

Applying Genetic Programming to Improve Interpretability in Machine Learning Models

Leonardo Augusto Ferreira, Frederico Gadelha Guimarães
Machine Intelligence and Data Science (MINDS) Lab
Department of Electrical Engineering
Universidade Federal de Minas Gerais, UFMG,
31270-000 Belo Horizonte – MG, Brazil
leauferreira@cpdee.ufmg.br, fredericoguimaraes@ufmg.br
ORCID 0000-0001-9238-8839

Rodrigo Silva
Department of Computer Science
Universidade Federal de Ouro Preto, UFOP
35400-000 Ouro Preto - MG, Brasil
rodrigo.silva@ufop.edu.br
ORCID 0000-0003-2547-3835

Abstract—Explainable Artificial Intelligence (or xAI) has become an important research topic in the fields of Machine Learning and Deep Learning. In this paper, we propose a Genetic Programming (GP) based approach, name Genetic Programming Explainer (GPX), to the problem of explaining decisions computed by AI systems. The method generates a noise set located in the neighborhood of the point of interest, whose prediction should be explained, and fits a local explanation model for the analyzed sample. The tree structure generated by GPX provides a comprehensible analytical, possibly non-linear, expression which reflects the local behavior of the complex model. We considered three machine learning techniques that can be recognized as complex black-box models: Random Forest, Deep Neural Network and Support Vector Machine in twenty data sets for regression and classifications problems. Our results indicate that the GPX is able to produce more accurate understanding of complex models than the state of the art. The results validate the proposed approach as a novel way to deploy GP to improve interpretability.

Index Terms—Interpretability, Machine Learning, Genetic Programming, Explainability

I. INTRODUCTION

Advances in Machine Learning (ML) and Deep Learning (DL) have had a profound impact in science and technology. These techniques have had many recent successes, achieving unprecedented performance in tasks such as image classification, machine translation and speech recognition, to cite a few. The remarkable performance of Artificial Intelligence (AI) methods and the growing investment on AI worldwide will lead to an ever-increasing utilization of AI systems, having a significant impact on society and everyday life decisions. However, depending on the model used, understanding why it makes a certain prediction can be difficult. This is particularly the case with the high performing DL models and ML models in general. The more complex the model, the more opaque its decisions are to human understanding. Black-box ML models are increasingly used in critical applications, leading to an

urgent need for understanding and justifying their decisions. This difficulty is an important impediment for the adoption of ML, particularly DL, in domains such as healthcare [1], criminal justice and finance [2]. The term Explainable AI (or XAI) has been adopted by the community to refer to techniques that help the understanding of decisions or results of AI artifacts by a given audience, which can be domain experts, regulatory agencies, managers, decision-makers, policy-makers or users affected by these decisions.

The interpretability problem can be understood as the technical challenge of explaining AI decisions, especially when the underlying AI technology is perceived as a black-box model. Humans tend to be less willing to accept decisions that are not directly interpretable and trustworthy [3]. Therefore, users should be able to understand the models outputs in order to trust them. There are two types of trusting, as described in [4]: one about the model and another about the prediction. Though similar, they are not the same thing. The first one is related to whether someone will choose a model or not, whereas the second relates to whether someone will make a decision relying on that prediction.

Interpretability, as pointed out in [5], is associated with a human perception i.e. the ability to classify something or someone according to their main characteristics. This idea applied to ML models would be to highlight the main features that contributed to a prediction. Other works, such as [6] and [7], define interpretability as “*the ability to explain or to present in understandable terms to a human*”. In other words, a model can be defined as explainable whether its decisions are easier for a human to understand.

Another issue involving interpretability goes beyond trusting some model or prediction. The European Union has recently deployed the General Data Protection Regulation (GDPR) as pointed out in [8] and [9]. GDPR directly deals with subjects related to European citizens’ data, for example: it prohibits judgments based solely on automated decisions [10]. European Union’s new GDPR has a major impact in deploying machine learning algorithms and AI-based systems. It restricts automated individual decision-making which “significantly affects” users. The law also creates a “right to explanation,”

This work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq); (ii) Coordination for the Improvement of Higher Education (CAPES) and (iii) Foundation for Research of the State of Minas Gerais (FAPEMIG, in Portuguese).

MINDS Laboratory – <https://minds.eng.ufmg.br/>

whereby someone has a right to be given an explanation for an output of the algorithm [11].

There are many important results of interpretability and explainability in the literature. For instance, interpretable mimic learning [12] is an approach in which the behavior of a slow and complex model is approximated by a faster and simpler model (also more transparent) with comparable performance. The idea is that by mimicking the performance of other complex models, one is able to compress the learning into a more interpretable model and derive rules or logical relationships. In this regard, it is possible to cite Lime (Local Interpretable Model-Agnostic Explanations) [4] and SHAP (SHapley Additive exPlanations) [13], which have been widely used for interpretability. Despite their success, both Lime and SHAP assume that a linear model is a good local representation of the original one. This simplification may cause, in some circumstances, a significant loss in the mimicking model accuracy, which may spoil the final interpretation.

In this paper we present an approach to interpretability based on Genetic Programming (GP), named Genetic Programming Explainer (GPX). GP has the ability to produce linear and nonlinear models increasing the flexibility of the aforementioned methods. The evolution process of the GP algorithm ensures the selection of the best features for the analyzed sample. Moreover, the tree structure representation naturally provides an analytical expression that reflects a local explanation about the model's prediction. The main goal is to produce an accurate mimicking model which preserves the advantages of having a closed mathematical expression such as readability and the ability of computing partial derivatives to assess the sensitivity of the output with respect to the input parameters. According to the taxonomy recently advocated by [14], the proposed approach can be categorized as a model agnostic technique for post-hoc explainability, able to provide a local explanation for a specific prediction output given by a complex black-box model.

In this work we have considered the following pre-trained complex ML algorithms that can be recognized as black-box models: Random Forest [15], Support Vector Machines (SVM) and a Deep Neural Network (DNN) [16] in a number of different data sets (10 regression and 10 binary classification problems available in public repositories). These methods have great performance in most ML problems, however, their explainability is low. For each pre-trained complex model, we compared (GPX) against other methods that could be used for generating local explanations: Lime and Decision Tree/Regression. The statistical analysis shows that GPX was able to better approximate the complex model, providing an interpretable explanation in terms of a symbolic expression. Genetic Programming brought us a new approach for interpretability with a local explanation for black-box models predictions. In addition, we present two case studies to illustrate the proposed methodology and serve as a guide for future use. The first one is about predicting home prices in Boston area and the other one measures the progression of Diabetes over the years.

In summary, the proposed approach aims to contribute to improving interpretability of black-box ML by using automatic generation of model-agnostic explanations that are able to fit the input-output decisions performed by the more complex model, which would be otherwise impossible to be derived by usual analytical methods. The GP algorithm is applied locally and provides an analytical and visual explanation in terms of a human readable expression represented as a tree. These evolved explanations can be easily interpreted and help understanding these complex decisions.

This paper is organized as follows. Section III-A reviews the main ideas of GP Algorithm and discusses how it will be applied for the purpose to provide interpretability. Section II introduces some concepts about interpretability and describes our approach according to these. Section III presents our methodology and the main idea of our solution for approaching the interpretability problem. Section V discusses the results of this article compared with the state of the art.

II. CONCEPTS OF INTERPRETABILITY

Humans are capable of making predictions about a subject and build a logical explanation to justify it. When a prediction is based on understandable choices, it gives the decision maker more confidence on the model [2]. On the other hand, Machine Learning and Deep Learning models are not able to provide the same level confidence. Their complexity and exorbitant number of parameters make them unintelligible for a human being and for most of the purposes they are seen as black-boxes. Humans tend to be resistant to techniques that are not well understood or that cannot be directly interpreted [5].

More recently, several strategies have been applied to understand how black-box models work. These strategies aim to decrease the opacity of artificial intelligence systems and to make these models more user friendly. In order to address the opacity of some machine learning models, we first must introduce some concepts of interpretability [7], [2]:

- **Comprehensibility:** refers to the ability of a learning algorithm to represent its learned knowledge in a human understandable fashion, in such a way that the knowledge representation can be comprehended.
- **Interpretability:** It is defined as the ability to explain or to provide the meaning in understandable terms to a human.
- **Explainability:** is associated with the notion of explanation as an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans.
- **Transparency:** A model is considered to be transparent if by itself it is understandable. For instance, a decision tree is a transparent model. The model must allow the reproduction of every calculation step, it needs to have a clear explanation of parameters and hyper-parameters and to provide an explanation about the learning process. Basically complex models can be considered as black-box models because they lack transparency.

- **Functionality:** the model must provide an understandable output, it needs to have visualization tools and to ensure a local explanation [4], [13]. The outputs must be presented in a user-friendly way.

Post hoc explainability techniques employ other methods after training step to analyze a given model. Different techniques can be used to enhance interpretability of a black-box model, such as text explanations, visual explanations, local explanations, explanations by example, explanations by simplification and feature relevance explanations [5].

More specifically, post hoc explainability by means of local explanations is an approach applied for a prediction or set of predictions of pre-trained black-box models, for example: Lime [4] and SHAP [13] fall in this category. Lime generates a local explanation model while SHAP identifies feature relevance explanations.

The approach presented in this paper can be considered as a post hoc technique since the GP algorithm is applied to a pre-trained model. The goal of the GP is to produce a local symbolic model to provide a visual explanation in terms of a human readable expression represented as a tree. Thus, in this sense, the proposed method generates a knowledge representation that can be comprehended, a local explanation model that is interpretable, transparent and functional.

Explainability and interpretability are extremely relevant issues nowadays. They are fundamental pieces for dealing with several philosophical and ethical aspects of the interaction between humans and AI artifacts.

III. EVOLVING EXPLANATIONS

A. Genetic Programming

Genetic Programming (GP) was developed by Koza [17] to evolve functional structures such as algebraic expressions, computer programs, and logical expressions [18]. GP works with a population of programs, usually initialized at random. In a GP there is a problem to be solved and the fitness of the individuals is related to how well they solve this problem. Thus, the GP algorithm evolves this population until it finds the best solution for the problem. Figure 1 presents all steps of the GP algorithm.

The GP algorithm is relatively simple to describe, as observed in Figure 1. The first step is to generate a random population with several individuals. The next step is to evaluate the fitness of each individual to measure how well an individual solves the problem at hand. If some individual satisfies the stopping criterion, the algorithm ends. Otherwise, it goes to the selection step in which the algorithm will favor the better individuals based on fitness evaluation. After the selection step a genetic operator is chosen randomly to generate new individuals [17], [18].

In this work, genetic programming is used to evolve non-linear symbolic expressions for regression and classification problems. These expressions are represented in the program as a tree over which the genetic operations of mutation, crossover and reproduction are defined. Figure 2 illustrates the tree representation of a mathematical expression.

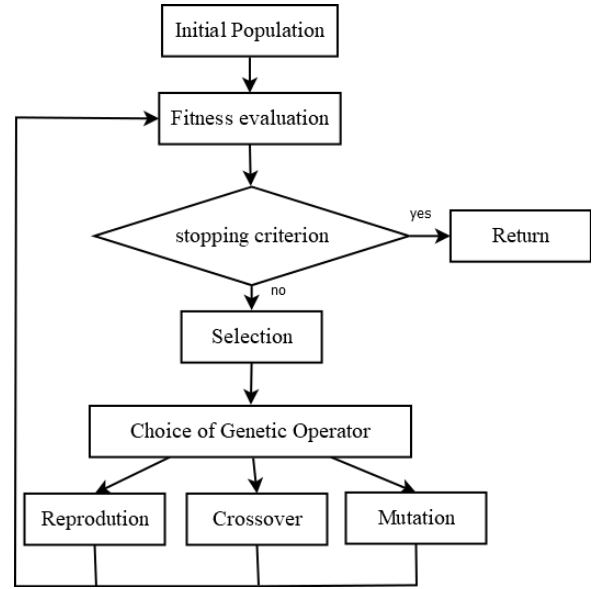


Fig. 1. This flowchart presents the steps of the evolutionary process in GP.

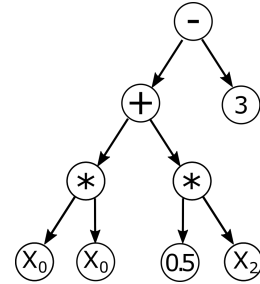


Fig. 2. This binary tree represents the expression: $f(x) = x_0^2 + 0.5x_2 - 3$. The terminal nodes are constants or input variables and the internal nodes are the functions or operations.

The *gplearn*¹ Python library was used as the GP search engine. This library extends from scikit-learn, a widely known Python library for ML.

B. GP approach to local explanation

In this section, the proposed approach to the interpretability problem is described. The example discussed here helps understanding the steps needed to achieve our local interpretation.

Let $\mathbf{x} \in \mathbb{R}^n$ be an input fed to a complex pre-trained machine learning model. The first step in our method is to generate m sample points around the input \mathbf{x} . This set of samples, called noise set, η , is created by sampling from a multivariate Gaussian distribution centered at \mathbf{x} with covariance matrix, $\Sigma = I_n \times \sigma$ where, I_n is the $n \times n$ identity matrix and σ is measured on training data.

The goal of GPX is to find the function, $f^* : \mathbb{R}^n \rightarrow \mathbb{R}$ which is easy to interpret and, at the same time, mimics the behavior of original complex model, $g : \mathbb{R}^n \rightarrow \mathbb{R}$, over the sample set, η . Formally, the problem to be solved by GPX can be defined as follows:

¹<https://gplearn.readthedocs.io/>

$$f^* = \arg \min_{f \in F, \mathbf{s}_i \in \eta} d([f(\mathbf{s}_1), \dots, f(\mathbf{s}_m)] - [g(\mathbf{s}_1), \dots, g(\mathbf{s}_m)]) \quad (1)$$

where:

- F : set of all admissible functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$;
- $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m\}$: samples form the noise set, η ;
- $g(\mathbf{s}_i)$: is the prediction given by the complex ML model;
- $f(\mathbf{s}_i)$ is the prediction generated by a given individual in the population; and
- $d(\cdot)$: some distance metric, usually the l_2 -norm or the root-mean-square error (RMSE).

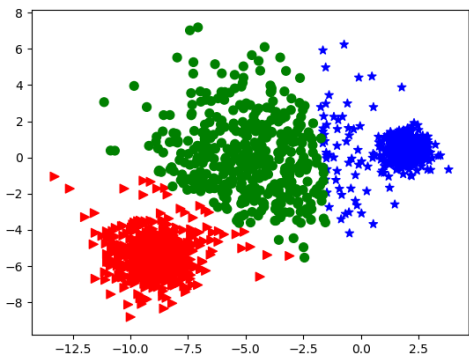


Fig. 3. Suppose a classification problem in which there are 1,500 samples in the training data set in 3 separate classes. Class 1 are triangles, class 2 circles and class 3 stars. This data set was created by scikit-learn tool, where the standard deviation in each class is 1.0, 2.5, 0.5 respectively.

As an example, Figure 3 illustrates some training data $X_{k \times n}$, where $k = 1,500$ samples and $n = 2$. Assume that a model, $g(\cdot)$, has been previously obtained from this data set.

After defining σ , it is possible to generate the noise set, η , located in the neighborhood of the point of interest \mathbf{x} whose prediction should be explained. Figure 4 shows \mathbf{x} and the noise set, η , generated randomly around \mathbf{x} .

The next step is to apply GP to try to find f^* as defined in Equation (1).

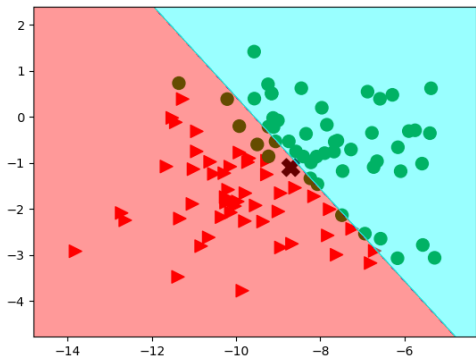


Fig. 4. The bold X represents the input \mathbf{x} for a complex ML model, whose prediction should be interpreted. The set η contains 100 samples classified by the complex model. This figure also presents the surface decision generated by GP algorithm.

Figure 5 illustrates the best representation found for f^* by the GP. For didactic reasons, this example was built with two dimensions only and the produced representation for f^* contains all the input features. It is important to highlight, however, that the GP is not obliged to use all the features. Thus, the produced model already indicates which variables are important. Besides, the returned closed mathematical expression allows the user to compute partial derivatives to assess the local importance of each feature.

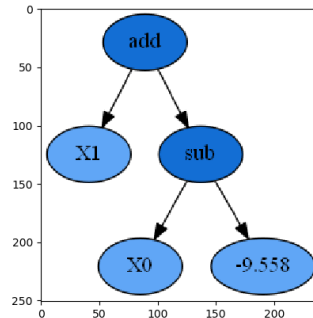


Fig. 5. A tree structure representing an algebraic expression, $f(s) = x_1 + (x_0 - (-9.558))$. This structure was generated by GPX and visualized via *graphviz*.

IV. EXPERIMENTAL METHODOLOGY

A. Data Sets

In this work twenty data sets were used. Ten for classification and ten regression. All data sets were extracted from well-known repositories such as *UCI Machine Learning Repository*², *OpenML*³ and *Kaggle*⁴. Our selection was based on features variability between data sets and number of downloads. Table I lists the chosen data sets for the classification problems and the corresponding number of used features.

TABLE I
CLASSIFICATION DATA SET

Data Set	Features
Diabetes [19]	8
Steel Plates Faults [20]	33
The Monk's Problems 2 [21]	6
Phoneme [22]	5
Blood Transfusion Service Center [23]	4
Ozone Level Detection [24]	72
Hill-Valley [25]	100
EEG Eye State [25]	14
Spambase [25]	57
ILPD (Indian Liver Patient Data Set) [26]	10

Table II presents the data sets chosen for the regression problems and the corresponding number of used features.

²<https://archive.ics.uci.edu/>

³<https://www.openml.org>

⁴<https://www.kaggle.com/>

TABLE II
REGRESSION DATA SETS

Data Set	Features
Diabetes [25]	10
Boston [27]	13
Fetch California Housing [27]	8
Bike Sharing in Washington D.C.	13
Red Wine Quality [28]	11
House Sales in King County, USA	18
GPU Kernel Performance [29]	14
Beer Consumption - São Paulo	5
Houses to rent data	8
Predicting Compressive Strength of Concrete	10

B. Complex, black-box, models

There are several machine learning models which can be considered black-box. In this work we have selected three of the most popular to serve as “complex models”. That is, model for which explanations have to be produced. These models are the Random Forest, Deep Neural Networks and Support Vector Machines. They are briefly introduced below.

Random Forest is a type of tree ensemble which have shown improvements for the generalization of a single decision tree and has achieved remarkable results in the literature [15], [30]. Our experiments set Random Forest with up two thousand trees. The more trees are used in the architecture the more complexity is brought to the understanding of the model.

Deep neural networks (DNNs) [16] are artificial neural networks (ANNs) with multiple layers between the input and output layers. The multi-layer architecture is inspired in the human brain structure and has shown to be quite effective in many difficult classification and regression problems [31]. This architecture, however, represents compositions of non-linear functions which are not easy to interpret.

Finally, Support Vector Machines (SVMs) [32] are classifiers which find a separating hyperplane, such that the distance on either side of that hyperplane to the next-closest data points is maximized. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. Support Vector Regression (SVR) is the extension of SVMs for regression problems [33]. The SVR tries to fit the error within a certain threshold and can also be considered opaque since it is hard to interpret its decisions, specially when nonlinear kernel functions are used.

C. Competing Explainers

In this section we present the competing explainers. Just like our method, the idea is to sample a noise set, η , around the point of interest, \mathbf{x} . The set η is then used to build a local comprehensible model which will be used as an explainer. We say that an explainer, f , “understands” the complex model, g , if its error on the noise set, η , relative to the complex model predictions is small. For regression problems, this concept may be translated into the Root Mean Squared Error between the model and the explainer as follows:

$$u_r(f) = \frac{1}{|\eta|} \sum_{\mathbf{s}_i \in \eta} (f(\mathbf{s}_i) - g(\mathbf{s}_i))^2 \quad (2)$$

For classification problems, it may be formalized as the accuracy of the explainer with respect to the complex models predictions.

$$u_c(f) = \frac{1}{|\eta|} \sum_{\mathbf{s}_i \in \eta} h(\mathbf{s}_i) \quad (3)$$

where,

$$h(\mathbf{s}_i) = \begin{cases} 1 & \text{if } f(\mathbf{s}_i) = g(\mathbf{s}_i) \\ 0 & \text{if } f(\mathbf{s}_i) \neq g(\mathbf{s}_i) \end{cases} \quad (4)$$

The first method, named Lime [4], generates an explainer based on a linear least squares method with l_2 -norm regularization [34]. This linear model is used into Lime in order to measure locally the feature importance.

Another good candidate for explainer is the Decision Tree (DT). A DT is, as the name implies, a tree where each node represents a feature, each branch represents a decision, and each leaf represents a prediction. By making a path from the root to the prediction leaf node, the user can obtain an explanation for that prediction.

The performance of Lime, DT and GPX (explained in section III-A) as explainers of Support Vector Machines, Neural Networks and Random Forests are discussed next.

V. TESTING EXPLAINERS ACCURACY

A. Experimental Setup

In this section we test the ability of Lime, GPX, DTs to understand the different complex models discussed in section IV-B. The experimental steps can be described as follows:

- 1) Divide all the data sets described in section IV-A into training and test data. We used 80% for training and 20% for test.
- 2) Train the complex models (described in section IV-B) with training data.
- 3) Using the trained complex models, predict the value of 100 random samples selected from the test set. At this point, we have six thousand predictions for the regression and classification problems together (20 data sets \times 100 predictions \times 3 complex models).
- 4) Build the three explainers (Lime, GPX, DT) for each of these six thousand predictions in order to measure which interpreter can better understand the black-box prediction. At this time we have 18,000 data in order to apply a statistical analysis.

The GPX hyper-parameters were set as follows:

- Population size: 100 individuals.
- Probability of crossover: 70%.
- Probability of hoist mutation: 5%. This mutation is called hoist mutation because the method chooses randomly a subtree and hoist it into the tree.

- Probability of point mutation: 10%. This mutation selects a random node to be replaced.
- Search interval: $[-100, 100]$.

The other hyper-parameters are set to the library default values.

The noise set, η (see section III-B), has the same one thousand samples for each explainer. Thus, the explainers always access same data. In order to measure how well the explainers understand the complex models (2) and (3) are used for the regression and classification problems, respectively.

B. Results

Table III presents the average and the standard deviation of the error, computed with (2), for each explainer across the regression problems.

TABLE III
EXPLAINERS ERROR

Explainer	Average Error	Error Standard Deviation
DT	0.083	0.329
GPX	0.065	0.508
Lime	7.577	36.913

A Permutation Pairwise Test⁵ was performed in order to test the hypothesis that the difference between the error means is zero. The results are shown in Table IV where P.adjust represents the P-Value adjusted with the Bonferroni Method. Considering a confidence level of 95%, it is possible to say that there is no difference in the mean error presented by the DT and the GPX. On the other hand, the results indicate that both DT and GPX, understand the complex model better than Lime.

TABLE IV
PAIRWISE PERMUTATION TEST FOR THE ERROR DIFFERENCE

Comparison	Stat	P-value	P.adjust
DT - GPX = 0	1.599	0.1099	0.3297
DT - Lime = 0	-11.01	3.515e-28	1.054e-27
GPX - Lime = 0	-11.03	2.668e-28	8.004e-28

Table V presents the average and the standard deviation of the accuracy, computed with (3), for each explainer across the classification problems.

TABLE V
EXPLAINERS ACCURACY

Explainer	Average Accuracy	Accuracy Standard Deviation
DT	0.852	0.104
GPX	0.899	0.130
Lime	0.658	0.334

A Permutation Pairwise Test⁶ was performed again in order to test the hypothesis that the difference between the accuracy means is zero. The results are shown in Table VI where P.adjust represents the P-Value adjusted with the Bonferroni

Method. Considering a confidence level of 95%, the results indicate that, in average, GPX was better than both DT and Lime in understanding the complex models. The DT, in turn, was better than Lime.

TABLE VI
PAIRWISE PERMUTATION TEST FOR THE ACCURACY DIFFERENCE

Comparison	Stat	P-value	P.adjust
DT - GPX = 0	-15.09	1.866e-51	5.598e-51
DT - Lime = 0	28.31	0	0.000e+00
GPX - Lime = 0	33.27	0	0.000e+00

The complete table of results as well as the analysis scripts are available at <https://github.com/leauferreira/GpX>.

Overall, these results show that, at least for the scenarios presented here, the Lime assumption that the complex model can be locally approximated by a linear model, is not always adequate. The DT and the GPX specially were superior methods in understanding the complex model.

Having shown that the proposed methodology can indeed produce accurate local models, in the next section, a demonstration of the use of GPX in practice is presented.

VI. CASE STUDY ON INTERPRETABILITY FOR RANDOM FOREST REGRESSOR

In this case study we selected the Boston and the Diabetes data sets, see Table II, and the *Random Forest Regressor* with 2,000 estimators (trees), from *scikit-learn*. Both data sets (Diabetes and Boston) can be found in *scikit-learn*

The Boston data set has 13 features, 506 samples and the target consists of home prices in Boston. The data set was randomly split into training and test sets with 80% of the samples for training and the remainder for testing. Apart from the number of trees the other hyper-parameters were set to the default values in the library. It is hard to analyze the joint decision performed by 2,000 trees, therefore explainability of Random Forest is low, although its performance is high.

After the training step we take the first sample in the test set, x_1 , and apply the process described in Section III-B in order to create a noise around x_1 . Then, the GP algorithm is trained with the noise set, $\eta = \{s_1, \dots, s_n\}$ where the targets are given by $g(s_i)$, $\forall s_i \in \eta$, and g is the *Random Forest Regressor* model.

The Boston data set consists of $d = 13$ features. However, as observed in Figure 6, after the evolution process only two features were chosen by the GP. They were:

- **PTRATIO** : pupil-teacher ratio by town
- **NOX**: nitric oxides concentration (parts per 10 million)

In (5) we changed **PTRATIO** to $x_{ptratio}$ and **NOX** to x_{nox} . The tree structure in Figure 6 represents the following equation:

$$f^*(s) = \frac{x_{ptratio}^2 x_{nox}}{28.390} \quad (5)$$

Figure 7 presents the result of the same process but in a different area, by considering sample x_2 . It is possible

⁵See <https://rdrr.io/cran/rcompanion/man/pairwisePermutationTest.html>

⁶See <https://rdrr.io/cran/rcompanion/man/pairwisePermutationTest.html>

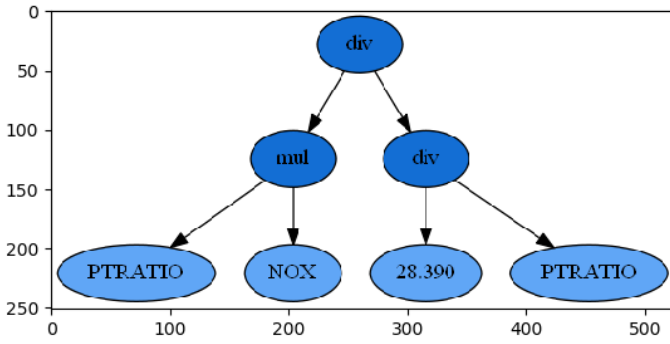


Fig. 6. GP algorithm output for the *Random Forest Regressor* prediction applied to instance x_1 in Boston data set.

to observe that different features were chosen during the evolutionary process. The features chosen for the solution shown in Figure 7 were:

- **PTRATIO**: pupil-teacher ratio by town
- **INDUS** proportion of non-retail business acres per town centres
- **LSTAT** lower status of the population

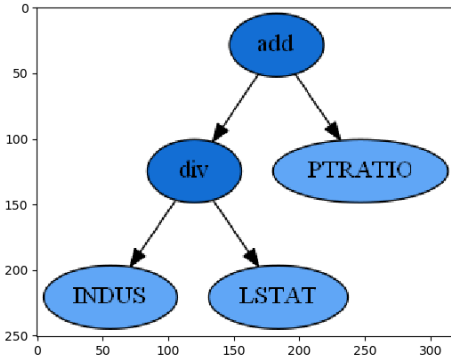


Fig. 7. GP algorithm output for the *Random Forest Regressor* prediction applied to instance x_2 in Boston data set.

The expression represented in Figure 7 can be defined as:

$$f^*(s) = \frac{x_{indus}}{x_{lstat}} + x_{ptratio} \quad (6)$$

Based on the equations (5) and (6) we can understand the behavior around x_1 and x_2 . First of all, it is possible to observe that the feature PTRATIO (pupil-teacher ratio by town) is relevant to define home prices in the neighborhood of both instances (x_1, x_2). However, in the neighborhood of x_2 , PTRATIO has more influence.

These results can help a decision-maker to understand which features contribute the most to the increase in the price in a neighborhood. Moreover, it is possible to know which features changed in order to increase or decrease home prices. In this regard, it is useful to compute the gradient of the model output with respect to the selected input parameters. Lets take equation (5) as an example. Its gradient is given by:

$$\nabla f^*(s) = \left[\frac{x_{ptratio}x_{nox}}{14.195}, \frac{x_{ptratio}^2}{28.390} \right]^T \quad (7)$$

By analyzing equations (5) and (7) it is easy to understand how and by how much each feature affects housing prices.

The Random Forest Regressor has also been applied to the Diabetes data set. This data set consists of 402 samples and 10 features. In this case, the target is a quantitative measure of disease progression. The GPX was set with the same hyper-parameters used for the Boston data set. Figure 8 presents the result for the Diabetes data set.

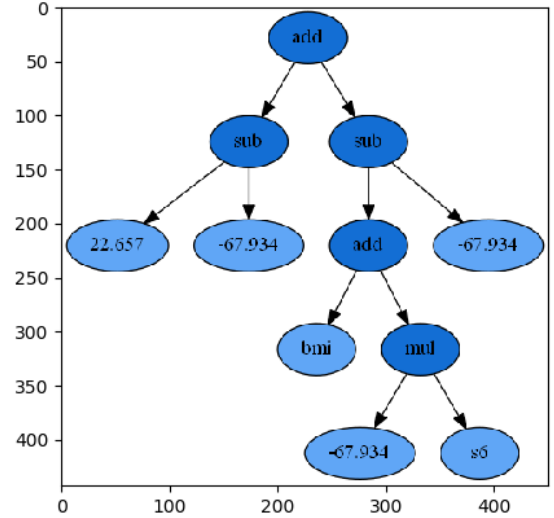


Fig. 8. GP algorithm output for the *Random Forest Regressor* prediction applied to instance x_{d1} in Diabetes data set.

The evolutionary process chose a final tree with two features:

- **bmi**: Body mass index
- **S₆**: blood serum measurements

The tree structure in Figure 8 represents the following expression:

$$-67.934x_{S_6} + x_{bmi} + 158.525 \quad (8)$$

where x_{bmi} is **bmi** and x_{S_6} is **S₆**.

This expression shows for instance that decreasing body mass index is a relevant feature to decrease diabetes progression for that specific patient.

VII. CONCLUSION

This paper presented an approach to the interpretability problem based on Genetic Programming. We discussed several concepts of interpretability and why this subject is relevant nowadays.

The GP algorithm used was able to produce a non-linear algebraic expression as output, which in turn provides many opportunities for the interpretability of the more sophisticated ML algorithms. It naturally selects the most important features

to build a local explanation around a given sample. Besides, the produced analytic expression allows for easy differentiation, which gives the sensitivity of the output with respect to each feature.

Examples using the classic Boston and Diabetes data sets show that the proposed approach can be seen as a source for interpretability and help the decision maker to understand better how the complex model is making a decision.

We submit the explainers GPX, Lime and Decision Tree to a stress test in order to measure which one can, locally, better understand the black-box model. The statistical analysis showed us that GPX, besides bringing a new approach to interpretability, presented better or at least similar results when compared with the state of the art.

REFERENCES

- [1] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," *Artificial Intelligence in Medicine*, vol. 24, no. 1, pp. 25 – 36, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S093336570100094X>
- [2] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao, T. D. Kelley, D. Braines, M. Sensoy, C. J. Willis, and P. Gurrum, "Interpretability of deep learning models: A survey of results," pp. 1–6, Aug 2017.
- [3] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, "Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 2018, pp. 1–8.
- [4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: <https://doi.org/10.1145/2939672.2939778>
- [5] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Towards medical xai," 10 2019.
- [6] P. Hall and N. Gill, *An Introduction to Machine Learning Interpretability: An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI*. O'Reilly Media, 2018.
- [7] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*, 2019, <https://christophm.github.io/interpretable-ml-book/>.
- [8] C. Tankard, "What the gdpr means for businesses," *Network Security*, vol. 2016, no. 6, pp. 5 – 8, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1353485816300563>
- [9] J. P. Albrecht, "How the gdpr will change the world," *European Data Protection Law Review*, vol. 2, no. 3, 2016. [Online]. Available: <https://doi.org/10.21552/EDPL/2016/3/4>
- [10] B. P. Evans, B. Xue, and M. Zhang, "What's inside the black-box? a genetic programming method for interpreting complex machine learning models," p. 1012–1020, 2019. [Online]. Available: <https://doi.org/10.1145/3321707.3321726>
- [11] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation";" *AI Magazine*, vol. 38, pp. 50–57, 2017.
- [12] Z. Che, S. Purushotham, R. G. Khemani, and Y. Liu, "Interpretable deep models for icu outcome prediction," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2016, pp. 371–380, 2016.
- [13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," pp. 4765–4774, 2017. [Online]. Available: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- [14] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82 – 115, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1566253519308103>
- [15] F. Livingston, "Implementation of breiman's random forest machine learning algorithm," *ECE591Q Machine Learning Journal Paper*, pp. 1–13, 2005.
- [16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. USA: Prentice Hall PTR, 1998.
- [17] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.
- [18] A. Gaspar-Cunha, R. Takahashi, and C. Antunes, *Manual de computação evolutiva e metaheurística*, ser. Ensino. Imprensa da Universidade de Coimbra / Coimbra University Press, 2012. [Online]. Available: <https://books.google.com.br/books?id=9Di5CwAAQBAJ>
- [19] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," *Proceedings. Symposium on Computer Applications in Medical Care*, p. 261–265, November 1988. [Online]. Available: <https://europepmc.org/articles/PMC2245318>
- [20] M. Buscema, S. Terzi, and W. Tastle, "A new meta-classifier," in *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, 2010, pp. 1–7.
- [21] S. B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. D. Jong, S. Dzeroski, S. E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. V. D. Welde, W. Wenzel, J. Wnek, and J. Zhang, "The monk's problems a performance comparison of different learning algorithms," *Tech. Rep.*, 1991.
- [22] J.-L. Voz, M. Verleysen, P. Thissen, and J.-D. Legat, "A practical view of suboptimal bayesian classification with radial gaussian kernels," in *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, ser. IWANN '96. Berlin, Heidelberg: Springer-Verlag, 1995, p. 404–411.
- [23] I.-C. Yeh, K.-J. Yang, and T.-M. Ting, "Knowledge discovery on rfm model using bernoulli sequence," *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 5866 – 5871, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417408004508>
- [24] K. Zhang and W. Fan, "Forecasting skewed biased stochastic ozone days: Analyses, solutions and beyond," *Knowl. Inf. Syst.*, vol. 14, no. 3, p. 299–326, Mar. 2008. [Online]. Available: <https://doi.org/10.1007/s10115-007-0095-1>
- [25] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [26] B. V. Ramana, M. S. P. Babu, and N. B. Venkateswarlu, "A critical comparative study of liver patients from usa and india: An exploratory analysis," 2012.
- [27] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
- [28] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decis. Support Syst.*, vol. 47, no. 4, pp. 547–553, 2009. [Online]. Available: <http://dblp.uni-trier.de/db/journals/dss/dss47.html#CortezCAMR09>
- [29] R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola, "Sobol tensor trains for global sensitivity analysis," *Reliability Engineering & System Safety*, vol. 183, pp. 311–322, 2019.
- [30] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [31] I. G. Goodfellow, Y. Bengio, and A. C. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [32] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152.
- [33] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.
- [34] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Math. Dokl.*, vol. 4, pp. 1035–1038, 1963.