International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1857-1863
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0269

AI-Driven Decision Making in Business Analytics

Veerendeswari J[1], Keerthana Priya M[2], Shushmita P[3], Varsha S S[4]
[1]Head of the Dept, Dept. of IT, Rajiv Gandhi College of Engg. & Tech, Kirumampakkam, Puducherry, India.
[2,3,4]UG Scholar, Dept. of IT, Rajiv Gandhi College of Engg. & Tech, Kirumampakkam, Puducherry, India.
Email ID: *jveerendeswari _it@rgcet.edu.in[1], Keerthanapriya472004@gmail.com[2], shushmitap2@gmail.com[3], varshasuresh7426@gmail.com[4].*

Abstract

Machine learning has the potential to transform industries, but many small and medium-sized enterprises (SMEs) struggle with the technical demands of building optimized models. To solve this, we propose an user-friendly framework powered by Automated Machine Learning (AutoML) tools. TPOT helps automate the complex task of choosing the right algorithms and hyperparameter tuning their settings, while PyCaret simplifies data preprocessing tasks such as feature engineering, class imbalance handling, and encoding. and allows quick testing of different models. Together, these tools make the entire machine learning process faster and more accessible even for those with limited experience. In a manufacturing case study, our approach improved prediction accuracy and cut down both time and cost. This solution supports scalable AI adoption and helps SMEs benefit from the power of intelligent automation.

Keywords: *Auto ML; Feature engineering; Hyperparameter tuning; Machine Learning automation; Small and medium-sized enterprises*

1. Introduction

This paper traverses the interlaced landscape between ML and AutoML sharing the joint effort between them. Based on broad data sets, ML uses algorithms to identify patterns and make predictions and decisions. AutoML addresses this by automating critical but often painstaking tasks across the ML pipeline to improve the efficiency and accessibility.

This survey aims to explore data preprocessing strategies in depth, focusing on identifying effective methods and addressing existing gaps. The study investigates creative methods to address present-day preprocessing issues, aiming to inspire future innovations in this area. It aims to derive meaningful insights by blending intricate data preparation tasks with the progressive tools of AutoML, helping enhance the quality of data-based decisions in machine learning systems the rapidly changing machine learning field. The survey reviews key studies, including notable works like "DataAssist" and "REIN," each offering important perspectives that shape this research journey. The reviewed studies tackle key challenges like unbalanced datasets, hyperparameter tuning, and the growing importance of advanced feature engineering emphasizing the necessity for cohesive approaches that bridge raw data with high-performing machine learning models. the rapid growth of machine learning, the studies reviewed in this survey provide valuable direction for creating automated approaches to tackle intricate data preprocessing issues. The upcoming architectural framework aims to introduce innovative, end-to-end methods for streamlining data preparation and automating critical tasks. Rooted in thorough analysis, the survey emphasizes identifying current issues and proposing future-ready strategies. Every study highlights a distinct element of data preprocessing, and collectively, they contribute to a deeper understanding of how to enhance this vital phase in machine learning.

2. Literature Survey

The study titled "Runtime Prediction of Machine Learning Algorithms in Automated Systems" by Parijat Dube and Theodoros Salonidis [1] explores the vital role of tools like DataAssist in shaping the future of AutoML workflows, especially in data preparation and cleaning. Its features—including exploratory data analysis, visualization, anomaly detection, and preprocessing—serve professionals in sectors such as economics, business, and forecasting, where data quality is critical. DataAssist accelerates preprocessing, reducing the time involved by more than half and enabling efficient and reliable machine learning workflows. Overall, it emphasizes data-focused processes, a crucial area often overlooked in the AutoML pipeline. **In his ponder, Suraj Juddoo [2]** investigates the pivotal steps included in repairing information inside Electronic Wellbeing Records (EHR) within the setting of Enormous Information frameworks. The paper sheds light on the challenges of keeping up information quality, especially in healthcare, where exact and clean information is basic for creating important bits of knowledge. It highlights how vital the information repair organize is in guaranteeing in general information quality, particularly when working with enormous and complex datasets. One vital understanding is the need of clarity around how current information rebuilding apparatuses perform when connected to large-scale datasets. The study adopts a well-organized review combined with experimental analysis to evaluate how effective different data repair techniques are Evaluating these methods using a prototype developed from previous studies brings a practical edge to the research, making its findings more applicable to real-world scenarios. The findings reveal that none of the existing tools or algorithms fully meet the demands of Big Data, underscoring the complexity and ongoing hurdles in this area. These findings, paired with recommendations for enhancing data repair approaches, serve as a strong foundation for guiding upcoming research and innovation in the domain. **AutoML continues to grow in popularity, raising the challenge of selecting the best tool among many. Ribeiroa and Orzechowski [3]** They evaluate four widely-adopted frameworks—Auto-Sklearn, Auto-Sklearn 2, H2O AutoML, and TPOT—on synthetic datasets constructed with DIGEN. While the tools show comparable overall performance, the study highlights subtle variations influenced by the nature of the datasets and the evaluation metrics applied. These findings are useful to understand each algorithm’s strength and guide better decision making for AutoML selection. **A Comprehensive Framework for Comparing Data Cleaning Methods in Machine Learning Pipelines: REIN Harald Schoening, Mohamed Abdelaal, and Christian Hammacher [4]** emphasize that high-quality data is essential for effective machine learning (ML). Real-world datasets often suffer from issues like inconsistencies, missing values, and duplicates, which can significantly impact model performance. While many data cleaning methods exist, they are rarely evaluated based on their actual effect on ML outcomes. To address this, the authors introduce REIN1, a benchmarking framework designed to assess 38 different error detection and repair techniques. These methods are tested across 14 publicly available datasets using various ML models. REIN offers valuable insights to help researchers choose the most effective cleaning strategies for robust ML pipelines. **In AutoCure: An Automated and Configuration-Free Pipeline for Tabular Data Preparation in Machine Learning, Schoening, Koparde, and colleagues [5]** introduce AutoCure, a fully automated and configuration-free pipeline designed to simplify tabular data preparation in machine learning workflows. It tackles the often-time-consuming challenge of preparing clean and usable data especially important in fields like healthcare, finance, and autonomous systems. AutoCure boosts data quality by combining an advanced data augmentation module with a unique ensemble-based error detection approach. It strengthens the model’s performance by smartly increasing the portion of clean, reliable data through synthetic enhancements. Its smooth integration with popular open-source tools like Auto-sklearn, H2O, and TPOT makes machine learning more accessible, helping practitioners across industries build better models with less effort.

3. Proposed Methodology

The study involved several key steps such as gathering relevant information, preparing datasets, and evaluating the performance of the model [6].

3.1 Data Processing Module

The role of cleaning and preparing unprocessed data for analysis is an important step in the data process. One of the fundamental aspects is the processing of the decision value, and the methods such as tanks or deletion are used to eliminate the lack of information. Scaling is another important task, especially if variables are measured on a different scale to implant the effect of a specific function on the analysis. In addition, to convert high -quality input into numerical format, coding of category variables is required, which can handle machine learning algorithms. This process keeps data integrity and allows the selected analysis to effectively obtain information [7-10]. In general, thorough purification and preparation of unprocessed data contributes to the adoption of information based on information in various fields, forming the foundation of reliable and reliable data analysis.

3.2 AutoML Core Module

The central module that integrates these windows offers a consistent and well -organized process, and each step contributes to the final result. Automation of life cycle to develop models and provide optimized high -performance machine learning models. The configuration of the machine learning model is optimized to improve performance. Functional functions to inhibit the functional function of the conversion and selection of the model that identifies the link and data template. Choosing another important underground model will help you choose the algorithm algorithm or ensemble that is best suited for this task. The central module that integrates these windows offers a consistent and well -organized process, and each step contributes to the final result. Automation of life cycle to develop models and provide optimized high -performance machine learning models [12].

3.3 Categorical Variable Encoding Standardization Module

Work of guaranteeing a consistent coding of variable categories is important in classification in the context of regression and machine learning. Qualitative variable categories need to be transformed into numerical values to ensure compatibility with various algorithms. The responsible module addresses this by employing encoding methods that maintain consistency across tasks. Common techniques include label encoding and one-hot encoding, in which binary columns indicate each category, which provides a distinct number label for every category, and target encoding, where categories are encoded based on the mean of the target variable. The module that continues to implement such a coding method ensures that the machine learning model is guaranteed to receive uniform input representation, contributing to the accuracy and reliability of prediction in regression scenarios and categories. This order is needed to create a reliable and interpreted model that can effectively study the template in category functions.

3.4 User Interface (UI) Module

Convenient interface for users and it acts as a portal, and practitioners can freely interact with the Automl system. Its primary function is to provide an accessible platform where users can input their data, define relevant parameters, and visualize the results of the automated machine learning process. Integrated visualization tools within the interface enable users to analyze and comprehend the outcomes of the automated process, enhancing transparency and promoting an interactive experience [11]. The interface abstracts the complexities of the underlying AutoML algorithms, making it suitable for users of different skill levels. Intuitive design makes it easy to upload data sets, display preference for functional or functional technology development, and move easily depending on system functions. Overall, the user-friendly interface enhances the usability of the AutoML system, facilitating effective collaboration between machine learning practitioners and the automated system for streamlined model development.

3.5 Report Generation Module

The distribution module is crucial for ensuring the seamless integration of models produced by automated systems. ts primary function is to streamline the transition from model development to real-world applications [14-15]. This module often includes features for model versioning, allowing practitioners to track and manage different iterations of models. Additionally, it addresses scalability concerns, ensuring that the deployed models can handle varying workloads and adapt to changing data volumes. Moreover, monitoring capabilities are integrated to keep track of model performance in real-time, enabling timely interventions if issues arise. The main function is to increase the reliability, scalability and maintenance of these models, ultimately providing stable and practical machine learning applications.

4. Architecture Diagram

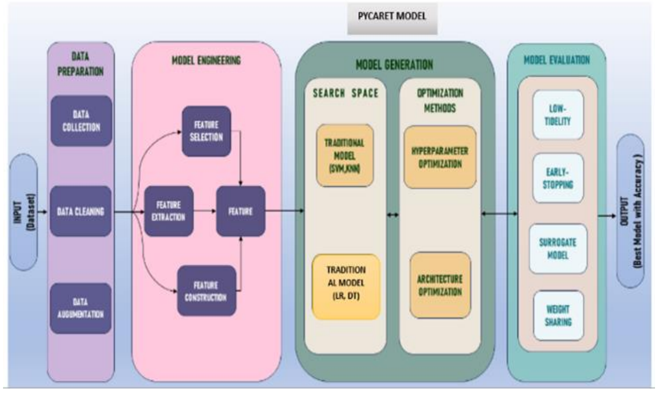


Figure 1 System Architecture

This architecture diagram can access the visual expression of the structure and it creates the components of the system or application. It typically includes various elements such as modules, databases, servers, and their interactions [16][18]. This diagram acts as a high level of field of the view, showing how to perform the approved functions and it connect and work with other parts of the system. This visual representation aids in understanding the overall design, dependencies, and flow of data or processes within the architecture. It is a valuable tool for communication among stakeholders, allowing developers, architects, and other team members to have a shared understanding of the system's structure and also helps in decision-making, troubleshooting, and system documentation. Figure 1 shows System Architecture.

5. Methodologies

Research work aims to study the complex parts of the automatic system to provide the potential to revolutionize side analysis, experiments and machine learning areas. As a special emphasis for eliminating defects in the existing method for pre -processing data, the system is placed as a promising approach to improve data recruitment and improve the results of other areas. This document provides more effective and effective applications for machine learning, which ultimately helps to overcome the innovative functions, experimental inspections and preliminary data processing of the system. The emphasis on improving datasets suggests a commitment to elevating the overall quality of input data, It is essential to machine learning models' ability to succeed. Figure 2 shows Methodologies.

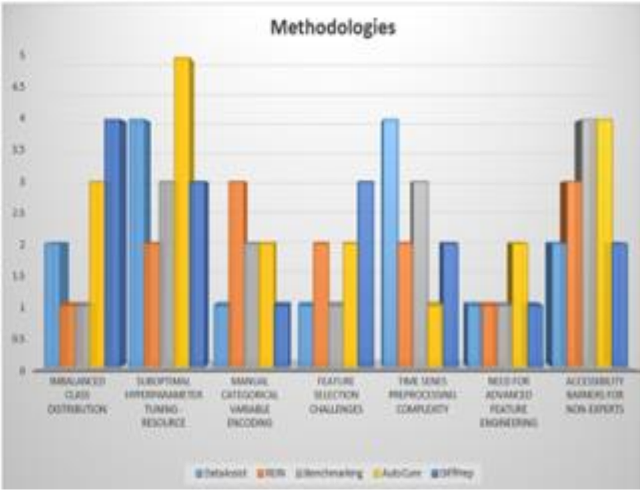


Figure 2 Methodologies

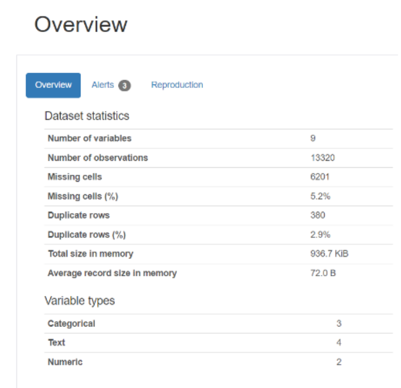
6. Result and Discussion

Automl is an extensive term that is involved in various methods and method to automate the ML model. The specific formulas and methods used in the Automl system may vary depending on the automatic tasks. Conclusion and output of this data set. This histogram analyzes the value of the data set. The value of the data set consists a variable without cells, stone, replication and total size [17].

6.1 Hyperparameter Tunning:

Improving machine learning models is mostly dependent on hyperparameter adjustment. It describes the process of selecting specific movable parameters that affect the model's learning from data before training. These parameters, which are manually set and have a big impact on the outcome, are not learned throughout the training process. The key objective is to identify the best set of parameter values that enhance the model’s accuracy, as measured by a specific evaluation metric. Commonly employed strategies include Random Search, which randomly investigates settings, and Grid Search, which checks all combinations in an organized manner. By adjusting these parameters, one can ensure that the model works effectively on unknown data by striking a reasonable balance between being too basic and too complex. Table 1 shows Individual Values from the Dataset.

Table 1 Individual Values from the Dataset



Overview

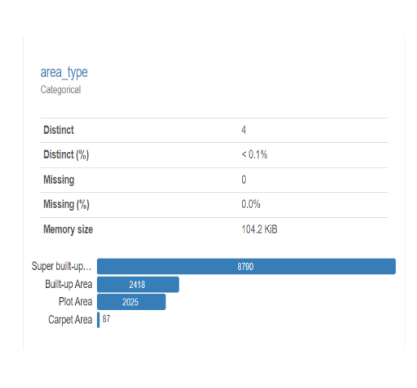
Overview | Alerts 3 | Reproduction

Dataset statistics

Number of variables	9
Number of observations	13320
Missing cells	6201
Missing cells (%)	5.2%
Duplicate rows	380
Duplicate rows (%)	2.9%
Total size in memory	936.7 KiB
Average record size in memory	72.0 B

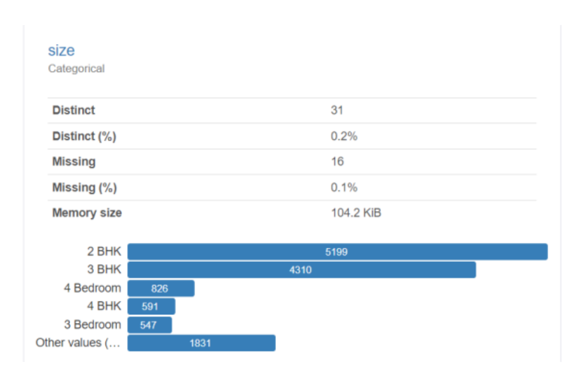
Variable types

Categorical	3
Text	4
Numeric	2



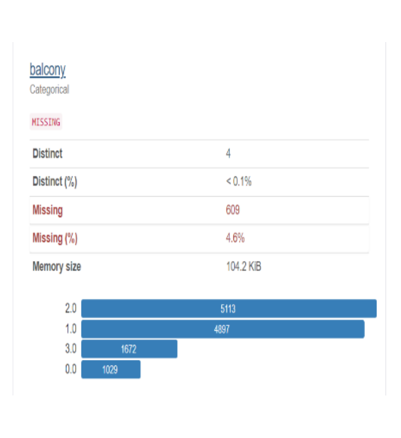
area_type
Categorical

Distinct	4
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	104.2 KiB



size
Categorical

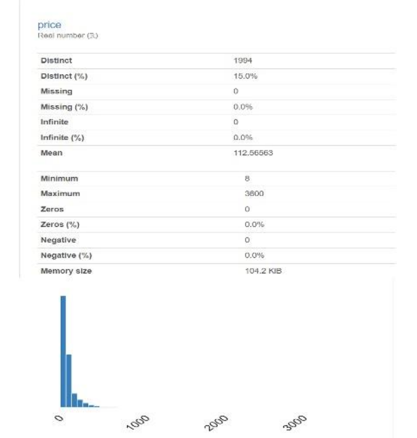
Distinct	31
Distinct (%)	0.2%
Missing	16
Missing (%)	0.1%
Memory size	104.2 KiB



balcony
Categorical

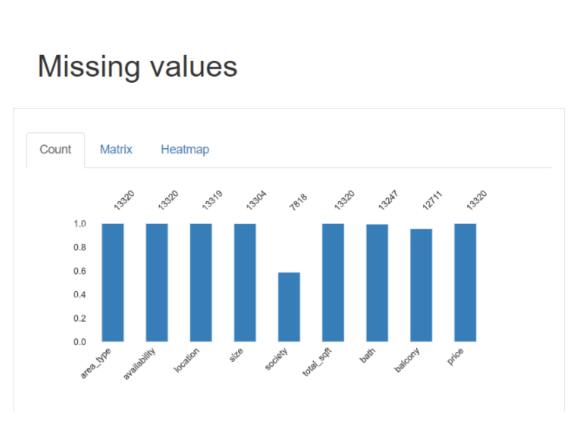
MISSING

Distinct	4
Distinct (%)	< 0.1%
Missing	609
Missing (%)	4.6%
Memory size	104.2 KiB



price
Real number (ℝ)

Distinct	1994
Distinct (%)	15.0%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	112.56563
Minimum	8
Maximum	3600
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	104.2 KiB



Missing values

Count | Matrix | Heatmap

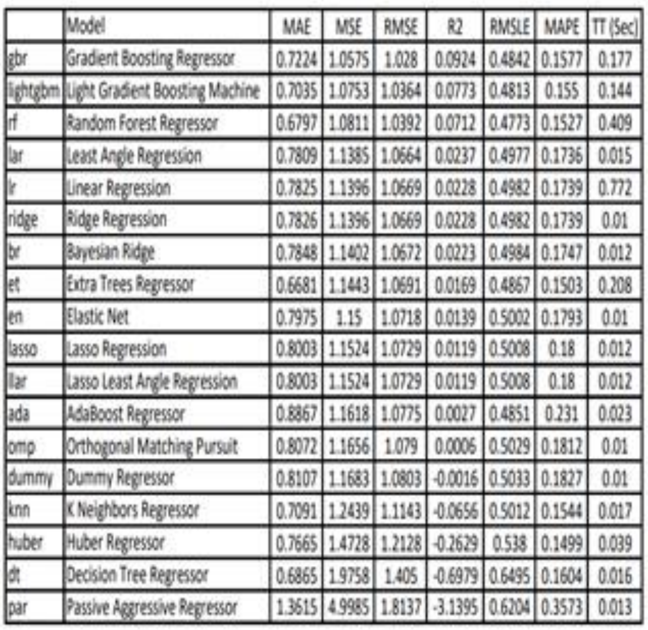
6.2 Feature Engineering

Feature engineering is a key step in building effective machine learning models. It centers on adjusting and structuring the input data to help the model recognize patterns more efficiently during training. This can include generating new features such as squaring an existing one to highlight nonlinear relationships or combining existing features to uncover deeper patterns. It also covers transforming data, like scaling numerical values to a standard range, handling missing values appropriately, or converting categorical data into numerical form. Another important part of the process is selecting the most relevant features while eliminating those that add little value. By simplifying the model, this method helps it make better predictions on new data while reducing the chance of overfitting.

6.3 Ensemble Methods

Ensemble methods are a valuable part of AutoML systems, where multiple models are combined to produce more accurate and stable predictions. Instead of relying on a single model, these techniques merge results from several, each offering different perspectives on the data. Popular strategies like bagging, boosting, and stacking help reduce errors, avoid overfitting, and improve overall performance. For instance, bagging blends results from models trained on different parts of the data, whereas boosting focuses on improving accuracy by learning from earlier model errors. Stacking goes a step further by using another model to learn how best to combine the outputs. By working together, these models often outperform any single one, making ensemble methods a powerful tool for building strong machine learning solutions [19].

6.4 Model Selection



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	0.7224	1.0575	1.028	0.0924	0.4842	0.1577	0.177
lightgbm	Light Gradient Boosting Machine	0.7035	1.0753	1.0364	0.0773	0.4813	0.155	0.144
rf	Random Forest Regressor	0.6797	1.0811	1.0392	0.0712	0.4773	0.1527	0.409
lar	Least Angle Regression	0.7809	1.1385	1.0664	0.0237	0.4977	0.1736	0.015
lr	Linear Regression	0.7825	1.1396	1.0669	0.0228	0.4982	0.1739	0.772
ridge	Ridge Regression	0.7826	1.1396	1.0669	0.0228	0.4982	0.1739	0.01
br	Bayesian Ridge	0.7848	1.1402	1.0672	0.0223	0.4984	0.1747	0.012
et	Extra Trees Regressor	0.6681	1.1443	1.0691	0.0169	0.4867	0.1503	0.208
en	Elastic Net	0.7975	1.15	1.0718	0.0139	0.5002	0.1793	0.01
lasso	Lasso Regression	0.8003	1.1524	1.0729	0.0119	0.5008	0.18	0.012
llar	Lasso Least Angle Regression	0.8003	1.1524	1.0729	0.0119	0.5008	0.18	0.012
ada	AdaBoost Regressor	0.8867	1.1618	1.0775	0.0027	0.4851	0.231	0.023
omp	Orthogonal Matching Pursuit	0.8072	1.1656	1.079	0.0006	0.5029	0.1812	0.01
dummy	Dummy Regressor	0.8107	1.1683	1.0803	-0.0016	0.5033	0.1827	0.01
knn	K Neighbors Regressor	0.7091	1.2439	1.1143	-0.0656	0.5012	0.1544	0.017
huber	Huber Regressor	0.7665	1.4728	1.2128	-0.2629	0.538	0.1499	0.039
dt	Decision Tree Regressor	0.6865	1.9758	1.405	-0.6979	0.6495	0.1604	0.016
par	Passive Aggressive Regressor	1.3615	4.9985	1.8137	-3.1395	0.6204	0.3573	0.013

Figure 3 Precision for Every Algorithm

Choosing the right model in AutoML depends on how well it performs based on specific metrics like accuracy, F1-score, or ROC-AUC. Accuracy is useful for balanced datasets, while F1-score works better when dealing with imbalanced data. Figure 4 shows Best Model Analysis. By testing various models and selecting the most appropriate model based on performance, autoML tools such as PyCaret simplify the process. Figure 3 shows Precision for Every Algorithm.

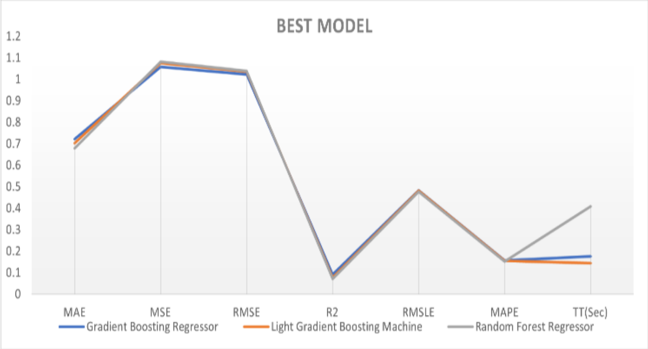


Figure 4 Best Model Analysis

Conclusion and Future Work

The field of machine learning data pre-processing is progressing swiftly, with former challenges such as imbalanced datasets and complex hyperparameter tuning—now offering valuable opportunities for innovation. This study emphasizes the need for smart, reliable strategies to effectively tackle these issues and support continuous growth in machine learning. The proposed AutoML system presents a streamlined and cohesive way to automate both data pre-processing and model deployment tasks. It includes specialized components for key tasks such as feature engineering, handling imbalanced data, hyperparameter tuning, and time-series processing—streamlining the process without compromising performance. With this integrated design, the system holds strong potential to accelerate and simplify model development, making it more accessible and efficient for a wide range of users. Future advancements in data pre-processing are expected to focus on a few critical areas. One major goal is to improve automation throughout the machine learning workflow streamlining processes, enhancing ease of use, and reducing manual effort. This would help users work more efficiently and effectively. Another important direction is making model behavior easier to understand by improving explain ability. Clearer insights into how predictions are made are essential for building trust and confidence, particularly when these models are applied in real-world scenarios.

Acknowledgements

The authors would like to thank Rajiv Gandhi College of Engineering and Technology for providing the necessary support and facilities to carry out this research work.

References

[1]. K. Goyle, Q. Xie, & V. Goyle, "DataAssist: A Machine Learning Approach to Data Cleaning and Preparation," eprint arXiv:2307.07119, 2023.

[2]. S. Juddoo, "Investigating Data Repair steps for EHR Big Data," in International Conference on Next Generation Computing Applications, 2022.

[3]. P. Ribeiro, P. Orzechowski, J. B. Wagenaar, & J. H. Moore, "Benchmarking AutoML

algorithms on a collection of synthetic classification problems," eprint arXiv:2212.02704, 2022.

[4]. M. Abdelaal, C. Hammacher, & H. Schoening, "REIN: A Comprehensive Benchmark Framework for Data Cleaning Methods in ML Pipelines," eprint arXiv:2302.04702, 2023.

[5]. F. Neutatz, B. Chen, Y. Alkhatib, J. Ye, & Z. Abedjan, "Data Cleaning and AutoML: Would an Optimizer Choose to Clean?" Eprint Springer s13222-022-00413-2, 2022.

[6]. M. Abdelaal, R. Koparde, & H. Schoening, "AutoCure: Automated Tabular Data Curation Technique for ML Pipelines," eprint arXiv:2304.13636, 2023.

[7]. S. Holzer & K. Stockinger, "Detecting errors in databases with bidirectional recurrent neural networks," OpenProceedings ZHAW, 2022.

[8]. P. Li, Z. Chen, X. Chu, & K. Rong, "DiffPrep: Differentiable Data Preprocessing Pipeline Search for Learning over Tabular Data," eprint arXiv:2308.10915, 2023.

[9]. M. Singh, J. Cambronero, S. Gulwani, V. Le, C. Negreanu, & G. Verbruggen, "DataVinci: Learning Syntactic and Semantic String Repairs," eprint arXiv:2308.10922, 2023.

[10]. R. Wang, Y. Li, & J. Wang, "Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integration and Preparation," eprint arXiv:2207.04122, 2.

[11]. B. Hilprecht, C. Hammacher, E. Reis, M. Abdelaal, & C. Binnig, "DiffML: End-to-end Differentiable ML Pipelines," eprint arXiv:2207.01269, 2022.

[12]. V. Restat, M. Klettke, & U. Störl, "Towards a Holistic Data Preparation Tool," in EDBT/ICDT Workshops, 2022.

[13]. H. Stühler, M. A. Zöller, D. Klau, A. Beiderwellen-Bedrikow, & C. Tutschku,"Benchmarking Automated Machine Learning Methods for Price Forecasting Applications," eprint arXiv:2304.14735, 2023.

[14]. P. Gijsbers, E. LeDell, S. Poirier, J. Thomas, B. Bischl, J. Vanschoren, An open source automl benchmark, 2019, 6th ICML Workshop on Automated Machine Learning AutoML@ICML2019; Conference date: 14-06-2019 Through 14-06-2019.

[15]. P. Gijsbers, M. L. P. Bueno, S. Coors, E. LeDell, S. Poirier, J. Thomas, B. Bischl, J. Vanschoren, Amlb: an automl benchmark (2022). doi:10. 48550/ARXIV.2207.12560.

[16]. I. Guyon, L. Sun-Hosoya, M. Boull´e, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W.-W. Tu, E. Viegas, Analysis of the AutoML Challenge Series 2015–2018, Springer International Publishing, Cham, 2019, pp. 177–219. doi:10.1007/978-3-030-05318-5_10.

[17]. Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, Smola AJ (2020) Autogluon-tabular: Robust and accurate automl for structured data. CoRR, abs/2003.06505

[18]. K. Van der Blom, A. Serban, H. Hoos, and J. Visser, “AutoML Adoption in ML Software,” 8th ICML Workshop on Automated Machine Learning, 2021.

[19]. T. T. Le, W. Fu, J. H. Moore, Scaling tree-based automated machine learning to biomedical big data with a feature set selector, Bioinformatics 36 (1) (2020)