The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2022

Volume 19, Pages 35-40

IConTech 2022: International Conference on Technology

End-to-End AutoML Implementation Framework

Muhammet Ali KADIOGLU
Istanbul Technical University

Abstract: Automated machine learning (AutoML) has been an active research area in recent years. Researchers investigate the potential of AutoML as more stakeholders want to maximize the value of their data. The methods are designed to increase the effectiveness of machine learning (ML), accelerate model development processes, and make it accessible for domain experts that are not ML professionals. The systems without the aid of humans are feasible with AutoML, an area that has been increasingly studied recently. Even though efficiency and automation are two of AutoML's key points, a number of critical steps still require human involvement, such as understanding the characteristics of domain-specific data, defining prediction problems, creating a suitable training dataset, and choosing a promising ML technique. A comprehensive and updated analysis of the state-of-the-art in AutoML is presented in the study. AutoML techniques, including hyperparameter optimization (HPO), feature engineering, and data preparation are presented. As-is prediction structure and AutoML-based benchmark model are compared to show how to implement these methods. It is stated what a real end-to-end machine learning pipeline looks like and which parts of the pipeline have already been automated. Our AutoML implementation framework has been introduced and presented as a road map for the entire ML pipeline. Several unresolved issues with the current AutoML techniques are discussed. The obstacles have been outlined that must be overcome in order to achieve this objective.

Keywords: Automated Machine Learning (AutoML), Hyperparameter Optimization (HPO), Data preparation, Machine learning pipeline

Introduction

A unique idea of automating the whole ML process has emerged to lower development costs. AutoML, which combines automation and ML, involves the automated structure of a pipeline with the limited computational budget. AutoML enable domain experts to automatically create applications without high level statistic or ML knowledge. It has become a hot topic in both industry and academia with the exponential growth of computing power (He, Zhao, & Chu, 2020). The professionals are becoming more dependent on AutoML techniques as advances in ML are released quicker than researchers can integrate them. This encourages researchers to study AutoML methods and systems rather than only ML algorithms (Milutinovic, et al., 2020). Studies that produce high quality outputs in the fields of natural language processing and image processing are raised with the recent rapid progress in deep learning methods (Kadıoğlu & Takcı, 2022). Even an open source AutoML tool like H2O made it possible to work on text data.

Providing cost advantage in development processes is one of the main financial goals of almost all technology companies. Various solution proposals have been developed on how to use this advantage (Kadıoğlu, 2021). Automating the processes is one of the significant solutions to get more work output with less qualified human resources in development. AutoML tools provide a simple web GUI or API to train many models or a powerful single model. It can be a helpful tool for either a novice or advanced machine learning practitioner. There is still a fair bit of expertise that is required to achieve state-of-the-art results. It simplifies the training and tuning of ML models by offering a single function to replace a process that would typically require tons of code lines. AutoML saves time to focus on data preprocessing, feature engineering, and model deployment (LeDell & Poirier, 2020).

There is not a unique optimum AutoML solution. AutoML techniques differ by their optimization method, generated pipelines, the library of algorithms they select from, used meta-learning to learn from runs on prior datasets, or performed post-processing (Gijsbers, et al., 2019). For this reason, it should be defined whether it is appropriate to develop a model with AutoML tools for the correct analysis of needs and to solve the business problem. The study provides a general overview of the AutoML and implementation framework for a ML pipeline. A proper understanding of AutoML tools and the solutions they offer is critical for positioning these tools correctly.

AutoML Methods

Hyperparameter Optimization

One of the AutoML fundamentals is to automatically configure hyperparameters to enhance performance. Especially there is a wide range of hyperparameter choices regarding the neural network's architecture, regularization, and optimization. The performance of ML algorithms and the reproducibility of scientific research can both be improved through automated hyperparameter adjustment. As one of these methods, Bayesian optimization, recently gained popularity in HPO by achieving new state-of-the-art outcomes in tuning deep neural networks. Bayesian optimization is an iterative algorithm with two key ingredients. These components are a probabilistic surrogate model and an acquisition function to decide which point to evaluate next. The surrogate model is fitted to all observations of the target function made so far in each iteration. Then the acquisition function, which uses the predictive distribution of the probabilistic model, determines the utility of different candidate points, trading off exploration and exploitation (Feurer & Hutter, 2019).

The two AutoML tools, Auto-WEKA (Thornton, Hutter, H. Hoos, & Leyton-Brown, 2013) and auto-sklearn (Feurer & Eggensperger, 2015) use Bayesian optimization to select and tune the algorithms. Auto-WEKA uses Bayesian optimization in a machine learning pipeline based on WEKA and auto-sklearn did the same using scikit-learn (Pedregosa, et al., 2011) and added meta-learning to warm-start the search with the best pipelines on similar datasets, as well as ensemble construction (Gijsbers, et al., 2019). The main idea behind the genetic algorithm, which is another optimization method and a relatively new technique, is to simulate how a species evolves by Darwin's "survival of the fittest" theory (Leardi, 2009). By using genetic programming, TPOT (Olson, et al., 2016) optimizes scikit-learn pipelines by starting with the simplest and improving them over time. Besides these methods, H2O AutoML stacks the best results of a random search in order to optimize H2O components.

Meta-Learning

Meta-learning covers all learning based on prior experience with other tasks. How different ML methods perform on a wide range of learning tasks are observed systematically. This significantly accelerates and improves the design of ML pipelines or neural architectures. Meta-data that describe prior learning tasks and previously learned models are collected. The first step includes the exact algorithm configurations and measurable properties of the task itself, also known as meta-features. The second one is to learn from this experience or meta-data. It is needed to learn from this prior meta-data, to extract and transfer knowledge that guides the search for optimal models for new tasks. There are three fundamental meta-learning methods based on the type of meta-data they leverage, from the most general to the most task specific. Learning from model evaluations includes recommending useful configurations and configuration search spaces, as well as transferring knowledge from empirically similar tasks. Learning from task properties characterizes tasks to more explicitly express task similarity and build meta-models that learn the relationships between data characteristics and learning performance. Learning from prior models covers how can be transferred trained model parameters between inherently similar tasks (Vanschoren, 2019). As an AutoML tool, auto-sklearn uses meta-learning to warm-start the Bayesian optimization procedure and includes ensemble construction which allows to use all classifiers that were found by Bayesian optimization (Gijsbers, et al., 2019).

Architecture Search

Automated neural architecture search methods can be categorized according to three dimensions. The first dimension, the search space, defines which architectures can be represented. Prior knowledge can reduce the size of the search space and simplify the search. However, this also may prevent finding novel architectural

building blocks that go beyond the current knowledge. The second dimension, the search strategy, details how to explore the search space. It aims to find well-performing architectures quickly, but at the same time, needs to be avoided premature convergence to a region of suboptimal architectures. The third dimension, performance estimation strategy, refers to the process of estimating predictive performance on unseen data. Because of to perform a standard training and validation of the architecture on data (it is the simplest option) is computationally expensive and limits the number of architectures that can be explored, recent research focuses on developing methods that reduce the cost of these performance estimations (Gijsbers, et al., 2019).

Automatic Feature Generation

Applying machine learning models to capture the correlations over different features has remarkably progressed. Self-attention mechanisms and graph neural networks are designed to generate high-order interactive features. automatic feature generation improves the effectiveness of the processes in several applications such as recommender systems. On the other hand, the explainability of these features is weakened due to the black-box nature of adopted neural networks. Search-based automatic feature generation methods are proposed to meet the demand for interpretability. Besides that, search-based methods required computation resources and training time can become intolerable when the scale of the considered data is large. Some of the studies in automatic feature generation try to find an optimal solution that possesses both feature interpretability and search efficiency. The open questions for automatic feature generation can be listed as how to exploit prior knowledge and how to balance the trade-off between the usefulness of generated features and their completeness, especially when the running time and computation resources are limited (Li, et al., 2021).

Solution Architecture

Data Pre-Processing

H2O AutoML provides automatic data pre-processing which includes automatic imputation, normalization, and one-hot encoding. Various automatic target encoding strategies has been benchmarked for high-cardinality features in experimental versions of the algorithm. H2O AutoML has a pre-processing option with minimal support for automated target encoding of high cardinality categorical variables (LeDell & Poirier, 2020).

Model Generation

H2O AutoML includes Gradient Boosting Machines (GBM), Random Forests (RF), Deep Neural Networks (DNN) and Generalized Linear Models (GLM). After training the base models, stacked ensembles is created as a class of algorithms that involves training a second level meta learner to find the best combination of the base models. Two stacked ensemble models, the "All Models" ensemble includes all the models, and the "Best of Family" ensemble includes the best performing model from each algorithm class/family, are trained using H2O’s Stacked Ensemble algorithm. Stacked Ensembles predict well if the base models are individually strong and make uncorrelated errors. Random search across a number of algorithm families produces a very diverse set of base models, and when paired with stacking, produces powerful ensembles. Generally, both ensembles produce better models than any individual model. By default, the meta learner in the Stacked Ensemble will be trained using the k-fold cross-validated predictions from the base learners. This version of stacking is called the Super Learner algorithm (Laan, J., Polley, & Hubbard, 2007) to represent an optimal system for learning.

Model Evaluation

AutoML tool includes a leader board which ranks all models by model performance. The leader board presents the information about model performance, training time and per-row prediction speed for each model trained in the AutoML run, ranked according to user preference. The models are ranked by a default metric based on the problem type. In binary classification problems, that metric can be AUC, and in multiclass classification problems, the metric can be MAPE. In regression problems, the default sort metric can be RMSE. H2O AutoML covers the leader board, provides a model list and prepares the best model to use in predictions.

Experiments

The current solution proposal is compared with the AutoML model to measure the contribution made to the business problem. If the business unit does not currently use a solution for the business problem and a solution proposal is to be developed from scratch, a benchmark model is developed that uses the methods in the business unit's skill set. In order to develop a benchmark model, the business needs should be understood correctly, the current conditions and capabilities of the business unit should be well defined, and the outputs of business and data analysis should be used as a substantial resource. Communication and mutual knowledge transfers between the business unit and the data science team are vital in modelling the current situation. The business unit's familiarity with data science as well as the data science team's knowledge about the business unit's operational processes can be a crucial factor in strengthening this mutual communication.

The benefit of the data science team to the business unit is measured by comparing the developed model performance with the estimation method used in the current structure. Where there is no estimation method used in the existing structure, a model that represents the current state can be designed using simple statistical methods, or a reference model can be developed using AutoML tools to set a more challenging target. Even in some business problems, the ML model developed by the data science team may not outperform the model developed with AutoML.

Table 1. Confusion matrix of As-Is model

	Predicted - False	Predicted - True
Actual - False	24853	6212
Actual - True	6020	1619

The confusion matrix provides to measure how well the model does the classification task by comparing the predicted values with the actual ones. It is used to evaluate the accuracy of a model created for a classification problem. The actual and predicted values in the confusion matrix create performance criteria such as accuracy, recall, precision, and F1-score and they are calculated by the following formulas. Test results that correctly or wrongly indicate the presence of a condition are used to define the performance of the classification (Kadıoğlu & Işıklı, 2022). The outputs of the benchmark models that show the as-is situation and developed using AutoML can be seen in Table 1 and Table 2.

Table 2. Confusion Matrix of AutoML Best Model

	Predicted - False	Predicted - True
Actual - False	19262	11803
Actual - True	1979	5660

For a classification problem, the AutoML solution framework has been implemented and the results are monitored. Target variable is defined, and the features which are learned by business units has been added to the dataset. Feature types are defined to the model and, response should be a factor for binary classification. AutoML uses DRF, GLM, XGBoost, GBM algorithms and produce the model depend on their AUC scores. H2O AutoML runs for 20 base models and Stacked Ensemble of all models is used as best model. The performance metrics of the AutoML model with better classification performance are shown in Table 3.

Table 3. Model Performance Scores f or Benchmark Model and AutoML Experiments

Experiments	AUC	F1-Score
As-Is Model	0.5034	0.2054
AutoML - Best Model	0.6804	0.4509

Conclusion

There are many open source AutoML tools allow novice users to create useful ML models. However, defining prediction problems has significant challenges. The data scientist must work with the domain expert to understand the context of the business problem. Many vital steps of this ML process are generally still done manually by the data scientist. This often requires a lot of work between the data scientists and domain experts, making the whole process more difficult and inefficient. Data science project methodologies emphasize the importance of collaboration between domain experts and data scientists and how it can be improved (Kadıoğlu & Takcı, 2022). To improve these collaborations, the difficulties can be overcome by following the relevant

main development areas. On the other hand, a significant portion of the ML process today requires involvement from a data scientist, making the whole process inefficient and inaccessible to a wider audience. First, formulating a prediction problem is challenging, and there is currently no established standard for accomplishing this task systematically. The challenges, automated task formulation, effective prediction engineering, and the recommendation of useful tasks, must be addressed to reach the eventual goal of an automated ML process (Karmaker, et al., 2021).

There are similar studies tries to reduce human bias in the search space and develop a solution to the automatic discovery of whole ML algorithms from basic operations with minimal restrictions on form (Real, Liang, So, & Le, 2020). In addition to these studies examined in the literature review, the performance of two solution architectures, which express the current situation and developed using AutoML, was examined and it was seen that AutoML constitutes a substantial starting point in data science projects.

Scientific Ethics Declaration

The author declares that he is solely responsible for the scientific, ethical, and legal aspects of the paper published in EPSTEM.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Technology (www.icontechno.net) held in Antalya/Turkey on November 16-19, 2022.

References

Feurer, M., & Hutter, F. (2019). Hyperparameter Optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Ed.), *Automated machine learning methods, systems, challenges* (pp 3-33). New York, NY: Springer.

Feurer, M., Klein, A., & Eggensperger, K. (2015). Efficient and robust automated machine learning. *Advances in neural information processing systems*, *28*, 2944-2952.

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019). An open source AutoML Benchmark. *6th ICML Workshop on Automated Machine Learning.*

He, X., Zhao, K., & Chu, X. (2020). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems,212*.

Kadıoğlu, M. A. (2021). A decision support tool to evaluate software outsourcing locations with STI indicators: A case study Pakistan. *European Journal of Science and Technology*, (24), 405-415.

Kadıoğlu, M. A., & Işıklı, E. (2022). *Discovering market insights from online product reviews through sentiment analysis*. Istanbul Technical University Institute of Science, Istanbul.

Kadıoğlu, M. A., & Takcı, H. (2022). A data science project management methodology: From development to production. *International Engineering and Technology Management Summit (ETMS 2022).*

Kadıoğlu, M. A., & Takcı, H. (2022). Converting call center recordings into valuable insights using sentiment analysis. *5th International Conference on Data Science and Applications (ICONDATA'22).*

Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., & Veeramachaneni, K. (2021). AutoML to date and beyond: Challenges and Opportunities. *ACM Computing Surveys (CSUR), 54(*8), 1-36.

Laan, V. d., J., M., Polley, E. C., & Hubbard, A. E. (2007). Super Learner. *Statistical applications in genetics and molecular biology*, *6*(1).

Leardi, R. (2009). Genetic algorithm-PLS as a tool for wavelength selection in spectral data sets. *Data Handling in Science Technology,23.*

LeDell, E., & Poirier, S. (2020). H2O AutoML: Scalable Automatic Machine Learning. *7th ICML Workshop on Automated Machine Learning (AutoML).*

Li, Y., Wang, Z., Xie, Y., Ding, B., Zeng, K., & Zhang, C. (2021). AutoML: From methodology to application. *30th ACM International Conference on Information & Knowledge Management,* (pp. 4853-4856).

Milutinovic, M., Schoenfeld, B., Martinez-Garcia, D., Ray, S., Shah, S., & Yan, D. (2020). On evaluation of AutoML systems. *7th ICML Workshop on Automated Machine Learning.*

Olson, R., Urbanowicz, R., Andrews, P., Lavender, N., Kidd, L., & Moore., J. (2016). Automating biomedical data science through tree-based pipeline optimization. *Applications of Evolutionary Computation: 19th European Conference, EvoApplications* pp. 123– 137. New York, NY: Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825-283.

Real, E., Liang, C., So, D., & Le, Q. (2020). AutoML-Zero: Evolving machine learning algorithms from scratch. *International Conference on Machine Learning*, 8007-8019.

Thornton, C., Hutter, F., H. Hoos, H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *19th ACM SIGKDD International conference on Knowledge discovery and data mining*, 847-855.

Author Information

Muhammet Ali Kadioglu
Istanbul Technical University
İstanbul, Turkey
Contact E-mail: *kadioglumuhammetali@gmail.com*

To cite this article:

Kadıoglu, M.A. (2022). End-to-end AutoML implementation framework. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 19*, 35-40.