International Journal of Smart Business and Technology
Vol. 12, No. 2 (2024), pp. 21-32
http://dx.doi.org/10.21742/ijsbt.2024.12.2.03

Enhancing Predictive Analytics Effectiveness with Evolutionary AutoML Pipelines

Florin Stoica[1*] and Laura Florentina Stoica[2]

[1*,2]*Department of Mathematics and Informatics, Faculty of Sciences, "Lucian Blaga" University, Sibiu, Romania*
[1*]*florin.stoica@ulbsibiu.ro*, [2]*laura.cacovean@ulbsibiu.ro*

Abstract

Machine learning (ML) has achieved considerable success in recent years, and an ever-growing number of disciplines rely on it. With Automated Machine Learning (AutoML) tools, organizations can unlock valuable new business insights, embed advanced AI capabilities in applications, and empower data scientists and nontechnical experts to build predictive models rapidly. AutoML tools are within broader MLOps (Machine Learning Operations) platforms, such as Oracle AutoML (OML4Py) or pure Python frameworks like FEDOT. We have built a simplified AutoML pipeline, focusing on hyperparameter optimization, based on the Optimal Multiple Kernel-Support Vector Machine (OMK-SVM) method. A benchmarking experiment was conducted to identify customers with a higher likelihood of switching from one streaming service to another movie streaming provider. The results revealed that our approach delivers best-in-class performance (SVM), and our evolutionary approach to hyperparameter optimization provides results comparable to those of the FEDOT framework.

Keywords: *SVM, AutoML, FEDOT, Customer churn*

1. Introduction

Automated Machine Learning (AutoML) provides tools and techniques to automate the end-to-end process of applying machine learning to real-world problems. AutoML makes machine learning model development accessible, efficient, and effective by reducing or eliminating the need for manual intervention in various stages of the ML workflow [1].

AutoML automates the entire ML pipeline, from data preprocessing and feature engineering to model selection, hyperparameter tuning, and evaluation. This approach allows users to build high-quality models without needing extensive ML expertise. Automating repetitive and resource-intensive tasks, AutoML reduces the time and cost associated with ML model development, allowing data scientists to focus on more strategic or creative aspects of their work [2].

The machine learning market is anticipated to experience significant growth in the coming years, driven by rising demand for predictive analytics in the economic sector [3]. With machine learning models becoming more embedded in business-critical applications, there is a growing need to enhance their accessibility and reproducibility. Since only models deployed in production can deliver real value, *time-to-market* is a crucial metric that must be optimized in any commercial ML project.

Article history:
Received (September 25, 2024), Review Result (October 30, 2024), Accepted (December 5, 2024)
*corresponding author

To shorten the time required for the deployment of high-performance Support Vector Machine (SVM) models, we propose a simplified AutoML pipeline based on the Optimal Multiple Kernel - Support Vector Machine (OMK-SVM) method [4] and genetic algorithms for hyperparameter optimization.

To evaluate the performance of our framework in comparison to others, we will use a dataset containing 175 features and 130,000 observations. This dataset is designed to identify customers who are more likely to churn from a fictional streaming service, MovieStream, to a competitor. The MovieStream service is introduced and described by Gubar [5].

Evaluating the performance of classification models is essential to selecting the best-performing models for production use in solving real-world problems. The classification models were assessed using Gain Charts and a range of performance metrics derived from the Confusion Matrix. To evaluate the performance of the model optimized by our method, we compared it with two other models developed using similar techniques.

The first model with which we made the comparison was OML4Py SVMG (Support Vector Machine Gaussian), which is a type of SVM model that uses a Gaussian kernel (also known as the Radial Basis Function or RBF kernel). This kernel allows the model to handle non-linear relationships by mapping input data into higher dimensions. It is suitable for complex classification and regression tasks with insufficient linear boundaries. The SVMG model was tuned using the AutoML framework OML4Py [6].

The second model for comparison was provided by the FEDOT framework [7]. FEDOT emphasizes pipeline-based optimization, where a pipeline represents the sequence of data transformations, feature selection, model training, and post-processing steps [8]. We chose this framework for comparison because it also uses genetic algorithms to search for the best pipeline configurations, aiming to find combinations that provide the highest predictive accuracy. The best model proposed by the FEDOT framework for the concrete problem addressed was RF (Random Forest).

The rest of this paper is organized as follows. Section 2 introduces AutoML use cases focused on optimizing machine learning models in real-world scenarios. Section 3 formally defines the AutoML pipeline optimization task and describes a simplified AutoML pipeline that uses the evolutionary hyperparameter optimization (OMK-SVM) method. Details about the metrics used in evaluating ML models are presented in Section 4. A comparative evaluation of the prediction models is presented in Section 5. Finally, Section 6 provides the conclusions and directions for future development.

2. AutoML in real-world scenarios

AutoML applications span various sectors and hold significant social and economic value. We will further illustrate the applicability of AutoML in multiple areas, including business analytics, demand forecasting, disease prediction, decision-making in advertising strategies, and forecasting product quality in manufacturing.

The paper of Nasseri et al., [9] compares tree-based ensemble methods and Long Short-Term Memory (LSTM) deep learning for retail demand prediction. It investigates the effectiveness of automated machine learning (AutoML) in generating demand forecasts, highlighting the strengths of each approach. The results indicate that while both methods perform well, tree-based ensembles offer competitive accuracy and are more interpretable, making them suitable for practical applications in retail.

Paladino et al., [10] evaluate various Automated Machine Learning (AutoML) tools for diagnosing and predicting heart disease. It examines the effectiveness of these tools in processing medical data, focusing on their accuracy, efficiency, and usability. The study compares different AutoML frameworks and highlights their strengths and weaknesses in heart disease applications. Ultimately, it concludes that while AutoML can enhance predictive capabilities, careful selection of tools and techniques is essential for optimal performance in medical contexts.

Kramer et al., [11] evaluates various AutoML tools against traditional statistical methods in the context of demand forecasting. The authors conduct empirical experiments comparing tools like Microsoft Azure Automated ML, Google Cloud AutoML Tables, and Dataiku Data Science Studio. They focus on short-term and long-term forecasting for two case studies, assessing the accuracy and efficiency of predictions. The findings suggest that AutoML can significantly improve forecasting accuracy, particularly for complex demand patterns, while streamlining the modeling process.

The chapter "Use AutoML to Predict Advertising Media Channel Sales" from Stripling and Abel [12] discusses leveraging AutoML to forecast sales across different advertising media channels. It guides readers through preparing data, selecting models, and evaluating results using low-code tools. The chapter emphasizes the advantages of AutoML in simplifying the predictive modeling process, making it accessible for non-experts, and enhancing decision-making in advertising strategies.

The research published by Kraub et al., [13] discusses the application of Automated Machine Learning (AutoML) in predictive quality for production environments. It identifies the challenges of scaling machine learning projects due to the need for manufacturing and data science expertise. The authors benchmark various AutoML systems, comparing their performance to traditional manual implementations in a production use case. The study highlights that while AutoML can provide results faster, it still requires some programming knowledge and may not consistently outperform manual approaches.

A closely related work to the one presented here is Schmitt [14], examining how AutoML can make Machine Learning (ML) more accessible in business contexts. Focusing on the H2O AutoML framework [15], it compares its performance to manually tuned ML models using real-world credit risk, insurance, and marketing datasets. The findings show that manual tuning yields slightly higher accuracy. AutoML is beneficial for rapid prototyping and democratizing ML in business, helping non-experts deploy reliable models quickly and bridging the ML expertise gap.

In the following, we will evaluate the effectiveness of the proposed AutoML pipeline in predicting customer churn. Reducing customer churn is crucial for businesses, as retaining existing customers is generally more cost-effective than acquiring new ones.

3. Formal Definition of the AutoML Pipeline Optimization Task

OML4Py framework accelerates the process of tuning ML models by employing a single-pass feed-forward approach that covers all four stages of the pipeline: data preprocessing, algorithm selection, adaptive data reduction tailored to the chosen algorithm, and hyperparameter tuning. Next, we provide the formal definition of the OML4Py pipeline optimization problem [16].

Let $A = \{A^{(1)},\ldots,A^{(R)}\}$ be a set of algorithms and let the hyperparameters of each algorithm $A^{(j)}$ have a domain $\Lambda^{(j)}$. Given a dataset D_{train} with N samples and K features, let $D_{train}^{(n,k)} \subset D_{train}$'s denote a subset with $k \le K$ features and $n \le N$ samples. Finally, let $L(A^{(j)}, D_{train}^{(n,k)})$ us denote the loss that the algorithm $A^{(j)}$ achieves $D_{train}^{(n,k)}$ when trained with hyperparameters $\lambda \in \Lambda^{(j)}$ where L there is any user-defined misclassification rate. The objective of OML4Py is to identify the optimal combination of algorithm A^*, data sample D^*_{train}, and hyperparameter setting λ^* by minimizing the average loss function L :

$$D^*_{train}, A^*, \lambda^* \in \underset{\substack{n\le N,k\le K \\ A^{(j)}\in A \\ \lambda\in\Lambda^{(j)}}}{\arg\min} L\ (A^{(j)}_{\lambda}, D^{(n,k)}_{train}) \tag{1}$$

Our approach is based on an adaptation for binary classification of the Optimal Multiple KernelSupport vector Regression (OMK-SVR) method [4]. Usually, the choice of the kernel is made empirically, and the standard Support Vector Machine (SVM) classifiers use a single kernel. Given that multiple kernels give better results than single ones, we used an evolutionary technique based on a breeder genetic algorithm for building an optimal multiple kernel utilizing a set of operations (+,*,exp). The evolutionary approach will also determine the optimal values of the SVM's parameters C and ε .

We have built a simplified AutoML pipeline whose focus was on hyperparameter optimization. Row sampling was used before the grid search procedure, the role of which is to initialize an SVM from the initial population of the genetic algorithm.

We further present a formalization of the CASH (Combined Algorithm Selection and Hyperparameter optimization) problem [17], starting from the general formulation (1) and considering that in our approach, we fixed the prediction algorithm, i.e. $A^* = SVM$ (with an optimal multiple kernel).

Given a dataset D_{train} with N samples, $D^{(n)}_{train} \subset D_{train}$ denote a subset with $n \le N$ samples. $L\ (SVM_{\lambda}, D^{(n)}_{train})$ and denote the loss the algorithm SVM achieves $D^{(n)}_{train}$ when trained with hyperparameters λ with domain Λ, where L there is any user-defined misclassification rate. The grid search procedure aims to find the hyperparameter setting $\lambda^{(0)}$ by minimizing the average loss function L :

$$\lambda^{(0)} \in \underset{\substack{n\le N \\ \lambda\in\Lambda}}{\arg\min} L\ (SVM_{\lambda}, D^{(n)}_{train}) \tag{2}$$

The aim of our simplified AutoML pipeline, depicted in Figure 1, is to find the best hyperparameter setting λ^* by minimizing the average loss function L :

$$\lambda^* \in \underset{\substack{\lambda^{(i)}\in\Lambda \\ i\le T}}{\arg\min} L\ (SVM_{\lambda^{(i)}}, D_{train}) \tag{3}$$

where T is the size of the genetic algorithm population. The pre-trained $SVM_{\lambda^{(0)}}$ is used to speed up the convergence process.

The fitness function is the prediction accuracy provided by the SVM model generated through training based on 10-fold cross-validation, using multiple kernels and SVM parameters encoded in the corresponding chromosome.

Suppose the time budget is exhausted before the stop condition is met (maximum number of generations or minimum accuracy). In that case, the set of hyperparameters with the best performance at that time is returned (λ^{best}).

The results presented in Section 5 were obtained for $n = N / 5$ in just over 120 minutes of execution (on an Intel i7-5500U 2.40GHz processor). Certainly, the accuracy of the results would be better for a subset with more samples, but the execution time would also increase.

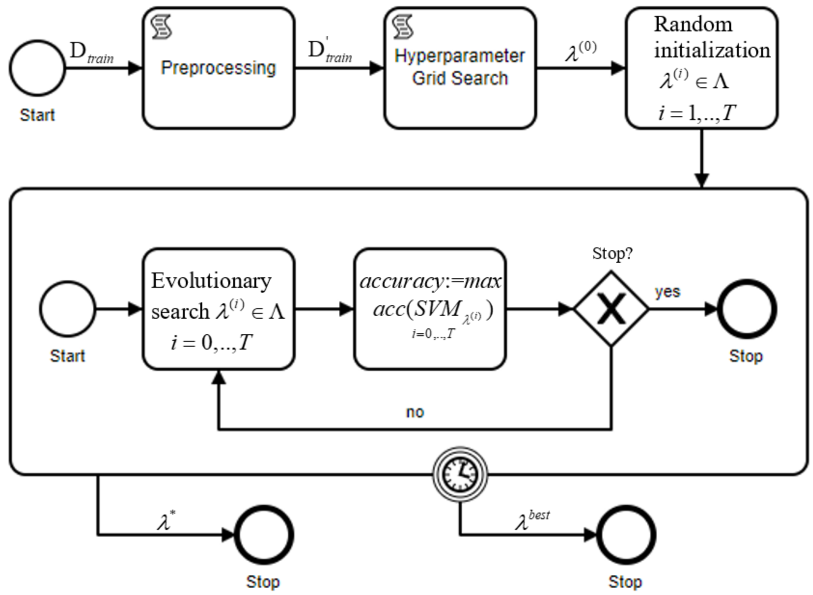

Figure 1. The simplified AutoML pipeline with evolutionary (OMK-SVM) hyperparameter optimization

We used a 4-kernel scheme, and the optimal multiple kernel obtained has the expression in equation 4:

$$K = (K_{RBF}^{\gamma_1} \times K_{SIG}^{\gamma_2}) \times (K_{RBF}^{\gamma_3} + K_{SIG}^{\gamma_4}) \quad (4)$$

It is calculated according to the tree structure shown in Figure 2, which contains the operators applied to the determined simple kernels.

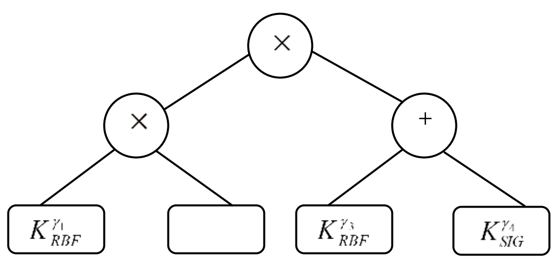

Figure 2. Tree representation of the multiple kernel

The leaves of the tree represent standard kernel functions:

- Gaussian RBF (Radial Basis Function): $K_{RBF}^{\gamma}(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$
- Sigmoid: $K_{SIG}^{\gamma}(x_i, x_j) = \tanh\left(\gamma \left\langle x_i, x_j \right\rangle + 1\right)$

Where $<\cdot,\cdot>$ is the inner product of vectors x_i x_j. The determined values of the parameters γ were: γ_1 = 20.59, γ_2 = 213.48, γ_3 = 205.45, and γ_4 = 376.92, respectively.

Figure 1 was created using the semi-formal Business Process Modeling Notation (BPMN) 2.0. BPMN is a standardized method for developing visual models that document business processes and convey the business requirements of workflows using a flowchart-like notation [18].

4. Model evaluation metrics

4.1. Model assessment using a Gain chart

Gain charts can be employed to assess predictive machine learning models by visualizing modeling statistics [19]. Gain is a metric that measures the effectiveness of a classification model, calculated as the ratio between the results obtained using the model and the results achieved without it (i.e., random outcomes).

In our case study, the machine learning models are employed to identify customers likelier to churn from MovieStream streaming services to a competing movie streaming company. We refer to a client with an actual churn value of "yes" as a "positive target." Each prediction is accompanied by a probability (prediction confidence), and the observations are organized in descending order based on the predicted probabilities of positive targets.

The cumulative gains chart illustrates the percentage of positive targets "gained" by targeting a specific percentage of the total cases. The dataset utilized to create the gain chart includes the following columns:

- CumulativeGain: This metric represents the cumulative number of positive targets ratio up to that percentage relative to the total number of positive targets.
- GainChartBaseline: This refers to the overall response rate, represented by a line that indicates the percentage of positive records we would expect to obtain if predictions were made randomly.
- OptimalGain: This denotes the ideal number of customers to target in a marketing campaign to minimize churn. The cumulative gain curve will level off beyond this point.

The Gain Chart can be utilized to analyze statistics produced by machine learning classification models, helping to identify the most effective model for use. CumulativeGain and OptimalGain will be used to evaluate the performance of the model. The closer the CumulativeGain line is to the top-left corner of the chart, the greater the gain. This indicates that a higher proportion of positive targets is reached with a smaller proportion of customers considered. OptimalGain is defined as the longest segment between the CumulativeGain and GainChartBaseline curves. The farther the CumulativeGain curve is positioned above the baseline, the greater the gain.

4.2. Performance metrics derived from the confusion matrix

A Confusion Matrix is a performance measurement tool used in machine learning to evaluate the accuracy of a classification model [1].

The confusion matrix provides a comprehensive overview of how well the model is performing by breaking down the results into four categories:

1. True Positives (TP): The number of cases where the model correctly identified a positive class.
2. True Negatives (TN): The number of cases where the model correctly identified a negative class.
3. False Positives (FP): The number of cases where the model predicted a positive class, but the actual class was negative.
4. False Negatives (FN): The number of cases where the model predicted a negative class, but the actual class was positive.

The structure of a confusion matrix for a binary classification problem is represented in Figure 3.

		Predicted	
		Positive	Negative
Actual	Positive	***TP***	***FN***
	Negative	***FP***	***TN***

Figure 3. Definition of confusion matrix components

The values of the components in the confusion matrix were used to calculate the classification metrics shown in Table 1. These metrics help assess the model's effectiveness in classifying data into distinct classes or categories.

Table 1. Performance measures in machine learning classification models

$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$	$Precision = \frac{TP}{TP + FP}$
$Recall = \frac{TP}{TP + FN}$	$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$

5. Comparative Evaluation of ML Models

To test our AutoML pipeline based on the evolutionary optimization of hyperparameters for SVM models with multiple kernels (OMK-SVM method), we chose two state-of-the-art frameworks: OML4Py and FEDOT, respectively.

OML4Py [16] optimized and tuned an SVM model with a Gaussian kernel (SVMG).

The FEDOT framework [7][8], which addresses the CASH problem also using an evolutionary approach, was used to provide an optimized model through its AutoML pipeline. The best model obtained was of the Random Forest (RF) type.

A comparison of the performance metrics for evaluating OMK-SVM, FEDOT RF, and OML4Py SVMG models is presented in Table 2.

Table 2. Performance metrics used to evaluate the effectiveness of the OMK-SVM, FEDOT RF, and OML4Py SVMG classification models

		Accuracy	Precision	Recall	F1 Score
OML4Py SVMG	Train	0.910	0.108	0.958	0.195
	Test	0.904	0.102	0.912	0.183

OMK-SVM	Train	0.999	0.969	0.941	0.955
	Test	0.992	0.795	0.463	0.585
FEDOT RF	Train	0.997	0.985	0.782	0.872
	Test	0.995	0.939	0.608	0.738

The Gain Chart and the Confusion Matrix obtained by applying the OMK-SVM model, the OML4Py SVMG model, and the FEDOT RF model on the test data set are shown in Figure 4, Figure 5, and Figure 6, respectively.

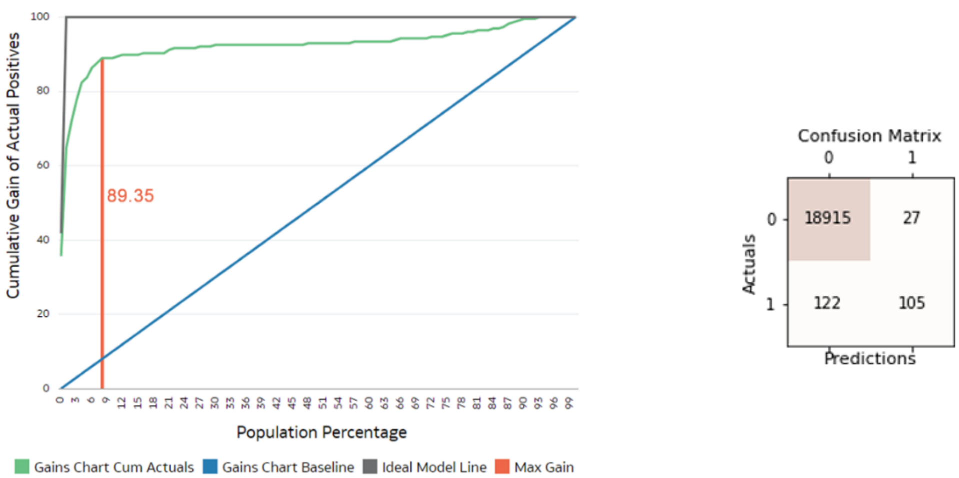

Figure 4. Performance of the OMK-SVM model on the test data

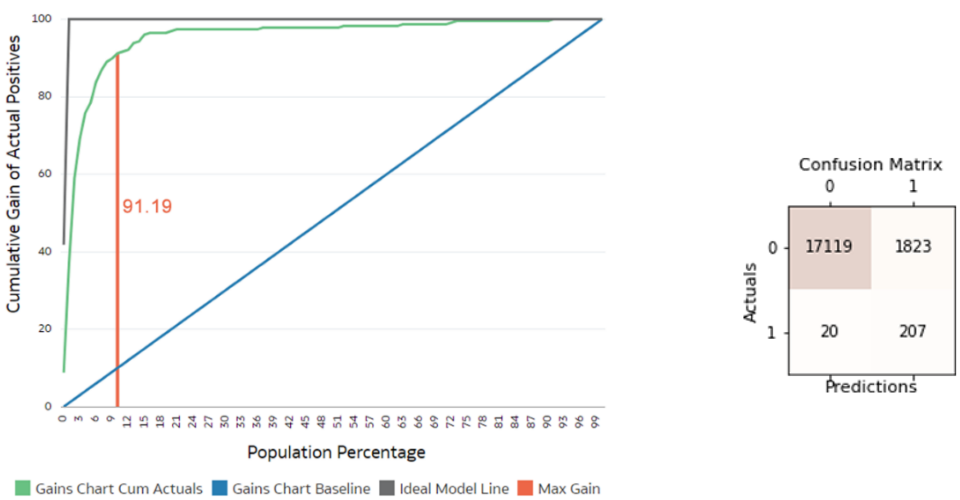

Figure 5. Performance of the OML4Py SVMG model on the test data

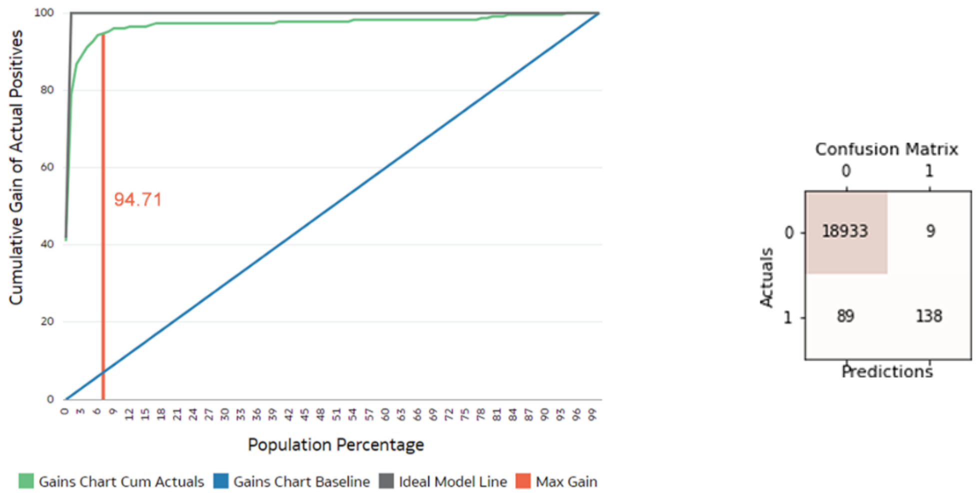

Figure 6. Performance of the FEDOT RF model on the test data

Figure 7 shows an assessment of the performances of the evaluated models using Gain Charts. The chart highlights the maximum gain and the percentage of the population for which it was reached for each model.

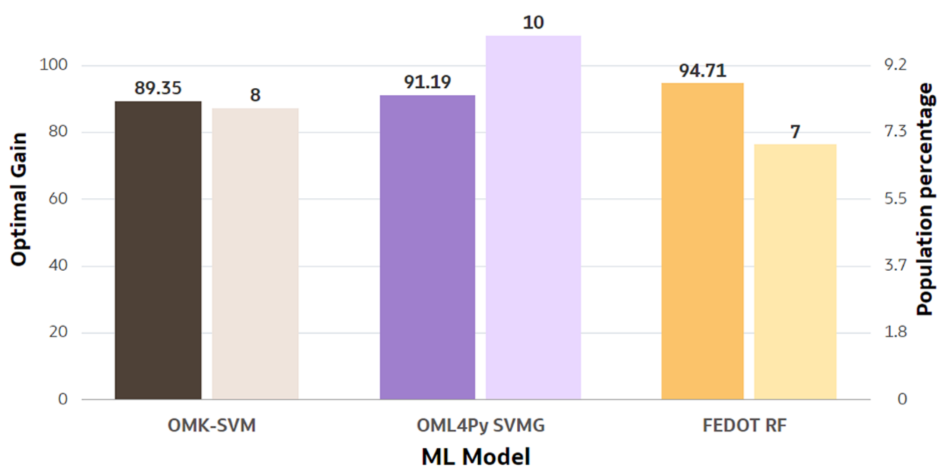

Figure 7. The maximum gain and the percentage of the population for which it was achieved

6. Conclusions and future work

Analyzing the data in Table 2, we notice that no model has the best values for all performance metrics. The OMK-SVM model offers better results than OML4Py SVMG for three of the four metrics considered, but the FEDOT RF model on test data outclasses it. FEDOT provides a superior model to OMK-SVM, with better results on test data for all four metrics used in the evaluation, according to Table 2. For the Recall metric, the best results are obtained by the OML4Py SVMG model.

However, the evaluation of the ML models with the help of the Gain chart indicates that the FEDOT RF model is the winner of this metric. The FEDOT model, applied in a marketing campaign addressed to customers with a higher likelihood of churning from the MovieStream streaming service, requires targeting only 7% of the population to obtain an optimal gain of 94.71% (the optimum number of customers to contact through the marketing campaign to avoid churning). In this ranking, the OMK-SVM model occupies the second place because it provides approximately the same value for optimal gain as the one provided by the OML4Py SVMG model, targeting only 8% of the population shown in Figure 7.

Developing effective ML pipelines is time-consuming and costly, requiring the specialized knowledge of data scientists and domain experts.

To accelerate the successful model deployment, we propose an AutoML pipeline based on an evolutionary approach dedicated to optimizing ML models built with the OMK-SVM method.

As a limitation of our study, we should mention that accuracy has been used as a score metric. This metric, used to assess the relative performance of optimized models, should be selected according to the problem being addressed and may influence the ranking of the top models identified.

The results allow us to conclude that the AutoML frameworks based on an evolutionary technique provide high-performance models. Still, they have the disadvantage of taking longer to optimize these models.

As future development directions, we intend to incorporate an extension into the proposed AutoML pipeline to enhance the explainability of prediction results [20].

References

[1] N. Shafiabady, N. Hadjinicolaou, F.U. Din, B. Bhandari, R.M.X. Wu, and J. Vakilian, “Using Artificial Intelligence (AI) to predict organizational agility,” PLOS ONE, vol.18, no.5:e0283066, (2023) DOI:10.1371/journal.pone.0283066

[2] M. Zoller and M.F. Huber, “Benchmark and survey of automated machine learning frameworks,” Journal of Artificial Intelligence Research, vol.70, pp.409-474, (2021) DOI:10.1613/jair.1.11854

[3] Research and Markets, “Automated Machine Learning (AutoML) Global Market Report,” February (2024) https://www.researchandmarkets.com/reports/5896115/global-automated-machine-learning-automl

[4] D. Simian, F. Stoica and A. Bărbulescu, “Automatic optimized support vector regression for financial data prediction,” Neural Computing and Applications, vol.32, pp.2383-2396, (2020) DOI:10.1007/s00521-019-04216-7

[5] M. Gubar, “Integrate, Analyze and Act on All Data using Autonomous Database,” (2024) https://apexapps.oracle.com/pls/apex/r/dbpm/livelabs/view-workshop?wid=889

[6] B. Tierney, “OML4Py – AutoML – An Example”, Oralytics, March 30, (2021) https://oralytics.com/2021/03/15/oml4py-automl-an-example/

[7] N.O. Nikitin, P. Vychuzhanin, M. Sarafanov, I.S. Polonskaia, I. Revin, I.V. Barabanova, G. Maximov, A.V. Kalyuzhnaya and A. Boukhanovsky, "Automated evolutionary approach for the design of composite machine learning pipelines," Future Generation Computer Systems, vol.127, pp. 109-125, (2022) DOI:10.1016/j.future.2021.08.022

[8] I.S. Polonskaia, N.O. Nikitin, I, Revin, P. Vychuzhanin and A.V. Kalyuzhnaya, "Multi-Objective Evolutionary Design of Composite Data-Driven Models," IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, June 28 – July 01, pp.926-933, (2021) DOI:10.1109/CEC45853.2021.9504773

[9] M. Nasseri, T. Falatouri, P. Brandtner and F. Darbanian, “Applying machine learning in retail demand prediction - A comparison of tree-based ensembles and long short-term memory-based deep learning,” Applied Sciences, vol 13, no. 19:11112, (2023) DOI: 10.3390/app131911112

[10] L. M. Paladino, A. Hughes, A. Perera, O. Topsakal and T. C. Akinci, “Evaluating the Performance of Automated Machine Learning (AutoML) Tools for Heart Disease Diagnosis and Prediction,” AI, vol. 4, no. 4, pp. 1036-1058, (2023) DOI:10.3390/ai4040053

[11] K. J. Kramer, N. Behn and M. Schmidt, “The potential of AutoML for demand forecasting,” Proceedings of the Conference on Production Systems and Logistics: CPSL 2022, Bremen, Germany, May 4-6, (2022) DOI:10.15488/12162

[12] G. Stripling and M. Abel, “Low-code AI: A practical project-driven introduction to machine learning,” O'Reilly, **(2023)**

[13] J. Krauß, B.M Pacheco, H.M. Zang, and R.H. Schmitt, "Automated machine learning for predictive quality in production," Procedia CIRP, vol. 93, pp. 443-448, (2020) DOI:10.1016/j.procir.2020.04.039

[14] M. Schmitt, "Automated machine learning: AI-driven decision making in business analytics," Intelligent Systems with Applications, vol.18:200188, (2023) DOI:10.1016/j.iswa.2023.200188

[15] E. LeDell and S. Poirier, “H2O AutoML: Scalable automatic machine learning”, 7th ICML Workshop on Automated Machine Learning (ICML 2020), July 12-18, (2020) https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf

[16] A. Yakovlev, H.F. Moghadam, A. Moharrer, K. Cai, N. Chavoshi, V. Varadarajan, S.R. Agrawal, S. Idicula, T. Karnagel, S. Jinturkar and N. Agarwal, “Oracle AutoML: A Fast and Predictive AutoML Pipeline," Proceedings of the VLDB Endowment, vol. 13, no.12, pp. 3166-3180, (2020) DOI:10.14778/3415478.3415542

[17] F. Hutter, L. Kotthoff and J. Vanschoren, “Automated Machine Learning. Methods, Systems, Challenges”, Springer, Switzerland, **(2019)**

[18] Business Process Model and notation™ (BPMN™) Version 2.0, The Object Management Group (OMG), January, (2014) https://www.omg.org/spec/BPMN

[19] V. Saini, "Model Evaluation using Lift and Gain Analysis - Lift and Gain Charts," November 21, (2022) https://varshasaini.in/model-evaluation-using-lift-and-gain-analysis-lift-and-gain-charts/

[20] S.M. Lundberg and Su-In Lee, "A unified approach to interpreting model predictions," Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, California, USA, December 4-9, pp.4768–4777, **(2017)**

This page is empty by intention.