COMPUTERS AND INFORMATICS
2757-8259 2023, Volume 3 Issue 1
Research Article

Genetic programming-based automated machine learning approach to solve regression problems

Maialen Murua
TECNALIA, Basque Research and Technology Alliance (BRTA), Mikeletegi 7, 2009 Donostia-San Sebastián, Spain, maialen.murua@tecnalia.com

Submitted: *14.02.2023*
Accepted: *20.02.2023*
Published: *30.06.2023*

* *Corresponding author*

Abstract:

Automated machine learning aims to optimize machine learning pipelines automatically given a dataset, task type and a target variable. This research analyzes the use of genetic programming to perform automated feature engineering in regression problems. It introduces a methodology to perform feature selection and to construct new features departing from the original feature set by combining and selecting features in the leaf nodes of the genetic programming tree. A multiple feature generation technique is proposed, where three different feature sets are tested with linear regression, Random Forest regressor and Gradient Boosting regressor. The proposed approach is applied to an industrial process dataset where the target variable is an indicator of the performance of the process. The experimental results reveal the ability of the method to reduce the cardinality of the original feature set while maintaining the performance of the learning models. Moreover, they show the ability of the newly constructed feature to better discriminate the target variable.

Keywords: *Automated machine learning, Feature engineering, Genetic programming, Predictive analytics*

Cite this paper as: Murua, M., Genetic programming-based automated machine learning approach to solve regression problems, *Computers and Informatics*, 2023, *3*(1), 19-25

1. INTRODUCTION

Predictive analytics have been widely used across a variety of domains. It reveals relationships and patterns within large volumes of data that can be used to predict behavior and events. Predictive models are commonly constructed using statistical techniques and artificial intelligence techniques, specifically machine learning (ML) techniques [1]. ML is a subfield of artificial intelligence that studies algorithms that learn from experience. ML has been broadly applied to many fields with high success, however this application has also shown that it requires considerable effort and knowledge by human experts in the field [2].

Automated ML (AutoML) aims to optimize ML pipelines automatically given a dataset, task type and a target variable [3]. This paradigm brings three main advantages: 1) releases data scientists from time consuming and trial error tests; 2) makes easier to organizations the development of ML technologies; 3) makes ML universally accessible to all domain scientists. An AutoML approach could tackle with one or more than one task that belongs to an ML pipeline. Main ML tasks are, data preprocessing, feature engineering (FE), model selection and hyperparameter optimization [4]. In this investigation, we will focus on the FE task. FE is the process of selecting, manipulating, and transforming raw data into features that can be used to generate predictive models. Different strategies have been used to perform FE [2,5], however they can be classified in two main groups: reinforcement learning and evolutionary algorithms (EAs). EAs are population-based optimization methods based on the theory of natural evolution [6]. The main types of EAs are genetic algorithms, genetic programming (GP), evolutionary programming, and estimation of distribution algorithms. The principal advantage of EAs, such as GP, against reinforcement learning is that they can perform feature construction and selection at the same time.

GP is an extension of a genetic algorithm, where a population of computer programs is evolved over a series of generations to solve a problem. GP uses complex representations such as trees with a huge variety of operators and functions allowing the generation of new features. Moreover, it can be also used as a feature selection method by using the features that are in leaf nodes of the GP tree.

Several research works can be found in the literature related to automated FE [7,8], though most of the works focus on classification problems. This investigation aims to analyze the potential of an AutoML approach based on GP to construct informative features that help to better discriminate target variables in regression problems.

The remainder of the paper is organized as follows. Section 2 introduces the AutoML approach, and the experimental setup employed in this investigation. Afterwards, Section 3 presents the experimental results obtained from the carried-out tests. Finally, Section 4 brings some conclusions.

2. AUTOMATED MACHINE LEARNING APPROACH

In this investigation, GP is used to implement FE in a regression problem. A GP algorithm starts with a population of individuals (GP trees), where the leaf nodes are original feature values and internal nodes are predetermined operators or functions. Each GP tree can be understood as a mathematical function used to generate new features. This is illustrated in Figure 1.

The function set in this case is {+, −, ×, /, sin} and the features in the leaf nodes are $\{f_{22}, f_{39}, f_5, f_{35}, f_{11}, f_{25}, f_{23}, f_{27}, f_{45}, f_9, f_{31}, f_{14}\}$. The new feature constructed from the GP tree is shown in Eq. (1).

$$f' = (f_{22}^2 - f_{39} - f_5)\, x \left[(f_{35} - f_{11}) x \frac{f_{25}}{f_{22}}\right] + \left[\frac{sin\, f_{45}\ x\ f_{27}^2}{\frac{f_9}{45} - \frac{f_{31}}{14}}\right] \quad (1)$$

2.1. Multiple feature generation

GP has been proposed to generate both single features and multiple features. In this work, we are going to generate multiple features using GP. At the end of a GP execution, the best individual or tree is used to generate different feature sets related to f' and the features in the leaf nodes. In this investigation, the following three are considered:

1. F_1: Original feature set including f'.
2. F_2: Features in the leaf nodes. In the example of Fig. 1, {f_{22}, f_{39}, f_5, f_{35}, f_{11}, f_{25}, f_{23}, f_{27}, f_{45}, f_9, f_{31}, f_{14}}.
3. F_3: Features in the leaf nodes of the best individual including the new constructed feature. In the example of Fig. 1, {f', f_{22}, f_{39}, f_5, f_{35}, f_{11}, f_{25}, f_{23}, f_{27}, f_{45}, f_9, f_{31}, f_{14}}.

2.2. Experimental setup

To evaluate the proposed approach *k*-fold cross-validation with $k = 10$ is used. The GP is executed with training data *k* times, where the coefficient of determination, R^2 is used to evaluate individuals. R^2 is one of the most common error metrics employed in regression problems [9]. It takes values between [0, 1], a value near to 1 indicates a perfect fit, and is thus a highly reliable model for future forecasts. Hence, we are going to consider the GP as a maximization problem. A predictive model M_{tree} is trained with training data using a single feature, f'.

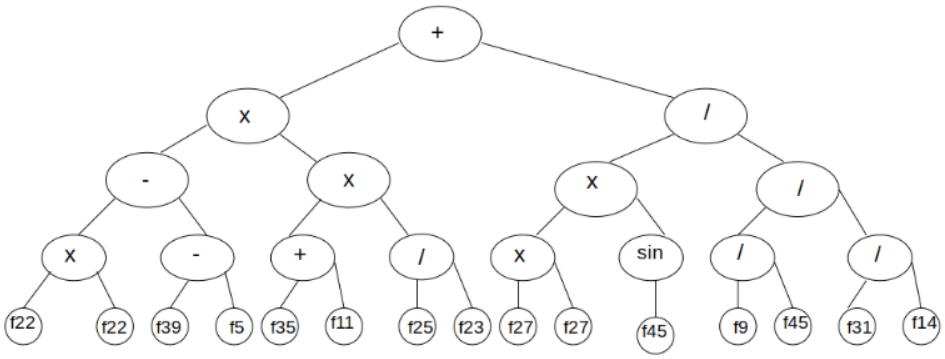

Figure 1. Example of a GP tree, where operators and functions are shown in intermediate nodes and features are shown in the leaf nodes.

The fitness value associated to a GP tree is the R^2 resulting from training results of M_{tree}. As mentioned, for each k execution the three feature sets are computed. Afterwards, the feature sets are used in testing data to compute the R^2. Finally, the average R^2 value obtained in the *k* folds is considered (illustrated in Fig. 2). As at each GP execution the initial population is different, to make a consistent evaluation, this process is repeated $t = 5$ times. In each run, the train and test sets are identical as they are built by the same seed. As a baseline, the same experimental setup is used without applying the FE. The three ML algorithms are executed in the same training and testing sets using k = 10-fold cross-validation and the original feature set.

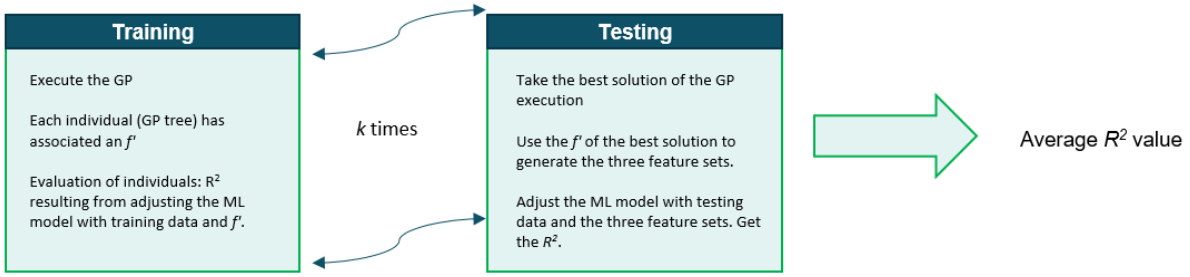

Figure 2. The proposed AutoML methodology for FE.

The genetic operators employed in the GP are tournament selection (with tournament size three), one point crossover and uniform mutation. Other details of the parameters considered are summarized in Table 1. The learning ML algorithms used in this investigation are linear regression (LR), Random Forest regressor (RF) [10] and Gradient Boosting regressor (GB) [11].

GP is implemented using the DEAP library and the ML algorithms using Sklearn library, both of Python programming language. In the case of the ML algorithms, default hyperparameters of Sklearn are employed.

Table 1. The employed parameters in GP executions.

Parameter	Parameter value
Maximum depth of the tree	5
Generations	40
Mutation rate	0.5
Crossover rate	0.1
Population size	10
Function set	+, -, x, /, *sin*, *cos*

3. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed approach is applied to an industrial process dataset. The dataset contains 9132 instances and 19 variables, 18 input variables and one output variable. The input variables are process parameters, and the output is an indicator of the performance of the process (Fig. 3).

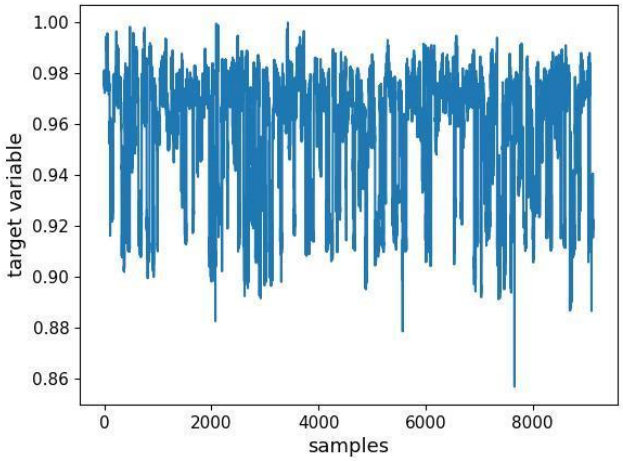

Figure 3. Target variable considered, normalized to [0,1].

Table 2 shows the R^2 measurements obtained in the testing sets when applying the approach proposed in Section 2. The best results obtained for the three ML algorithms per each feature set are indicated in bold. As can be observed, the best results were achieved for RF, continued by GB and LR. This phenomenon was expected as the LR is a simple algorithm with the ability of only capturing linear relationships. The R^2 values obtained for the baseline and the three models are the following ones: LR = 0.868, RF = 0.975 and GB = 0.94.

The best results for all runs and the three ML algorithms are achieved for $\boldsymbol{F_1}$, continued by $\boldsymbol{F_3}$ and $\boldsymbol{F_2}$. This makes sense as $\boldsymbol{F_1}$ is the feature set with the highest cardinality. In the case of set $\boldsymbol{F_1}$, almost identical R^2 values were achieved in the five executions for the three models. In addition, almost identical R^2 values were also achieved when comparing to the baseline approach.

Regarding to $\boldsymbol{F_2}$, worse results were obtained compared to $\boldsymbol{F_1}$. This is more noticeable in the case of LR with average value of R^2 = 0.264 in the $t = 5$ runs. However, in the case of RF and GB quite better results are obtained with average values of R^2 = 0.844 and R^2 = 0.675, respectively. Table 3 shows the maximum cardinality of $\boldsymbol{F_2}$ achieved in $k = 10$ folds for the three ML algorithms and $t = 5$ runs. It can be observed that the maximum cardinality is smaller than the 50% of the cardinality of the whole feature set except for RF $t = 3$ and $t = 5$.

Table 2. Obtained R^2 measurements for the three ML models and $\boldsymbol{F_1}$, $\boldsymbol{F_2}$ and $\boldsymbol{F_3}$ feature sets for t =5 runs.

	LR			*RF*			*GB*		
t	$\boldsymbol{F_1}$	$\boldsymbol{F_2}$	$\boldsymbol{F_3}$	$\boldsymbol{F_1}$	$\boldsymbol{F_2}$	$\boldsymbol{F_3}$	$\boldsymbol{F_1}$	$\boldsymbol{F_2}$	$\boldsymbol{F_3}$
1	*0.868*	*0.294*	*0.309*	***0.975***	*0.790*	*0.933*	*0.949*	*0.697*	*0.793*
2	*0.869*	*0.279*	*0.370*	***0.975***	*0.901*	*0.923*	*0.949*	*0.565*	*0.691*
3	*0.869*	*0.281*	*0.612*	***0.975***	*0.749*	*0.916*	*0.949*	*0.778*	*0.788*
4	*0.869*	*0.179*	*0.222*	***0.975***	*0.851*	*0.943*	*0.949*	*0.724*	*0.766*
5	*0.867*	*0.288*	*0.452*	***0.975***	*0.931*	*0.942*	*0.949*	*0.611*	*0.768*

In the case of $\boldsymbol{F_3}$, an improvement is observed in the R^2 values, when compared to R^2 values of $\boldsymbol{F_2}$. It should be noted that the only difference between the two feature sets is the variable f'. In some cases, such a LR t = 3, t = 5 and t = 2, the difference of R^2 values between $\boldsymbol{F_2}$ and $\boldsymbol{F_3}$ have been perceptible.

Table 3. Maximum $|\boldsymbol{F_2}|$ achieved in k = 10 folds for the three ML algorithms and t = 5 runs.

	LR	RF	GB
1	8	6	8
2	5	6	3
3	5	11	5
4	5	8	5
5	4	10	7

Fig. 4 shows the frequency of the features considered in the original set in the k = 10 folds and t = 5 repetitions for the three ML models. It can be seen that the most repeated features in the three models are features 1 and 14. On the contrary, the less frequent ones are features 12 and 15. The most repeated features appear for the RF model, which is the model that achieved best performance. This analysis serves to identify the most critical process variables that affect the target variable.

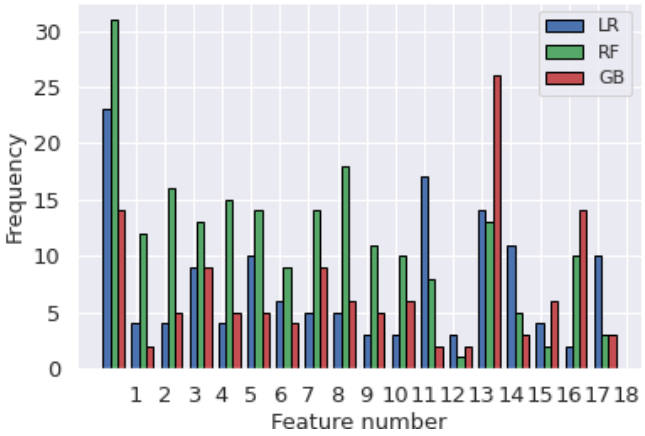

Figure 4. Original features repeated in the leaf nodes of the best GP tree in the k =10 folds and t = 5 repetitions for the three ML models.

4. CONCLUSION

This investigation introduces an AutoML approach to compute FE in regression problems using GP. It serves both to construct new features from the original set of features and to perform feature selection. The experimental results show that there is no improvement in the model's performance when adding the new constructed feature to the original feature set. Nevertheless, although the results are worse with F_2 than with the original feature set, F_2 contains significantly less features and still maintains good performance in many cases. Moreover, when applying the new constructed feature to F_2 the results are improved significantly.

It can be concluded that the proposed approach clearly serves to perform feature selection and that the new constructed feature has the ability to discriminate the target variable. This approach should be appropriate to apply to data with high dimensional set of input variables with the proposed feature set F_3.

This study reveals the need to go deeper into the FE using other techniques, such as reinforcement learning and to extend the analysis by comparing the proposed approach in this investigation with others. In addition, this approach could be complemented with another automated ML task, such as hyperparameter optimization. Note, that in this investigation default parameters of the ML algorithms were employed and that hyperparameter optimization can considerably improve the performance of the models.

Acknowledgment

This work was supported by the Project BISUM under Grant ELKARTEK 2021 through Basque Government.

REFERENCES

[1] Candanedo, I.S, Nieves, E.H, González, S.R, Martín, M, Briones, A.G. Machine learning predictive model for industry 4.0. In: 13th International Conference on Knowledge Management in Organizations, KMO 2018; August 6-10, 2018: Springer, Cham, pp. 501-510.

[2] Khurana, U, Samulowitz H, Turaga, D. Feature engineering for predictive modeling using reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence; February 2-7, 2018: AAAI Press, pp. 3407–3414.

[3] Chen, Y.-W, Song, Q, Hu, X. Techniques for automated machine learning. ACM SIGKDD Explorations Newsletter 2021; 22: 35–50 <https://doi.org/10.1145/3447556.3447567>

[4] Olson, R.S, Bartley, N, Urbanowicz R.J, Moore, J.H. Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016; July 20-24, 2016: Association for Computing Machinery, pp. 485–492.

[5] Viegas, F, Rocha, L, Gonçalves, M, Mourão, F, Sá, G, Salles, T, Andrade G, Sandin, I. A genetic programming approach for feature selection in highly dimensional skewed data. Neurocomputing 2018; 273: 554–569. <https://doi.org/10.1016/j.neucom.2017.08.050>

[6] Eiben, A, Smith, J. Introduction to Evolutionary Computing. Berlin: Springer, 2003.

[7] Khurana, U, Turaga, D, Samulowitz, H, Parthasrathy, S. Cognito: Automated feature engineering for supervised learning. In: IEEE 16th International Conference on Data Mining Workshops (ICDMW); December 12-15, 2016: IEEE, pp. 1304-1307.

[8] Lucas, Y, Portier, P.E, Laporte, L, He-Guelton, L, Caelen, O, Granitzer M, Calabretto, S. Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. Future Generation Computer Systems 2020; 102: pp.393-402 < https://doi.org/10.1016/j.future.2019.08.029 >

[9] Naser M, Alavi, A.H. Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. Architecture, Structures and Construction 2021; 1–19 < https://doi.org/10.1007/s44150-021-00015-8>

[10] Shanmugasundar, G, Vanitha, M, Čep, R, Kumar, R, Kalita, K, Ramachandran, M. A comparative study of linear random forest and AdaBoost Regressions for modeling non-traditional machining. Processes 2015; 9: 1-14 <https://doi.org/10.3390/pr9112015>

[11] Bataineh, A.S.A. A gradient boosting regression-based approach for energy consumption prediction in buildings. Advances in Energy Research 2019; 6: 91-101 < https://doi.org/10.12989/eri.2019.6.2.091 >