and train suitable ML models based on the dataset provided.
   Rating (1–10): ______
   Justification:
   ____

8. **Hyperparameter Tuning**
   The application effectively automates the process of model tuning and optimization.
   Rating (1–10): ______
   Justification:
   ____

9. **Model Validation**
   The application includes automated procedures for evaluating and validating the trained models using appropriate techniques.
   Rating (1–10): ______
   Justification: