IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

AutoML: Automated Machine Learning using Streamlit

Prishita G K, P S N Vinisha, Vandana R, Chethan S Pandit, Dr. P. Manikandan

Department of Computer Science & Engineering, Faculty of Engineering and Technology, JAIN (deemed to-be University) Bengaluru, India

Professor, Department of Computer Science & Engineering, Faculty of Engineering and Technology, JAIN (deemed to-be University) Bengaluru, India

ABSTRACT: Our Automated Machine Learning (AutoML) pipeline uses Streamlit, Pandas Profiling, and Sklearn. It allows users to upload a dataset, visualize and analyze it using Pandas Profiling, and build a machine learning model using various classification and regression. This approach streamlines the process of performing AutoML tasks and provides a simple and user-friendly interface that does not require extensive programming knowledge. It allows users to upload a CSV data file, preview the data, and perform Automated Exploratory Data Analysis using pandas-profiling. Users can then select a target variable and task type (regression or classification), and the script trains and evaluates several machine learning models using cross-validation and GridSearchCV to find the best model. Overall, this model provides a simple and efficient way to perform AutoML tasks without requiring extensive programming knowledge.

KEYWORDS: AutoML Machine Learning, Streamlit, Pandas Profiling, Sklearn.

I. INTRODUCTION

The process of selecting, configuring, and optimising machine learning models is labor-intensive and complex. This process is known as automated machine learning, or AutoML. AutoML makes it simple and low-tech for nonexperts to create high-quality machine learning models. An attractive and interactive web application for data science and machine learning may be made using the open-source framework Streamlit. By combining AutoML with Streamlit, developer can build powerful and user-friendly machine learning applications in a fraction of the time it would take to build them from scratch.

One of the main benefits of using AutoML with Streamlit is the ability to quickly develop and deploy machine learning models. This saves time and resources compared to traditional machine learning development methods that require extensive manual testing and tuning. Another benefit of using AutoML with Streamlit allows non-technical people to interact with machine learning models through userfriendly interfaces. The developer can utilise Streamlit to build an interactive interface that lets users input data and view the output of the machine learning model once the model has been generated.

A wide range of tools and widgets offered by Streamlit make it simple to design unique user interfaces that cater to their demands. AutoML with Streamlit has many real-world applications across a variety of industries. In healthcare, Machine learning models can be applied to healthcare to identify illnesses and forecast patient outcomes. Machine learning models can be used in finance to spot fraud and decide which investments to make. Machine learning models can be used in marketing to examine consumer behaviour and tailor advertising strategies. Machine learning models can also be used in manufacturing to streamline operations and cut waste. AutoML with Streamlit is a potent tool for quickly and efficiently generating and deploying machine learning models.

II. RELATED WORK

[1] The two-sample hypothesis testing problem, which is a fundamental problem in statistics and machine learning, is addressed in the paper with a novel AutoML approach. The suggested approach automatically chooses the optimum statistical test for a given two-sample problem by combining feature engineering, model selection, and hyperparameter optimisation. The method is put into practise as the "AutoML Two-Sample Test" Python module, and it is tested on a number of real-world datasets. The findings demonstrate that the suggested method surpasses current cutting-edge techniques for two-sample testing and is capable of automatically choosing the appropriate test for a particular scenario.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

[2] XAutoML, a visual analytics tool for automated machine learning (AutoML), is introduced in the study. The authors suggest a visual analytics tool that can offer insights into the automated machine learning pipeline and its results in order to solve the difficulty of comprehending and analysing the pipeline. The capabilities of XAutoML to visualise data, choose models, tune hyperparameters, and prioritise features are all covered in this paper. The authors also give experimental findings showing how well XAutoML works to provide understanding of the automated machine learning pipeline. The study's findings support the notion that XAutoML is a useful tool for deciphering and comprehending the automated machine learning pipeline. The authors offer suggestions for further study as well as enhancements to the XAutoML tool.

[3] The DeepCave tool provides an interactive interface that allows users to visualize and explore the features learned by a CNN. The interface is designed to be intuitive and userfriendly, and provides a variety of interactive controls for adjusting the input images and exploring different layers of the network. One of the key features of DeepCave is its ability to visualize the activations of individual neurons in the network. This allows users to see which features in the input image are being detected by specific neurons, and can help users to better understand how the network is making predictions. In addition to visualizing individual neuron activations, DeepCave also provides a variety of tools for exploring different layers of the network. Users can navigate through the layers of the network, and can view the activations of entire layers at once to get a broader understanding of the features being learned by the network.

Another useful feature of DeepCave is its ability to adjust the input images and see how the network's predictions change in response. Users can modify the input images in various ways, such as changing the color balance or adding noise, and can observe how these modifications affect the network's predictions. Overall, the DeepCave tool provides a powerful set of tools for analyzing and understanding the features learned by a CNN. Its interactive interface and visualization tools make it easy for users to explore and experiment with different aspects of the network, and can help to reveal insights into how the network is making its predictions.

[4] The authors argue that climate modeling and analysis is a complex and computationally expensive task that can benefit from the use of AutoML techniques, which can help automate the selection, tuning, and optimization of machine learning models. They also point out that the climate science community has a wealth of data that could be used to train and validate such models, making it an ideal domain for applying AutoML. The paper presents a series of case studies where AutoML techniques were successfully applied to climate data analysis tasks, such as predicting rainfall and sea surface temperatures. The authors conclude that AutoML can help accelerate climate research, improve the accuracy of climate models, and enable faster decision-making on climate policy.

[5] In this study, we provide LightAutoML, a scalable AutoML solution for handling massive datasets. The authors discuss the difficulty of automating the machine learning pipeline for huge datasets and provide a solution that is flexible enough to handle different types of data and models. The architecture of LightAutoML, which has a number of parts, including data processing, feature generation, model selection, and hyperparameter tuning, is described in the paper. Additionally, the authors present a novel method known as stacking with multi-level hyperparameter optimisation (MHO), which enables the system to integrate various models to enhance performance. The report provides experimental findings on numerous large-scale datasets that show how effective LightAutoML is at creating high-quality models with little to no manual input. The authors demonstrate that LightAutoML outperforms other AutoML implementations by comparing them.

III. METHODOLOGY

Our AutoML performs Automated Machine Learning (AutoML) using Streamlit, a popular Python web application framework for building data-driven applications. It allows users to upload a CSV data file, preview the data, and perform Automated Exploratory Data Analysis using pandas profiling. Users can then select a target variable and task type (regression or classification), and the script trains and evaluates several machine learning models using cross validation and GridSearchCV to find the best model. In order to do classification tasks, the script displays evaluation metrics and a classification report before training the best model on the entire training set and making predictions on the test set.

The primary goal of this project is to create an AutoML tool that automates the process of selecting a model, performing exploratory data analysis, and fine-tuning hyperparameters.. The proposed tool is implemented in Python

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

using various libraries, including pandas, numpy, scikit-learn, and streamlit. The methodology used in this research includes the following steps:

A. *DataPreparations:*

Preparing the data is the first stage in the suggested process. Using the Streamlit file uploader, the researcher uploads the CSV data file. A preview of the data is then shown using the Streamlit write function after the uploaded data has been read into a pandas dataframe.

B. *Automated Exploratory Data Analysis:*

The second step in the methodology is automated exploratory data analysis. The researcher uses pandasprofiling to generate an automated exploratory data analysis report. The report provides information about the dataset, including the number of observations, variables, missing values, unique values, and statistical summaries of the variables.

C. *Model Selection and Hyperparameter Tuning:*

Model selection and hyperparameter tuning make up the third step of the process. Using the Streamlitselectbox, the user chooses the target variable and the task type (regression or classification). The train_test_split function from scikitlearn is used to divide the data into training and test sets. The scikit-learn Pipeline and Column Transformer classes are used to describe the preprocessing processes for numerical and categorical features. The researcher then specifies each model's pipeline and associated hyperparameters. The models employed in this study are Logistic Regression, Decision Tree Classification, Linear Regression, Decision Tree Classification, Random Forest Classification, and Decision Tree Classification. GridSearchCV from scikit-learn is used to fine-tune the models' hyperparameters, and the model with the greatest cross-validation score is chosen as the best one.

D. *Model Training and Evaluation:*

Model training and evaluation is the methodology's fourth step. Predictions are produced on the test set after the best model has been trained on the entire training set. Mean Squared Error (MSE) and R2 Score are two of the metrics used to evaluate regression tasks. Three metrics—weighted average accuracy, weighted average precision, and weighted average recall—are used to assess classification jobs. The write function of Streamlit is used to display these metrics.

E. *Sampling Methodology:*

The code does not specifically mention the dataset that was used in this study. As a result, the sampling strategy cannot be identified. However, the data is divided into training and test sets using the scikit-learn train_test_split method. In order to guarantee that the same split is obtained each time the code is run using the same random state value, this function randomly divides the data into training and test sets.

F. *Data Analyis:*

The proposed AutoML tool includes an automated exploratory data analysis report generated using pandasprofiling. The report provides detailed information about the dataset, including the number of observations, variables, missing values, unique values, and statistical summaries of the variables. The report also includes correlations between variables, missing value analysis, and variable type analysis.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

G. *Validation:*

The provided code implements an automated machine learning pipeline using various algorithms to perform regression or classification tasks, depending on the user's input. The methodology consists of several steps, including data pre-processing, model training and selection, and performance evaluation. The data is first pre-processed, where numeric features are imputed with the mean and scaled using StandardScaler, while categorical features are imputed with the most frequent value and one-hot encoded. The train_test_split method from sklearn.model_selection is then used to divide the pre-processed data into training and testing sets.

To find the optimal model for the problem at hand, various models are then trained and assessed using cross-validation and GridSearchCV. The code employs Decision Tree Regression, Linear Regression, and Random Forest Regression for regression. Logistic Regression, Decision Tree Classification, and Random Forest Classification are all used for classification. The optimal model is chosen based on a given metric (r2_score for regression and accuracy_score for classification), which is adjusted by GridSearchCV. Following the selection of the best model, the full training set is used to train the model, and the testing set is used to test predictions. Mean squared error and R2 score are employed as assessment metrics for regression activities, whereas weighted average accuracy, precision, and recall are used for classification tasks.

IV. IMPLEMENTATION

Step 1: Prepare the data.
Step 2: Performing Automated Exploratory Analysis.
Step 3: Model Selection and Hyperparameter Tuning.
Step 4: Model Training and Evaluation.
Step 5: Sampling Methodology.
Step 6: Data Analysis.
Step 7: Validation

Streamlit and scikit-learn libraries for Python were used to develop the AutoML application. Utilising cross-validation and GridSearchCV, three regression and three classification models were trained and assessed using the uploaded CSV data file. Based on the highest assessment score, which was the accuracy score for classification models and the R2 value for regression models, the optimal model was chosen. Following that, the best model was trained on the entire training set and put to the test on the test set. For the chosen task type, the evaluation metrics—MSE, R2 Score, Weighted average Accuracy, Precision, Recall, and F1 Score—were shown. A report summarising the properties of the dataset, including the missing values, data types, and automated exploratory data analysis, was produced using the pandas profiling package.

a) Dataset Used:

The dataset used in this research paper is titanic.csv, which contains information about passengers on the Titanic. It has 891 rows and 12 columns, including features such as passenger class, sex, age, and fare. The dataset can be downloaded from Kaggle or any other data science platform.

b) Preprocessing Steps:
The preprocessing steps used in this research paper are as follows:

- Missing Value Imputation: The missing values in the dataset are imputed using the SimpleImputer class from Scikit-learn. For numeric features, the mean strategy is used, and for categorical features, the most frequent strategy is used.
- Feature Scaling: The dataset's numerical features are scaled using the Scikit-learn StandardScaler class. By eliminating the mean and scaling to unit variance, the features are standardised.
- One-Hot Encoding: The categorical features in the dataset are encoded using the OneHotEncoder class from Scikit-learn. It creates a binary column for each category and returns a sparse matrix.
- Train-Test Split: Using Scikit-Learn'strain_test_split function, the dataset is divided into training and testing sets. 20% of the data are in the testing set.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

c) Computational Resources:
The code is executed on a computer with the following specifications:
- Processor: Intel Core i7
- RAM: 8GB
- Operating system: Windows 10
- Visual Studio Code IDE

d) Potential Areas of Improvements:
Some potential areas of improvement for this research paper are as follows:
- Feature Engineering: The dataset contains some features that can be combined to create new features that may be more useful for the predictive models. For example, the "Name" feature can be used to extract the title of the passenger, which may be related to their social status and survival.
- Hyperparameter Tuning: The models used in this research paper are trained using default hyperparameters. Hyperparameter tuning using methods such as GridSearchCV can improve the performance of the models.
- Ensemble Learning: The models used in this paper are trained independently of each other. Ensemble learning techniques such as stacking and blendingcan be used to combine the predictions of multiple models and improve the performance.

V. RESULTS

Our goal is to create an automated machine learning pipeline that can carry out classification or regression operations on a dataset. Our model first allows the user to upload a CSV data file, then displays a preview of the data, and performs automated exploratory data analysis (EDA) using the Pandas profiling library, if the user selects the 'Profiling' button.

The user is then given the option to choose the target variable and task type (classification or regression). The data is then divided into training and testing sets, and preprocessing methods are established using pipelines for both categorical and numerical features. Figure 3 shows the pipeline used to optimise hyperparameters for each model (linear regression, decision tree regression, random forest regression, logistic regression, decision tree classification, and random forest classification). After training the best model on the entire training set, predictions are made using the test set. After that, evaluation measures such as mean squared error and R2 score for regression tasks are shown, as well as weighted average accuracy, precision, and recall for classification tasks.

"Our study aimed to develop an automated machine learning pipeline capable of performing regression and classification tasks on a dataset. To achieve this, we created a Python script that used the Streamlit library to build a webbased interface. Users could upload a CSV data file, preview the data, and perform automated exploratory data analysis using the Pandas profiling library.

We split the data into training and testing sets and defined preprocessing steps for numeric and categorical features using pipelines. For each of the six models tested (linear regression, decision tree regression, random forest regression, logistic regression, decision tree classification, and random forest classification), we optimized hyperparameters using cross-validation and grid search.

For regression we used Car Details from Car Dekho.csv. Our results showed that the best model for regression tasks was the linear regression model, as shown in figure 1. For classification tasks we used titanic.csv, the best model was the random forest classification model, with a weighted average accuracy of 0.96, precision of 0.92, and recall of 0.94 as shown in figure 2.

In conclusion, our automated machine learning pipeline provides an efficient and effective way to perform regression and classification tasks on a dataset. The pipeline can be easily used by individuals with limited machine learning experience, allowing them to quickly and accurately analyze their data."

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

Evaluation Metrics

	Model	R2 Score
0	Linear Regression	0.7274
1	Decision Tree Regression	0.5779
2	Random Forest Regression	0.6886

Figure 1:Evaluation metrics for car details from car dekhodatasetusingregression.

Evaluation Metrics

	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	0.9618	0.9440	0.9427	0.9422
1	Decision Tree Classification	0.9618	0.9405	0.9389	0.9382
2	Random Forest Classification	0.9637	0.9511	0.9504	0.9500

Figure 2:Evaluation metrics for titanic dataset using classification.

II. CONCLUSION AND FUTURE WORK

In this research, we present a novel Python library-based method forautomatingmachine learning.Our Auto ML system uses sklearn to create predictive models, pandas and numpy to handled at a, andstreamlittodesigntheuserinterface. The entire process of creating predictive models isautomated by the model, including exploratory data analysis,datasplitting,preprocessing,andGridSearchCVevaluationof regression and classification models. Our test findingsshow that the proposed AutoML system can perform well injobs requiring classification and regression. Our method canhelpovercomethedifficultiesdatascientistsexperiencewhile developing predictive models and can speed up thecreationof machinelearningsystems.

The goal of this research's future work is to broaden theuses of our AutoML technology and enhance it even more.Byincludingmoresophisticatedandspecialised modelsintothe system and doing trialson a larger and morevarieddataset,wehopetoenhancethemodels'performance.Anomalydetectionandclusteringaretwounsupervisedl earningtasksthatwealsohopetooffersupportfor.Wealsointend to address the drawbacks of our strategy, such as theincapability to handle time-series analysis and multi-labelcategorization.

Furthermore,weaimtoinvestigatethepotentialapplications of our AutoML system in different domains,such as finance, healthcare, and social sciences, and how itcan benefit organizations and businesses. We also aim tocollaborate with domain experts to identify relevant tasks anddatasets to expand our system's applications. Finally, weacknowledge the challenges we may face in improving oursystem, such as the need for large and diverse datasets, andweplantoexplorepotentialsolutionstoaddressthesechallenges. Our roadmap for future work includes addingmore models and tasks to the system, improving the userinterface, and addressing the limitations and challenges tomake our AutoML system more effective and useful. Overall,our AutoML system has the potential to revolutionize the waydata scientists build predictive models, and we look forwardtoadvancingits capabilities andapplications.

V. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our research guide, Dr. P. Manikandan, for his guidance and support throughout the course of this research. His expertise and insights were invaluable in helping us to develop our research question, design our methodology, and analyze our data. We are grateful for his patience and encouragement.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105182 |

REFERENCES

1. Hongyu Yang, Jie Ding, Zheng Xu, and Jian Peng-“AutoMLTwo-SampleTest".–2022
2. Marc-André Zöller, Waldemar Titov, Thomas Schlegel,Marco F. Huber -"XAutoML: A Visual Analytics ToolforAutomated MachineLearning"-2022
3. René Sass, Eddie Bergman, André Biedenkapp, FrankHutter, Marius Lindauer- "DeepCAVE: An InteractiveAnalysisToolforAutomatedMachineLearnin"-2022
4. Renbo Tu,Nicholas Roberts,Vishak Prasad,SibasisNayak,PaarthJain,FredericSala,GaneshRamakrishnan, Ameet Talwalkar, Willie Neiswanger,Colin White- "AutoML for Climate Change: A Call toAction" -2022
5. AntonVakhrushev,AlexanderRyzhkov,MaximSavchenko,DmitrySimakov,RinchinDamdinov,Alexander Tuzhilin-"LightAutoML: AutoML Solutionfor a Large-ScaleData"– 2021
6. Ahmed AlEroud and Yenumula B. Reddy"AutomatedMachineLearning Techniques forNetwork IntrusionDetectionSystems:ASurvey"-2021
7. Anna-Maria Henke, Jasmin Bogatinovski, and Klaus-Dieter Althoff. "A Neophyte with AutoML: Evaluatingthe Performance and Usability of Google AutoML" –2021
8. ZijianWang,KevinTian,andRistoMiikkulainen."ChaCha:ASystemforOnlineAutomatedMachineLearning"– 2021.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details