Acta Informatica Pragensia
2026, Volume 15, Issue 1, 72–89
https://doi.org/10.18267/j.aip.288

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

Automated Machine Learning in Action: Performance Evaluation for Predictive Analytics Tasks

Nicolas Leyh

TUM School of Management, Technical University of Munich, Munich, Germany

Corresponding author: Nicolas Leyh (nicolas.leyh@tum.de)

Editorial Record

First submission received:
June 19, 2025

Revisions received:
August 10, 2025
August 23, 2025

Accepted for publication:
August 25, 2025

Academic Editor:
Stanislav Vojir
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Leyh, N. (2026). Automated Machine Learning in Action: Performance Evaluation for Predictive Analytics Tasks. *Acta Informatica Pragensia*, 15(1), 72–89. https://doi.org/10.18267/j.aip.288

Abstract

Background: As organizations increasingly seek data-driven insights, the demand for machine learning (ML) expertise outpaces the current workforce supply. Automated Machine Learning (AutoML) frameworks help close this gap by streamlining the ML pipeline, making advanced modeling accessible to non-specialists.

Objective: This study evaluates the performance of four open-source AutoML frameworks—Auto-Keras, Auto-Sklearn, H2O, and TPOT—in predictive analytics, focusing on both binary and multiclass classification. The goal is to identify performance strengths and limitations under varying dataset conditions and propose improvements for framework optimization.

Methods: Quantitative experimental research design was employed. 22 publicly available datasets were selected from established benchmarking sources, covering diverse predictive analytics challenges. Framework performance was assessed across twelve data segments, defined by characteristics such as sample size, feature count, and categorical feature proportion. Evaluation metrics included AUC for binary and accuracy/F1 for multiclass classification tasks, with standardized runtime constraints applied to ensure comparability.

Results: The findings show that H2O delivered strong results across diverse datasets, particularly for binary classification. However, no single framework achieved superior performance across all data segments. Auto-Sklearn performed well in multiclass classification, especially with higher feature counts, while Auto-Keras and TPOT demonstrated variable outcomes depending on dataset complexity. Performance declined notably in scenarios with high categorical proportions, severe class imbalance, or extensive missing values.

Conclusion: This study demonstrates that AutoML frameworks can substantially support predictive analytics but exhibit distinct strengths and limitations under specific data conditions. While H2O proved most robust overall, targeted refinements such as enhancing feature selection in Auto-Keras and improving categorical variable handling in Auto-Sklearn could further optimize performance. The findings provide actionable insights for both practitioners selecting frameworks and developers enhancing AutoML design, highlighting the need for ongoing innovation to ensure adaptability to complex predictive analytics tasks.

Index Terms

AutoML frameworks; Performance benchmarking; Predictive analytics; Binary and multiclass classification; Model interpretability.

1 INTRODUCTION

Automated Machine Learning (AutoML) frameworks have significantly improved accessibility to machine learning (ML) models, allowing non-experts to apply complex analytics techniques (Balaji & Allen, 2018).

This advancement allows for the effective bridging of the demand-supply gap in ML expertise within organizations (SAS Institute, 2022). The functional goal of AutoML frameworks is hereby to automate the tasks within the machine learning pipeline (Zöller & Huber, 2021). This automation and accessibility is crucial for organizations wanting to leverage ML without the need for deep technical knowledge (Eldeeb et al., 2023; Schmitt, 2023).

Existing research has explored the efficacy of AutoML frameworks across diverse datasets. Recently, Eldeeb et al. (2023) examined how design decisions affect AutoML framework performance, while Schmitt (2023) focused on comparing H2O AutoML's performance to manual model development. Furthermore, Gijsbers et al. (2023) underscore the necessity for standardized benchmarking, introducing the open-source AMLB benchmark tool to avoid common pitfalls in AutoML comparisons and enable detailed performance analysis across numerous classification and regression tasks. Additionally, previous research from Balaji & Allen (2018), Gijsbers et al. (2019), Truong et al. (2019), Zöller & Huber (2021), Ge (2020), and Tuggener et al. (2019) on AutoML framework performance across datasets from diverse domains provide profound theoretical foundations, however, are missing applicability due to continuous updates and improvements of the AutoML frameworks. Finally, there has been research done with a very specific focus on tailored applications and fields such as materials science (Conrad et al., 2022), genomics (Khan et al., 2023), and healthcare (Lenkala et al., 2023), which, while impactful, lack broad applicability especially in business contexts.

Despite these advancements in AutoML research, a critical gap remains in the up-to-date evaluation of these frameworks specifically for predictive analytics tasks. Although prior studies have assessed AutoML performance across diverse domains, they have not addressed the unique challenges posed by predictive analytics datasets – characterized by complex structures, high dimensionality, and pronounced class imbalances (Eldeeb et al., 2023). This omission is particularly consequential given the increasing reliance on predictive analytics for strategic business decision-making, where accurate forecasting of trends and behaviors is essential (Asniar & Surendro, 2019). Moreover, the systematic impact of key data characteristics - such as the proportion of categorical features and the degree of class imbalance – on AutoML performance remains underexplored (Schmitt, 2023; Conrad et al., 2022). This study addresses these gaps by evaluating the performance of four prominent AutoML frameworks on predictive analytics datasets. Model-specific limitations are analyzed and targeted improvements proposed to better manage these data challenges. In doing so, our research not only advances the theoretical AutoML framework understanding but also delivers actionable insights for optimizing adaptive, high-performance AutoML solutions in diverse business environments (Khan et al., 2023; Lenkala et al., 2023).

As a result, this study investigates the two research questions: Which among the four AutoML frameworks—AutoKeras, Auto-Sklearn, H2O, and TPOT—achieves the highest performance in binary and multiclass classification tasks when applied to predictive analytics datasets? Which specific data characteristics (e.g., class imbalance, categorical feature proportion) impact AutoML performance in predictive analytics, and what targeted optimizations could mitigate these issues?

To address the research questions at hand, a quantitative approach is employed, utilizing an experimental research design (Bell, 2009) to systematically manipulate data characteristics and analyze their impact on the performance of the AutoML frameworks.

Concerning the theoretical implications of the research, this study provides an up-to-date evaluation of AutoML framework performance, focusing on binary and multiclass classification tasks within predictive analytics. By demonstrating performance fluctuations across various data conditions, this research builds upon the findings of Gijsbers et al. (2019), Truong et al. (2019), Eldeeb et al. (2023), and Tuggener et al. (2019), who have also noted the variability in AutoML framework performance. This study also contributes to the ongoing discourse on the impact of data characteristics - such as categorical feature proportions, class imbalance, and missing values - on AutoML efficacy and extends it to the field of predictive analytics. By identifying these limitations and proposing targeted technical improvements, it extends the work of researchers like Gijsbers et al. (2019) and Truong et al. (2019) to provide actionable insights for optimizing AutoML. Lastly, the study highlights the advancement of AutoML framework performance in multiclass classification tasks, both supporting and contradicting the results of Eldleeb et al. (2019), Zöller & Huber (2021), and Conrad et al. (2022).

1.1 AutoML frameworks

Automated machine learning (AutoML) frameworks represent an advancement in the field of machine learning (ML) deployment and accessibility. These frameworks are aimed for automating the repetitive, complex, and time-consuming tasks within a ML pipeline (Schmitt, 2023). The structure of AutoML frameworks is built on the classical ML pipeline, which involves the following key phases: data collection, exploratory data analysis, data preprocessing, feature engineering, model training, and model evaluation (Zöller & Huber, 2021).

Following the data collection, the exploratory data analysis phase can start to utilize AutoML frameworks to understand data characteristics and guide subsequent preprocessing steps (Khan et al., 2023). Afterward, the AutoML frameworks automate data preprocessing tasks such as handling missing values and normalizing data (Zöller & Huber, 2021). The fourth phase leverages the frameworks to streamline feature engineering (Bilal et al., 2022). In the fifth phase the AutoML frameworks automate the selection of the ML models and their hyperparameter optimization (Zöller & Huber, 2021). This is followed by the last phase, where the AutoML frameworks evaluate the trained models using appropriate metrics (Topsakal & Akıncı, 2023).

1.2 Performance of AutoML frameworks

The performance of AutoML frameworks is primarily influenced by the framework's design, however time constraints, computational resources, and dataset characteristics also have a significant impact. Especially for the dataset characteristics – as also the research focus of this paper - larger datasets increase the complexity of feature engineering and computational demands, while issues like class imbalances, categorical variables, and missing values can significantly degrade model performance (Topsakal & Akıncı, 2023). Existing literature underscores the varying strengths and weaknesses of AutoML frameworks across different datasets and classification tasks. Given the architectural differences and prior benchmark results of the four AutoML frameoworks chosen, the following hypotheses were formulated to explore their relative strengths under varying dataset conditions.

***H1**: H2O-AutoML outperforms other frameworks under diverse dataset conditions in binary classification tasks for predictive analytics tasks.*

> In binary classification, H2O-AutoML emerges as particularly effective due to its use of stacked ensemble methods and random search for hyperparameter tuning, making it efficient and reliable for this task type across varied datasets (Truong et al., 2019; Gijsbers et al., 2023; Schmitt, 2023). Studies confirm that H2O maintains robust performance under diverse dataset conditions and generally excels in binary settings, a result of its optimized ensemble approach, which balances model accuracy with computational efficiency (Balaji & Allen, 2018; Truong et al., 2019). Gijsbers et al. (2023) note that H2O's stability holds even as dataset complexity increases, though performance may be affected with highly imbalanced classes if extensive tuning is absent. Given its ability to deliver reliable accuracy and speed across benchmarks, H2O-AutoML's architecture allows it to manage variability in binary classification effectively.

***H2**: Auto-Sklearn performs the best with multiclass classification tasks within predictive analytics, excelling particularly where the dataset feature count is high.*

> Auto-Sklearn is notably effective in multiclass classification due to its meta-learning and ensemble strategies, which allow it to optimize pipeline configurations flexibly for datasets with complex categorical distributions (Balaji & Allen, 2018; Eldeeb et al., 2023). This adaptability, enhanced by ensemble construction that combines top-performing models, enables Auto-Sklearn to achieve high performance across feature-rich multiclass datasets, as identified by Gijsbers et al. (2019, 2023) and Eldeeb et al. (2023). Comparisons in Zöller & Huber (2021) and He et al. (2021) show that Auto-Sklearn outperforms frameworks lacking meta-learning, such as H2O, especially when categorical diversity is high, supporting its leading position in multiclass classification. Gijsbers et al. (2023) further highlight Auto-Sklearn's stability across multiclass benchmarks, affirming its suitability for complex multiclass classification tasks.

***H3**: Larger sample sizes with high feature counts enhance AutoML framework performance for predictive analytics tasks, while low categorical proportions and shape ratios help maintain stable performance.*

> Beyond individual framework designs, research shows that larger sample sizes with high feature counts enhance AutoML framework performance by enabling more robust generalization and reducing overfitting

risks. Gijsbers et al. (2023) demonstrate that those datasets provide richer patterns, allowing models to capture data distributions more effectively, particularly benefiting scalable frameworks like Auto-Sklearn and H2O. Increased feature count helps models learn generalizable patterns, improving prediction accuracy and stability across test sets, a critical advantage for frameworks relying on automated feature selection and model tuning (Elshawi et al., 2019). With ample data, frameworks like TPOT and AutoGluon also mitigate performance fluctuations caused by smaller datasets' sensitivity to random splits, as highlighted by Balaji & Allen (2018). Although larger sample sizes require more computational power, frameworks efficiently balance this demand with enhanced accuracy on typical hardware, according to Truong et al. (2019). Balanced categorical proportions and shape ratios, while not enhancing performance, support stability by preventing overfitting and ensuring efficient computational use (Ge, 2020; Tuggener et al., 2019).

H4: *Dataset complexities - high class imbalance, and missing values - reduce AutoML framework performance for predictive analytics tasks.*

A range of studies indicate that dataset complexities, particularly high class imbalance, and missing values, generally degrade the performance of AutoML frameworks. Frameworks with such class imbalance often skew predictions towards the majority class, undermining accuracy for minority classes, as documented by Tuggener et al. (2019) and Eldeeb et al. (2023). Additionally, missing values add further complexity, as imputation processes can consume resources and introduce biases, affecting model accuracy, as noted by He et al. (2021) and Zöller & Huber (2021). These dataset complexities collectively strain AutoML frameworks and are particularly limiting under standard hardware constraints.

1.3 Performance of AutoML frameworks for predictive analytics

Predictive analytics involves the application of statistical models and ML algorithms to historical data to make informed predictions about future events (Grover et al., 2018). In this context, Automated Machine Learning (AutoML) has enhanced predictive analytics by automating complex tasks such as feature engineering, algorithm selection, and hyperparameter tuning, thereby reducing the need for extensive manual effort and domain expertise (Khan et al., 2023). However, challenges persist, particularly concerning the interpretability of analytics models generated by AutoML systems. While these automated processes can produce highly accurate models, understanding the rationale behind their future predictions can be difficult, posing challenges in regulated industries where explainability is crucial (Coors et al., 2021). To address these issues, ideas such as intagrating explainable AI (XAI) techniques, such as SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-agnostic Explanations), into AutoML frameworks for specifically predictive analytics is essential to enhance model transparency and foster trust among stakeholders (Velmurugan et al., 2020; Salih et al., 2023). An additional challenge is that predictive analytics datasets often contain high dimensionality, intricate feature interactions, and noisy or imbalanced data, all of which negatively influence machine learning algorithms' capacity to extract accurate insights (Lee et al., 2022).

Empirical studies have demonstrated that AutoML frameworks can achieve predictive performance comparable to, and sometimes surpassing, manually crafted models (Eldeeb et al., 2023; Ferreira et al., 2021). For instance, Conrad et al. (2022) benchmarked four AutoML frameworks on twelve small-sample datasets from specifically the materials science, finding that AutoML not only matched manual modeling in accuracy but also highlighted issues such as robustness to small data. Despite these advancements, challenges such as handling complex data, ensuring model interpretability, and managing the risk of overfitting remain (Ray et al., 2021). Addressing these issues is crucial for the broader adoption and effectiveness of AutoML in predictive analytics.

2 METHODOLOGY

The research employs a quantitative approach and adopts an experimental research design (Bell, 2009) to systematically manipulate data characteristics and analyze their influence on the AutoML framework performance.

2.1 AutoML framework selection

The selection of AutoML frameworks in this study was based on a criteria-driven approach designed to balance research rigor with practical feasibility. Given the prohibitive cost and time requirements of evaluating all available

frameworks, a representative subset was chosen to capture a range of optimization approaches without overwhelming resources (Gijsbers et al., 2023). The focus was placed on open-source frameworks to maintain transparency and accessibility for both academic and industry applications (Kavanagh, 2004). The selected frameworks - Auto-Keras, Auto-Sklearn, H2O AutoML, and TPOT - were also chosen for their variety in optimization methods, including Bayesian, and random search, which allow for a comprehensive evaluation without redundancy. Importantly, the selection includes frameworks developed by both academia and industry. Further, usability and robust documentation were prioritized to support broad accessibility and applicability in business contexts, ensuring that these frameworks empower users with minimal coding and configuration requirements while enhancing machine learning model accessibility (Balaji & Allen, 2018).

Auto-Keras (Version 1.1.0) (Jin et al., 2023) leverages scikit-learn (Pedregosa et al., 2011), an open-source ML library for statistical modeling, TensorFlow (Abadi et al., 2016), a framework for ML model development and deployment, and Keras (Chollet et al., 2015), a user-friendly interface for TensorFlow to automate the exploration of deep neural network architectures. Utilizing Bayesian optimization and hyperparameter tuning, Auto-Keras searches through various layer combinations and design features to identify the most effective deep neural network architectures for specific tasks (Jin et al., 2023).

Auto-Sklearn (Version 0.15.0) uses meta-learning to speed up the slow initialization phase in Bayesian optimization across a broad hyperparameter scope. By retaining optimization knowledge from multiple tasks and pre-training on over 100 OpenML datasets, Auto-Sklearn records optimal models and dataset characteristics. These records help assess new datasets' proximity in meta-feature space, influencing the initial benchmarks for Bayesian optimization (Feurer et al., 2015). Auto-sklearn also applies weights to different models and utilizes ensemble selection during testing to increase efficiency. By harnessing the abilities of base learners, ensembles improve model performance (Dong et al., 2020).

H2O (Version 3.44.0.2) employs a three-phase optimization process to enhance classification performance (LeDell & Poirier, 2020). It begins with a fixed grid search to establish strong baseline models, followed by a random search within predefined models for expanded exploration, and concludes with a stacked ensemble approach, which combines top-performing models to enhance predictive accuracy across dataset variations (Schmitt, 2023). H2O supports automated preprocessing, including missing value imputation and categorical encoding, reducing the need for manual intervention (H2O.ai, 2023). Its broad model search space, covering deep neural networks, gradient boosting machines, random forests, and generalized linear models, ensures robust adaptability to different dataset structures (Truong et al., 2019).

TPOT (Version 0.12.1) employs genetic programming to evolve scikit-learn pipelines through iterative generations. It starts with a random population of pipelines, which evolves over generations based on performance against a specified loss function. The top-performing pipelines are selected for further evolution. During this process, TPOT explores a vast search space, experimenting with various preprocessing steps, feature selection methods, and ML models to identify the most effective pipelines for the designated task (Olson et al., 2016).

2.2 Dataset selection and collection

Secondary and external predictive analytics datasets in tabular format were chosen from prior AutoML (Gijsbers et al., 2019; Balaji & Allen, 2018; Truong et al., 2019; Eldeeb et al., 2023; Zöller & Huber, 2021) and ML benchmark studies, for example OpenML100 (Bischl et al., 2017) and OpenML-CC18 (Bischl et al., 2019).

Table 1. *List of selected datasets.*

ID	Class. Type	Name	Classes	Sample Size	Features
29	Binary	credit-approval	2	690	16
31	Binary	credit-g	2	1000	21
44	Binary	spambase	2	4601	58
1461	Binary	bank-marketing	2	45211	17

ID	Class. Type	Name	Classes	Sample Size	Features
1462	Binary	banknote-authentification	2	1372	5
1504	Binary	steel-plates-fault	2	1941	34
1590	Binary	adult	2	48842	15
4135	Binary	Amazon-employee-access	2	32769	10
4534	Binary	PhishingWebsites	2	11055	31
6332	Binary	cylinder-bands	2	540	40
23381	Binary	dresses-sales	2	500	13
23517	Binary	Numerai28.6	2	96320	22
40701	Binary	churn	2	5000	21
40978	Binary	Internet-Advertisements	2	3279	1559
40981	Binary	Australian	2	690	15
6	Multiclass	letter	26	20000	1
54	Multiclass	vehicle	4	846	1
1468	Multiclass	cnae-9	9	1080	1
40668	Multiclass	connect-4	3	67557	43
40670	Multiclass	dna	3	3186	181
40975	Multiclass	car	4	1728	7
40984	Multiclass	segment	7	2310	1

The datasets included have 500 to 100,000 items, ensuring a balance between computational feasibility and real-world scalability (Feldman et al., 2020). This range allows for the examination of AutoML performance across small-to-moderate dataset sizes commonly encountered for predictive analytics tasks in business and industry settings while excluding extreme cases that may require specialized tuning (Bischl et al., 2021). The feature count range (5-1,600) was chosen to reflect typical predictive analytics datasets in structured data environments, ensuring an evaluation of frameworks across varying degrees of feature complexity (Grinsztajn et al., 2022). Furthermore, the datasets have to be compatible with 10-fold cross-validation to allow for the research at hand, ensuring that results are robust and not overly dependent on a specific data split. Datasets that are part of larger datasets were excluded to maintain sample independence and prevent redundancy in training, which could artificially inflate performance metrics. Only datasets with comprehensive source and reference information were included to ensure reproducibility and verifiability, allowing future researchers to replicate and extend the findings. Each dataset requires multiple attributes for effective classification, preventing trivial classification tasks that could distort the evaluation of AutoML frameworks (Lampert et al., 2013). To ensure a realistic representation of predictive analytics challenges, datasets created by binarizing classification tasks were excluded, as they often oversimplify problems and do not reflect typical real-world predictive modeling scenarios. Similarly, datasets with over 5,000 features post one-hot encoding were excluded to mitigate overfitting risks and computational inefficiencies, which can disproportionately impact certain AutoML frameworks. Finally, datasets with extreme class imbalances, where the minority class constitutes less than 5% of the majority class, were removed to avoid biased performance evaluation and ensure models can effectively learn from both majority and minority classes.

The data was accessed through the secondary sources OpenML (Vanschoren et al., 2014), the UC Irvine ML repository (Kelly et al., 2023), and Kaggle (Kaggle, 2023). Then, they were downloaded using the OpenML Python package (Feurer et al., 2021), which stores the datasets in CSV format, accompanied by descriptive text files. Table 1 presents the lists of selected datasets for binary and multiclass classification.

2.3 Data preprocessing

The holdout method was applied to divide the data into training and test sets with an 80:20 ratio. Specifically, 80% of the data was designated for training to optimize model parameters, while the remaining 20% was reserved for evaluating the model's generalization ability (Fels et al., 2023). This split strategy is widely recognized for preventing overfitting and ensuring that AutoML models perform reliably across different data distributions (Tan et al., 2021). A unique random seed was used for each split to maintain reproducibility.

To ensure compatibility and effective data preparation for each AutoML framework, specific preprocessing requirements were addressed, particularly in handling missing values, encoding categorical data, and transforming data types (Bilal et al., 2022; Pfisterer et al., 2019).

The AutoML frameworks demonstrated varied capacities for managing missing values during model training. H2O and Auto-Keras incorporate internal imputation methods for resolving missing data. H2O applies built-in imputation strategies, addressing missing values with techniques tailored to the model's requirements (H2O.ai, 2023). Auto-Keras uses strategies such as imputation or row exclusion before training (Jin et al., 2023). By contrast, TPOT and Auto-Sklearn rely on the underlying machine learning models chosen to handle missing data, meaning that if a selected model does not support missing data processing, TPOT and Auto-Sklearn require manual intervention for data completion (Olson et al., 2016; Feurer et al., 2015; Pio et al., 2023). In these cases, missing values were systematically analyzed, and imputation techniques were applied depending on the missingness pattern to maintain data integrity.

Additionally, encoding categorical variables was necessary to ensure all input data conformed to framework requirements, as most AutoML frameworks support only numerical inputs. For TPOT, categorical features were converted to numerical values using one-hot encoding through Python's pandas library function get_dummies(), which creates binary variables for each category (The pandas development team, 2020). Auto-Sklearn required transforming target data types from numerical to categorical using LabelEncoder from scikit-learn, allowing it to recognize categorical labels for classification (Pedregosa et al., 2011; Jolly, 2018). AutoML frameworks like H2O, which supports basic data type recognition, simplify preprocessing by automatically managing these transformations (H2O.ai, 2023). However, frameworks such as TPOT and Auto-Keras lack automated encoding support, making it necessary to manually encode categorical features into numerical formats for compatibility (Jin et al., 2023).

Finally, data type identification and transformation play a crucial role in preprocessing since AutoML frameworks typically require numeric input data. H2O's ability to automatically identify and convert basic data types facilitates a more streamlined workflow without extensive user intervention. In contrast, scikit-learn-based frameworks like TPOT and Auto-Sklearn demand that users manually specify data types for each column, requiring categorical data to be transformed into numerical formats prior to model training (Feurer et al., 2015; Olson et al., 2016). For these frameworks, data type conversion involved manually transforming categorical data into integer or binary formats to enable input acceptance. This added layer of preprocessing highlights the need for user intervention, particularly in TPOT and Auto-Sklearn, as they do not provide comprehensive preprocessing automation (Bilal et al., 2022; Pfisterer et al., 2019).

2.4 Framework setup

To maintain consistency and comparability, a standardized configuration for all AutoML frameworks was implemented. Default hyperparameter values and search spaces were used for each framework, ensuring that results were based on the frameworks' out-of-the-box capabilities without extensive fine-tuning. A maximum runtime of 15 minutes was defined for each framework to ensure computational efficiency while maintaining strong predictive performance. Empirical studies like Conrad et al. (2022) have shown that longer runtimes do not yield superior results. For Auto-Keras, which lacks a built-in runtime constraint, a custom Class TimeoutCallback was employed

to enforce this time limit (Jin et al., 2023). A consistent random seed was also set across all frameworks to guarantee reproducibility and control over stochastic processes during model training (LeDell & Poirier, 2020).[1]

Parallel processing was enabled for faster computations (Karras et al., 2023), and a 10-fold cross-validation resampling strategy was applied, wherein the dataset is split into 10 subsets, each iteratively used as a test set while the remaining subsets serve as training data. This strategy ensures that each data point is fully utilized, optimizing data usage and improving the robustness of error estimates (James et al., 2023). Notably, Auto-Keras required customization to handle parallel processing and cross-validation due to its limited support for these features (Jin et al., 2023).

Additionally, the optimization metric was selected according to the classification task, with AUC (Area Under the Curve) designated for binary classification and accuracy and the weighted-average F1 score for multiclass classification (see Section 3.6). This metric choice establishes the objective function to guide the optimization algorithm in prioritizing the best-performing models and hyperparameter configurations. However, certain frameworks, such as H2O, lack support for commonly used metrics like accuracy in multiclass settings, which can introduce inconsistencies in optimization. Specifically, H2O defaults to the mean per-class error as its primary metric for multiclass classification tasks (H2O.ai, 2023). To maintain consistency across frameworks despite these variations, accuracy was set as the evaluation metric where feasible. Additionally, for data segments with high class imbalances, the weighted-average F1 score was applied as an evaluation metric to address the limitations of accuracy alone.

Finally, all experiments were executed on a local machine equipped with an Intel 11th Gen CPU featuring 8 physical cores and 16 logical threads, 32 GB RAM, and running Windows 11 Pro 64-bit. The implementation environment utilized Python 3.11.4, ensuring compatibility with all selected AutoML frameworks and their dependencies. This setup provided a controlled and consistent technological infrastructure across all experiments, eliminating performance variations due to hardware or operating system differences and enabling reproducibility.

2.5 Framework training

Each AutoML framework uses its supported algorithms to optimize hyperparameters and identify the optimal model. This is then tested for generalization on the remaining 20% test data. The hyperparameter optimization methods include random search and build ensembles for H2O and Auto-Sklearn (H2O.ai, 2023, Feurer et al., 2015), neural architecture search for Auto-Keras (Jin et al., 2023), and genetic algorithms to evolve model traits for TPOT (Olson et al., 2016). If optimization metrics underperform, additional fine-tuning of parameters was conducted to improve the model's performance assessment.

2.6 Framework Evaluation

For binary classification tasks, the evaluation metric Area Under the Curve (AUC) of the Receiver Operator Characteristic (ROC) is used. In multiclass classification tasks, accuracy is the primary metric, except in scenarios with class imbalances. This is because high accuracy in multiclass scenarios can be misleading if it results from frequent class prediction rather than genuine model accuracy (Sun et al., 2009). Therefore, for evaluations considering class imbalances, the weighted-average F1 score is used, adjusting for class distribution, and providing a more balanced measure of model performance (Mathew et al., 2023). These metrics are widely recognized in academic research for their effectiveness in assessing binary and multiclass classification tasks by AutoML frameworks.

The evaluation dimensions of the research are bifurcated based on dataset classification into binary and multiclass types (Truong et al., 2019). It specifically focuses on the data segment dimension, examining how frameworks handle different data characteristics. The dataset criteria are sample size, feature count, shape ratio, categorical percentage, class ratio, and missing percentage, as detailed in Table 2. Sample size refers to the total number of items or entries within the dataset. Feature count reflects the number of dataset features, while shape ratio measures the sample size relative to feature count. Categorical percentage shows the proportion of categorical features, class ratio indicates

[1] Additionally, 60-minute runtimes and robustness tests were conducted to assess framework performance over time and under different time constraints. However, as the extended runtime limit didn't yield significant improvements or degradation, and robustness tests showed similar results, they were not included in this research paper. These results can be shared upon request.

the sample distribution across classes, and missing percentage evaluates the amount of missing data. These segments allow for precise assessment of each framework's capability to handle different predictive analytics dataset characteristics.

The thresholds for sample size, categorical percentage, and missing percentage were adopted from Truong et al. (2019). The feature count threshold was adjusted to 50, aiming to encompass a broader range of samples in segments exceeding this threshold, thereby enhancing the dataset's representativeness. The class ratio threshold, used to gauge class imbalance, was set at 10% due to the absence of a well-defined cutoff in both Truong et al. (2019) and related ML performance evaluation literature (Buda et al., 2018), where standards vary significantly. Regarding the shape ratio, which assesses the suitability of feature distribution in predictive modeling, it was established at 50 based on guidance that a higher ratio predicts classifier performance more effectively (Buda et al., 2018). This adjustment ensures that the data structure optimally supports accurate model training and evaluation.

Table 2. *Data segments.*

Data Segment	Threshold	Below Threshold	Above Threshold
Sample Size	10,000	Less than 10,000 samples	More than 10,000 samples
Feature Count	50	Less than 50 features	More than 50 features
Shape Ratio	50	Less than 50	More than 50
Categorical Percentage	One third	Less than one-third	More than one-third
Class Ratio	10%	High imbalance (<10%)	Low imbalance (>10%)
Missing Percentage	One third	Less than one-third	More than one-third

3 RESULTS

Figure 1 and 2 present the performance of the four evaluated AutoML frameworks across different data segments for binary and multiclass classification tasks. Regarding the visuals, each row represents a specific data segment characteristic, with the leftmost plot of each pair corresponding to datasets below the given threshold (e.g., low feature count) and the rightmost plot representing datasets above the threshold. The boxplots illustrate the distribution of model performance for each AutoML framework, with Auto-Keras, Auto-Sklearn, H2O, and TPOT shown along the x-axis. The line inside the box represents the median, while the interquartile range (IQR) is the difference between the first and third quartiles. Whiskers extend to data within 1.5 times the IQR, with outliers depicted by circles. Finally, the analysis of the data segment with missing percentage of values was omitted for multiclass classification, as all datasets evaluated contained no missing entries.

3.1 Binary classification results

Notably, heightened variability and a decline in performance were observed with increasing percentage of categorical features. This trend stemmed from variations in the categorical value encoding methods. Across smaller datasets (less than 10,000 samples), all frameworks demonstrated consistent performance, with H2O and TPOT achieving the highest AUC scores and lowest variability. As dataset size increased, Auto-Keras and Auto-Sklearn showed heightened variability and diminished predictive power.

A decline in performance was observed for datasets with fewer than 50 features, potentially due to feature expansion during encoding. For instance, the "dresses-sales" dataset initially contained 13 features but expanded to 155 after encoding. This increase led to diminished performance across all AutoML frameworks. However, due to only 13% of datasets exceeding the 50-feature threshold, a comprehensive comparison of feature variability is limited.

AutoML frameworks performed better and showed reduced variability when dataset size exceeded 50 times the feature count, likely due to improved predictive capabilities with higher shape ratios. Auto-Keras, H2O, and TPOT maintained consistent performance across shape ratios, while Auto-Sklearn exhibited increased variability.

Class imbalance and high proportions of missing values challenged most frameworks, although the predominance of balanced datasets limited a comprehensive variability comparison. Specifically, in datasets with a high missing

percentage, Auto-Keras showed inferior performance, whereas other frameworks maintained relatively consistent performance levels.

Concerning ***H1***, the results support this hypothesis, with H2O demonstrating superior average performance and the highest AUC values, especially under diverse dataset conditions involving a high percentage of categorical values, large sample sizes (over 10,000), and high feature counts. This performance advantage likely stems from H2O's unique technical design, which integrates stacked ensembles and random search for hyperparameter tuning (LeDell & Poirier, 2020; Schmitt, 2023). Therefore, H2O's layered approach leverages the strengths of multiple model types providing robustness across diverse dataset characteristics (Truong et al., 2019; Gijsbers et al., 2023). Finally, H2O's automated preprocessing, including missing value handling, supports stability across datasets where other frameworks might struggle.

The only exception to this trend occurred in datasets with low class imbalance, where H2O's performance was similar to other frameworks, showing no marked advantage. To improve H2O's performance specifically in scenarios with low class imbalance in binary classification tasks, a few technical adjustments could be considered. One approach is to incorporate adaptive resampling techniques, like SMOTE (Synthetic Minority Over-sampling Technique), during training to address low class imbalance more effectively, ensuring minority classes are better represented (Khan et al., 2023). Another potential improvement is to fine-tune the hyperparameters of individual models within the ensemble, focusing on models particularly sensitive to imbalance, such as decision trees, which may be adjusted to handle rare events more robustly (Eldeeb et al., 2023). Finally, leveraging a weighted ensemble approach that assigns greater weight to classifiers excelling with imbalanced data can optimize predictions for minority classes without compromising overall performance (Omar et al., 2023).

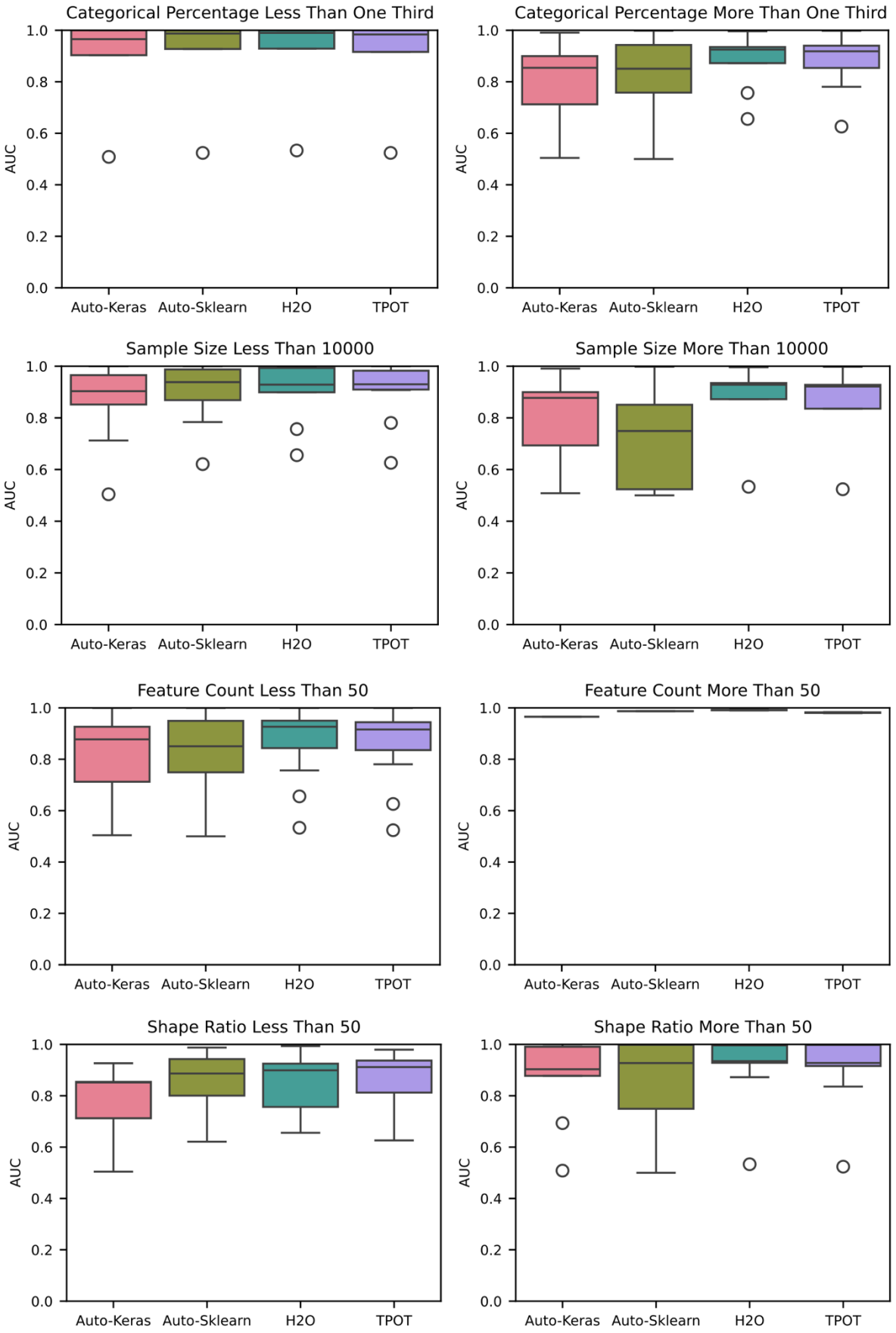
3.2 Multiclass classification results

In the multiclass classification tests, TPOT failed to produce results for the connect-4 dataset within the allocated time, while Auto-Sklearn exhibited below-average performance. To minimize bias, these outcomes were excluded from the figures. Additionally, H2O and Auto-Keras failed to complete 6% of experiments within the 15-minute limit, and these results were also disregarded.

For datasets with high categorical feature percentages, H2O outperformed other frameworks, while datasets with low categorical percentages showed comparable results across frameworks, except for slightly lower accuracy in Auto-Keras. Performance across different sample sizes revealed no consistent trend: Auto-Keras performed similarly across smaller and larger datasets, though its variability increased with larger datasets. Auto-Sklearn performed better with sample sizes over 10,000 but completed only about half of these experiments. In contrast, H2O excelled with smaller datasets (under 10,000), suggesting it is better optimized for smaller data volumes in multiclass classification, while TPOT showed improved performance with larger sample sizes but also completed only half of these experiments.

In the low feature count segment (fewer than 50 features), H2O and Auto-Sklearn demonstrated consistent results, while Auto-Keras and TPOT had slightly lower accuracy and higher variability. For datasets with more than 50 features, Auto-Sklearn, H2O, and TPOT maintained high and stable accuracy, likely due to their robust handling of large feature spaces through ensemble methods and advanced optimization techniques (Gijsbers et al., 2023; Schmitt, 2023; Truong et al., 2019). In contrast, Auto-Keras showed lower accuracy and greater variability, potentially due to its reliance on neural networks and Bayesian optimization, which may struggle with high-dimensional spaces lacking ensemble or meta-learning strategies (Jin et al., 2023; Zöller & Huber, 2021; He et al., 2021).

Shape ratio did not significantly impact performance, as most frameworks produced similar results across both shape ratio segments. In analyzing class imbalance, Auto-Keras showed minimal variability, performing similarly across high and low imbalance levels. Auto-Sklearn and TPOT performed better with low class imbalance, likely benefiting from ensemble techniques that thrive with balanced classes (Gijsbers et al., 2023; Zöller & Huber, 2021). H2O's stacked ensemble method provided stability with low imbalance but struggled under high imbalance, suggesting it could benefit from SMOTE or similar methods for enhanced handling of minority classes (Dablain et al., 2022), aligned with the insights for binary classification.


Categorical Percentage Less Than One Third
Categorical Percentage More Than One Third
Sample Size Less Than 10000
Sample Size More Than 10000
Feature Count Less Than 50
Feature Count More Than 50
Shape Ratio Less Than 50
Shape Ratio More Than 50
AUC
0.0
0.2
0.4
0.6
0.8
1.0
Auto-Keras
Auto-Sklearn
H2O
TPOT

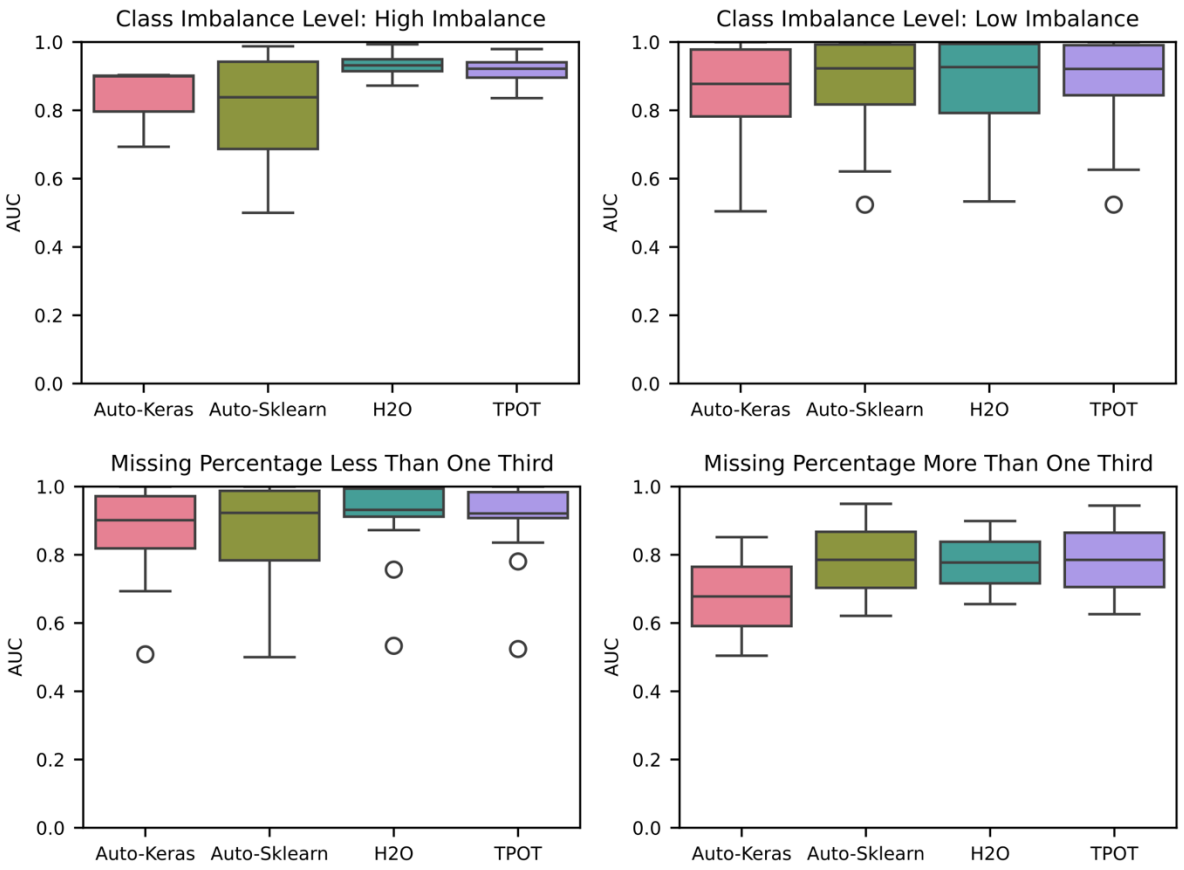
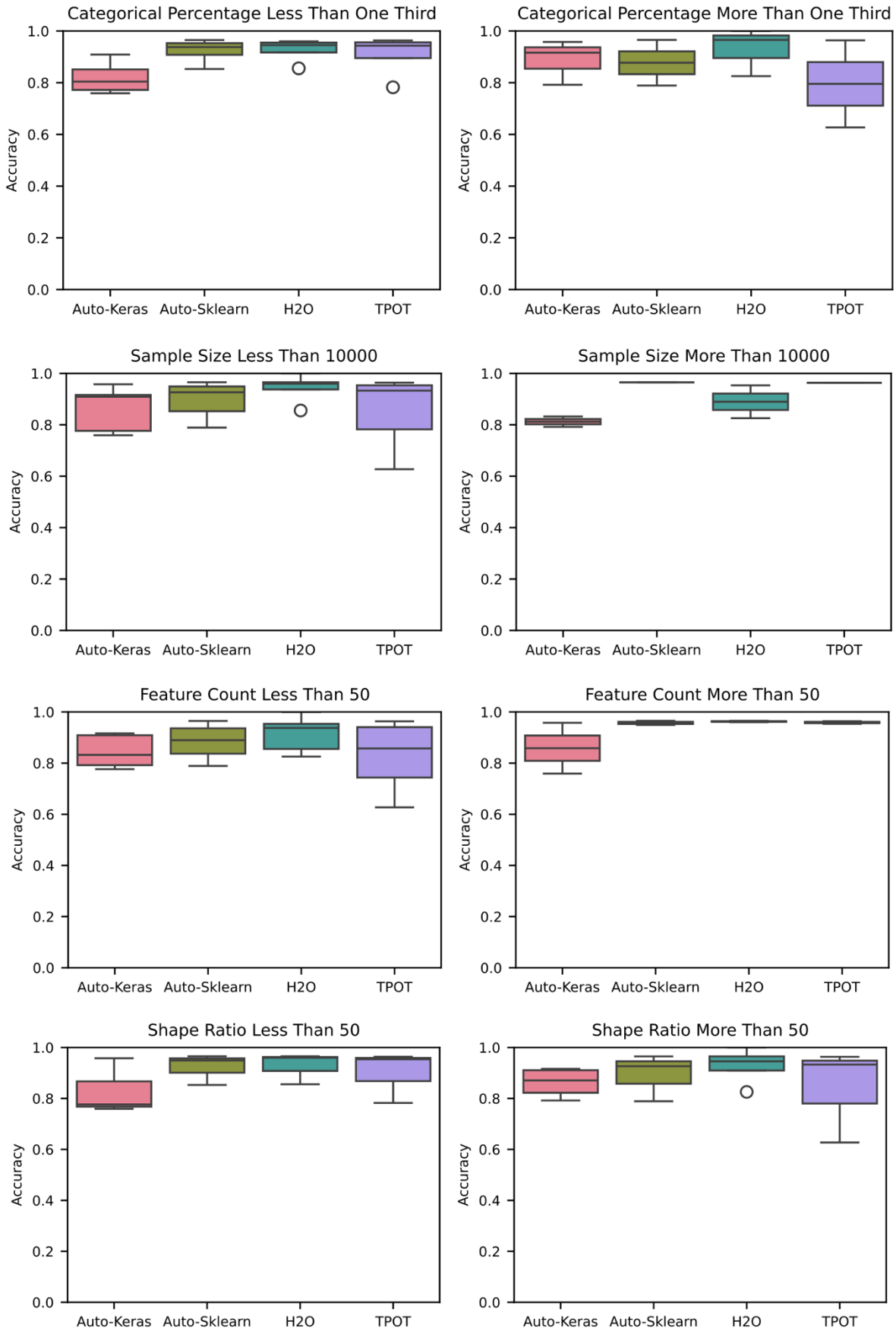


Figure 1. *Binary classification results.*

The analysis of multiclass classification results provides mixed support for ***H2***. Auto-Sklearn demonstrated strong overall performance but did not consistently outperform H2O and Auto-Keras, particularly in datasets with high categorical percentages and extensive feature counts. In high categorical percentage segments, Auto-Sklearn's performance was comparable to H2O and Auto-Keras, counter to the hypothesis of Auto-Sklearn's superiority (Balaji & Allen, 2018; Eldeeb et al., 2023). Additionally, in datasets with more than 50 features, Auto-Sklearn's results closely aligned with H2O and TPOT, challenging its anticipated advantage in high feature count scenarios (Gijsbers et al., 2023; Zöller & Huber, 2021).

While Auto-Sklearn's meta-learning and ensemble techniques aim to optimize performance across diverse multiclass tasks, H2O demonstrated equivalent or better results in several data segments, particularly with smaller datasets (under 10,000 samples) (Schmitt, 2023; Truong et al., 2019). These observations suggest that while Auto-Sklearn's architecture is effective, H2O's ensemble-based design may offer it a competitive edge in handling multiclass tasks under various dataset conditions (LeDell & Poirier, 2020).

To enhance Auto-Sklearn's competitiveness in multiclass classification, particularly with high categorical percentages and feature counts, refining its meta-learning component to focus on configurations suitable for complex categorical structures could improve search efficiency (Feurer et al., 2021). Integrating more advanced feature selection mechanisms would help manage high-dimensional categorical data, supporting stability (Feurer et al., 2015). Additionally, adopting a more diverse ensemble strategy could improve performance in complex multiclass scenarios by aligning Auto-Sklearn's capabilities with H2O's robust ensemble techniques (LeDell & Poirier, 2020).


Categorical Percentage Less Than One Third
Categorical Percentage More Than One Third
Sample Size Less Than 10000
Sample Size More Than 10000
Feature Count Less Than 50
Feature Count More Than 50
Shape Ratio Less Than 50
Shape Ratio More Than 50
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
Auto-Keras
Auto-Sklearn
H2O
TPOT

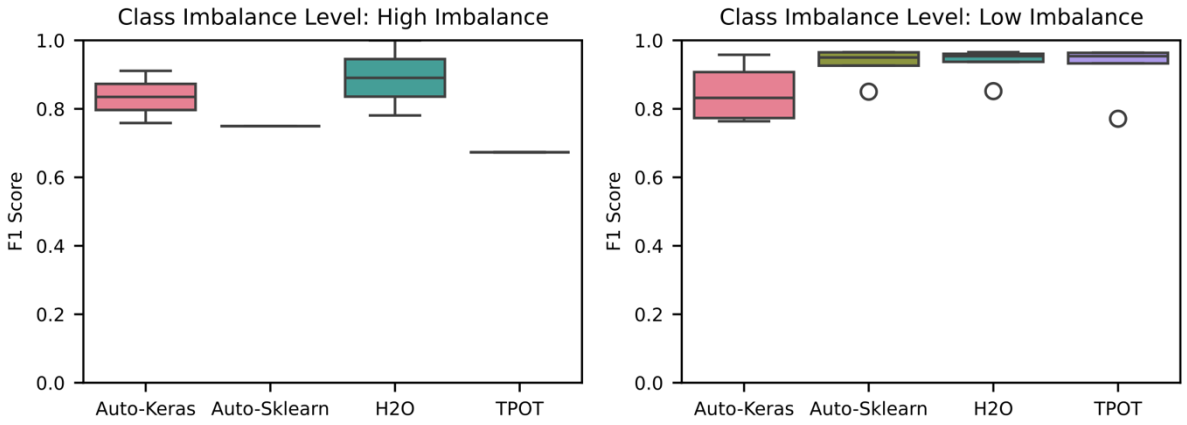


Figure 2. *Multiclass classification results.*

3.3 Cross-classification effects of dataset characteristics

Following the binary and multiclass classification results, this section addresses the cross-classification effects of dataset characteristics in predictive analytics, focusing on hypotheses *H3* and *H4*.

H3 is partially supported as higher feature counts and low categorical proportions enhanced performance, while effects of sample size and shape ratio were mixed, as detailed below. Higher feature counts led to improved performance and reduced variability across most frameworks, particularly H2O, Auto-Sklearn, and TPOT, likely due to their ensemble and feature selection mechanisms (Gijsbers et al., 2023; Truong et al., 2019). AutoKeras, however, showed consistent results across low and high feature counts, possibly due to its reliance on neural network structures, which could benefit from enhanced feature selection (Feurer et al., 2015). Regarding sample size, the results were inconsistent. Binary classification showed better performance with smaller datasets, contrary to expectations, with only H2O maintaining stability across sizes, likely due to its adaptive search and ensemble strategies (LeDell & Poirier, 2020). In multiclass classification, Auto-Sklearn and TPOT performed better with larger sample sizes, while H2O and AutoKeras showed higher accuracy with smaller samples, indicating possible scalability constraints for larger multiclass data (Schmitt, 2023; Balaji & Allen, 2018). For categorical proportions, low proportions (<33%) supported stable performance across frameworks, except AutoKeras, which showed similar results regardless of categorical density, suggesting a need for enhanced categorical handling (Jin et al., 2023). Shape ratio influenced performance in binary classification, with high shape ratios (>50) leading to more stable outcomes, except for Auto-Sklearn, which showed increased variability - likely due to its meta-learning approach's sensitivity to class distribution variations (Feurer et al., 2021; Eldeeb et al., 2023).

H4 is also partially supported. High class imbalance generally reduced AutoML performance, but certain frameworks demonstrated unexpected stability or improvement, and missing values clearly degraded accuracy. In multiclass classification, high class imbalance reduced accuracy across frameworks, aligning with prior studies that identify imbalanced classes as challenging due to bias toward majority classes (Tuggener et al., 2019; Eldeeb et al., 2023). However, binary classification results diverged: frameworks achieved high AUC scores with low imbalance but showed substantial variability, indicating unstable performance under minimal class imbalance. Interestingly, H2O and TPOT demonstrated more consistent performance with high imbalance in binary classification, likely due to their ensemble approaches which can better aggregate diverse predictions (LeDell & Poirier, 2020; Truong et al., 2019). Auto-Sklearn performed better with low imbalance, leveraging meta-learning to optimize configurations for balanced data, whereas Auto-Keras showed higher but less stable scores with high imbalance (Feurer et al., 2015). For binary classification, datasets with more than one-third missing values consistently showed degraded performance across AutoML frameworks, reinforcing the importance of preprocessing to minimize missing data. This outcome suggests that practitioners should prioritize data quality and completeness to maximize AutoML effectiveness, as frameworks are particularly sensitive to large portions of missing data across classification types. This analysis was limited to binary classification, as multiclass classification datasets contained no missing entries.

4 DISCUSSION

The research provides an up-to-date evaluation of AutoML framework performance, focusing on binary and multiclass classification tasks within predictive analytics. In addition to determining which framework excels across various data characteristics, the study addresses the technical limitations contributing to each framework's performance variability and identifies targeted improvements. The findings reveal that H2O outperformed other frameworks in both classification tasks, especially in datasets with high categorical proportions and feature counts, with comparable performance only in cases of low class imbalance. To further enhance H2O's accuracy in these imbalanced conditions, adaptive resampling techniques like SMOTE could be integrated. Especially for multiclass classification, TPOT and Auto-Keras lagged behind in achieving optimal performance for small sample size datasets, and a shape ratio of less than 50 segments. Additionally, specific limitations in binary classification, such as high performance variability of Auto-Sklearn and Auto-Keras with large sample sizes and low feature count datasets, highlight areas for framework enhancement. The technical analyses suggest that adaptive feature selection, optimized handling of categorical variables, and ensemble configurations could improve the consistency and efficiency of these frameworks.

The study contributes to the existing discussion on AutoML framework performance. It provides an up-to-date evaluation across binary and multiclass classification tasks and extends it into the field of predictive analytics. It demonstrates that no single AutoML framework outperforms others across all data segments. While H2O exhibited higher average performance for both classification types, the research reveals nuanced performance variations based on specific dataset attributes. This builds upon benchmark studies by Gijsbers et al. (2019), Truong et al. (2019), and Eldleeb et al. (2023). Furthermore, consistent with the findings of Gijsbers et al. (2019) and Truong et al. (2019), the study identifies weaknesses in AutoML frameworks when handling imbalanced datasets and with high categorical percentages, while also proposing targeted technical improvements, such as adaptive resampling, to address these limitations. Additionally, by building upon the research of Gijsbers et al. (2019) and Zöller & Huber (2021), this study uncovers further technical nuances in the performance of H2O and Auto-Keras in multiclass classification tasks, such as Auto-Keras's improved accuracy when handling datasets with a higher proportion of categorical features. The observed divergence in performance between H2O and Auto-Keras in multiclass classification tasks contrasts with previous findings by Eldleeb et al. (2019), Zöller & Huber (2021), and Conrad et al. (2022), highlighting how framework performance evolves with changes in model configurations and technological advancements. Finally, by exploring areas for technical enhancement within frameworks, such as improving H2O's handling of low class imbalance through resampling techniques, the study emphasizes the need for ongoing development in AutoML frameworks to meet diverse data challenges. Understanding these factors' impact on AutoML frameworks is vital for guiding future research on machine learning automation.

The research underscores several managerial implications. Firstly, it highlights AutoML's ability to increase access to machine learning tools across varying levels of expertise. However, while these frameworks reduce the technical barriers to ML adoption, their reliance on black-box optimization and automated decision-making raises concerns about interpretability, model bias, and the potential for misapplication by non-expert users (Barbudo et al., 2023, Eldeeb et al., 2023; Schmitt, 2023). Organizations should carefully consider the trade-offs between automation and the need for human oversight. Additionally, the research insights into AutoML framework performance under varying data complexities aid practitioners in selecting suitable frameworks for different applications (Halvari et al., 2020). This study also underscores the limitations of current AutoML systems in handling complex datasets, emphasizing the ongoing necessity of human expertise in data preprocessing (Barbudo et al., 2023). Finally, for AutoML framework developers, the technical evaluation and improvement suggestions - such as incorporating adaptive resampling techniques - provide actionable strategies for enhancing framework reliability in challenging data scenarios, like low class imbalance or high categorical proportions.

This study has certain limitations that influence the generalizability of its findings. The focus solely on supervised classification tasks with tabular data limits broader applicability, prompting future studies to explore AutoML frameworks efficacy in unsupervised, self-supervised or reinforcement learning, and unstructured data scenarios within predictive analytics. Additionally, the manual selection of datasets could introduce sampling biases, emphasizing the need for a more diverse dataset selection process (Barbudo et al., 2023). The incomplete analysis of certain datasets for multiclass classification, such as the connect-4 dataset, resulted in a smaller sample size, limiting the robustness of the research findings.

Regarding future research, scholars could test the recommendations presented in this study to improve AutoML framework performance across various data segments. Additionally, the questions could be addressed on how the data preprocessing capabilities can be enhanced to improve AutoML framework performance (Bilal et al., 2022) and how hybrid approaches that integrate AutoML with domain-driven feature engineering and model selection to balance automation with interpretability can be established. Finally, exploring deep reinforcement learning in predictive analytics present promising research directions in business analytics (Bertsimas & Kallus, 2019).

5 CONCLUSION

This study examined the performance of four AutoML frameworks across diverse datasets for predictive analytics tasks, highlighting that no single framework consistently outperforms others. Future research should explore ways to enhance AutoML's adaptability and assess its role in scaling machine learning applications. While AutoML can help bridge the supply-demand gap for ML expertise, it is not without challenges. The black-box nature of many frameworks poses interpretability concerns, particularly in regulated industries where transparency is critical. To ensure effective deployment, organizations should complement AutoML adoption with workforce upskilling and robust oversight to mitigate potential risks.

ADDITIONAL INFORMATION AND DECLARATIONS

Acknowledgments: I would like to express my sincere gratitude to Han-Yin Chen who supported this study, mainly within the AutoML framework setup, and training. Additionally, I would like to thank Prof. Dr. Theresa Treffers for her review and support throughout the research process.

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: The author confirms being the sole contributor of this work.

Statement on the Use of Artificial Intelligence Tools: The author acknowledges the support provided by an artificial intelligence (AI) tool, specifically OpenAI's ChatGPT 4.0 model, which was utilized solely for refining and improving the manuscript to enhance its clarity and flow. It is important to note that the AI model was not used for any empirical research purposes.

Data Availability: The data that support the findings of this study are available from the corresponding author.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J. M., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, Ł., Kudlur, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*. https://doi.org/10.48550/arXiv.1603.04467

Asniar, & Surendro, K. (2019). Predictive analytics for predicting customer behavior. In *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)* (pp. 230–233). IEEE. https://doi.org/10.1109/ICAIIT.2019.8834571

Balaji, A., & Allen, A. (2018). Benchmarking automatic machine learning frameworks. *arXiv:1808.06492*. https://doi.org/10.48550/arXiv.1808.06492

Barbudo, R., Ventura, S., & Romero, J. R. (2023). Eight years of AutoML: Categorisation, review and trends. *Knowledge and Information Systems*, 65(12), 5097–5149. https://doi.org/10.1007/s10115-023-01935-1

Bell, S. (2009). Experimental design. In *International Encyclopedia of Human Geography* (pp. 672–675). Elsevier. https://doi.org/10.1016/B978-008044910-4.00431-4

Bertsimas, D., & Kallus, N. (2019). From predictive to prescriptive analytics. *Management Science*, 66(3), 1025–1044. https://doi.org/10.1287/mnsc.2018.3253

Bilal, M., Ali, G., Iqbal, M. W., Anwar, M., Malik, M. S. A., & Kadir, R. A. (2022). Auto-PreP: Efficient and automated data preprocessing pipeline. *IEEE Access*, 10, 107764–107784. https://doi.org/10.1109/ACCESS.2022.3198662

Bischl, B., Casalicchio, G., Feurer, M., Gijsbers, P., Hutter, F., Lang, M., Mantovani, R. G., Van Rijn, J. N., & Vanschoren, J. (2017). OpenML Benchmarking Suites. *arXiv:1708.03731*. https://doi.org/10.48550/arXiv.1708.03731

Bischl, B., Casalicchio, G., Feurer, M., Gijsbers, P., Hutter, F., Lang, M., Mantovani, R. G., Van Rijn, J. N., & Vanschoren, J. (2019). OpenML Benchmarking Suites. *arXiv:1708.03731*. https://doi.org/10.48550/arXiv.1708.03731

Bischl, B., Casalicchio, G., Feurer, M., Gijsbers, P., Hutter, F., Lang, M., Mantovani, R. G., Van Rijn, J. N., & Vanschoren, J. (2021). OpenML benchmarking suites. In *35th Conference on Neural Information Processing Systems*. NeurIPS.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Chollet, F. (2015). *Keras*. https://keras.io

Conrad, F., Mälzer, M., Schwarzenberger, M., Wiemer, H., & Ihlenfeldt, S. (2022). Benchmarking AutoML for regression tasks on small tabular data in materials design. *Scientific Reports*, 12(1), Article number 19350. https://doi.org/10.1038/s41598-022-23327-1

Coors, S., Schalk, D., Bischl, B., & Rügamer, D. (2021). Automatic Componentwise Boosting: An Interpretable AutoML System. *arXiv:2109.05583*. https://arxiv.org/abs/2109.05583

Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9), 6390–6404. https://doi.org/10.1109/TNNLS.2021.3136503

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2019). A survey on ensemble learning. *Frontiers of Computer Science*, 14(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z

Eldeeb, H., Maher, M., Elshawi, R., & Sakr, S. (2023). AutoMLBench: A comprehensive experimental evaluation of automated machine learning frameworks. *Expert Systems with Applications*, 243, 122877. https://doi.org/10.1016/j.eswa.2023.122877

Feldman, D., Schmidt, M., & Sohler, C. (2020). Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3), 601–657. https://doi.org/10.1137/18M1209854

Fels, A. E., Mandi, L., Kammoun, A., Ouazzani, N., Monga, O., & Hbid, M. L. (2023). Artificial intelligence and wastewater treatment: A global scientific perspective through text mining. *Water*, 15(19), Article 3487. https://doi.org/10.3390/w15193487

Ferreira, L., Pilastri, A., Martins, C. M., Pires, P. M., & Cortez, P. (2021). A comparison of AutoML tools for machine learning, deep learning and XGBoost. In *2021 International Joint Conference on Neural Networks (IJCNN)*, (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN52387.2021.9534091

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, 28. NIPS.

Feurer, M., Van Rijn, J. N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., & Hutter, F. (2021). OpenML-Python: An extensible Python API for OpenML. *Journal of Machine Learning Research,* 22, 1–5.

Ge, P. (2020). Analysis on approaches and structures of automated machine learning frameworks. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)* (pp. 474–477). IEEE. https://doi.org/10.1109/CISCE50729.2020.00106

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019). An open source AutoML benchmark. *arXiv:1907.00909*. https://doi.org/10.48550/arXiv.1907.00909

Gijsbers, P., Bueno, M. L., Coors, S., LeDell, E., Poirier, S., Thomas, J., ... & Vanschoren, J. (2024). AMLB: an AutoML Benchmark. *Journal of Machine Learning Research*, 25, 1–65.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *36th Conference on Neural Information Processing Systems (NeurIPS 2022).* NeurIPS.

Grover, V., Chiang, R. H. L., Liang, T., & Zhang, D. (2018). Creating strategic business value from big data analytics: A research framework. *Journal of Management Information Systems*, 35(2), 388–423. https://doi.org/10.1080/07421222.2018.1451951

H2O.ai. (2023). H2O AutoML. https://h2o.ai/platform/h2o-automl/

Halvari, T., Nurminen, J. K., & Mikkonen, T. (2020). Testing the robustness of AutoML systems. *arXiv:2005.02649*. https://doi.org/10.4204/EPTCS.319.8

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). K-fold cross-validation. In *An introduction to statistical learning with applications in Python* (1st ed., pp. 206–208). Springer.

Jin, H., Chollet, F., Song, Q., & Hu, X. (2023). AutoKeras: An AutoML library for deep learning. *Journal of Machine Learning Research*, 24(6), 1–6.

Jolly, K. (2018). *Machine learning with scikit-learn quick start guide: Classification, regression, and clustering techniques in Python*. Packt Publishing.

Kaggle. (2023). Kaggle. https://www.kaggle.com/

Karras, A., Karras, C., Schizas, N., Avlonitis, M., & Sioutas, S. (2023). Automl with bayesian optimizations for big data management. *Information*, 14(4), Article 223. https://doi.org/10.3390/info14040223

Kavanagh, P. (2004). The open source definition. In Elsevier eBooks (pp. 321–322).

Khan, A. A., Dwivedi, P., Mugde, S., Sajidha, S., Sharma, G., & Soni, G. (2023). Toward automated machine learning for genomics: Evaluation and comparison of state-of-the-art AutoML approaches. In *Data Science for Genomics*, (pp. 129–152). Elsevier. https://doi.org/10.1016/B978-0-323-98352-5.00017-3

Kelly, M., Longjohn, R., & Nottingham, K. (2023). The UCI machine learning repository. https://archive.ics.uci.edu

Lampert, C. H., Nickisch, H., & Harmeling, S. (2013). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 453–465. https://doi.org/10.1109/TPAMI.2013.140

LeDell, E., & Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. In *7th ICML Workshop on Automated Machine Learning*. ICML. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf

Lee, C. S., Cheang, P. Y. S., & Moslehpour, M. (2022). Predictive analytics in business analytics: Application of Decision Tree in Business Decision Making. *Advances in Decision Sciences*, 26(1), 1–29. https://doi.org/10.47654/v26y2022i1p1-30

Lenkala, S., Marry, R., Gopovaram, S. R., Akinci, T. C., & Topsakal, O. (2023). Comparison of automated machine learning (AutoML) tools for epileptic seizure detection using electroencephalograms (EEG). *Computers*, 12(10), Article 197. https://doi.org/10.3390/computers12100197

Mathew, J., Kshirsagar, R., Abidin, D. Z., Griffin, J., Kanarachos, S., James, J., Alamaniotis, M., & Fitzpatrick, M. E. (2023). A comparison of machine learning methods to classify radioactive elements using prompt-gamma-ray neutron activation data. *Scientific Reports*, 13(1), Article 9948. https://doi.org/10.1038/s41598-023-36832-8

Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C. R., & Moore, J. H. (2016). Automating biomedical data science through tree-based pipeline optimization. In *Applications of Evolutionary Computation*, (pp. 123–137). Springer. https://doi.org/10.1007/978-3-319-31204-0_9

Omar, I., Khan, M., Starr, A., & Abou Rok Ba, K. (2023). Automated prediction of crack propagation using H2O AutoML. *Sensors*, 23(20), Article 8419. https://doi.org/10.3390/s23208419

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Pfisterer, F., Thomas, J., & Bischl, B. (2019). Towards human centered AutoML. *arXiv:1911.02391*. https://doi.org/10.48550/arXiv.1911.02391

Pio, P. B., Rívolli, A., De Carvalho, A. C. P. L. F., & García, L. (2023). A review on preprocessing algorithm selection with meta-learning. *Knowledge and Information Systems*, 66(1), 1–28. https://doi.org/10.1007/s10115-023-01970-y

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in Python: main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4), Article 193. https://doi.org/10.3390/info11040193

Ray, P., Reddy, S. S., & Banerjee, T. (2021). Various dimension reduction techniques for high dimensional data analysis: a review. *Artificial Intelligence Review*, 54(5), 3473–3515. https://doi.org/10.1007/s10462-020-09928-0

Salih, A., Raisi-Estabragh, Z., Boscolo Galazzo, I., Radeva, P., Petersen, S. E., Menegaz, G., & Lekadir, K. (2023). A perspective on explainable artificial intelligence methods: SHAP and LIME. *arXiv preprint arXiv:2305.02012*. https://arxiv.org/abs/2305.02012

SAS Institute. (2022). How to solve the data science skills shortage. *SAS Institute*. https://www.sas.com/content/dam/SAS/documents/technical/education/en/solve-data-science-skills-shortage-uk-113039.pdf

Schmitt, M. (2023). Automated machine learning: AI-driven decision making in business analytics. *Intelligent Systems with Applications*, 18, 200188. https://doi.org/10.1016/j.iswa.2023.200188

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4), 687–719. https://doi.org/10.1142/S0218001409007326

Tan, J., Yang, J., Wu, S., Chen, G., & Zhao, J. (2021). A critical look at the current train/test split in machine learning. *arXiv:2106.04525*. https://doi.org/10.48550/arXiv.2106.04525

The pandas development team. (2020). Pandas-dev/pandas: pandas (2.2.1) [Software]. *Zenodo*. https://doi.org/10.5281/zenodo.3509134

Topsakal, O., & Akıncı, T. Ç. (2023). Classification and regression using automatic machine learning (AutoML) – Open source code for quick adaptation and comparison. *Balkan Journal of Electrical and Computer Engineering*, 11(3), 257–261. https://doi.org/10.17694/bajece.1312764

Truong, A., Walters, A., Goodsitt, J., Hines, K. E., Bruss, C. B., & Farivar, R. (2019). Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 1471–1479). IEEE. https://doi.org/10.1109/ICTAI.2019.00209

Tuggener, L., Amirian, M., Rombach, K., Lörwald, S., Varlet, A., Westermann, C., & Stadelmann, T. (2019). Automated machine learning in practice: State of the art and recent results. In *2019 6th Swiss Conference on Data Science (SDS)* (pp. 31–36). IEEE. https://doi.org/10.1109/SDS.2019.00-11

Vanschoren, J., Van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2), 49–60. https://doi.org/10.1145/2641190.2641198

Velmurugan, M., Ouyang, C., Moreira, C., & Sindhgatta, R. (2020). Evaluating explainable methods for predictive process analytics: A functionally-grounded approach. *arXiv preprint arXiv:2012.04218*. https://arxiv.org/abs/2012.04218

Zöller, M. A., & Huber, M. F. (2021). Benchmark and survey of automated machine learning frameworks. *Journal of Artificial Intelligence Research*, 70, 409–472. https://doi.org/10.1613/jair.1.11854

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951