A Comprehensive AutoML Solution for Automated Data Preprocessing and Model Deployment

Palanivel N[#1], Vigneshwaraan B[#2], Sobanraj B[#3], Ragavan P[#4]
Department of Computer Science and Engineering,
Manakula Vinayagar Institute of Technology, Puducherry, India.
palanivelcse@mvit.edu.in [#1], vigneshwaraancse2020@mvit.edu.in [#2], sobanrajcse2020@mvit.edu.in [#3],
ragavancse2020@mvit.edu.in [#4].

Abstract:* An important turning point in the field of machine learning has been reached with the convergence of data preparation and automated machine learning (AutoML). AutoML has become a reliable solution for tackling major issues with data preprocessing approaches because of its capacity to automate the coordination of different machine learning processes. This study covers a wide range of important topics related to data preparation, including feature selection, time-series preprocessing, manual encoding mistakes, class imbalance, and inefficient hyperparameters. AutoML's revolutionary effect on simplifying crucial data preparation procedures is one of its main contributions to data preprocessing. Data preparation has historically been a labor-and time-intensive procedure that calls for specialised knowledge and physical involvement at different points in the process. But many of these jobs may now be completed automatically because to the development of automated algorithms, which has significantly increased productivity and efficiency. Furthermore, by making data preprocessing more approachable for both specialists and non-experts, AutoML has democratised the field. Through the automation of intricate processes like feature selection and hyperparameter tweaking, AutoML technologies enable users to concentrate on more advanced parts of model creation, such formulating problems and interpreting outcomes. In addition to quickening the rate of invention, this democratisation of data preprocessing encourages increased cooperation and knowledge exchange within the machine learning community.

Keywords- ***AutoML, Data, Preprocessing, MachineLearning, Hyperparameters, Feature selection, Report generation, Data Visualization***

I. INTRODUCTION

The presented paper navigates the intricate relationship between Machine Learning (ML) and AutoML, emphasizing their symbiotic collaboration. ML, relying on vast datasets, employs algorithms to discern patterns, make predictions, and facilitate decision-making. AutoML complements this by automating essential yet tedious tasks, enhancing the efficiency and accessibility within the ML pipeline.

In the pursuit of refining data preprocessing methodologies, the survey proposes a comprehensive exploration of research papers and contributions. This extensive study encompasses a spectrum of papers that delve into innovative solutions bridging the gap between current challenges and the future promises of data preprocessing. The objective is to extract crucial insights by merging the intricacies of data preprocessing with the transformative potential of AutoML, aiming to advance data-driven decision-making in the evolving ML landscape. Highlighted within this survey are various research papers, from "DataAssist" to "REIN," each contributing vital principles to guide this quest. These papers shed light on the complex terrain where challenges such as imbalanced data, hyperparameter optimization nuances, and the need for advanced feature engineering converge, necessitating holistic solutions to bridge the divide between data and model.

As the frontiers of machine learning continue to expand, the principles extracted from these research papers act as guiding beacons, urging the crafting of automated solutions capable of addressing multifaceted challenges. Rooted in this extensive literature survey, the forthcoming architectural overview promises not only innovation but a transformation of the status quo. It aims to provide all-encompassing, end-to-end solutions for data preprocessing and the automation of pivotal tasks. The survey's principles revolve around problem identification and the search for innovative solutions, bridging the gap between present challenges and the promising future of data preprocessing. Each research paper in the survey addresses a specific facet of data preprocessing, collectively contributing to a comprehensive understanding of this critical domain.

II. RELATED WORKS

Runtime Prediction of Machine Learning Algorithms in Automated Systems Parijat Dube; Theodoros Salonidis [1] DataAssist sounds like a promising addition to the landscape of automated machine learning (AutoML) tools. By focusing on data preparation and cleaning, it addresses a crucial aspect of the machine learning workflow that is often overlooked by existing tools. The key features of DataAssist seem to align well with the needs of data scientists and analysts, particularly in industries where data quality is paramount, such as economics, business, and forecasting. By providing functionalities for exploratory data analysis, visualization, anomaly detection, and data preprocessing, DataAssist streamlines the process of preparing data for modeling, potentially saving significant time and effort for practitioners. Moreover, the ability to export cleaned and preprocessed datasets for integration with other AutoML tools or user-specified models enhances its versatility and interoperability within existing workflows. This flexibility is crucial for accommodating different preferences and requirements in data analysis pipelines. Overall, DataAssist appears to fill a significant gap in the existing landscape AutoML tools by prioritizing data-centric tasks and offering comprehensive support for data preparation and cleaning. Its potential to save over 50% of the time typically spent on these tasks underscores its value proposition for practitioners across various domains.

Suraj Juddoo investigates data repair steps for EHR Big Data.[2] This paper addresses a significant challenge with relation to big data systems, particularly focusing on Electronic Health Records (EHR). The emphasis on optimizing data quality methodologies aligns well with the growing importance of leveraging high-quality data for meaningful insights, especially in sensitive domains like healthcare. The recognition of the data repair stage as a critical component of the The data quality life cycle involves crucial, as addressing dirty data is often a complex and resource-intensive task. The acknowledgment an ignorance of how well-performing data restoration tools and algorithms work in the context of big data is an important observation, highlighting the need for specialized solutions in this domain. The systematic examination of data repair techniques and tools, then an experiment-based evaluation, is a robust methodology for gaining insights into their effectiveness. The comparison with a prototype built from previous study results adds a practical dimension to the evaluation. The finding that For Big Data, no algorithm or tool was found to be exceptionally sufficient emphasizes the challenges in this domain. However, identifying some algorithms and tools as marginally better than others provides valuable insights for potential improvements. The recommendations for enhancing data repair algorithms and tools for Big Data represent a valuable contribution to the field, guiding future research and development efforts.

Assessing the performance of AutoML algorithms using a set of simulated classification tasks Henrique Pedro Ribeiroa and Patryk Orzechowski [3] This paper explores the growing landscape of the popularity of machine learning automated (AutoML) programs can be attributed to their great performance and versatility in solving a wide range of issues. The challenge lies in choosing the most suitable AutoML algorithm for a given problem amid the increasing options available. To address this, the study examines the output of four well-known AutoML algorithms using their Diverse and generative ML benchmarking (DIGEN): Auto-Sklearn, H2O AutoML, Auto-Sklearn 2, and Tree-based Pipelines Optimizing Tool (TPOT). Synthetic datasets called DIGEN are used to demonstrate the advantages and disadvantages of popular machine learning algorithms. The outcomes demonstrate how successfully AutoML detects pipelines across datasets. While the majority of AutoML algorithms demonstrated similar performance, subtle differences emerged based on specific datasets and evaluation metrics, providing valuable insights into their comparative effectiveness.

A Whole-System Benchmarking Structure for Data Cleaning Techniques in Machine Learning Pipelines: REIN Christian Hammacher, Harald Schoening, and Mohamed Abdelaal [4] The paper emphasizes the crucial role of machine learning (ML) in daily life and emphasizes how important high-quality data is throughout the ML application lifecycle. It acknowledges the common discrepancies present in real-world tabular data, include inconsistencies, duplication, outliers, missing values, and pattern violations, which often arise during data collection, transfer, storage, or integration. Despite numerous data cleaning methods addressing these issues, the paper points out a gap in considering downstream ML model requirements. To bridge this gap, the work introduces a comprehensive benchmark named REIN1, aiming must carefully evaluate how various ML models are affected by data cleaning techniques. The benchmark addresses key research questions, exploring the necessity and efficacy of data cleaning in ML pipelines. The evaluation involves 38 error detection and repair methods, ranging from simple to advanced. To provide comprehensive insights, the benchmark employs a broad range of machine learning models that were trained on 14 publicly-accessible datasets that span multiple domains and include both synthetic and realistic error characteristics.

AutoCure: Machine Learning Pipeline Automatic Tabular Data Curation Method Ahmad Schoening, Harald Schoening, and Rashmi Koparde [5] The paper introduces Data curation pipeline AutoCure is innovative and requires no setting designed to address the persistent challenge of data preparation in machine learning applications across domains like autonomous driving, healthcare, and finance. The need for expert knowledge and considerable time investment in navigating the extensive search space for suitable data curation and transformation tools is a recognized hurdle in model development. AutoCure stands out by synthetically enhancing the clean data fraction by combining a data augmentation module with an inventive adjustable ensemble-based error detection technique. Notably, its configuration-free nature streamlines the implementation process, making it accessible for integration using free and open-source resources such as Auto-sklearn, H2O, and TPOT, therefore advancing the general democratization of machine learning.

Reciprocal neural networks for bidirectional mistake detection in databases Holzer and Stockinger, Kurt [6] This paper presents an innovative architecture leveraging bidirectional recurrent neural networks for the purpose of error detection in databases. Through experiments conducted on six distinct datasets, The outcomes demonstrate how well this strategy performs in comparison to cutting-edge mistake detection technologies. Specifically, the average F1-scores across all datasets demonstrate the effectiveness of the proposed architecture. Notably, the system exhibits a lower standard deviation, indicating greater robustness compared to existing methods. An additional advantage is the system's ability to achieve high F1-scores without the need for supplementary data augmentation techniques. This signifies the potential of the introduced bidirectional recurrent neural network architecture as a robust and efficient solution for error detection in diverse database scenarios.

III. PROPOSED METHODOLOGY

A variety of technologies were employed in the study, including information collection, dataset preparation, and model evaluation.

DATA PREPROCESSING MODULE:

The role of cleaning and preparing raw data for analysis is a crucial step in the data science workflow, as it significantly influences the accuracy and reliability of subsequent analyses and machine learning models. This responsibility involves a series of tasks aimed at ensuring the data is presented in a format that is appropriate and consistent for meaningful interpretation. One fundamental aspect is handling missing values, where techniques such as imputation or deletion are employed to address the absence of information. Scaling features is another essential task, particularly when variables are measured on different scales, to prevent certain features from disproportionately influencing the analysis. Additionally, encoding categorical variables is necessary to transform qualitative input into a numerical form that machine learning algorithms can handle. This process helps maintain the integrity of the data and ensures that the chosen analytical techniques can effectively derive insights. Overall, the meticulous cleaning and preparation of raw data form the foundation for robust and reliable data analyses, facilitating informed decision-making in various domains.

AUTOML CORE MODULE:

The central module orchestrating the entire AutoML process serves as the backbone of the automated machine learning workflow, playing a pivotal role in coordinating and managing various tasks. This module integrates sub-modules that collectively contribute to the comprehensive AutoML pipeline, ensuring a streamlined and efficient process. Among these sub-modules, hyperparameter tuning is responsible for optimizing the configuration settings of machine learning models to enhance their performance. Feature engineering involves transforming and selecting features to improve the capacity of the model to identify links and patterns in the data. Model selection, another critical sub-module, aids in choosing the most suitable algorithm or ensemble of algorithms for a given task. By consolidating these sub-modules, the central module makes ensuring the AutoML process is coherent and well-coordinated, with each step contributing to the final result. of automating the model development lifecycle and delivering optimized, high-performing machine learning models.

CATEGORICAL VARIABLE ENCODING STANDARDIZATION MODULE:

The task of ensuring consistent encoding of categorical variables is vital in both regression and classification tasks within the context of machine learning. Categorical variables, representing qualitative data, need to be converted into a numerical representation so that it can work with other 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. By implementing these encoding methods consistently, the module ensures that the machine learning models receive uniform input representations, fostering accuracy and reliability in predictions across both regression and classification scenarios. This consistency is essential for creating robust and interpretable models that can effectively learn patterns from categorical features.

USER INTERFACE (UI) MODULE:

The user-friendly interface serves as the gateway for practitioners to interact seamlessly 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. Through an intuitive design, practitioners can effortlessly upload datasets, specify preferences for hyperparameters or feature engineering, and easily navigate through the system's functionalities. The interface abstracts the complexities of the underlying AutoML algorithms, making it suitable for users of different skill levels. Visualization tools incorporated into the interface enable users to interpret and comprehend the outcomes of the automated processes, fostering a transparent and interactive user experience. 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.

REPORT GENERATION MODULE:

The deployment module plays a crucial role in facilitating the seamless integration of AutoML-generated models into production environments. Its primary function is to streamline the transition from model development to real-world applications. 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. By encompassing these functionalities, the deployment module enhances the reliability, scalability, and maintainability of AutoML-generated models in production, ultimately supporting the practical and sustainable application of machine learning solutions.

ARCHITECTURE DIAGRAM:

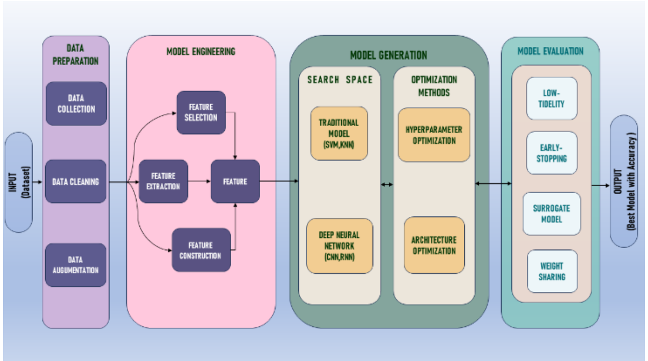


Fig 1: System Architecture

An architecture diagram makes the structure's visual representation available and components of a system or application. It typically includes various elements such as modules, databases, servers, and their interactions. The diagram serves as a high-level overview, illustrating how different parts of the system are connected and work together to achieve the intended functionalities. 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, helping in decision-making, troubleshooting, and system documentation.

The paper aims to explore the intricate details of the AutoML system, providing an in-depth analysis of its capabilities, experimental results, and its potential to revolutionize the field of machine learning. With a particular focus on addressing the shortcomings in existing data preprocessing methodologies, the system is positioned as a promising approach to enhancing datasets and subsequently improving findings across diverse domains. The paper likely delves into the system's innovative features, experimental validations, and how it contributes to overcoming challenges in data preprocessing, ultimately paving the way for more effective and efficient machine learning applications. 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.

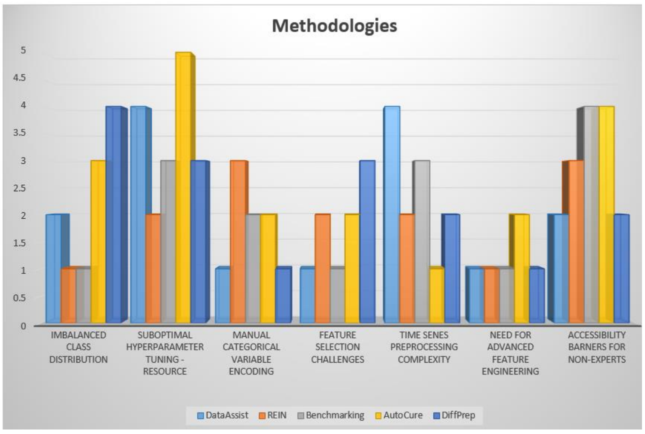
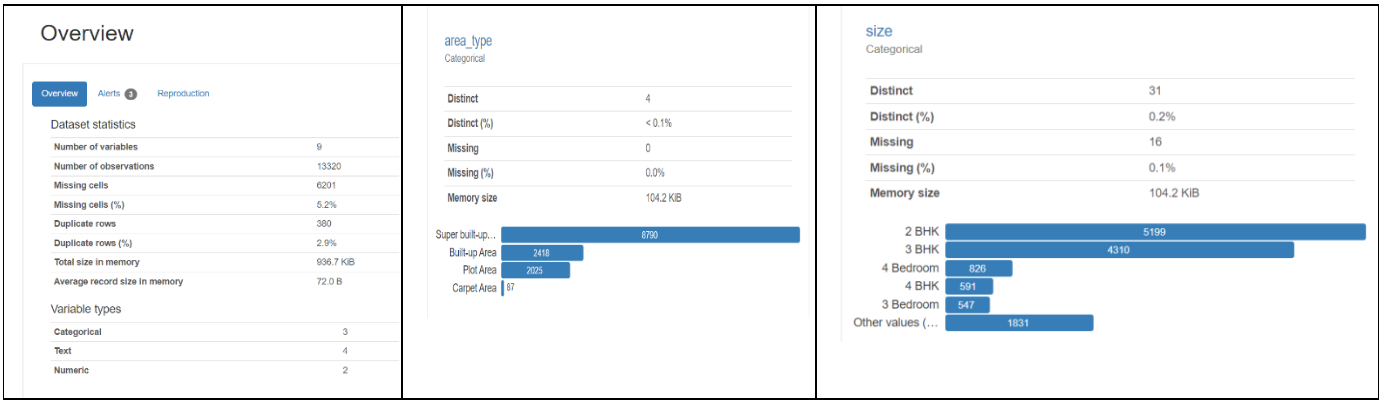


Fig 2: Methodologies

IV. RESULT AND DISCUSSION

AutoML (Automated Machine Learning) is a broad term that encompasses a variety of techniques and methodologies to automate the procedure for creating ML models. The specific formulas and methods used in AutoML systems can indeed vary based on the tasks being automated.



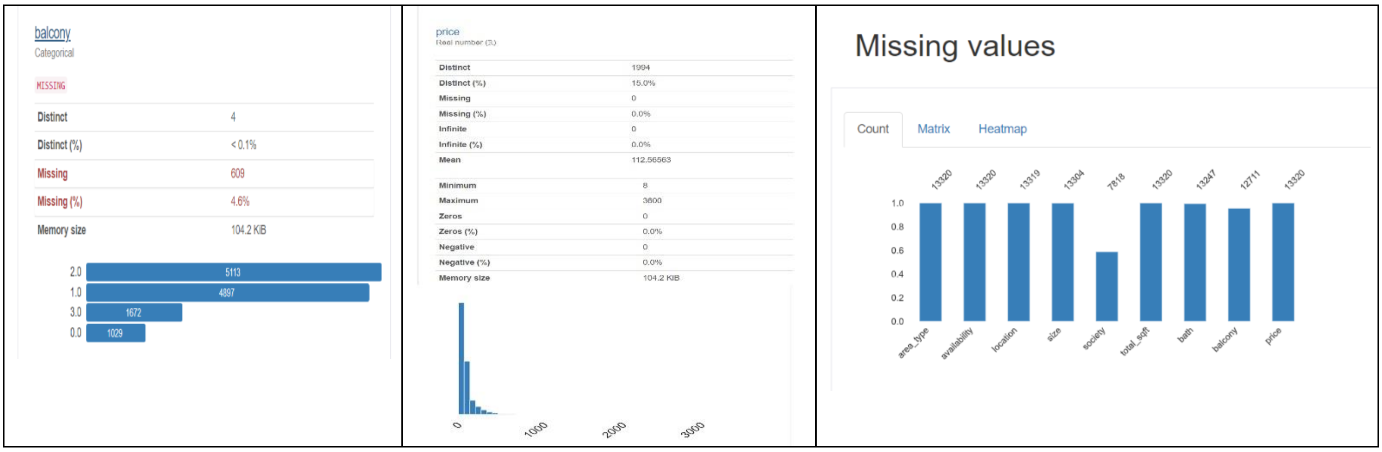


Table 1: **Individual Values from the Dataset.**

These are some outputs and overview of the given dataset. The given bar graphs are the analysed values of the given dataset which consits of no. of variables, Missing cells, Duplicate rows, Total Size in Memory and Missing Values.

HYPERPARAMETER TUNNING:

A crucial part of optimising machine learning models is hyperparameter tweaking, which is essential to deciding the models' performance and capacity for generalisation. Hyperparameters are external settings that need to be specified prior to the training process, in contrast to the model's internal parameters, which are learned during the training process. The learning rate in gradient descent, the number of hidden layers in a neural network, and the kind of kernel in a support vector machine are a few examples of hyperparameters. Finding the ideal set of hyperparameter values to achieve the best model performance on unobserved data is the main goal of hyperparameter tweaking. Usually, this is accomplished by looking through a predetermined space of potential hyperparameter configurations and assessing the effectiveness of each configuration using a selected performance indicator. Two methods for hyperparameter tuning that are frequently utilised are grid search and random search. Grid search is systematically going through a predetermined grid of hyperparameter variables and assessing the model's performance for each possible combination. Grid search can be computationally expensive, particularly for high-dimensional hyperparameter spaces, despite being methodical and guaranteeing the discovery of the ideal configuration inside the search area. Conversely, random search assesses the model's performance for each randomly selected configuration and chooses hyperparameter values at random from the established search space. When searching for optimal hyperparameter combinations, random search is frequently more computationally efficient than grid search, requiring fewer evaluations. It does not, however, ensure that the ideal arrangement will be found. To sum up, hyperparameter tuning is an essential stage in the optimisation of machine learning models, which seeks to identify the ideal external setting combination that maximises the model's performance. Two popular methods for this purpose are random search and grid search, each with pros and cons of their own. The selection of one of these methods is contingent upon various parameters, including the extent of the hyperparameter space, available processing capacity, and the targeted optimisation efficiency.

$$Best\ Hyperparameter\ Value = \arg max_{Hyperparameter\ Values}\ Model\ Performance\ Metric$$

Grid search examines a preset set of combinations of hyperparameters methodically, exploring the entire search space. In contrast, random search randomly samples configurations, offering a more stochastic approach. To ensure that the model performs as well as possible on data that hasn't been seen before, both approaches try to find a compromise between generalization and model complexity. A key component of fine-tuning models is hyperparameter tuning for specific tasks, ultimately enhancing their predictive capabilities and robustness.

FEATURE ENGINEERING:

In order to improve the data representation and, eventually, the model performance, feature engineering is an essential phase in the creation of machine learning models. It includes a wide range of methods designed to modify the input features in order to better fit them with the learning algorithms. Creating new features from scratch or from outside sources is one aspect of feature engineering. This could mean introducing domain-specific knowledge to generate useful features, or integrating numerous features to create interactions. For instance, new elements like the proportion of bedrooms to bathrooms or the overall area of the home may be added to a dataset that contains data on housing costs in order to gather more insights. Changing current features to make them more suited for the modelling methods is another component. In order to properly capture non-linear interactions, this may involve applying transformations such polynomial or logarithmic transformations or scaling numerical features to a standardised range through normalisation or standardisation. Furthermore, choosing a subset of features that are most instructive and pertinent for the task at hand is a common step in the feature engineering process. Principal component analysis (PCA) and other dimensionality reduction approaches, such as feature importance ranking, could be used to accomplish this. The model can concentrate on the most discriminative characteristics by decreasing the dimensionality of the feature space, which may enhance generalisation performance as well as computational efficiency. All things considered, feature engineering is a complex process that calls for ingenuity, domain knowledge, and a thorough comprehension of the data and the modelling assignment. When implemented correctly, it can greatly improve machine learning models' performance by giving them access to more pertinent and instructive input characteristics. Therefore, it is believed that feature engineering is essential to the building of successful machine learning models.

$$New\ Feature = Feature^2$$

Creating new features may involve combining or synthesizing existing features to capture higher-order relationships or patterns in the data. Transformation of features can include normalizing or scaling numerical features, handling missing values, or encoding categorical variables. Additionally, finding and keeping the most essential features is the goal of feature selection while discarding less important ones, reducing dimensionality and potentially mitigating overfitting.

ENSEMBLE METHODS:

AutoML frequently leverages ensemble methods as a powerful strategy to boost overall model performance. Ensemble methods involve combining predictions from multiple individual models, often of diverse architectures or trained with different subsets of data. The goal is to exploit the complementary strengths of various models, mitigating individual weaknesses and improving overall predictive accuracy. Common ensemble techniques include bagging, boosting, and stacking. Bagging, such as in Random Forests, aggregates predictions from several decision trees that were trained using arbitrary portions of the data improved robustness and decreased overfitting.

$$Ensemble\ Prediction = \frac{1}{n}\sum_{i=1}^{n} Model_i\ (Input\ Data)$$

Boosting, exemplified by algorithms like AdaBoost or Gradient Boosting, sequentially trains models, with each subsequent model focusing on correcting the errors of its predecessor, leading to increased accuracy. Stacking combines predictions from different models using a meta-model, learning to weigh individual model outputs optimally. Ensemble methods are effective in handling complex relationships within data, increasing model stability, and generalizing well to unseen instances, making them a valuable tool in the AutoML toolkit for achieving superior predictive performance.

MODEL SELECTION:

The selection of the optimal model is a crucial step in the field of AutoML, and it is largely dependent on the assessment of several performance measures. These metrics are essential for comparing and evaluating various models since they function as quantitative indications of a model's predictive efficacy. The area under the Receiver Operating Characteristic (ROC) curve, accuracy, and F1-score are a few of the frequently employed metrics. Probably the easiest indicator to understand is accuracy, which is the percentage of properly predicted cases among all the instances. It is especially helpful when all classes in the dataset are equally significant or when a straightforward, understandable performance metric is required. It provides a clear indicator of overall accuracy. Conversely, the F1-score achieves a compromise between recall and precision. The percentage of accurately anticipated positive cases among all predicted positive cases is known as precision, whereas the percentage of properly predicted positive cases among all real positive cases is known as recall. Because the F1-score offers a single metric that accounts for both aspects of model performance, it is a good option for situations where the costs of false positives and false negatives differ. The F1-score is the harmonic mean of precision and recall. The trade-off between the genuine positive rate (sensitivity) and the false positive rate (specificity) at various thresholds is measured by the area under the ROC curve. A higher area denotes better performance, and it gives insight into how well a model can distinguish between classes. This statistic is very helpful for evaluating how well models perform in tasks involving binary categorization. The type of problem at hand determines which statistic should be prioritised. Accuracy might be a good option for datasets that are balanced and have equal representation for each type. Nonetheless, the F1-score is frequently chosen in situations with imbalanced classes, where one class much outnumbers the other or classes, since it assigns greater weight to the minority class and hence more accurately represents the model's performance. In the end, choosing the right assessment metric is essential to precisely evaluating and contrasting model performance in AutoML workflows.

$$Best\ Model = \arg max_{Models}\ Model\ Performance\ Metric$$

AutoML systems often perform a systematic search over hyperparameter configurations, and the model that performs the best according to the selected metric is the one that gets deployed. These metrics guide the AutoML process, ensuring the chosen model aligns with the specific objectives and requirements of the given machine learning task.

	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

Fig 3: Precision for Every Algorithm

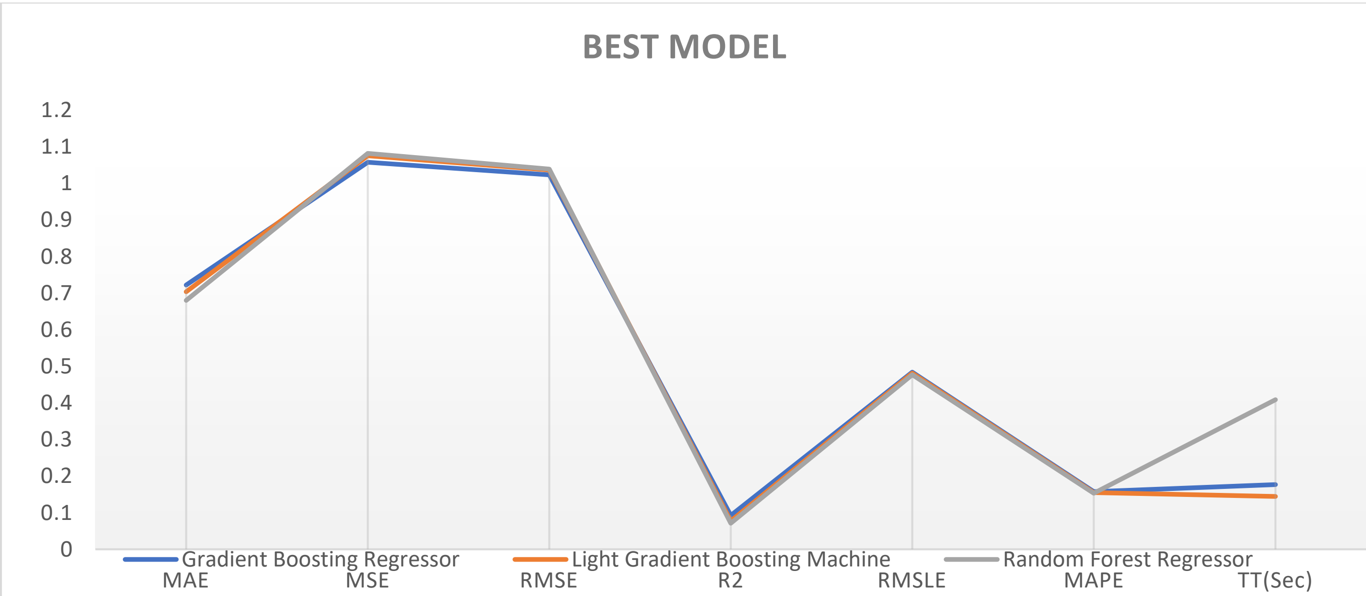


Fig 4: Best Model Analysis

V. CONCLUSION

The field of machine learning data preprocessing is about to undergo a significant metamorphosis. The problems and inefficiencies that have been identified, which vary from unequal class distributions to poor hyperparameter optimisation, present formidable obstacles as well as exciting opportunities for innovation. The aforementioned architectural study highlights the paramount significance of resilient solutions in overcoming these challenges and advancing the machine learning domain towards enhanced effectiveness and efficiency. The AutoML system is a groundbreaking solution that has emerged as a major component of this evolution. This all-inclusive technology is made to automate model deployment and data preprocessing, which is a paradigm change in the way machine learning processes are planned. Its main goal is to simplify the difficulties involved in data preparation while upholding a strict level of analytical integrity. The AutoML system stands out due to its specially designed components, each of which is made to tackle a particular problem that arises during data preprocessing. This system provides a complete and unified solution for processing complex time-series data, managing skewed data distributions, optimising hyperparameters, and feature engineering. The AutoML system can transform the field of data preprocessing in machine learning by combining these different activities into a unified framework. By providing practitioners with a potent toolkit to improve the calibre and effectiveness of their modelling endeavours, it stimulates innovation and advances the field. All things considered, the AutoML system is a huge step towards expanding the frontier of data-driven decision-making and democratising access to sophisticated machine learning capabilities.

FUTURE WORK:

There is a strong commitment to tackling the challenges present in this crucial phase of machine learning processes in the next developments in the field of data preprocessing. Increasing automation capabilities is one of the main goals in order to make the pipeline as a whole more accessible and effective. To reduce the need for manual intervention and promote a more seamless user experience, this calls for a concentrated effort to further streamline procedures. The automation of processes like transformation, feature engineering, and data cleansing frees up practitioners' time for higher-level decision-making and model improvement. Furthermore, a strong focus is placed on improving model explainability methods. This means developing and improving techniques that allow users to have a greater understanding of the reasoning behind the choices and forecasts made by machine learning models. Improved explainability not only encourages users to have more faith in these models but also gives them the ability to understand their internal workings better, which leads to more informed decisions and wider adoption in a variety of fields. In the end, these double pledges to automate and explainability are important steps in improving the effectiveness, openness, and usability of machine learning applications in practical settings.

REFERENCE

[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] S. Guha, F. A. Khan, J. Stoyanovich, & S. Schelter, "Automated Data Cleaning Can Hurt Fairness in Machine Learning-based Decision Making," in IEEE 39th International Conference on Data Engineering, 2023.

[11] R. Wang, Y. Li, & J. Wang, "Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integration and Preparation," eprint arXiv:2207.04122, 2022.

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

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

[14] M. Nashaat, A. Ghosh, J. Miller, & S. Quader, "TabReformer: Unsupervised Representation Learning for Erroneous Data Detection," eprint https://doi.org/10.1145/3447541, 2021.

[15] F. Calefato, L. Quaranta, F. Lanubile, & M. Kalinowski, "Assessing the Use of AutoML for Data-Driven Software Engineering," eprint arXiv:2307.10774, 2023.

[16] 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

[17] M. Feurer, A. Klein, J. Eggensperger, Katharina Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, in: Advances in Neural Information Processing Systems 28 (2015), 2015, pp. 2962–2970.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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

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

[24] 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) 250–256.