

Classification and Regression Using Automatic Machine Learning (AutoML) – Open Source Code for Quick Adaptation and Comparison

Oguzhan Topsakal and Tahir Cetin Akinci


Abstract— This paper presents a comprehensive exploration of automatic machine learning (AutoML) tools in the context of classification and regression tasks. The focus lies on understanding and illustrating the potential of these tools to accelerate and optimize the process of machine learning, thereby making it more accessible to non-experts. Specifically, we delve into multiple popular open-source AutoML tools and provide illustrative examples of their application. We first discuss the fundamental principles of AutoML, including its key features such as automated data preprocessing, feature engineering, model selection, hyperparameter tuning, and model validation. We subsequently venture into the hands-on application of these tools, demonstrating the implementation of classification and regression tasks using multiple open-source AutoML tools. We provide open-source code samples for two data scenarios for classification and regression, designed to assist readers in quickly adapting AutoML tools for their own projects and in comparing the performance of different tools. We believe that this contribution will aid both practitioners and researchers in harnessing the power of AutoML for efficient and effective machine learning model development.

Index Terms— AutoML, Machine Learning, Artificial Intelligence, Code, Adaptation, Sample, Classification, Regression


I. INTRODUCTION

MACHINE LEARNING (ML) has experienced substantial advancements in recent years, becoming an indispensable tool in an expanding array of academic and industrial disciplines [1-3]. The efficacy and accuracy of machine learning models are contingent on a multitude of factors, each of which necessitates significant expertise and labor.

OGUZHAN TOPSAKAL, Department of Computer Science, Florida Polytechnic University, 33805, FL, USA. (e-mail: otopsakal@floridapoly.edu).

 <https://orcid.org/0000-0002-9731-6946>

TAHIR CETIN AKINCI, Department of Electrical Eng. Istanbul Technical University (ITU), Turkey, and University of California Riverside (UCR), CA, USA (e-mail: tahircetin.akinci@ucr.edu).

 <https://orcid.org/0000-0002-4657-6617>

Manuscript received Jun 11, 2023; accepted July 31, 2023.

DOI: [10.17694/bajece.1312764](https://doi.org/10.17694/bajece.1312764)

Such factors include the preliminary task of data preprocessing and cleansing, the selection and creation of pertinent features, the determination of an optimal model family, the tuning of model hyperparameters, and, in the case of deep learning applications, the design of neural networks. Furthermore, a critical evaluation of the results derived from these models forms an integral component of the process [4, 5].

With the primary objective of democratizing the application of machine learning models, research has been undertaken to automate these tasks, thus eliminating the need for extensive knowledge and skill in machine learning principles. The culmination of these endeavors has resulted in the development of Automated Machine Learning (AutoML) tools [1,6]. These tools are designed to enable individuals without specialized machine learning expertise to successfully implement and gain valuable insights from machine learning models.

Automated Machine Learning (AutoML) refers to the process of automating the end-to-end process of applying machine learning to real-world problems. Traditionally, building a machine learning model involves a lot of manual work such as feature selection, model selection, hyperparameter tuning, etc., which require considerable expertise and time [7]. AutoML aims to automate these manual, time-consuming aspects of machine learning, making the technology more accessible and efficient. It's designed to make machine learning more user-friendly by enabling individuals without specific knowledge in the field to build models, improve model efficiency, and speed up the process.

Some of the steps AutoML can automate include [8-11]:

- *Data preprocessing*: It helps in cleaning the data and selecting the right features to feed into the model. This might include imputing missing values, handling categorical variables, scaling and normalizing data, and more.
- *Feature engineering*: It is the process of creating new features from existing ones to improve the performance of the model. AutoML can create and choose the most effective features to use.
- *Model selection*: There are many types of machine learning models (e.g., decision trees, linear models, neural networks, ensemble models), and each has its strengths and weaknesses depending on the problem at hand. AutoML

can automatically test different models on your data to find the one that works best.

- *Hyperparameter tuning*: This involves finding the optimal configuration for a given model to maximize its performance. AutoML can systematically try many combinations of hyperparameters to find the best ones.
- *Model validation and selection*: After training multiple models, AutoML can evaluate their performance and select the best one.
- *Prediction and interpretation*: Finally, AutoML can generate predictions using the best model and provide the functionality to understand the insights into the model's decisions.

AutoML systems can be especially useful for businesses and researchers with large datasets but without the time or resources to manually build and tune machine learning models. However, it's worth noting that while AutoML can help automate many tasks, understanding the fundamentals of machine learning is still important to interpret results and ensure ethical and responsible use of the technology.

Many AutoML tools have been developed to achieve various tasks at different levels of performance. With all AutoML tools, users should consider their specific needs, the nature of their data, and the problem they're trying to solve when deciding which AutoML tool to use. To compare the AutoML tools for the task at hand, researchers or practitioners should learn each tool's capabilities and develop the code to test the task with the AutoML tool.

II. SELECTED AUTOML TOOLS

We have six most frequently used, well-known AutoML tools to study in this paper.

A. AutoGluon

AutoGluon is an open-source Automated Machine Learning (AutoML) library developed by Amazon. It aims to enable easy-to-use and easy-to-extend AutoML with robust machine learning techniques and advanced features.

AutoGluon automates various steps of the machine learning pipeline such as data preprocessing, feature engineering, model selection, model training, and hyperparameter tuning. It also supports automatic ensembling and stacking of models, a powerful technique for achieving higher predictive performance.

AutoGluon is particularly known for its efficiency and flexibility. It provides users the option to have full control over the machine learning process while also allowing them to leverage automation capabilities when needed.

AutoGluon is designed to work with different types of data including tabular data, image data, and text data. AutoGluon's functionality has demonstrated competitive performance in several machine learning competitions and has been utilized in real-world applications, making it a strong contender in the AutoML field [12].

B. AutoKeras

AutoKeras is an open-source Automated Machine Learning (AutoML) library developed by DATA Lab at Texas A&M University. It's built on top of the popular deep learning library Keras, and it aims to make machine learning accessible to non-experts and improve the efficiency of experts in model development [13].

The primary focus of AutoKeras is to automate the process of model selection and hyperparameter tuning, thereby reducing the manual, often time-consuming, trial-and-error involved in designing optimal neural network architectures.

The main features of AutoKeras include automated model architecture search, preprocessing (ex. handle missing data, categorical data), and hyperparameter tuning.

AutoKeras supports multiple types of data and tasks, including image classification, text classification, and regression problems, and continues to evolve and expand its capabilities to encompass a broader range of applications [14-16].

C. Auto-Sklearn

Auto-Sklearn is an open-source library in Python for performing Automated Machine Learning (AutoML). It is an extension of the popular Scikit-learn machine learning library and is built to automate the process of selecting the right machine learning model and tuning its hyperparameters [17].

The main features of Auto-sklearn include automated model selection, automated hyperparameter tuning, automated preprocessing. Auto-Sklearn uses ensemble methods to combine the predictions of multiple models, which can often lead to better predictive performance. Auto-sklearn is designed to be used with a minimal amount of code and has been engineered to fit into the Scikit-learn ecosystem, making it easy for those who are already familiar with Scikit-learn to start using Auto-sklearn [18, 19, 20].

D. H2O

H2O's AutoML is an automated machine learning tool developed by H2O.ai, which is well-known for its scalable and fast machine learning platform. The H2O AutoML library provides automated model selection, hyperparameter tuning, and ensemble learning capabilities, thus simplifying the process of building machine learning models [21].

The key features of H2O's AutoML include automated model selection, automated hyperparameter tuning, and automated ensemble learning: H2O's AutoML automatically trains two kinds of ensemble models at the end of its run - one is a simple ensemble using uniform weights, and the other uses stacking, a technique that uses a meta-learning algorithm [22-24] to learn how to best combine the predictions from multiple models. H2O's platform is designed for scalability and can handle large datasets and complex computations efficiently.

E. PyCaret

PyCaret is an open-source, low-code machine learning library in Python [24-25] that allows you to go from preparing your data to deploying your model within seconds. It is designed to expedite the process of building machine learning

pipelines for both binary and multiclass classification problems, regression problems, and clustering, among others [24-28].

The key features of PyCaret's AutoML include preprocessing, model selection, hyperparameter tuning, model analysis, and ensembling. It also provides a method for stacking models where a meta-model is trained on the predictions of base models.

F. TPOT

TPOT (Tree-based Pipeline Optimization Tool) is an open-source Automated Machine Learning (AutoML) tool developed in Python. The main goal of TPOT is to automate the process of building optimal machine learning pipelines, making it easier for researchers and data scientists to build efficient machine learning models [29].

One of the key features of TPOT includes automated pipeline optimization using genetic programming. TPOT also has the functionality of automated feature preprocessing, feature selection, model selection, and hyperparameter tuning.

TPOT is built on top of Scikit-learn, a popular machine learning library in Python. This makes it compatible with Scikit-learn's extensive range of functions and features.

Although TPOT automates the machine learning process, it also allows users to customize the search space and other parameters for the genetic programming algorithm, providing a balance between automation and control. After TPOT finds the optimal pipeline, it can export the corresponding Python code. This enables users to understand the pipeline's structure and make further customizations if necessary.

TPOT is computationally intensive and may require a lot of time and computational resources to find the optimal pipeline, particularly for large and complex datasets [29-32].

III. CODE SAMPLES

In this study, we also provide open-source code samples for regression and classification tasks for several well-known AutoML tools. The availability of code samples for AutoML tool will help compare and adapt the tools. The code samples provide practical examples of how to use the tool. This is especially useful for beginners or those transitioning from other tools, as it helps them understand how to apply the tool to real-world scenarios. Moreover, the code will help the researcher save time by letting them utilize the code rather than writing code from scratch. The samples also showcase different features of the tool, illustrating how to use and combine these features to achieve the desired results. A working example of the code is again beneficial when users encounter problems or errors while using the tool, they can refer to the code samples to check how certain functions or features should be used correctly. We hope the sample code will encourage adaptation by reducing the barrier to entry for using the tool.

Jupyter Notebook pages including sample code for AutoML tools AutoGluon, AutoKeras, Auto-Sklearn, H2O, PyCaret and TPOT can be downloaded from the GitHub page [33]. The code has been tested on Google Colab and includes samples for both regression and classification. The dataset used for regression includes features to predict used car values [34]. The dataset

used for classification includes the data from the well-known Kaggle's Titanic competition [35].

IV. DISCUSSION

Adopting Automated Machine Learning (AutoML) tools into the machine learning process comes with several challenges:

- *Interpretability and transparency:* While AutoML tools can make machine learning more accessible, they can also act as "black boxes" where it's hard to understand why a particular decision or prediction has been made. This can make it difficult for users to trust the system and can be a barrier to adoption, especially in domains where explainability is critical, like healthcare or finance.
- *Lack of customization:* AutoML tools are designed to automate and simplify many steps in the machine learning process, but this can also limit their flexibility. Advanced users or those with specific needs may find that they don't have as much control or ability to customize the model as they would with traditional machine learning approaches.
- *Resource consumption:* AutoML methods often involve exploring a large number of different models and hyperparameters, which can be computationally intensive and time-consuming, especially for larger datasets or more complex models.
- *Data privacy and security:* Like all machine learning approaches, AutoML needs access to potentially sensitive data to train models. Ensuring that this data is used and stored securely is a critical challenge.
- *Quality of input data:* The quality of the results produced by AutoML is heavily dependent on the quality of the input data. AutoML can automate many parts of the machine learning process, but it may still struggle with poorly-formatted, inconsistent, or biased data. Users of AutoML tools may not always have the expertise to recognize and address these issues.
- *Evaluation of results:* While AutoML tools can automate the process of evaluating and comparing different models, interpreting these results can still be challenging, especially for non-experts. Users may need a solid understanding of machine learning concepts to make sense of the results and choose the best model for their needs.

In spite of these challenges, the field of AutoML is rapidly evolving, and many ongoing research efforts are aimed at addressing these issues. The ultimate goal is to create systems that can automate as much of the machine learning process as possible, while still being transparent, customizable, and easy to use.

V. CONCLUSION

In this research paper, we have embarked on a deep investigation into the realm of Automated Machine Learning (AutoML) tools, specifically focusing on their application in

classification and regression tasks. Our exploration aimed to reveal the immense potential of AutoML tools in streamlining and optimizing the process of machine learning, essentially democratizing the field by making it more approachable to non-experts.

Our discourse covered the fundamental principles of AutoML, elucidating key features such as automated data preprocessing, feature engineering, model selection, hyperparameter tuning, and model validation. We extended this theoretical understanding into practical application, demonstrating the implementation of AutoML tools in real-world classification and regression tasks.

In doing so, we illustrated the significant capabilities of these tools, highlighting their ability to quickly adapt to varied data scenarios and problem statements. Our findings confirmed the proficiency of AutoML tools in efficiently and effectively developing machine learning models.

To further aid in the comprehension and utilization of AutoML, we have made available a broad set of open-source code. This repository is designed to provide immediate assistance to readers, enabling them to swiftly adapt AutoML tools for their projects and perform comparative analysis between different tools.

In conclusion, our study underpins the value and impact of AutoML tools in the modern data-driven era. We posit that these tools will play a crucial role in the future of machine learning, enabling professionals and researchers alike to harness their power for efficient and effective model development. We hope that our contributions in this study will empower more individuals to embrace and exploit AutoML, leading to novel insights and breakthroughs in various fields.

REFERENCES

- [1] Patil, P. S., Kappuram, K., Rumao, R., & Bari, P. (2022, May). Development Of AMES: Automated ML Expert System. In 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) (Vol. 1, pp. 208-213). IEEE.
- [2] Glasby, L. T., Whaites, E. H., & Moghadam, P. Z. (2023). Machine Learning and Digital Manufacturing Approaches for Solid-State Materials Development. *AI-Guided Design and Property Prediction for Zeolites and Nanoporous Materials*, 377-409.
- [3] Pugliese, R., Regondi, S., & Marini, R. (2021). Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management*, 4, 19-29.
- [4] Lu, S. C., Swisher, C. L., Chung, C., Jaffray, D., & Sidey-Gibbons, C. (2023). On the importance of interpretable machine learning predictions to inform clinical decision making in oncology. *Frontiers in Oncology*, 13, 780.
- [5] Manduchi, E., & Moore, J. H. (2021). Leveraging automated machine learning for the analysis of global public health data: a case study in malaria. *International Journal of Public Health*, 31.
- [6] Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges* (p. 219). Springer Nature.
- [7] Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020, March). Automated machine learning: The new wave of machine learning. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 205-212). IEEE.
- [8] Singh, V. K., & Joshi, K. (2022). Automated Machine Learning (AutoML): An overview of opportunities for application and research. *Journal of Information Technology Case and Application Research*, 24(2), 75-85.
- [9] Heizmann, M., Braun, A., Glitzner, M., Günther, M., Hasna, G., Klüber, C., ... & Ulrich, M. (2022). Implementing machine learning: chances and challenges. *at-Automatisierungstechnik*, 70(1), 90-101.
- [10] Majidi, F., Openja, M., Khomh, F., & Li, H. (2022, October). An Empirical Study on the Usage of Automated Machine Learning Tools. In 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 59-70). IEEE.
- [11] Mengi, G., Singh, S. K., Kumar, S., Mahto, D., & Sharma, A. (2023, February). Automated Machine Learning (AutoML): The Future of Computational Intelligence. In International Conference on Cyber Security, Privacy and Networking (ICSPN 2022) (pp. 309-317). Cham: Springer International Publishing.
- [12] Erickson, Nick & Mueller, Jonas & Shirkov, Alexander & Zhang, Hang & Larroy, Pedro & Li, Mu & Smola, Alexander. (2020). AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. ArXiv, abs/2003.06505.
- [13] M. Feurer, A. Klein, K. Eggenberger, J. T. Springenberg, M. Blum, and F. Hutter, "Auto-sklearn: Efficient and Robust Automated Machine Learning," *Automated Machine Learning*, pp. 113–134, 2019, doi: https://doi.org/10.1007/978-3-030-05318-5_6.
- [14] Alaiad, A., Migdady, A., Al-Khatib, R. E. M., Alzoubi, O., Zitar, R. A., & Abualigah, L. (2023). Autokeras Approach: A Robust Automated Deep Learning Network for Diagnosis Disease Cases in Medical Images. *Journal of Imaging*, 9(3), 64.
- [15] Filippou, K., Aifantis, G., Papakostas, G. A., & Tsekouras, G. E. (2023). Structure Learning and Hyperparameter Optimization Using an Automated Machine Learning (AutoML) Pipeline. *Information*, 14(4), 232.
- [16] Vincent, A. M., & Jidesh, P. (2023). An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms. *Scientific Reports*, 13(1), 4737.
- [17] Jin, H., Chollet, F., Song, Q., & Hu, X. (2023). AutoKeras: An AutoML Library for Deep Learning. *Journal of Machine Learning Research*, 24(6), 1-6.
- [18] Lee, S., Kim, J., Bae, J. H., Lee, G., Yang, D., Hong, J., & Lim, K. J. (2023). Development of Multi-Inflow Prediction Ensemble Model Based on Auto-Sklearn Using Combined Approach: Case Study of Soyang River Dam. *Hydrology*, 10(4), 90.
- [19] Feurer, M., Eggenberger, K., Falkner, S., Lindauer, M., & Hutter, F. (2022). Auto-sklearn 2.0: Hands-free automl via meta-learning. *The Journal of Machine Learning Research*, 23(1), 11936-11996.
- [20] Shi, M., & Shen, W. (2022). Automatic Modeling for Concrete Compressive Strength Prediction Using Auto-Sklearn. *Buildings*, 12(9), 1406.
- [21] LeDell, E., & Poirier, S. (2020, July). H2o automl: Scalable automatic machine learning. In Proceedings of the AutoML Workshop at ICML (Vol. 2020).
- [22] Singh, K., & Malhotra, D. (2023). Meta-Health: Learning-to-Learn (Meta-learning) as a Next Generation of Deep Learning Exploring Healthcare Challenges and Solutions for Rare Disorders: A Systematic Analysis. *Archives of Computational Methods in Engineering*, 1-32.
- [23] Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-Plan: Automated machine learning via hierarchical planning. *Machine Learning*, 107, 1495-1515.
- [24] Whig, P., Gupta, K., Jiwani, N., Jupalle, H., Kouser, S., & Alam, N. (2023). A novel method for diabetes classification and prediction with Pycaret. *Microsystem Technologies*, 1-9.
- [25] Huynh, T., Mazumdar, H., Gohel, H., Emerson, H., & Kaplan, D. Evaluating the Predictive Power of Multiple Regression Models for Groundwater Contamination using PyCaret-23489.
- [26] Liu, X., Wu, J., & Chen, S. (2023). Efficient hyperparameters optimization through model-based reinforcement learning with experience exploiting and meta-learning. *Soft Computing*, 1-18.
- [27] Pol. U.R. and Sawant, T.U. "Automl: building an classification model with PyCaret", *YMER*, vol. 20, pp. 547-552, Dec. 2021, doi: 10.37896/YMER20.11/50
- [28] N. Sarangpure, V. Dhamde, A. Roge, J. Doye, S. Patle and S. Tamboli, "Automating the Machine Learning Process using PyCaret and Streamlit," 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/INOCON57975.2023.10101357.
- [29] R. S. Olson et al., "TPOT", Accessed on March 3, 2023, Available: <http://epistasislab.github.io/tpot/>
- [30] Chen, X., Xu, J., Zhou, H., Zhao, Y., Wu, Y., Zhang, J., & Zhang, S. (2023). Tree-based machine learning models assisted fluorescent sensor array for detection of metal ions based on silver nanocluster probe. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 297, 122738.

- [31] Xiang, C. Y., Gao, F., Jakovlić, I., Lei, H. P., Hu, Y., Zhang, H., ... & Zhang, D. (2023). Using PhyloSuite for molecular phylogeny and tree-based analyses. *iMeta*, 2(1), e87.
- [32] Grjazniha, M. (2023). Performance and Competitiveness of Tree-Based Pipeline Optimization Tool (Doctoral dissertation).
- [33] Github: <https://github.com/research-outcome/automl-sample>
- [34] Regression Dataset Based on a Used Card Dataset at Kaggle: <https://www.kaggle.com/datasets/lepchenkov/usedcarscatalog>
- [35] Classification Dataset Adapted from Kaggle Titanic Competition: <https://www.kaggle.com/competitions/titanic>

BIOGRAPHIES



OGUZHAN TOPSAKAL (Senior Member IEEE) received his B.S. in Computer Engineering in 1996 from Istanbul Technical University, Turkey. He received his M.S. and Ph.D. in Computer Science from the University of Florida in 2003 and 2007.

After gaining extensive experience in the software industry, Dr. Topsakal is currently an assistant professor in the Computer Science department at the Florida Polytechnic University since 2018. He teaches courses related to Machine Learning,

Algorithm Design, Databases, and Mobile Development. Dr. Topsakal's research interests include applications of machine learning and deep learning in the medical field.



TAHIR CETIN AKINCI (Senior Member IEEE) pursued his Bachelor's degree in Electrical Engineering in 2000, followed by his Master's and Ph.D. degrees in 2005 and 2010, respectively.

From 2003 to 2010, he worked as a Research Assistant at Marmara University in Istanbul, Turkey. Dr. Akinci is currently a full professor in the Electrical Engineering Department at ITU in 2021. Dr. Akinci assumed the role of a visiting scholar at the University of California Riverside (UCR). His research interests including artificial

neural networks, deep learning, machine learning, cognitive systems, signal processing, and data analysis. In 2022, Dr. Akinci was honored with the International Young Scientist Excellence Award as well as the best researcher award for his exceptional research achievements.