© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717446

AUTOML - A Desktop Application for Algorithm Selection and Optimization

SATHYA S[1], ABDUL RASHID ABDUL GAFFAR[2], AMEER FAHATH[3], VISHAL S[4]
[1]*Assistant Professor Department of Artificial Intelligence & Data Science Science Dhanalakshmi Srinivasan University, Tichy, India*
[2,3,4]*Department Of Artificial Intelligence & Data Dhanalakshmi Srinivasan University, Trichy, India*

Abstract- ***Automated Machine Learning (AutoML) has emerged as a practical approach to lowering the entry barrier for applying machine learning techniques by automating repetitive and technically complex steps such as data preprocessing, model selection, and evaluation. This paper presents AutoMLApp v1, a desktop-based automated machine learning application developed as a student-led applied research project. The system is implemented in Python, with a graphical user interface (GUI) built using PyQt5 and a machine learning backend based on the Scikit-learn library. The current version focuses on supervised classification tasks, providing automated dataset handling, model training across multiple algorithms, performance comparison using standard evaluation metrics, and learning curve visualization for training behavior analysis. The system architecture is designed with extensibility in mind, allowing future integration of regression tasks and hyperparameter optimization. Experimental results demonstrate that AutoMLApp v1 can effectively identify suitable classification models for user-provided datasets, making it a useful educational and prototyping tool. This work emphasizes practical system design, applied experimentation, and learning-oriented contributions rather than claiming state-of-the-art performance.***

Keywords - Automated Machine Learning, AutoML, PyQt5, Scikit-learn,Classification, Desktop Application, Educational Systems

I. INTRODUCTION

Machine learning (ML) has become a foundational technology across domains such as healthcare, finance, manufacturing, and intelligent systems. Despite its widespread adoption, effective application of ML often requires substantial expertise in data preprocessing, algorithm selection, model evaluation, and iterative experimentation. Automated Machine Learning (AutoML) aims to reduce this complexity by automating key stages of the ML pipeline, thereby improving accessibility and productivity.

Most existing AutoML solutions are cloud-based or library-centric, such as Auto-sklearn or TPOT, which require programming proficiency and are less suitable for beginners or rapid experimentation in offline environments. In academic settings, especially at the undergraduate level, there is a need for interactive, transparent, and desktop-based systems that allow students to understand ML workflows while benefiting from automation.

This paper introduces AutoMLApp v1, a desktop-based AutoML system developed by a third-year B.Tech student in Artificial Intelligence and Data Science. The application focuses on classification tasks and provides an end-to-end automated workflow through a user-friendly GUI. The primary objectives of this work are:

1. To design a modular and extensible AutoML architecture suitable for desktop environments.
2. To implement an automated classification pipeline using scikit-learn.
3. To provide performance comparison and learning curve visualization for educational analysis.

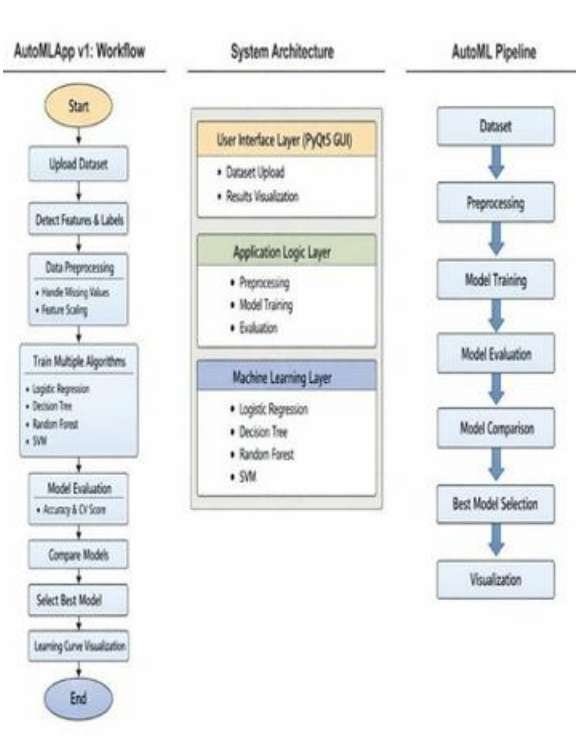
II. RELATED WORK

AutoML research has gained significant attention in recent years. Auto-sklearn integrates Bayesian optimization with meta-learning to automate algorithm selection and hyperparameter tuning. TPOT employs genetic programming to optimize ML

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717446

pipelines, while H2O AutoML provides an enterprise-oriented automated framework.

While these systems are powerful, they are primarily code-driven and often operate as black-box solutions, limiting their educational transparency. GUI-based ML tools such as WEKA provide visual interaction but offer limited automation and modern ML pipeline flexibility. AutoMLApp v1 positions itself between these approaches by combining automation with user interaction and interpretability, specifically targeting student learning and applied experimentation



III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall System Design

AutoMLApp v1 follows a modular architecture consisting of three primary layers:

- User Interface Layer: Handles user interaction, dataset upload and result visualization.
- Application Logic Layer: Manages preprocessing, model training and evaluation flow.
- Machine Learning Layer: Implements classification algorithms using scikit-learn.

This separation of concerns improves maintainability and enables future expansion

B. AutoML Workflow

The automated workflow implemented in AutoMLApp v1 consists of the following steps:

4. Dataset upload through the GUI.
5. Automatic detection of features and target labels.
6. Basic preprocessing, including handling missing values and feature scaling where applicable.
7. Training of multiple classification algorithms.
8. Evaluation using standard classification metrics.
9. Comparison and selection of the best-performing model.
10. Visualization of learning curves.

Learning curves are generated using Matplotlib to visualize model performance as a function of training set size. These plots help users analyze underfitting or overfitting behavior.

C. Classification and Regression Pipeline Design

Although v1 implements only classification, the system is conceptually designed to support both classification and regression. Separate pipelines are planned, each with task-specific preprocessing, algorithms, and evaluation metrics. This conceptual separation will be explicitly implemented in AutoMLApp v2

IV. IMPLEMENTATION DETAILS

A. PyQt5 GUI Architecture

The GUI is developed using PyQt5 and follows a signal-slot architecture. Key components include dataset upload dialogs, parameter display panels, model comparison tables, and visualization windows. The GUI is designed to abstract internal ML complexity while still exposing meaningful outputs to the user.

B. Scikit-Learn Model Integration

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717446

AutoMLApp v1 integrates multiple Scikit-learn classification algorithms, such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. Each model is trained using a consistent train-test split to ensure fair comparison. The modular design allows easy addition or removal of algorithms.

C. Improvements and Original Contributions Over the Base Paper

This section explicitly outlines how the proposed work AutoMLApp v1 extends and improves upon the base AutoML literature, and clarifies the novel contributions introduced in this IEEE paper.

1. Shift from Theoretical Survey to Applied System

The base paper primarily focuses on a theoretical and survey-oriented analysis of Automated Machine Learning

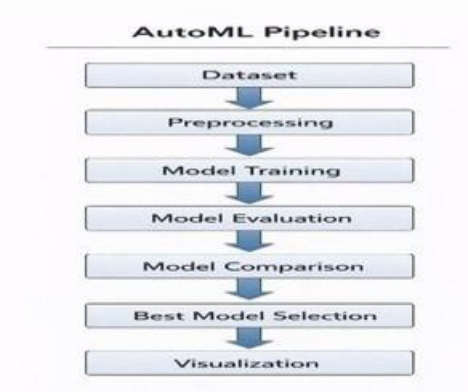
(AutoML), with emphasis on meta-learning frameworks, benchmark evaluations, and comparative performance analysis of existing AutoML systems. In contrast, this work presents AutoMLApp v1, a fully implemented and operational desktop-based AutoML system

Unlike prior studies that discuss AutoML concepts at an algorithmic or framework level, this paper demonstrates a practical realization of an end-to-end AutoML pipeline, enabling real dataset ingestion, model training, evaluation, and visualization through a unified application.

2. Desktop-Based GUI-Oriented AutoML System

Most existing AutoML solutions discussed in the base paper are either cloud-based or library-centric, requiring programming expertise and online infrastructure. AutoMLApp v1 introduces a desktop-based graphical user interface (GUI) built using PyQt5, allowing offline usage and interactive experimentation.

This design choice significantly improves accessibility for students and beginners by abstracting code-level complexity while preserving transparency of the machine learning workflow.



3. Educational and Learning-Oriented Positioning

While the base paper targets research-centric performance optimization, this work explicitly positions AutoMLApp v1 as an educational and prototyping tool. The system is designed to expose users to automated preprocessing steps, model comparison logic, and training behavior through learning curves. The paper intentionally avoids claiming state-of-the-art performance and instead emphasizes learning value, system design, and applied experimentation.

4. Explicit Modular System Architecture

AutoMLApp v1 introduces a clearly defined three-layer modular architecture consisting of: (1) User Interface Layer, (2) Application Logic Layer, and (3) Machine Learning Layer. This architectural separation is not detailed in the base paper and provides a maintainable and extensible foundation for future enhancements such as regression pipelines and hyperparameter optimization.

5. Concrete Implementation Details

Unlike the base paper, which remains largely conceptual, this work provides implementation-level details, including:

- PyQt5 signal-slot based GUI design.
- Integration of multiple scikit-learn classification models.
- Consistent train-test splitting for fair model comparison.
- Automated generation of learning curve visualizations using Matplotlib

6. Honest Limitations and Structured Future Work

The paper explicitly documents current limitations of AutoMLApp v1, such as restriction to classification tasks and absence of hyperparameter tuning. In contrast to generic future directions in the base paper, this work proposes a clear and structured roadmap for AutoMLApp v2, including:

- Separation of classification and regression pipelines.
- Integration of Grid Search and Random Search.
- Enhanced preprocessing and feature engineering.
- Improved scalability and robustness.

V. SUMMARY OF IMPROVEMENTS

In summary, this work improves upon the base paper by transitioning from a theoretical AutoML discussion to a practical, desktop-based, and educational AutoML system, supported by real implementation, experimentation, and system-level analysis. These contributions position AutoMLApp v1 as a distinct and original applied research effort suitable for undergraduate-level IEEE publication.

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

A. Classification Experiments

Experiments were conducted on standard tabular datasets provided by users. The datasets were split into training and testing subsets using a fixed ratio. No aggressive hyperparameter tuning was applied, as the goal was to evaluate baseline automated performance.

B. Model Comparison and Evaluation Metrics

Model performance was evaluated using accuracy and crossvalidation score. The system presents comparative results across algorithms and automatically identifies the best-performing model based on selected metrics. The results demonstrate that AutoMLApp v1 can reliably train and compare multiple models, providing reasonable baseline performance without manual intervention.

VII. FUTURE WORK

Future development will focus on AutoMLApp v2, which aims to:

11. Explicitly segregate classification and regression pipelines.
12. Integrate automated hyperparameter tuning techniques such as Grid Search and Random Search.
13. Add advanced preprocessing and feature engineering options.
14. Improve result visualization and reporting. Enhance scalability and robustness for larger datasets

VIII. CONCLUSION

This paper presented AutoMLApp v1, a desktop-based automated machine learning application developed as a studentled project. By combining PyQt5 for GUI development and Scikit-learn for ML backend functionality, the system provides an accessible and educational AutoML platform focused on classification tasks. While not intended to compete with advanced AutoML frameworks, AutoMLApp v1 demonstrates the feasibility and value of applied AutoML system development in academic settings. The planned enhancements in v2 aim to further extend its capabilities and practical relevance

REFERENCES

[1] F. Hutter, L. Kotthoff, and J. Vanschoren, Automated Machine Learning, Springer, 2019.

[2] M. Feurer et al., "Efficient and Robust Automated Machine Learning," Advances in Neural Information Processing Systems, 2015.

[3] R. Olson and J. Moore, "TPOT: A Tree-Based Pipeline Optimization Tool," AutoML Workshop, ICML, 2016.

[4] P. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[5] Riverbank Computing, "PyQt5 Documentation," 2023.

[6] A. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly Media, 2019