Explaining Symbolic Regression Predictions

Renato Miranda Filho
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
Instituto Federal de Minas Gerais
Sabará, Brazil
renato.miranda@dcc.ufmg.br

Anisio Lacerda
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
anisio@dcc.ufmg.br

Gisele L. Pappa
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
glpappa@dcc.ufmg.br

***Abstract*—The outgrowing application of machine learning methods has raised a discussion in the artificial intelligence community on model transparency. In the center of this discussion is the question of model explanation and interpretability. The genetic programming (GP) community has systematically pointed out as one of the major advantages of GP the fact that it produces models that can be interpreted by humans. However, as other interpretable supervised models, the more complex the model becomes, the less interpretable it is. This work focuses on post-hoc interpretability of GP for symbolic regression. This approach does not explain the process followed by a model to reach a decision. Instead, it justifies the predictions it makes. The proposed approach, named Explanation by Local Approximation (ELA), is simple and model agnostic: it finds the nearest neighbors of the point we want to explain and performs a linear regression using this subset of points. The coefficients of this linear regression are then used to generate a local explanation to the model. Results show that the errors of ELA are similar to those of the regression performed with all points. It also shows that simple visualizations can provide insights to the users about the most relevant attributes.**

***Index Terms*—Interpretability, Symbolic Regression, Explanations**

I. Introduction

The outgrowing application of machine learning methods in sensitive areas of our daily lives, including medicine, financial activities and the criminal justice system, has raised a productive discussion in the artificial intelligence community on model transparency, justice and accountability [1]–[3]. In the center of these discussions lies the question of model interpretability and explanation [4], [5].

Genetic programming (GP) is a method that has been used for quite some time to generate models. It is a bio-inspired method, where a population of solutions representing models is evolved for a number of generations. The models are evaluated according to a fitness function, and probabilistic selected to undergo mutation and crossover operators. One of the main advantages of models evolved by GPs is the fact that the evolved models can be interpreted by humans [6], [7]. However, in the same way as other supervised models that are considered interpretable – including decision trees, decision rules or linear regression – the more complex the model becomes, the less interpretable it is.

When discussing model interpretability, most applications involving interpretable models rarely analyze the simplicity or easiness of use of the generated models. Most of the times, model complexity (defined by the number of tree nodes, for example, in the case of tree-represented symbolic regression or rule induction with GP) is reported or optimized during model generation together with accuracy to result in simpler models. But hardly ever a specialist or human is involved in the loop to say whether a model is interpretable or not.

In this direction, there has been a fruitful discussion on how we define model interpretability [8]. Some papers relate interpretability to trust, where trust is related to understanding but can also refer to the confidence in the model accuracy. Others use interpretability as a synonym of understandability or intelligibility, and relate it to the concept of how the model actually works and makes decisions. Finally, there are the so called post-hoc interpretations, which explain predictions without going into details on how the models work.

This work focuses on the *interpretability of GP* when producing models for symbolic regression. As shown in our experimental study, when non-linear relationships are present in the data, even for synthetic datasets with two attributes (explainable variables), as the maximum depth of the GP tree increases (potentially increasing the model complexity) and the root mean square error (RMSE) decreases, the number of nodes in the tree grows from an average of 5 nodes (maximum depth 2) to more than 65 nodes (maximum depth 6) – see Table III.

In this direction, this paper proposes to use post-hoc interpretation to understand the outcomes of symbolic regression models. Post-hoc interpretability does not explain the process followed by a model to reach a decision. Instead, it *justifies* the prediction using, for example, text explanations, visualization or explanations by example [8]. The proposed method follows explanations by examples, as they resemble the way humans use analogy to explain their decisions. In medicine, for example, many diagnosis are "explained" by other case studies.

The proposed approach, named Explanation by Local Approximation (ELA), is simple, effective and model agnostic (i.e., it can be applied to any other regression model): it finds the nearest neighbors of the point we want to explain and

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

performs a linear regression using this subset of points. The coefficients of this linear regression are then used to generate a local explanation to the model. Note that this approach, together with GP, has the advantage of intrinsically performing an attribute selection during the GP evolution, and then using a local approximation to justify the predictions of the method.

We perform a quantitative analysis of the proposed approach in a set of 10 synthetic benchmarks with known non-linear relations between the predictive and target variables. We then present a qualitative analysis in a real-world dataset where we do not need very specialized knowledge to interpret the resulting justifications, as evaluation of the interpretability of the models is complicated and, for most datasets, require specialized expertise.

Results show that the errors of the local approximations are similar to those of the regression performed with all points. It also shows that simple visualizations can provide insights to the users about the most relevant model attributes. Finally, we also concatenate the results of many local interpretations into a single, global explanation, which can also aid the process of understanding predictions.

The remainder of this paper is organized as follows. Section 2 presents related work on interpretability. Section 3 introduces the proposed method, while Section 4 presents the quantitative and qualitative experiments. Finally, Section 5 draws conclusions and point out direction of future research.

II. Related Work

Given a classification or regression problem, the model it produces is defined as transparent (or white box [9]) if it is locally or globally interpretable on its own, i.e., understandable by humans [4].

It is fair to assume that the models that currently generate the best accuracy values for the classification and regression tasks are those known in the literature as black box models [4]. In these models, such as artificial neural networks and Support Vector Machines, a data input is given to the model and an output is obtained, but there is not a simple explanation that can be given to human beings about the process that was performed to obtain the output. They are systems that do not reveal their internal mechanisms [9]. Questions we might want to be able to answer regarding these models are: What does the generated prediction mean? What motivated a certain prediction? Is there any causal relationship? Was the decision-making process free from bias and prejudice? Can I trust the result? How can I improve the model?

Several works have been carried out in an attempt to answer these questions, either in the sense of formalizing the problem of model interpretability or proposing solutions to make algorithms transparent.

The authors in [9] present a taxonomy of the methods used for interpretability. These methods can be classified according to several criteria, and can be: i) Intrinsic, if models are represented by simple structures considered to be interpretable – such as short decision trees or linear models, or Post Hoc, where methods are applied to analyze the model produced after training; ii) Model-Specific, if the interpretation occurs directly from the model, e.g., through the analysis of the weights of a regression or Model-Agnostic, if they can be used together with any machine learning model; and iii) Methods that provide Local interpretation, explaining an individual forecast, or Global, if they explain the entire behavior of the model.

According to these criteria, the method presented in this work is Post Hoc, Model-Agnostic (although the analyzes are focused on problems of symbolic regression with genetic programming) and is capable of providing both local and global explanations.

In [10] there is a discussion about when it is really necessary to have an explanation of a model and also how the explanations can be evaluated. The authors argue that often issues involving interpretability are used as a means to understand other possible issues, such as, for example, justice, impartiality, security, reliability and causality. In this way, the concern with interpretability is adequate for situations in which the problems are incompletely specified. In the evaluation stage, three possibilities are proposed: i) Application level evaluation (real task): domain experts participate in the evaluation; ii) Human level evaluation (simple task): laymen are used to test more general functions of the quality of an explanation; and iii) Function level evaluation (proxy task): it does not require humans and uses, for example, previous knowledge. In this case, it is assumed that someone has tested it with humans before. For example, it is known that humans can interpret the simple results of linear regression or decision rules, so this knowledge is used during evaluation. The method proposed in this work fits in this last category.

In the scenario of solutions to make algorithms transparent, most of past works focused on the classification problem. Among these works, some approach the problem in a non-agnostic way [11], [12] and others in an agnostic way by means of, for example, local linear approximations [13], by rules-based structures [14], [15] or game theory [16].

We highlight the work developed by [13], where a technique called LIME is presented to explain the predictions returned by general classifiers. Their method learns a local interpretable model around the prediction to be explained. They also present a method to explain the whole model instead of individual predictions. However, they propose to use perturbations in data entries in order to understand the local behavior of the model and to explain individual predictions. Although it can also be used for regression, the presented approach clearly focuses on classification problems. It is important to note that perturbations on data may lead to fictitious inputs that are invalid depending on the regression scenario. In this paper, we opt to use only the training set to explain the local behavior of predictions, that is, we do not generate potentially invalid instances. In addition, the proposed method is also able to provide a global view of the relevance of each attribute to obtain the output by means of an importance metric.

III. Explanation by Local Approximation

This section describes ELA (Explanation by Local Approximation), a method based on local explanations used to help justifying predictions made by a more complex regression model. This section introduces the method in the context of symbolic regression with GP, although ELA can be easily generalized to any other type of regression method.

Let us assume we have a training set $T = p_i = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and a test set $T' = p_j = \{(\mathbf{x}_j)\}_{j=1}^{m}$ — with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. ELA works by generating a non-linear regression model (e.g., symbolic regression) for predicting the value y' of data points in T', and then performs a local approximation with a linear regression method to generate explanations for the predictions generated by the first model, as illustrated in Algorithm 1.

Algorithm 1 Explanation by Local Approximation (ELA)

```
Require: $T$ (train), $T'$ (test), $k$ (# of neighbors)
1: $SR$ = Run GP($T$)
2: for each p $\in T'$ do
3:     $y'$= SR(p)
4:     $NN$ = Find $k$ nearest neighbours of $p$ considering only
       attributes in $SR$
5:     $LR$ = Run linear regression($NN$)
6:     Compute importance of attributes in $LR$
7:     Show local explanation for predicting $y'$
8: end for
9: Plot global explanation
```

The algorithm receives as input the training and a test sets, T and T', and the value k of the number of neighbours to be considered in the local explanation. First, we run the GP in the training set to find the function that describes the in T (line 1). Having this function, we explain (or justify) its prediction for each test point considering a local linear regression over a set of k neighbour points.

Given a test point p, we first use the original GP model SR to find the predicted value y' (line 3). Each GP function is represented by a tree, generated respecting a maximum tree depth. The GP functions are evaluated using the RMSE as the fitness fucntion. A tournament selection is used to select the individuals that undergo crossover and mutation operators.

Next, we find the k neighbours of p (line 4 and Equation 1). We use a d'-dimensional Euclidean distance to find the k nearest neighbors, as shown in Equation 2. Note that d' is a subset of all attributes x that are present in the function SR returned by the GP, and is in the interval $0, .., d$, where d is the number of predictive attributes.

$$NN_p = \underset{p_i}{\arg\min}\{dist(p, p_i)\} \tag{1}$$

$$dist(p, p_i) = \sum_{h=1}^{d'} \sqrt{(p^h - p_i^h)^2} \tag{2}$$

Having the k nearest points of an examples p, we use a linear regression method to find the function that best represents these k points (line 5). This linear equation is able to provide a local explanation of the prediction given to p considering its neighbours. Being a linear equation, the interpretability of LR is more straightforward than the interpretability of the function returned by the GP. Of course there are exceptions, specially if the number of nodes of the GP tree is small. However, note that this method is recommended for cases where the original function SR is difficult to be read and understood by a human.

Next, we analyze the coefficients of LR by calculating a measure of importance of each attribute x to the final prediction (line 6). Importance is defined as the contribution that each attribute exerts to the final value of the output of p, given by y'. As shown in Equation 3, we look at the module of the importance of each attribute by multiplying its coefficient by its value in p and normalizing it. This measure of importance, where the signal is ignored, is able to provide extra information about the proportional contribution of each attribute during the regression task, both from a local or a global. Recall that the user can also access the coefficients found by the local linear regression directly, providing an additional way of explaining the result obtained by the regressor.

$$Importance_{x_i} = \frac{|coefficient_{x_i} \times x_i| \times 100}{\sum_{h=1}^{d} |coefficient_{x_h} \times x_h|} \tag{3}$$

In order to provide a visual interpretation of the result to the user, we use the following procedure (line 7). We start by checking how different are the outputs of the K training examples in NN from the test point of interest p. As they are neighbours, we expect than to be similar. We consider that a difference higher than 10% of the maximum range of the output values in T should be disregarded. For example, if the outputs y in T are in the interval [1,10], a variation of 1 in the neighborhood is considered as the threshold of similarity for visual explanations. This threshold is used to reduce the risk of considering noisy data in the explanations, damaging the method. For the subset of neighbours in NN within this defined threshold, we look at the variation of the values of each attribute x in the local explanation. The process aforementioned is repeated for each test set, providing a justification for each prediction.

Finally, the proposed method also presents an global explanation of the behavior of the symbolic regression obtained when we have a large set of explanations for different test points. In order to obtain that, all test points are discretized according to the output variable in z equally spaced intervals. For each interval, the average importance of the subset of input attributes x is calculated. The result is then plotted on a graph of stacked areas, given insights of the importance of different attributes to the whole dataset (see Figure 3).

Figure 1 illustrates the results of ELA using one of the synthetic datasets considered in the experimental analyzes reported in Section IV. The real function, shown in black in the figure, is defined by Equation 4. The test point p we want an explanation for is represented by the black diamond, and the

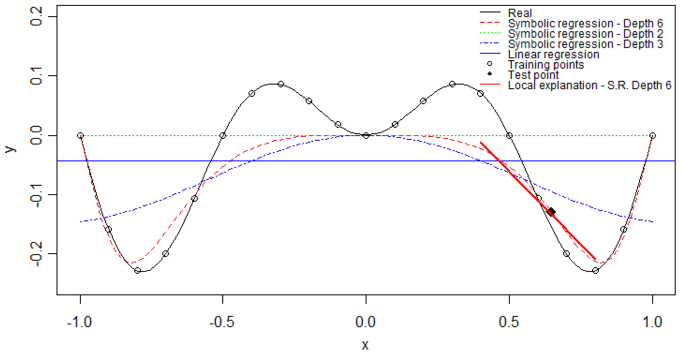

Fig. 1. Example of the method for the synthetic dataset keijzer-1.

red line going over the black diamond was generated by ELA. The other functions shown in the figure were generated by the GP using different maximum depths (a parameter that has a direct impact on model interpretability) and a simple linear regression using the whole set of training points in T. Note that the line produced by ELA follows the same direction of the function generated by the symbolic regression with depth 6, also represented in red.

$$f(x) = 0.3x\sin(2\pi x) \tag{4}$$

IV. Experimental Analysis

This section presents an experimental analysis of the interpretability of the proposed method. The symbolic regression method was implemented using the Python package Deap [17]. The method was also implemented in Python and is available for download[1].

The results of the proposed method are compared to the original function found by the GP and to a linear regression method run with no regularization, an L1 and an L2 normalization [18]. These models were chosen as they are considered interpretable regression methods, as the coefficients give us a notion of attribute importance.

Results are analyzed in three phases. First we perform an analysis of the different methods using all training points, considering both RMSE and model complexity. Next, we make a quantitative analysis of ELA using 10 synthetic datasets. Finally, we perform a qualitative case study considering a real-world dataset, namely the *wineRed* dataset.

A. Experimental Setup

We tested the proposed method in a set of 10 synthetic and one real-world dataset, which will be later used in a qualitative case study. The datasets are described in Table I, which shows the total number of attributes (predictive and target (Attr.), number of data points used in the training (Train) and test (Test) phases. All synthetic datasets were originally presented in [19]. The real dataset is available at the UCI repository [20], [21].

TABLE I
Datasets used in the experiments

Dataset	Attr.	Train	Test	Nature
keijzer-1	2	21	2001	Synthetic
keijzer-4	2	101	101	Synthetic
keijzer-7	2	100	991	Synthetic
vladislavleva-1	3	100	2025	Synthetic
vladislavleva-2	2	100	221	Synthetic
vladislavleva-3	3	600	5083	Synthetic
vladislavleva-4	6	1024	5000	Synthetic
vladislavleva-5	4	300	2700	Synthetic
vladislavleva-7	3	300	1000	Synthetic
vladislavleva-8	3	50	1089	Synthetic
wineRed	12	1279	320	Real

The GP functions are defined by the binary operations of addition (+), subtraction (-) and multiplication (x), in addition to the analytic quotient (AQ), which has the general properties of division but without discontinuity (see Equation 5). The terminals are the predictive attributes of the datasets.

$$Aq(a, b) = \frac{a}{\sqrt{1 + b^2}} \tag{5}$$

After a preliminary parameter tuning, the GP was executed with an initial population of 1,000 individuals evolved for 250 generations, using a tournament selection of size 7. The probabilities of crossover and mutation were defined as 0.8 and 0.2, respectively. The depth limit of the trees was varied to show the differences in the number of nodes of the solutions found and their impact in interpretability.

Considering the non-deterministic character of the results obtained by GP, all tests presented were executed 30 times, and the average RMSE and number of tree nodes are reported.

B. Regression with all points

We first analyze the error results of the symbolic regression and linear regression methods when run with all test points. This is important for two reasons: first, to show we need more than a linear regression to solve the problem; second, to assure the results of the local explanations will not increase the test error.

Table II shows the mean RMSE followed by their standard deviations in the training and test sets, respectively. Three maximum depths of GP trees were considered: 2, 3 and 6, to show the impact of the complexity of the model to the RMSE. The linear regression was run without any regularization and using both L1 and L2. The parameter that defines the regularization force (alpha) of the methods with L1 and L2 was set at 1.0.

Observe that the results of RMSE of the GP improved as we increased the depth of the tree. The results of the linear regression with and without regularization present no statistical difference for the synthetic datasets. The results of the GP with depth 6 are the best among all tested methods in all the 10 synthetic datasets.

However, note that synthetic datasets are used to give us a chance to analyze interpretability. In more complex data, the GP usually outperforms LR. As we are interested in model

[1] https://github.com/renatomir/ELA-WCCI2020

TABLE II
MEAN RMSE OF THE MODELS.

Dataset	TRAINING SET						TEST SET					
	Genetic Programming			Linear Regression			Genetic Programming			Linear Regression		
	Depth 2	Depth 3	Depth 6	LR	L1	L2	Depth 2	Depth 3	Depth 6	LR	L1	L2
keijzer-1	0.120 (0.000)	0.090 (0.002)	**0.041 (0.012)**	0.110	0.110	0.110	0.120 (0.000)	0.080 (0.002)	**0.042 (0.012)**	0.110	0.110	0.110
Keijzer-4	0.320 (0.000)	0.320 (0.000)	**0.272 (0.057)**	0.320	0.320	0.320	0.320 (0.000)	0.320 (0.000)	**0.272 (0.057)**	0.320	0.320	0.320
Keijzer-7	0.990 (0.000)	0.515 (0.075)	**0.150 (0.061)**	0.410	0.410	0.410	0.960 (0.000)	0.510 (0.074)	**0.149 (0.060)**	0.380	0.380	0.380
Vladislavleva-1	0.131 (0.005)	0.103 (0.010)	**0.055 (0.014)**	0.130	0.230	0.130	0.151 (0.004)	0.127 (0.008)	**0.096 (0.025)**	0.190	0.210	0.190
Vladislavleva-2	0.320 (0.000)	0.319 (0.004)	**0.228 (0.057)**	0.320	0.320	0.320	0.300 (0.000)	0.300 (0.003)	**0.217 (0.053)**	0.300	0.300	0.300
Vladislavleva-3	1.090 (0.000)	1.087 (0.005)	**0.913 (0.132)**	1.090	1.090	1.090	1.010 (0.000)	1.006 (0.005)	**0.855 (0.123)**	1.000	1.010	1.000
Vladislavleva-4	0.193 (0.005)	0.181 (0.003)	**0.158 (0.011)**	0.190	0.190	0.190	0.209 (0.004)	0.196 (0.008)	**0.166 (0.014)**	0.190	0.190	0.190
Vladislavleva-5	0.346 (0.034)	0.290 (0.069)	**0.118 (0.075)**	0.600	0.610	0.600	0.508 (0.043)	0.443 (0.088)	**0.214 (0.113)**	0.840	0.840	0.840
Vladislavleva-7	3.372 (0.011)	2.648 (0.385)	**1.738 (0.239)**	3.670	3.680	3.670	3.763 (0.014)	3.031 (0.423)	**2.070 (0.288)**	4.050	4.040	4.050
Vladislavleva-8	1.719 (0.002)	1.490 (0.056)	**0.764 (0.108)**	1.760	1.780	1.760	2.225 (0.020)	2.166 (0.085)	**1.537 (0.346)**	2.280	2.280	2.280

interpretability, Table III shows the complexity of the functions obtained by all methods. For the GP, the complexity is given by the number of nodes present in the trees of the returned individuals. For the linear regression methods, we report the number of coefficients different from 0.

Note that, for all datasets, the number of nodes of the GP trees when with maximum tree depth 2 and 3 are low, but the RMSE is high when compared to the version run with maximum tree depth 6. Also observe that the linear regression with L1, for most cases, returns a constant as the function that describes the data.

When a maximum tree depth of 6 is used, allowing for more complex models to be generated, the resulting number of nodes is high, which would preclude a level of interpretability suitable for the function to be understandable by a human being. Thus, from now on, our efforts are focused on locally approaching the curve obtained by the GP - D6 in order to have an interpretation of the model produced.

TABLE III
COMPLEXITY OF THE FUNCTIONS FOUND BY THE METHODS.

Dataset	Average size of the best individual			Non-zero coefficients		
	GP (D2)	GP (D3)	GP (D6)	LR	LR L1	LR L2
keijzer-1	4.9	13.6	67.9	0	0	0
keijzer-4	4.3	6.3	76.4	1	0	1
keijzer-7	7.0	15.0	78.5	1	1	1
vladislavleva-1	7.0	13.5	55.9	2	0	2
vladislavleva-2	3.9	7.2	67.7	1	0	1
vladislavleva-3	4.0	10.0	70.3	2	0	2
vladislavleva-4	6.4	11.1	50.0	5	0	5
vladislavleva-5	7.0	13.4	45.2	3	0	3
vladislavleva-7	7.0	14.8	75.5	2	0	2
vladislavleva-8	7.0	14.5	79.2	2	0	2

C. Quantitative Evaluation of ELA

ELA finds a local explanation to the model predictions based on the nearest neighbours of the point being predicted. This section tests the performance of the method and compares with the performance of the original functions obtained with all points.

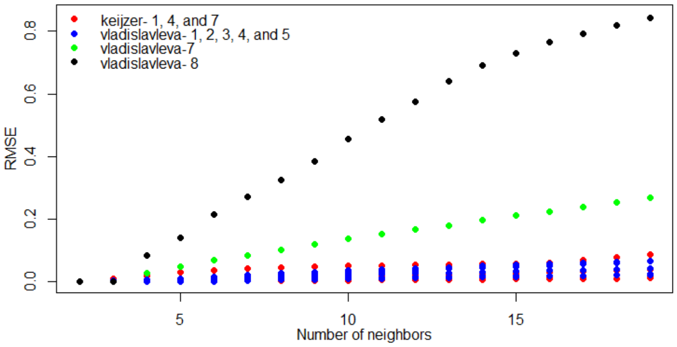

Fig. 2. RMSE variation of the local explanations with different numbers of neighbors

Influence of the value of k: We first make an analysis of the impact of the single parameter the proposed method has on the synthetic datasets, which is the number k of neighbors in the training set that will be considered in the local linear regression. We varied the value of k from 2 to 19.

As observed in Figure 2, the smaller the number of neighbors the smaller the error, but the higher the chances of overfitting the model to a very small set of points and generating a "false" explanation. In order to allow for a better generalization without significantly increasing the error, a value of k equals to 5 will be used in further experiments, as it represents a good trade-off between error and generalization.

Results of local approximation: Table IV shows the results of the RMSE found when comparing the real values of the training points used as neighbours for the local approximation by ELA with the predictions found by GP and ELA, and also compares the values predicted by GP and ELA. Note that these results consider the average RMSE of the points used to generate the local approximation for all test points. For example, in a hypothetical scenario with 10 test points and k equals 5, we have an average RMSE over 50 different training points. The rationale behind these results is to evaluate the impact that local models have in the errors of the training points. Observe that, for most cases, the real versus GP and real versus ELA errors are very similar, showing both functions are not very different in those regions of the space. On the other hand, it does not make sense to calculate the errors on the test set, as the predictions are made by the original model produced by the GP, and only the explanation uses this local model.

TABLE IV
MEAN RMSE OF NEIGHBORHOOD OF TEST POINTS.

Dataset	Real/GP	Real/ELA	GP/ELA
keijzer-1	0.039	0.057	0.029
keijzer-4	0.181	0.181	0.003
keijzer-7	0.128	0.130	0.006
vladislavleva-1	0.044	0.044	0.008
vladislavleva-2	0.144	0.144	0.005
vladislavleva-3	0.496	0.495	0.014
vladislavleva-4	0.128	0.128	0.000
vladislavleva-5	0.086	0.088	0.008
vladislavleva-7	1.607	1.607	0.045
vladislavleva-8	0.635	0.654	0.145
wineRed	0.610	0.610	0.000

As we can see, the RMSE average results between the real data and ELA are very close to the ones comparing the real data with the GP predictions. Additionally, the difference between the GP and ELA RMSE is considerably lower, in all cases, than the mean RMSE between GP and the real data. From that we can conclude that the proposed method is actually locally describing the behavior of the function obtained by the GP.

Finally, we show an example of the results found by ELA for a one-dimensional dataset, keijzer-1, with different regression methods and using ELA. As previously mentioned, the function this dataset represents is defined in Equation 4. As previously reported, the GP returned tree with an average number of nodes of 65.8 when running with maximum tree depth 6. The linear regression for the same data set, either by the common method or using L1 and L2, resulted in the following line:

$$LR(T) = -0.04397 \tag{6}$$

We chose the following point to explain the prediction given by the GP: $(0.647, -0.129)$. After selecting its 5 nearest neighbors, the function found was:

$$LR(NN) = -0.489x + 0.183 \tag{7}$$

Figure 1 shows that the local explanation we generate approximates well the equation obtained by the symbolic regression and is much simpler to interpret than the original function.

D. Qualitative case study: Wine dataset

TABLE V
WINE: COMPLEXITY OF THE FUNCTIONS FOUND BY THE GP METHODS.

Average size of the best individual		
GP (D2)	GP (D3)	GP (D6)
7.0	14.5	52.3

Here we analyze the dataset *wineRed*. We chose this dataset for two main reasons. First, it presents a nonlinear relationship between the attributes and the output. Next, the subject of the dataset, which describes the characteristics of wines of the red type and the output the quality, with notes varying between 0 and 10, can be interpreted with a low degree of expertise in the problem.

The set of attributes used to describe each instance in *wineRed* is as follows: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, and alcohol.

Analogously to what was observed in the synthetic datasets, in *WineRed* there is a relationship between the growth of the maximum tree depth of the GP and the complexity of the model in terms of the average size of the best individual (Table V) and smaller RMSE averages (Table VI).

Table VII shows an example of explanation found by ELA in GP (D6), where we show the test point and the value of the variable to be predicted. In this case, for a wine receiving note 6, there is an error of 0.223 (6.453 for GP and 6.676 for ELA). The table lists, apart from the coefficients of the linear regression produced by ELA, the importance of the attributes in the analyzed test instance.

A relevant feature that we can highlight at this point is that the ELA tool provides an additional interpretability feature that is the importance of attributes for regression of the test instance.Looking at the values of attribute importance, we can highlight two: alcohol (75.9%) and fixed acidity (8.8%).

The last lines in Table VII show the range of the values of the attributes of neighbors considering a maximum range of 10% of variance in the output. By verifying the modifications of the values of the attributes we can understand how wines similar to the presented could be manufactured. For example, the residual sugar attribute could range between 1.6 and 3.3.

Finally, since we have a large set of explanations for different test points, we can also provide an *overall explanation of the model* using a graph of stacked areas[2], as shown in the Figure 3, where the x axis represents the output of the method (in this case, the quality of the wine) and the y axis the relative importance of that attribute for a wine receiving that note. In this graph the attributes are sorted following the same order they are listed in Table VII: the first attribute is in the lower portion of the graph is fixed acidity in dark blue, the second is volatile acidity in dark green and so on, up to alcohol shown in red at the top of the graph area. Looking at this graph we observe that, regardless of the quality attributed to a wine, in general the most relevant attributes are alcohol (red) and total sulfur dioxide (green). The alcohol feature has an importance above 20% regardless of the quality of the wine. The Total sulfur dioxide feature is more important for intermediate quality wines, more than 30%, when compared to high and low quality wines. In addition, we also observed that the Fixed acidity and pH features have a behavior opposite to Total sulfur dioxide, that is, they are more relevant in the

[2] https://www.chartjs.org/

TABLE VI
WINERED: MEAN RMSE OF THE GP MODELS.

TRAINING SET			TEST SET		
Depth 2	**Depth 3**	**Depth 6**	**Depth 2**	**Depth 3**	**Depth 6**
0.724 (0.037)	0.672 (0.020)	**0.647 (0.013)**	0.739 (0.035)	0.691 (0.030)	**0.667 (0.016)**

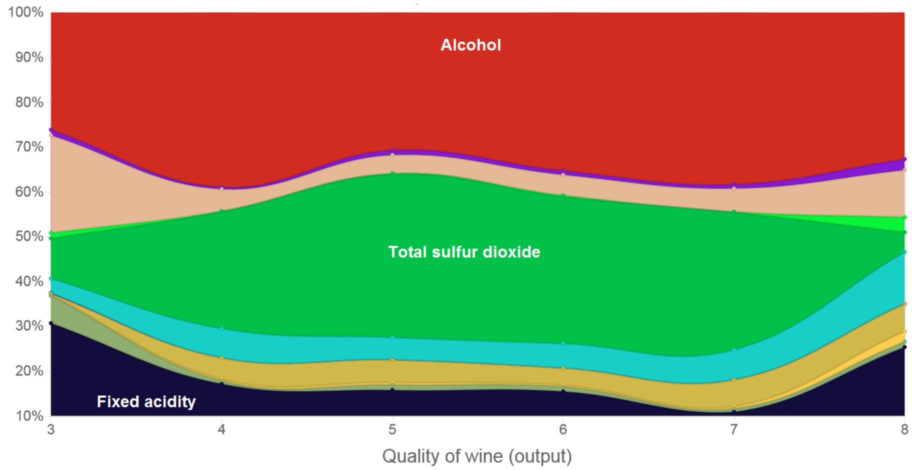

Fig. 3. Global explanation for attributing quality to a wine. The x-axis represent the note the wine received, the y-axis the relative importance of the feature when that note is attributed to a wine.

TABLE VII
INFORMATION OF THE REGRESSION METHODS AND ELA FOR THE WINERED DATASET.

Test point analyzed
= [10.8, 0.29, 0.42, 1.6, 0.084, 19, 27, 0.99545, 3.28, 0.73, 11.9]
Real value = 6.0
Value predicted by GP= 6.453
Value predicted by Local Explanation= 6.676
Coefficients of linear regression
= [-0.09, 0.32, 0.05, -0.17, -0.04, 0.03, -0.01, 0.02, 0.11, 0.18, 0.74]
Intercept= -1.8424176877057139
Importance of attributes
= [8.82, 0.79, 0.17, 2.39, 0.03, 5.46, 1.82, 0.14, 3.24, 1.17, 75.95]
Variation of attributes among neighbors within a maximum range of 10% of variance (6.4313 - 6.6138):
7.9 <= fixed acidity <= 10.8
0.2 <= volatile acidity <= 0.33
0.35 <= citric acid <= 0.42
1.6 <= residual sugar <= 3.3
0.054 <= chlorides <= 0.084
6.0 <= free sulfur dioxide <= 19.0
15.0 <= total sulfur dioxide <= 27.0
0.99458 <= density <= 0.99545
3.28 <= pH <= 3.32
0.73 <= sulfates <= 0.8
11.8 <= alcohol <= 12.0

qualification of lower and upper wines and less important in intermediate wines.

Sulfur dioxide is used in wine-production as a preservative due to its anti-oxidative and anti-microbial properties, but also as a cleaning agent for barrels and winery facilities. Studies have shown that wines may have their sensorial attributes deteriorated (e.g., oxidise) if the concentration of free sulfur dioxide falls below a particular critical level, specific for each particular wine. Also, wines with higher pH values may deteriorate at higher critical levels of free sulfur dioxide [22].

We can also note that the fixed acidity attribute is more important in the task of assigning notes for low and high quality wines, losing its importance among intermediate quality wines.

V. CONCLUSIONS AND FUTURE WORK

This work presented ELA, a method capable of generating interpretable explanations for the results provided by regression algorithms. Differently from other approaches previously presented in the literature, the proposed method uses the neighborhood concept of a certain test point of interest to carry out a local linear regression and identify how much each input attribute influences the output. The strategy adopted also provides ranges by which the attributes can be changed locally, bringing more information for interpretation. In addition, a graph-based view of stacked areas is proposed to provide an overview of the overall behavior of the model. The experiments showed that the explanations provided have a strong approximation with the results obtained by the symbolic regression method in terms of RMSE, besides providing useful explanations for understanding the results.

As future work we intend to explore the proposed method in a wider range of real-world data and verify their level of interpretability with experiments supported by lay users and also specialists of the domain. Another line of investigation to improve the method is studying the impact of a variable neighborhood size, set according to the local density of the training set. For example, the denser the more neighboring instances could be used in the approximation. Finally, a more detailed study of the range allowed for variation of the attributes can also improve the method, given the user more accurately information about the variations of attributes values close to the test point.

Acknowledgments

The authors would like to thank FAPEMIG (through the grant no. CEX-PPM-00098-17), MPMG (through the project Analytical Capabilities), CNPq (through the grant no. 310833/2019-1), CAPES, MCTIC/RNP (through the grant no. 51119), H2020 (through the grant no. 777154) and IFMG - Campus Sabará for their partial financial support.

References

[1] S. Hajian, F. Bonchi, and C. Castillo, “Algorithmic bias: From discrimination discovery to fairness-aware data mining,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 2125–2126. [Online]. Available: http://doi.acm.org/10.1145/2939672.2945386

[2] E. O. of the President and P. H. Press, *Big Data: A Report on Algorithmic Systems, Opportunity, and Civil Rights*. CreateSpace Independent Publishing Platform, 2016. [Online]. Available: https://books.google.com.br/books?id=qV_3vQAACAAJ

[3] M. E. Ahsen, M. U. S. Ayvaci, and S. Raghunathan, “When algorithmic predictions use human-generated data: A bias-aware classification algorithm for breast cancer diagnosis,” *Information Systems Research*, 2018.

[4] R. Guidotti *et al.*, “A survey of methods for explaining black box models,” *ACM Comput. Surv.*, vol. 51, no. 5, pp. 93:1–93:42, Aug. 2018. [Online]. Available: http://doi.acm.org/10.1145/3236009

[5] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1 – 38, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370218305988

[6] P. G. Espejo, S. Ventura, and F. Herrera, “A survey on the application of genetic programming to classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 2, pp. 121–144, 2010.

[7] T. McConaghy and G. Gielen, “Canonical form functions as a simple means for genetic programming to evolve human-interpretable functions,” in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 855–862.

[8] Z. C. Lipton, “The mythos of model interpretability,” *Queue*, vol. 16, no. 3, p. 31–57, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3236386.3241340

[9] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[14] ——, “Anchors: High-precision model-agnostic explanations,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[10] F. Doshi-Velez and B. Kim, *Considerations for Evaluation and Generalization in Interpretable Machine Learning*. Cham: Springer International Publishing, 2018, pp. 3–17. [Online]. Available: https://doi.org/10.1007/978-3-319-98131-4_1

[11] C. Lv and D.-R. Chen, “Interpretable functional logistic regression,” in *Proceedings of the 2Nd International Conference on Computer Science and Application Engineering*, ser. CSAE ’18. New York, NY, USA: ACM, 2018, pp. 82:1–82:5. [Online]. Available: http://doi.acm.org/10.1145/3207677.3277962

[12] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6541–6549.

[13] 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: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778

[15] E. Pastor and E. Baralis, “Explaining black box models by means of local rules,” in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC ’19. New York, NY, USA: ACM, 2019, pp. 510–517. [Online]. Available: http://doi.acm.org/10.1145/3297280.3297328

[16] E. Strumbelj and I. Kononenko, “An efficient explanation of individual classifications using game theory,” *J. Mach. Learn. Res.*, vol. 11, pp. 1–18, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756007

[17] F.-A. Fortin. *et al.*, “DEAP: Evolutionary algorithms made easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[19] M. Keijzer, “Improving symbolic regression with interval arithmetic and linear scaling,” in *Proceedings of the 6th European Conference on Genetic Programming*, ser. EuroGP’03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 70–82. [Online]. Available: http://dl.acm.org/citation.cfm?id=1762668.1762676

[20] D. Dheeru and E. Karra Taniskidou, “UCI machine learning repository,” 2017. [Online]. Available: https://archive.ics.uci.edu/ml/

[21] 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, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2009.05.016

[22] P. GODDEN *et al.*, “Wine bottle closures: physical characteristics and effect on composition and sensory properties of a semillon wine 1. performance up to 20 months post-bottling,” *Australian Journal of Grape and Wine Research*, vol. 7, no. 2, pp. 64–105, 2001. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-0238.2001.tb00196.x