arXiv:2404.05908v1 [cs.LG] 8 Apr 2024

Interpretability in Symbolic Regression: a benchmark of Explanatory Methods using the Feynman data set

Guilherme Seidyo Imai Aldeia[1*] and Fabrício Olivetti de França[1]

[1*]Center for Mathematics, Computation and Cognition, Federal University of ABC, Av. dos Estados, Santo Andre, 09210580, SP, Brazil.

*Corresponding author(s). E-mail(s): guilherme.aldeia@ufabc.edu.br; Contributing authors: folivetti@ufabc.edu.br;

Abstract

In some situations, the interpretability of the machine learning models plays a role as important as the model accuracy. Interpretability comes from the need to trust the prediction model, verify some of its properties, or even enforce them to improve fairness. Many model-agnostic explanatory methods exists to provide explanations for *black-box* models. In the regression task, the practitioner can use *white-boxes* or *gray-boxes* models to achieve more interpretable results, which is the case of symbolic regression. When using an explanatory method, and since interpretability lacks a rigorous definition, there is a need to evaluate and compare the quality and different explainers. This paper proposes a benchmark scheme to evaluate explanatory methods to explain regression models, mainly symbolic regression models. Experiments were performed using 100 physics equations with different interpretable and non-interpretable regression methods and popular explanation methods, evaluating the performance of the explainers performance with several explanation measures. In addition, we further analyzed four benchmarks from the GP community. The results have shown that Symbolic Regression models can be an interesting alternative to white-box and black-box models that is capable of returning accurate models with appropriate explanations. Regarding the explainers, we observed

that Partial Effects and SHAP were the most robust explanation models, with Integrated Gradients being unstable only with tree-based models. This benchmark is publicly available for further experiments.

Keywords: Symbolic Regression; Explanatory Methods; Feature Importance Attribution; Benchmark.

1 Introduction

The large diffusion of Artificial Intelligence and, more specifically, Machine Learning (ML), has become a reality over the last decades. ML has been seen as a promising application across multiple fields, such as judicial system [1], healthcare [2], scientific discovery in natural sciences [3], and credit lending [4]. Many state-of-the-art methods can achieve better performance than experts of the field with high dimensional and non-linear models [5–9]. Although ML can present high accuracy in several applications, it is arguable that many ML methods generate models that are incomprehensible for a human practitioner. These difficult-to-understand models, also called black-box models, may have undesirable consequences when applied in real-world applications.

The main motivations for explaining predictions and understanding the models are the trust (or lack of) that the model predictions will not lead to catastrophic results; to ensure fairness in the decision process, and to better understand the phenomena under study. The field that studies interpretability is known as *eXplainable Artificial Intelligence* (XAI), where interpretability plays a central role. The recent focus of XAI works was on either improving the accuracy of innate interpretable models or extracting explanations from black-box models.

The motivation for using black-box models is that their accuracy is usually higher than simpler and interpretable models [10]. One way to extract explanations from these kinds of models is to use an explanatory method [11–13]. These explanatory methods highlight certain aspects of the model with strategies such as a visual guide of the decision process, human-readable explanation, counterfactual explanation, perceived importance of each feature, among others. Particularly regarding feature importance, an explanatory method can return the aggregated importance of each feature to the decision process, the importance of the features for a single prediction, and the behavior of the feature importance as a function of its own value.

Alternatively to using black-box models and explaining them with explanatory methods, we can use more interpretable models that still provide accurate predictions. One such example is the Symbolic Regression (SR) [14], a regression method that searches for analytical models that best fit the training data. SR is a well-established field that presents competitive performance when compared to state-of-the-art regression algorithms, especially in some domains where a mathematical expression can describe the studied phenomena (*i.e.,*

natural science domain). The downside is that sometimes SR is performed with a higher computational cost than traditional regression techniques [15, 16].

Therefore, SR can be further explored in the XAI field when considering the regression task. Previous works studied the application of SR in XAI tasks, such as [14], where the authors introduce a model-specific explainer for Symbolic Regression models, exploiting the fact that it returns an analytical model. Another benefit of working with analytical models is the possibility of enforcing certain properties to the generated model, such as monotonicity, convexity, or symmetry [17]. Different methods have been proposed to improve performance and interpretability of the SR results [18], and SR has a potential of obtaining easier-to-interpret equations [18–23]. In the interpretability context, there are works that uses Symbolic Regression as an explanatory method, being used to generate simpler models to explain black-box predictions [20, 24].

One problem with SR is that, while some authors claim it as being more interpretable than black-box approaches, the interpretability is usually measured as the size of the generated expressions [25], a vague notion of interpretability that only suits methods that return mathematical expressions. Because of that, there is still a gap between the promising application of SR in contexts where interpretability plays a central role. Thus, the explanatory methods could be more explored.

This paper attempts to provide insights and to investigate the quality of interpretability in the context of Symbolic Regression, extending the work in [14]. The main objective is to investigate the interpretability of Symbolic Regression models when compared with other regression methods by using different feature importance explanatory methods.

For this purpose, we evaluate two different Symbolic Regression algorithms and compare them to other regression methods in the interpretability spectrum — from a white-box linear regression to a black-box Multi-Layer Perceptron Neural Network — with multiple explanatory methods. We created synthetic data sets from known physical equations as a proxy to train and evaluate the ML and explanatory methods. Several measures in the literature are revisited and adapted to measure interpretability quality in the experiments. As a by-product, we make all the source code available in the form of a python module called *iirsBenchmark*[1].

The *iirsBenchmark* creates a unified experimental design to generate results to compare interpretability measures with different regression methods. We expect to achieve the following objectives:

O1 Provide an evaluation method to assess the performance of different explanatory methods for the regression task through defining and evaluating different quality measures;
O2 Compare two different symbolic regression methods against many popular regression methods that range from the white-box to the black-box spectrum of interpretability.

---

[1]Open source module available at https://github.com/gAldeia/iirsBenchmark.

In summary, our contributions to the scientific community are *i)* the *iirs-Benchmark*, an open-source framework to measure explanations quality that unifies the interface of many popular explanatory methods; *ii)* the revision and proposal of measures to quantitatively evaluate the quality and robustness of explanations; *iii)* a robust background to lay the groundwork of interpretability in the symbolic regression context; and *iv)* an extensive comparative study of explanatory methods, using several regression methods and two hand-picked symbolic regression methods. We expect our work to highlight the benefits of using symbolic regression as an alternative to black-box methods, providing a framework to evaluate explanatory methods with special attention in the symbolic regression context. We discuss the evaluation of explanatory methods and propose a framework expected to help researchers present new and more robust methods by providing new perspectives when evaluating, reporting, and analyzing the results.

The remainder of this paper is organized as follows. Section 2 presents a revision focusing on benchmarking symbolic regression and explanatory methods. Section 3 presents the theoretical background to the interpretability field in symbolic regression. Section 4 presents popular feature importance explanatory methods for the regression task. Section 5 presents measures to evaluate the quality and robustness of explanations. Section 6 presents the experimental methods and the python package used in this paper. Section 7 reports the results, which are discussed in Section 8. Finally, Section 9 revisits the objectives and concludes the work, summarizing the findings and pointing out new directions.

2 Related work

The term *black-box* model is widely adopted to describe ML models that are complex and lack transparency. There are different definitions for what a black-box model is: a component that does not reveal anything from its inner design, structure or implementation [13] (or it reveals the structure, but is too complex to understand [25]); a model derived purely from data with no knowledge about its inner working [26]; a model that is not inherently interpretable (*i.e.* they need *post-hoc* explanatory methods); or a model that is incomprehensible to humans or proprietary [27, 28]. All different definitions relate to how well it is possible to understand the resulting model without additional tools (is it "transparent" enough to let us see inside? Do its individual components make sense? Can a human simulate the decision process of the model, or be able to predict its output by looking at it?). In contrast, *white-box* models are defined as a: model that can be decomposed into individual parts with an explicit meaning on the problem domain [29, 30]; or a model that does not need any external processes to determine the meaning of its decision process [24, 27].

However, why do we need to understand high-performance ML models? Understanding its behavior can help detect wrong assumptions derived from

biased data, understand the decision process, and gain insights into the domain problem. For example, in [31] the authors discuss a judicial system that associated a higher likelihood of black people committing future crimes; and in [32] the authors proposed automated experiments to detect biased models, which helped them find a biased behavior in a job recommender system, that suggested lower-paying roles to women. Many other examples were reported in [25] showing unfair and discriminatory models against minorities and women.

A debate has started about the relevance of understanding the underlying process guiding the decisions of black-box models, both in the scientific community as well as in the public setting [13].

Nevertheless, the term *interpretability* still lacks a formal definition, even though the intuitive definition can be reasonable. In [33] the authors noticed that in many works, the terms are used without specifying their meaning. Some authors adopt *interpretability* and *explainability* interchangeably [34]. Interpretability is seen as: a passive characteristic of a model, in the sense of how much it makes sense to a human [35]; the ability to explain or present in comprehensible ways the model to a human [27, 36]; a domain-specific notion, where an interpretable model is useful to someone or obeys domain knowledge [28]; the description of the inner functioning of the system in a comprehensible way to humans [37]; or the ability to explain or provide meaning without the necessity of additional information [25].

To improve interpretability, one can opt to use white-box models, those considered intrinsically interpretable. Examples are linear models, decision trees, and decision rules [28]. The downside of using simple and interpretable methods is that they sometimes present a less accurate model when compared to black-box models [10]. Between the white-box and black-box endpoints of the interpretability spectrum, several models cannot be classified as either of those, also called gray-box models, sometimes seen as a compromise between high accuracy and more interpretable models — although some authors, for example, in [28], argue that there is not necessarily a trade-off between them. There is also a belief that complex models are more accurate, which does not hold in every situation. The use of more interpretable models should be preferred when interpretability is a concern. Notice that a white/gray/black-box classification is sometimes subjective, and any white-box model can become gray or black-box depending on the situation, for example, a linear model with many features or a decision tree for a regression problem.

Another possibility is to use a *post-hoc Explanatory Method*, that returns an *explainer model* used to generate *explanations* to help understanding an ML model. The explanation can be a feature importance attribution, visual, prototype, or counterfactual explanation. The explanation is used to work around the lack of transparency of the model. One popular explanation is the feature importance which usually returns a vector with numerical values indicating how much each feature contributes to the model prediction. Despite its popularity, it is still an open question how to validate the performance of

feature importance explanatory methods since interpretability lacks a rigorous definition [33].

   While interpretability is an important concern and an active field of study, we also notice that some authors argue otherwise. In [38], the authors questions if interpretability in the healthcare context is necessary, then designed, proposed, and evaluated an ML tool that had other mechanisms besides interpretability to provide a trustable and accountable system. In [39] the authors criticize the quality of the current explanatory methods. They argue that evidence already shows existing biases toward over-trusting computer systems — a system that provides explanations can increase the confidence in it, thus leading to decreased vigilance and auditing of these systems.

   Even though there are many explanatory methods proposed recently, some works in the literature report results showing that explanatory approaches do not provide a better understanding of the problem. In [40], the authors try to explain deep learning models using explanatory methods, which point to all regions of the image and relies on non-trivial proxies, being unable to explain the model behavior. To obtain more robust explanatory methods, we first need to provide a proper evaluation methodology, which is considered an open question in the field. Although many feature importance methods were proposed and widely adopted, literature reports several problems on using some of the state-of-the-art interpretability methods, as stated in many recent works [28, 41–47]. To cite some, in [44], experiments have shown that explainers are prone to select non-important features as the most important feature. In [45] the authors compared different explanatory methods, finding that state-of-the-art explainers are sensitive to irrelevant variations in the input data, resulting in very different explanations. Authors in [46] made criticisms to a class of explanatory methods based on permutations, showing that complex ML methods learn correlation structures that are disrupted when permuting values. Permutation-based methods break the relationship of the features on the black-box model, presenting wrong explanations. In [47] the authors have shown that different feature importance explanatory methods return different and opposing explanations for similar inputs.

   Wrong explanations can have multiple causes, such as a misprediction of the black-box model (we must assume that the prediction model is not perfect), an inaccurate explainer (due to lack of data around a specific point), among others. This means that evaluation of explanations must consider that poor performance is not necessarily related to the explainer but also the explained model. In [43], the authors argue that the explanatory methods can rely on learned artifacts from the black-box model instead of actual knowledge learned from the data. Even if the explanatory method is able to generate flawless explanations, we notice that the prediction model can use non-rational correlations. For example, in [48] authors inspected an interpretable rule-based algorithm for predicting pneumonia death risk and found that their model presented a high correlation between patients with asthma having a more

negligible death risk, and, after some investigation, they found a causal relationship between the medicine used by asthma patients and the lower risk of death. Authors in [47] show that classification accuracy positively correlates with explanation accuracy. Many other works [46, 47, 49] have shown that learned interactions of the black-box model could not be correctly handled by the explainers, leading to wrong explanations.

It is still considered an open question how to validate the faithfulness of the explanations concerning the explained model [44]. In addition, the lack of data with available ground-truth makes it hard to quantitatively measure explanation quality [14, 42, 50], making experiments prone to experimenter biases. In order to evaluate the correctness of explanations, having a ground-truth is essential. Several works in the literature explores and proposes different ways of generating synthetic data sets to evaluate explanations [14, 41, 46, 47, 49, 51]. When working with real-world data [42–45, 50, 52], the ground-truth is established heuristically. Another important view of this issue is that, since explanations are meant to be used by humans, there should be a human-centered evaluation for such explanations. Some research groups argue that the XAI field should actively work together with Human-Computer Interface and Social Science fields to move towards intelligible interpretations of prediction models [53].

Regarding symbolic regression methods, prior works [54, 55] raised different questions concerning the evaluation of symbolic regression in the literature, pointing out problems such as the usage of toy data sets; lack of reproducibility; and poor experimentation, statistics, and reporting. In [54] the authors surveyed the Genetic Programming (GP) community and presented some points to consider when designing symbolic regression benchmarks. Apart from that, one of the promising applications of symbolic regression is discovering new physics equations and describing systems by finding appropriate expressions. Introduced recently in [56] and further investigated in [57], the Feynman benchmark comprehends 100 problems that can be explored by symbolic regression, which has been proven to be a complex task to solve these problems by finding the exact expression without domain-specific improvements to current algorithms, providing a challenging task such as the other GP problems, but with real-world physics equations.

3 Symbolic Regression

Regression Analysis is the task of estimating the expected value of a dependent feature Y, also known as target-value or outcome, conditioned to a set of independent features [58]. The main goal is to understand the structural relationships between the dependent and the independent features. It is based on the assumption that an unknown function $f(\mathbf{X}) = Y$ describes the relationship between the independent and dependent features. From a modeling perspective, many regression algorithms parts from some assumptions about this

relationship and chooses an appropriate model $\widehat{f}(\mathbf{X}, \beta)$, where β is adjusted to minimize the approximation error of the model[59].

A simple and well-known model is the linear regression, that predicts an outcome with the linear function:

$$\widehat{f}(\mathbf{X}, \beta) = \beta_0 + \mathbf{X}^T \boldsymbol{\beta}. \tag{1}$$

The interpretation of this model can be straightforward in the context of feature importance: we can say that by increasing X_i in one unit, the prediction will increase in β_i units; or that the i-th feature contributed with $\beta_i X_i$ to the outcome from the reference prediction β_0.

Despite the straightforward interpretation, the linearity assumption limits the achievable accuracy of this model for many data sets. This can be alleviated with manual feature engineering by adding feature interaction and non-linearity. Another possibility is to use one of the linear model generalizations such as Generalized Linear Models and Generalized Additive Models.

Symbolic Regression (SR) is another alternative to generating a regression model. It usually starts from a randomly initialized free-form mathematical expression and performs the search and optimization of both model structure and its inner coefficients [60]. The search space of SR is constrained by how the mathematical expressions are represented and the primitive set of mathematical structures [19]. This approach has more potential to find interpretable solutions than black-box approaches, as argued by many authors of the field [18–20, 24].

The most common representation is the expression tree, where each node is either a **leaf node**, holding a constant value or feature symbol; or an **inner node**, representing an n-ary function f, followed by n child nodes. For a given observation $\mathbf{X}$, the terminal symbols are evaluated as the constant value it holds, or as the value that a feature X_i assumes on $\mathbf{X}$. The inner nodes are evaluated as the application of the f function over its m children $t_1, t_2, \ldots, t_m$, returning $f(t_1, t_2, \ldots, t_m)$.

Symbolic Regression can be seen as an optimization problem in which we want to find the sub-optimal expression $g*$ that minimizes a cost function from the set $\mathbf{G}$ of all representable mathematical expressions.

3.1 Genetic Programming for Symbolic Regression

Genetic Programming is an evolutionary algorithm commonly used for Symbolic Regression [61]. The search for the sub-optimal expression starts with a random population of expression trees, created using a primitive set with functions nodes **f** and constant nodes **c**, with tree depth limited by constraints Γ. While a stop criteria (*e.g.* a fixed number of generations) is not met, the algorithm creates the next generation as children from the current population that will compete to replace the previous individuals with probabilities taken from their fitness evaluated over the training data. When the stop criteria is

---

Algorithm 1 Canonical Symbolic Regression

Input: Function set **f**, constant set **c**, tree constraints Γ, stop criteria Ω, population size p, train data $(\mathcal{X} = \{(x_1, x_2, \ldots, x_d)_i\}_{i=1}^n, \mathbf{Y} = \{y_i\}_{i=1}^n)$
Output: Symbolic expression $\widehat{f}$
1: pop $\leftarrow$ [GENERATE_RANDOM_TREE(**f**, **c**, Γ) for _ $\in$ [1, 2, ..., p]];
2: **while** Criteria Ω is not meet **do**
3:    parents $\leftarrow$ [SELECT_N_PARENTS(pop) for _ $\in$ [1, 2, ..., p]];
4:    children $\leftarrow$ [CROSSOVER(p1, p2, ...) for (p1, p2, ...) $\in$ parents];
5:    mutants $\leftarrow$ [MUTATE(c) for c $\in$ children];
6:    pop $\leftarrow$ REPLACE(pop ++ mutants, $\mathcal{X}$, $\mathbf{Y}$);
7: **end while**
8: **return** arg max [FITNESS(p, $\mathcal{X}$, $\mathbf{Y}$) for p $\in$ pop];

---

met, then it returns the best expression of the last existing generation. The whole process is summarized in Algorithm 1.

In this approach, both the function structure and the free parameters are created and adjusted by the simulated evolutionary process. In the original SR implementation, by Koza [61], the algorithm finds values for the free parameters using two different strategies: the Ephemeral Random Constant (ERC), which creates constants by taking values at random from a pre-defined interval; or by using pre-defined constant values (such as π, Euler's constant or arbitrary numerical values).

Several free parameter adjustment methods were proposed later, with approaches based on *gradient descent* or *hill-climbing* optimization methods, as identified by the literature review in [62].

3.2 State-of-the-Art Symbolic Regression

In [63], the authors proposed an extension to the original GP, an algorithm named GP-NLS. In this extension, the authors added two additional steps right before evaluating the fitness of an expression. This implementation is currently available in the Operon C++ Framework [64].

In [16], the authors compared different regression algorithms with other contemporary symbolic regression methods in a large set of benchmark problems, including the Feynman data sets presented in [56].Overall, the Operon GP-NLS was the best-ranked algorithm, indicating that it provides the best average performance for different problems.

The first step expands every individual in the population by adding an offset and a scaling node at the root of the tree, also adding a coefficient to every feature node. The second step adjusts the newly added free parameters by solving a non-linear least-squares problem, using the Levenberg–Marquardt algorithm, adjusting the values of all free parameters. Fig. 1 shows an expression tree before and after the expansion. Notice that this modification does
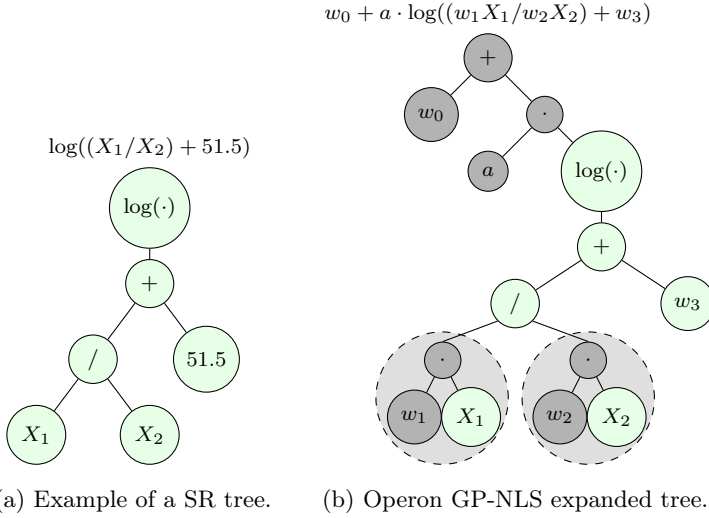
(a) Example of a SR tree.   (b) Operon GP-NLS expanded tree.

Fig. 1: Operon GP-NLS expanded tree (new nodes in gray, original tree nodes in a lighter color). The offset node w_0 is summed with the original tree, now scaled with the scale node a. Every feature has a coefficient associated with it (w_1, w_2). The original coefficient — a fixed value — is transformed into a free parameter w_3. The Levenberg-Marquardt algorithm will find optimal values for all free parameters in the expression tree ($\{w_0, a, w_1, w_2, w_3\}$)

.

not persist during the crossover and mutation — the evolutionary operators are applied only to the original tree.

We represent an Operon tree by $M_{g,\theta}$, where g is the differentiable function of the expanded tree T'n and θ is the vector of free parameters to be optimized in T'. Let $H : \mathbb{R}^p \to \mathbb{R}^n$ be a function that evaluates the parameter θ and returns the difference between the output of the model $M_{g,\theta}$ and the real value for observations $\mathcal{X}$:

$$H(\theta) = G(M_{g,\theta}, \mathcal{X}) - \mathbf{Y}, \tag{2}$$

with $G : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ being a function that evaluates the prediction of the model $M_{g,\theta}$ for the observations in $\mathcal{X}$.

The optimal coefficients can be found using the Jacobian matrix of H, through an interactive process of gradient descent, in which each interaction uses a step $\Delta\theta$ obtained by the linearization $H(\theta + \Delta\theta) \approx H(\theta) + J(\theta)\Delta\theta$. This process can be seen as a non-linear optimization problem.

The Operon GP-NLS algorithm follows the same steps as in Alg. 1, with the addition of creating and adjusting the free parameters. The initial population is created using the PTC2 initialization method [65], which allows controlling both depth and number of nodes in the generated trees while favoring balanced trees.

3.3 Interaction-Transformation representation

One way to improve the simplicity of the models found by SR algorithms is to constrain the search space to function forms, thus excluding complex structures such as function chaining. The *Interaction-Transformation* (IT) representation [19] was proposed to constrain the search space to an affine combination of non-linear transformations applied to interaction terms.

For a problem with d input features $\mathbf{X} = (X_1, X_2, \cdots, X_d)$ the IT representation is any regression model of the form:

$$\hat{f}(\mathbf{X}, \beta) = \beta_0 + \sum_{j=1}^{t} \beta_j \cdot g_j(p_j(\mathbf{X}, \mathbf{k}_j)), \tag{3}$$

where $\beta_j \in \mathbb{R}$ is the coefficient of the j-th term, and β_0 is the expression intercept. The function $g_j : \mathbb{R} \to \mathbb{R}$ is any unary function, called transformation function, and p_j is the interaction function, defined as:

$$p_j(\mathbf{X}, \mathbf{k}) = \prod_{i=1}^{d} X_i^{k_i}, \tag{4}$$

with $\mathbf{k} = \{k_1, k_2, \ldots, k_d\} \in \mathbb{Z}^d$ is the strength of interaction for the feature X_i in the interaction function index j.

In [21] the authors proposed the Interaction-Transformation Evolutionary Algorithm (ITEA) to search for the IT expression that best fits the data. The ITEA searches for transformation functions g_j and strengths k_{ij} for each j. Ordinary least squares can determine the values of β. ITEA follows a similar evolutionary process as in Alg. 1 but applies only the mutation operator to create the children expressions. The mutation operators can expand, shrink, or make local adjustments to an IT expression:

- **Expand:** mutations that add a new pair $(g_i, \mathbf{k}_j)$ to the expression. The new IT term can be a random one or can be the positive/negative element-wise combination of two existing terms on the expression;
- **Shrinkage:** performs the removal of a random existing pair $(g_i, \mathbf{k}_j)$ of the expression;
- **Local modification:** randomly changes one value in the interaction strengths $\mathbf{k}_j$ for a random j without adding or removing any new structures.

4 Explanatory Methods

Explanatory methods can alleviate the lack of transparency of black-box models when an explanation of the decision process is required. They are called *post-hoc explanatory method* since they are created after fitting the ML model. These methods can also be useful for white-box models in situations where the model becomes unintelligible due to an *a priori* feature engineering or
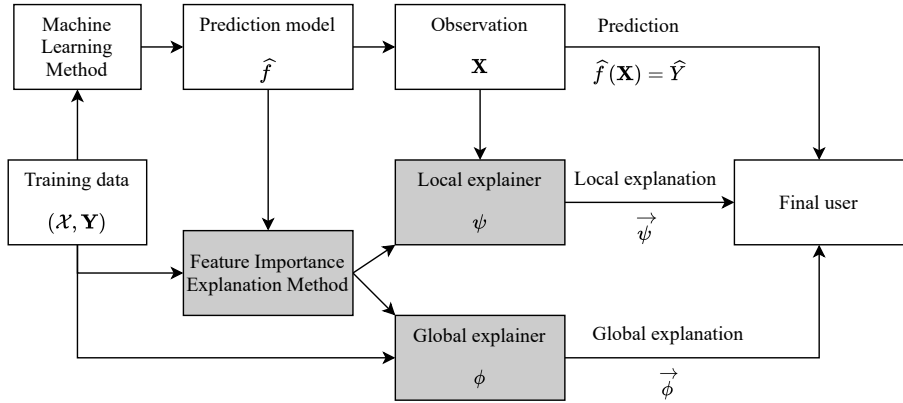
Fig. 2: Diagram illustrating the relationships between the black-box with its predictions and the explainer with its explanations. The ML prediction model is used as an input together with the training data. Then, it generates feature importance explainers to help understand the model. Adapted from [66].

due to the high dimensionality of a data set. The explanatory method usually returns an *explainer model* capable of providing explanations on demand for new examples or summarizing the behavior of the ML model.

One popular strategy for explaining a model is the feature importance explanation. It generates an explanatory function that inputs the predictor and the training data. Then, the function provides explanations by creating a vector assigning a numerical value for each feature, indicating their relative importance to the prediction. This explanation can be local when the explainer is used for single observations, or global when the behavior of the predictor is summarized as the expected importance of each feature.

Definition 1 (Local and Global feature Importance Explanations) Given a regressor $\widehat{f} : \mathbb{R}^d \to \mathbb{R}$ trained with a data set $(\mathcal{X} = \{(x_1, x_2, \ldots, x_d)_i\}_{i=1}^n, \mathbf{Y} = \{y_i\}_{i=1}^n)$, a **local explainer** is a model $\psi : \mathbb{R}^d \to \mathbb{R}^d$ that returns a d-dimensional vector where the i-th position contains a value with the importance of the i-th feature to a particular observation $\mathbf{x} \in \mathbb{R}^d$. Similarly, a **global explainer** is a model $\phi : \mathbb{R}^{n \times d} \to \mathbb{R}^d$ that, given a data set $\mathcal{X}$ it returns a d-dimensional vector with the aggregated importance of each feature.

Fig. 2 shows a diagram expressing the relationship between the black-box model, the explanatory method, and their respective outputs. The explanatory model is generated using the training data and the black-box model.

The remaining of this section will present popular explanatory methods for feature importance.

4.1 Permutation Importance

The Permutation Importance method was first described by Breiman when introducing the Random Forests algorithm [67]. This method measures how much the coefficient of determination (R^2) of the model reduces when we permute the values of a feature on the training data. This permutation is repeated several times to measure the expected variation of R^2. The main idea is that the random permutation simulates the removal of this feature and how much that impacts the prediction model.

Definition 2 (Permutation Importance – Global Importance Explanation) Let s be the R^2 score of the prediction model $\widehat{f}$. The importance of a feature j is calculated as:

$$\phi_j(\widehat{f}, \mathcal{X}) = s - \frac{1}{K} \sum_{k=1}^{K} \mathrm{R}^2(\widehat{f}(\tilde{\mathcal{X}}_{k,j}), \mathbf{Y}), \tag{5}$$

where the R^2 of $\widehat{f}$ is evaluated over $\tilde{\mathcal{X}}_{k,j}$, a copy of the matrix of problem features for all observations $\mathcal{X}$ with the j-th column randomly shuffled, and K is the number of iterations where the matrix $\tilde{\mathcal{X}}_{k,j}$ was recreated.

4.1.1 Local Interpretable Model-agnostic Explanations (LIME)

LIME [68] provides a local explanation for a single observation by fitting a linear model using random samples generated within the neighborhood of the observation. The linear model is used to obtain the feature importance. A mathematical formulation of LIME was presented in [12]. This method approximates the black-box predictions using an interpretable model and only supports local explanations in the original version.

Definition 3 (LIME – Local Importance Explanation) Given an observation $\mathbf{x}$ and a predictor $\widehat{f}$, LIME generates a neighborhood with normal distribution $N_{\mathbf{x}}$ and fits a local linear model for a number of r sampled observations $\mathcal{D}' = \{(\mathbf{x}'_i, \widehat{f}(\mathbf{x}')_i)\}_{i=1}^{r}$. The feature importances are calculated by minimizing the objective function:

$$\psi(\widehat{f}, \mathbf{x}) = \arg\min_{g \in G} L(\widehat{f}, g, \pi_{N_{\mathbf{x}}}) + \Omega(g), \tag{6}$$

where $L(\widehat{f}, g, \pi_{N_{\mathbf{x}}})$ is a loss function that evaluates how good $\widehat{f}$ is locally approximated by g in the local neighborhood of $\mathbf{x}$, $\pi_{N_{\mathbf{x}}}$ being a kernel that weights the neighborhood with its distance from the original point, G is the set of interpretable linear models, and Ω is a penalty function measuring the model complexity. The local explanation $\overrightarrow{\psi}$ is the vector of coefficients from the local linear model ξ.

4.1.2 Explain by Local Approximation (ELA)

ELA [69] is a local explainer based on LIME and was first introduced using a symbolic regression as the regression model. This method also uses a linear model to obtain local explanations for a regressor but, different from LIME, it does not create new samples with a normal distribution around the given observation. Instead, it uses the k-nearest neighbors to create the model. While this has the advantage of using real-world data instead of artificially created points, it assumes that we have representative training data. The authors also propose a global explanatory method in the original paper, but it is a visual explanation; thus, it is not considered in this work.

Definition 4 (ELA – Local Importance Explanation) Given an observation $\mathbf{x}$ and a predictor $\widehat{f}$, ELA generates a subset $\mathcal{N}_k$ of the available data $\mathcal{X}$ by collecting the k-nearest neighbors of $\mathbf{x}$ using the euclidean distance to fit a local linear model. If the model has an intrinsic feature selection property, the euclidean distance is calculated only with the features selected by the regression model. As in LIME, the coefficients of this linear model represent the importance of each feature.

4.1.3 SHapley Additive exPlanations (SHAP)

SHAP [12] is an algorithm that estimates the *Shapley* values as a feature importance measure, based on coalition game theory, where players can contribute positively or negatively to a result of a game. To find the total contribution of each player, and since players can interact between themselves, the Shapley values represent the overall contribution that the player presents, based on all possible coalition of teams. The Shapley value indicates how much a feature contributes to the current prediction w.r.t. the average prediction.

Definition 5 (SHAP – Local Importance Explanation) Let F be the set of all features, and $S \subseteq F$ a subset of the features, and $\mathbf{x}$ an observation of interest. For each feature j, the difference in prediction with and without this feature is evaluated. This is computed for all possible subsets $S \subseteq F \setminus \{j\}$. Finally, the weighted mean between all possible coalitions is computed, returning the Shapley Value interpreted as the feature importance:

$$\psi_j(\widehat{f}, \mathbf{x}) = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [\widehat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) - \widehat{f}_S(\mathbf{x}_S)]. \qquad (7)$$

In the original SHAP framework implementation, the authors calculate the global explanation as the mean of absolute local explanations for all observations in the training data.

Definition 6 (SHAP – Global Importance Explanation) Let $\mathcal{X}$ be the training data of the predictor. The global importance for a feature j is the mean of absolute local importance for all observations in D:

$$\phi_j(\widehat{f}, \mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} |\psi_j(\widehat{f}, \mathcal{X}_i)|. \tag{8}$$

4.1.4 Shapley Additive Global importancE (SAGE)

In [70] the authors proposed the SAGE explainer, an extension to SHAP for a global explanation. SAGE values measure the uncertainty reduction when incorporating a certain X_j in different subsets of features $\mathbf{X}_S$ to the model, and are interpreted as follows:

- The total sum of the values results in the predictive power of the model;
- An feature j will have zero effect if X_j is conditionally independent of $\widehat{f}(\mathbf{X})$ given all possible subsets of features in $\mathbf{X}$;
- The SAGE values represent the weighted average of uncertainty reduction for the predictions when a feature X_j is considered in the model.

Definition 7 (SAGE global explanation) Let a regressor be $\widehat{f} : \mathbb{R}^d \to \mathbb{R}$ trained with a data set $\mathcal{D} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^{n}$, and $\mathcal{X}$ the matrix of the d problem features for all the n observations. Also, let F be the set of all features in the problem, and $S \subseteq F$ a subset of the features. The SAGE value is calculated the same way as the *Shapley* values but using the mutual information I of predictions instead of the prediction model:

$$\phi_j(\widehat{f}, \mathcal{X}) = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [v_f(S \cup \{j\}, \mathcal{X}) - v_f(S, \mathcal{X})], \tag{9}$$

where v_f is a function that evaluates the predictive power of the subset of features S by comparing the average prediction (absence of all features) with the prediction considering the subset S:

$$v_f(S, \mathcal{X}) = \underbrace{\mathbb{E}\left[I(f_\emptyset(\mathcal{X}_{*,\emptyset}), \mathbf{Y})\right]}_{\text{Average prediction}} - \underbrace{\mathbb{E}\left[I(f_S(\mathcal{X}_{*,S}), \mathbf{Y})\right]}_{\text{Prediction considering } S}, \tag{10}$$

where I is the mutual information of a model f_S using the features in S and the target values of the training set.

Both SHAP and SAGE perform expensive computations to estimate the original SHAP value, which is hard to compute thus requires approximations to alleviate the computational burden since the number of permutations grows exponentially with the dimensionality and are estimated by various methods.

4.1.5 Morris Sensitivity

The Morris Sensitivity [71] calculates the importance of each feature based on their *elementary effect* (EE) – a measurement of how much the output changes with a slight change in the feature value. The values globally represent how sensitive the method is to variations in each feature. This explainer is implemented in InterpretML framework [72].

Definition 8 (Morris Sensitivity – Global Feature Explanation) Let the *Elementary effect* be calculated as:

$$\text{EE}_j = \frac{\widehat{f}(X_1, X_2, \ldots, X_j + \Delta, \ldots, X_n) - \widehat{f}(X_1, X_2, \ldots, X_n)}{\Delta}, \tag{11}$$

where Δ is the step size, picked in a way that $X_j + \Delta$ is still within the features domain.

The Morris Sensitivity global importance performs k random paths through the parameter grid, from different starting points in the data set, modifying the features one at a time, computing the mean EE from all trajectories of the feature of interest:

$$\phi_j(\widehat{f}, \mathcal{X}) = \frac{1}{k_j} \sum_{r=1}^{k_j} \text{EE}_k^r, \tag{12}$$

where k_j is the number of steps where X_j has changed.

4.1.6 Integrated Gradients

Integrated Gradients [73] is a local explainer created to calculate an attribution mask for image predictions made by Deep Learning models. The importance of each feature is calculated using the Aumann-Shapley [74] value that extends the Shapley value for infinitely many players. This method also requires that the user specifies a reference point $\mathbf{x}'$ instead of assuming the average prediction. This has the advantage of generating a better explanation for continuous features and using a realistic reference point. On the other hand, it requires calculating the gradient of the model w.r.t. its features.

Definition 9 (Integrated Gradients – Local Feature Explanation) The Integrated Gradient of the j-th feature for a given observation $\mathbf{x}$ is:

$$\psi_j(\widehat{f}, \mathbf{x}) = (x_j - x_j') \int_{\alpha=0}^{1} \frac{\partial \widehat{f}(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial X_i} d\alpha \tag{13}$$

where $\mathbf{x}'$ is the baseline point, and α represents a continuous path between the point being explained and the reference.

A Riemman trapezoidal sum approximates the integral, and the derivatives can be obtained numerically by a finite difference method. The values represent the gradients along the path for the predictor to go from the mean prediction (when $\mathbf{x}' = \overline{\mathbf{X}}$ is the mean vector for each feature) to the final prediction.

4.1.7 Partial Effects

The econometric and social science fields commonly use the concept of partial derivatives to investigate the behavior of the independent features and their interactions. There are many works that proposes its usage as an interpretability framework in the context of regression analysis, such as [75–79], to name a few.

The Partial Effect (PE)[2] measures how an infinitesimal change (or a discrete change of one unit, for discrete features) in an independent feature affects the dependent feature when its co-features are fixed on specific values [78], expressing the magnitude of the associations between an independent and a dependent feature, having an intuitive interpretation of its value [81].

Although not a new technique, the Partial Effect could be more explored in symbolic regression as a model-specific explanatory method since it is possible to automatically differentiate symbolic regression models, assuming all the functions from the primitive set are differentiable. In [14] the authors proposed the usage of Partial Effects as a model-specific explainer.

Definition 10 (Partial Effects – Local Feature Explanation) The local Partial Effect importance for a given observation $\mathbf{x}$ is given by the gradient of the model evaluated at $\mathbf{x}$:

$$\psi_j(\widehat{f}, \mathbf{x}) = \frac{\partial}{\partial X_j} \widehat{f}(\mathbf{x}), \tag{14}$$

where $\widehat{f}$ is a regression analysis model.

Similar to SHAP, we can calculate the Global explanations as the average of the absolute importance. The global importance of a feature X_s is made by marginalizing its co-features to a representative value and evaluating the mean of all partial derivatives with values that X_s can assume in the given data.

Definition 11 (Partial Effects Global Feature Explanation) The global explanation for a feature j is obtained by marginalizing all co-features and taking the mean absolute value of the local partial effects:

$$\phi_j(\widehat{f}, \mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\partial}{\partial X_j} \widehat{f}(\mathcal{X}_{i,j}, \mathcal{X}_{i,C}) \right|, \tag{15}$$

where $\mathcal{X}_{i,C}$ is the co-features marginalized.

5 Measuring explanations quality

---

[2]Also called *Marginal Effects* in the literature, but this term can be misleading, as mentioned in [80].

Measuring the quality of the explanations is an open question [42, 44, 50] as the definition of interpretability lacks mathematical rigor [27]. One possibility is to include humans in the loop to measure their perception of the quality of the explanations. Nevertheless, the results can be prone to biases due to the experimental design, leading the experimenters to consider multiple additional factors. Also, working with humans can limit the scalability of the framework, compromising the number of results needed for large-scale performance validation. Including humans in the loop of development and deployment of ML systems also requires proper ways of inspecting and analyzing the models, thus needing better benchmarks to validate the proposed explanatory methods.

In the literature, there are few measures proposed to evaluate the quality of the explanations focused on their robustness [82–85]. These measures are based on the principle that a high-quality explanation should not have noticeable variation caused by subtle perturbations in the explained point.

Another possibility explored in this work is to measure the quality of the explanation given an imperfect regression model. For this purpose, we can apply the explainers to a ground-truth model and compare how much the explanations differ when applied to different regression models. The following subsections will explain each of the measures considered in this work.

5.1 Robustness of explanations

Robustness measures evaluate explainers in how subtle perturbations in the observations affect the explanations and are suited to evaluate local explainers only. It is based on the *desiderata* that a good-quality explanation should not vary with a small perturbation on the explained observation.

5.1.1 Stability

The stability measure [83] is the degree to which the local explanation changes for a given point compared to its neighbors. A significant value implies that when the feature being explained changes in a small proportion, the feature importance responds in a more significant proportion, indicating that the explainer is not reliable since it is not locally stable.

Definition 12 (Stability measure) The stability of an explanatory model for an observation $\mathbf{x}$ is calculated by:

$$\mathrm{S}(\widehat{f}, \psi, \mathbf{x}) = \mathbb{E}_{\mathbf{x}' \sim N_{\mathbf{x}}} \left[||\psi(\widehat{f}, \mathbf{x}) - \psi(\widehat{f}, \mathbf{x}')||_2^2 \right], \tag{16}$$

with $N_{\mathbf{x}}$ being the neighborhood of $\mathbf{x}$.

The stability evaluates the mean distance between the explanation for the original observation and for the explanation of all sampled neighbors. In [82] the authors uses a normal distribution to generate the neighborhood $N_{\mathbf{x}} = \mathcal{N}(\mathbf{x}, \sigma)$ with a previously defined value for σ.

The problem when using a normal distribution is that the neighborhood can have a different distribution from the training data, especially in a high-dimensional case. As such, we propose two modifications for this measure:

1. The neighborhood should be generated using a multivariate normal distribution $\mathcal{N}_d(\mu, \boldsymbol{\Sigma})$. This should make the neighborhood more representative to the training data. This requires a variance-covariance matrix Σ, estimated using the training data.
2. The neighborhood spread should use the mean vector of each feature, multiplied by a neighborhood-range parameter λ as the value of σ since a fixed value is too conservative.

To summarize, we propose the neighborhood to be calculated by:

$$N_{\mathbf{x}} = \mathcal{N}_d(\mathbf{x}, \lambda \cdot \text{cov}(\mathcal{X})), \tag{17}$$

where $\mathcal{X}$ is the training matrix where each line is an observation and each column is a feature, $\mathbf{x} \in \mathbb{R}^d$ is the observation of interest, and $\text{cov}(\mathcal{X})$ is the variance-covariance matrix evaluated over the training data. The parameter λ controls the size of the neighborhood and can be found experimentally, set to a fixed value, or calculated over the training data.

5.1.2 (in)fidelity

Taking the subset of most relevant features for an observation $\mathbf{x}$, it is expected that the explanation will attribute high values for those features rather than the others when subtle changes occur in $\mathbf{x}$. The idea of infidelity [84] is to measure the difference between two terms:

i. The dot product between a significant perturbation $\mathbf{p} \in \mathbb{R}^d$ to a given observation $\mathbf{x}$ we are trying to explain and its corresponding explanation, and
ii. The difference in the prediction between the perturbed and original observations.

Definition 13 (Infidelity measure)

$$\text{INFD}(\widehat{f}, \psi, \mathbf{x}) = \mathbb{E}_{\mathbf{p} \in \mu_{\mathbf{p}}} \left[\left(\underbrace{\mathbf{p}^T \psi(\widehat{f}, \mathbf{x})}_{i} - \underbrace{(\widehat{f}(\mathbf{x}) - \widehat{f}(\mathbf{x} - \mathbf{p}))}_{ii} \right)^2 \right] \tag{18}$$

where ψ is an importance attribution explainer, $\widehat{f}$ is a black-box model, and $\mathbf{p}$ is a random feature representing perturbations around the point of interest $\mathbf{x}$ generated by a probability measure $\mu_{\mathbf{p}}$.

A good explanation should be robust to changes in the prediction in response to significant perturbations. This measure is helpful to evaluate if the explanation is robust to miss-specifications or noise in the given observation.

A significant value of infidelity means that the explainer changes more abruptly when small changes in the observation occur, resulting in very different feature importance vector, and thus is not reliable. In this paper, we sample $\mathbf{p}$ with $\mu_{\mathbf{p}} \sim \mathbf{x} - \mathcal{N}_d(\mathbf{x}, \lambda \cdot \mathrm{cov}(\mathcal{X}))$.

5.1.3 Jaccard Stability

The Jaccard Index was used in [85] to measure the stability of their proposed explainer called S-LIME. The Jaccard Index is often used as a similarity metric between two sets, and it is defined as the size of the intersection between the sets by the size of the union of these sets. If they are both equal, the Jaccard Index will have a value of 1. To use this metric for stability, we verify whether the set of the k most important features, measured by the magnitude of their importance, changes within a neighborhood.

Definition 14 (Jaccard Stability) Let $\mathbf{x}$ be a observation being explained, and let $\tilde{\psi}(\mathbf{x})_k$ denote the subset of k most important features for $\mathbf{x}$. The Jaccard Stability measure is calculated by:

$$\mathrm{J}(\widehat{f}, \psi, \mathbf{x}) = \mathbb{E}_{\mathbf{x}' \sim N_{\mathbf{x}}} \left[\frac{|\tilde{\psi}(\mathbf{x})_k \cap \tilde{\psi}(\mathbf{x}')_k|}{|\tilde{\psi}(\mathbf{x})_k \cup \tilde{\psi}(\mathbf{x}')_k|} \right], \tag{19}$$

with $N_{\mathbf{x}}$ being the neighborhood of $\mathbf{x}$, and k being the size of the subset.

5.2 Quality of explanations with imperfect predictions

Different explanatory methods have different interpretations of the importance values. For example, a Shapley value ψ_i of a feature means how much it contributed to the difference in prediction from the average prediction. On the other hand, for Partial Effects, it means how the prediction will change if we make a slight variation to the input value. We cannot compare two explanatory methods directly since there is no guarantee of correspondence between explanations.

Nevertheless, since explanations are generated from regression models that are usually an imperfect approximation of the true model, we can measure how much information is lost when trying to explain these imperfect models. For this purpose, we assume the existence of a ground-truth explanation that we can use to compare the *i)* error in direction (whether the importance is negative or positive) and the *ii)* magnitude of the explanation extracted from the regression models; given that we have access to the ground-truth function that generated the data set.

When used to measure global quality, only one explanation vector is compared. When measuring local quality, each observation in the test data is evaluated with the quality measureand the average quality is reported.

5.2.1 Cosine Similarity

The cosine similarity, first used in [51] to measure the quality of an explanation, returns a value between -1 and 1 representing if the explanations are in opposed or same directions. A cosine similarity of 0 means that the explanations are orthogonal.

Definition 15 (Cosine Similarity between two explanations) Let $\overrightarrow{\phi}$ be a vector of feature importance generated by a global explanatory method ϕ (or, similarly, let $\overrightarrow{\psi}$ be a vector of local importance for an observation $\mathbf{x}$ generated by a local explanatory method) for a regression model $\widehat{f}$, and let $\overrightarrow{e}$ be the correct explanation generated from applying one of the explanation methods to the ground-truth model. The cosine similarity is calculated by:

$$\cos(\overrightarrow{e}, \overrightarrow{\psi}) = \frac{\overrightarrow{e} \cdot \overrightarrow{\psi}}{\|\overrightarrow{e}\|\|\overrightarrow{\psi}\|}. \tag{20}$$

5.3 Normalized Mean Squared Error

The Normalized Mean Squared Error measures the squared difference between the ground-truth explanation and the evaluated explanation normalized by the true explanation variance. This captures whether the explanations differ not only in direction but also in magnitude.

Definition 16 (Normalized Mean Squared Error) Let $\overrightarrow{\phi}$ be a feature importance vector generated by a global explanatory method ϕ (or, similarly, let $\overrightarrow{\psi}$ be a vector of local importance for an observation $\mathbf{x}$ generated by a local explanatory method) for a regression model $\widehat{f} : \mathbb{R}^d \to \mathbb{R}$, and let $\overrightarrow{e}$ be an expected explanation (ground-truth). The Normalized Mean Squared Error (NMSE) is the MSE divided by the variance of $\overrightarrow{e}$:

$$\text{NMSE}(\overrightarrow{e}, \overrightarrow{\phi}) = \frac{\sum_{i=1}^{d}(e_i - \phi_i)^2}{\sum_{i=1}^{d}(e_i - \bar{e})^2}, \tag{21}$$

where $\bar{e}$ is the average of the feature importance values in $\overrightarrow{e}$.

6 Methodology

The experiments will follow the methodology illustrated in Fig. 3. In short, we will follow three steps.

First, we use the generating functions to create: *i)* a predictor using the original equation as the prediction function, *ii)* a training data set with 1000 observations, and *iii)* a test data set with 30 observations generated using the Latin Hypercube Sampling [86] method based on the training data. Both

Fig. 3: Scheme of the whole benchmark process to generate the results. The Feynman equations are used to generate the train and test data. Then the regression methods are fine-adjusted through a gridsearch process to finally be trained and used as input to feature importance methods.

train and test data do not contain any noise to prevent adding artifacts to the results.

The second step uses the training data to tune each regressor with a grid-search using 3-fold cross-validation for each data set. The grid-search optimizes the configuration that maximizes the R^2 score.

Finally, the third step executes the benchmark experiments using the optimal configurations obtained for each regression method. The models are generated using the training data and explained using different explanatory methods. Whenever the regression algorithm is stochastic, we repeat this procedure 30 times for every data set.

We will analyze the results by measuring the prediction error of each regression model, followed by the quality measures of the explainer models. For local explanations, we will use the robustness measure and the quality measure and, for global explanations, we will only report the quality measure as the robustness measures require a local neighborhood. We will report heatmaps of the results for different combinations of regression and explanatory models.

The neighborhood-range factor used for the local explanations was 0.001, and the number of top features for the Jaccard Stability was $k = 1$.

6.1 Data sets

For the overall results, we use the Feynman benchmark, introduced in [56], containing 100 equations from mechanical, electromagnetism, and quantum physics, with different degrees of complexity. The reason for choosing this benchmark is that it contains problems that describe actual phenomena, where

Table 1: Description of the additional 4 GP benchmark problems. $U(l_b, u_b, n_s)$ denotes a uniform distribution with bounds $[l_b, u_b]$ and n_s observations, and $E(start, stop, step)$ is an evenly spaced grid in the interval $[start, stop]$ and step size given by *step*. We adapted the number of generated observations in relation to the original publication to make the results and comparisons compatible with the Feynman benchmark.

Name	Formula	Features	Train set/Test set
Korns-11	$6.87 + 11\cos(7.23x^3)$	$\{x, y, z, v, w\}$	$U(-50, 10, 1000)$ $U(-50, 10, 100)$
Korns-12	$2 - 2.1\cos(9.8x)\sin(1.3w)$	$\{x, y, z, v, w\}$	$U(-50, 10, 1000)$ $U(-50, 10, 100)$
Vladislavleva-4	$\frac{10}{5+\sum_{i=1}^{5}(x_i-3)^2}$	$\{x_i\}_{i=1}^5$	$U(0.05, 6.0, 1000)$ $U(-0.5, 10, 100)$
Pagie-1	$\frac{1}{1+x^{-4}} + \frac{1}{1+y^{-4}}$	$\{x, y\}$	$E(-5, 5, 0.01)$ $E(-5, 5, 0.1)$

Fig. 4: Joint distribution of the number of features and the number of nodes required to build the expression tree for the iirsBenchmark data sets. The lighter portion of the histograms represents the GP benchmark.

the problem features have physical or mathematical meaning. The original data contained $1,000,000$ observations, from which we randomly picked $1,000$ data points. The test set was generated using the Latin Hypercube.

For a more detailed analysis, we selected four benchmark problems commonly used in GP literature described in Table 1.

Fig. 4 shows the distribution of the number of features for these 104 data sets.

Table 2: Regression methods available on *iirsBenchmark* and used in the experiments.

Regressor	Interpretability	Method type
XGB	black-box	Tree boosting
RF	black-box	Tree bagging
MLP	black-box	Feed-forward Neural network
SVM	black-box	Vector machine
k-NN	gray-box	Instance method
Operon	gray-box	Symbolic regression
ITEA	gray-box	Symbolic regression
Linear regression	white-box	Regression modeling
LASSO regression	white-box	Regression modeling
Single decision tree	white-box	Decision tree

6.2 Regression and explanatory methods

Table 2 shows all the regression algorithms used in this experiment. All of them, except for ITEA[3] and Operon[4], are available via scikit-learn module[5].

The tested configurations for each regressor are reported in Table 3. For symbolic regression methods ITEA and Operon, the functions sets were, respectively, {log, sqrt, id, sin, cos, tanh, exp, expn, arcsin} and $\{+, -, \times, /, \exp, \log, \text{sqrt}, \text{square}, \sin, \cos, \tanh, \text{asin}, \text{constant}, \text{variable}\}$.

Table 4 shows the explanatory methods used in this experiment together with their main characteristics. We also created a *Random Importance* explainer that supports local and global explanations, attributing random ranks of importance for each feature every time it is called. This explainer will serve as a baseline as it is expected to be the worst explainer.

For the robustness measures, we also report the values obtained in the ground-truth model, called *Feynman explainer*. This is the result of applying the explainer models to the ground-truth expression, representing the robustness of the true model.

6.3 Result analysis

The results involving multiple data sets are reported by presenting the median and the Interquartile Range (IQR). While boxplots implicitly report the Interquartile Range (IQR) — being the size of the filled box in the graphs —, tables and heatmaps reports the results as $\tilde{m} \pm \text{IQR}$, where $\tilde{m}$ is the median of the group and IQR is the respective IQR. Those descriptors were chosen due to robustness to outliers while still providing a good summary of the distribution.

We use *Critical Diagrams* [87] to represent the acceptance or rejection of the null hypothesis. The critical diagrams sorts the compared groups by

---

[3]https://github.com/gAldeia/itea-python.
[4]https://github.com/heal-research/operon.
[5]https://github.com/scikit-learn/scikit-learn.

Table 3: Regressors considered in the experiments, with their hyper-parameters to be adjusted through the gridsearch procedure. The Linear regressor is the only that had no adjustment.

Regressor	Hyper-parameter	Gridsearch values
XGB	n_estimators	$[100, 200, 300]$
	min_samples_split	$[0.01, 0.05, 0.1]$
RF	n_estimators	$[100, 200, 300]$
	min_samples_split	$[0.01, 0.05, 0.1]$
MLP	hidden_layer_sizes	$[(50,), (50, 100,), (100,), (100, 100,)]$
	activation	$['identity', 'logistic', 'tanh', 'relu']$
SVM	kernel	$['linear', 'rbf', 'poly']$
	degree	$[1, 2, 3, 4]$
k-NN	n_neighbors	$[3, 5, 7, 9, 11, 17, 19, 23, 29, 31]$
Operon	population_size	$[100, 250, 500]$
	generations	$[100, 250, 500]$
ITEA	popsize	$[100, 250, 500]$
	gens	$[100, 250, 500]$
LASSO	alpha	$[0.001, 0.01, 0.1, 1, 10]$
Decision Tree	max_depth	$[5, 10, 15]$
	max_leaf_nodes	$[5, 10, 15]$

Default values of hyper-parameters were omitted. All the regressors have their respective python module, and omitted parameters default values can be found on each regressor documentation

Table 4: Explanation methods available in *iirsBenchmark* and used in the experiments.

Explainer	Agnostic	Local	Global	Method type
Permutation Importance	**Y**	N	**Y**	feature removal
SHAP	**Y**	**Y**	**Y**	feature removal
SAGE	**Y**	N	**Y**	feature removal
LIME	**Y**	**Y**	N	Approximate by linear model
ELA	N	**Y**	N	Approximate by linear model
Morris sensitivity	**Y**	N	**Y**	Sensitivity analysis
Integrated Gradients	**Y**	**Y**	N	Gradient analysis
Partial Effects (PE)	N	**Y**	**Y**	Gradient analysis
Random Importance	**Y**	**Y**	**Y**	Random attribution

their average ranks, and two or more methods are connected by a horizontal line if there is no statistical significance between them. The statistical test used was the Wilcoxon signed-rank, with the *p*-values corrected by the Holm-Bonferroni method, recommended when testing only one null hypothesis that all groups have no significant differences between themselves [88]. The correction method mitigates the alpha inflation when multiple comparisons are made. The statistical significance threshold considered is $p < 0.05$, evaluated after correction.

Finally, since the computational cost of the explanation may be relevant to some applications, we also report the runtime of each explainer for both local and global explanations.

6.4 *iirsBenchmark*

All regressors and explanatory methods included in the experiments were wrapped up in a package module with a syntax similar to the Python ML library scikit-learn in the *iirsBenchmark*. The python module also provides a script to execute the experiments and all post-processing scripts.

All regression methods were previously implemented either by the scikit-learn module or by their corresponding authors following scikit guidelines, all of which have a structured unification. The *iirsBenchmark* only extends those regressors by adding more class methods and static attributes useful in the interpretability context. The explainers being evaluated lack a common interface. In spite of that, we unified their interface by implementing standardized classes in *iirsBenchmark*.

7 Experimental Results

In the first part of this section, we report and analyze the results obtained with the Feynman benchmark. We present the aggregated results of all combinations of regression models and explanation models. The following subsection shows the results obtained on the four select challenging benchmarks from SR literature, in which we will make a more detailed analysis of the obtained results.

7.1 Feynman benchmark

In this subsection, we will have a broader view of the results for a benchmark set that does not present a challenging scenario for the non-linear regression models. We expect to observe explanations close to what we would obtain using the generating function.

7.1.1 Model accuracy

Fig. 5 shows the boxplot of the MAE and NMSE of each method considering the entirety of the Feynman benchmark, with the critical difference diagram of the average rank for each method. As we can see from these plots, ITEA and Operon found the best performing regression models in the benchmark. The critical difference diagrams show a significant difference between ITEA and Operon, even though both obtained models are close to the maximum observed accuracy. The three models considered interpretable in the literature

(a) MAE on test   (b) NMSE on test

(c) Critical Diagram for MAE on test   (d) Critical Diagram for NMSE on test

Fig. 5: MAE (smaller is better) and NMSE (smaller is better) boxplots for all regressors, vertically ordered from the best to worst median values. The Critical Diagram below each plot indicates the absence of statistical significance between groups connected by a horizontal bar.

(a) Expression sizes   (b) Critical Diagram for expression sizes

Fig. 6: Expression sizes as number of nodes (smaller is better) for regression models. The regression methods are vertically ordered from the best to worst.

were the worst-performing models in this benchmark, corroborating with the idea of the trade-off between interpretability and accuracy.

Another relevant aspect of the interpretability of the model is the expression size. Larger expressions are harder to read and understand. Fig. 6 shows the boxplot of the distribution of expression sizes obtained by Lasso, Linear, Operon, and ITEA. Lasso and Linear obtained the smaller expressions since their size is bounded by the number of features, followed by Operon with a median size of 50. ITEA median size was 100 with an upper quartile much higher than Operon. This means that, despite the difference in accuracy between ITEA and Operon, Operon managed to find smaller expressions than ITEA, possibly increasing its readability by human experts.

(a) Hit rate percentage on Feynman data sets

(b) Venn diagram of hit rate percentage for the two symbolic regression methods

Fig. 7: Hit rate on Feynman data sets calculated for regression methods that return mathematical expressions.

Complementing the analysis of the expression size, Fig. 7a shows a bar plot of the hit rate of each of these methods. Given the original expression f and the SR model $\widehat{f}$, we perform a symbolic simplification of the expression $\widehat{f} - f$, considering a hit if the result of the simplification is zero. The hit rate is the percentage of expressions obtained by each model that corresponds precisely to the ground-truth generating function.

As expected, ITEA obtained a higher hit rate than Operon since almost half of this benchmark is representable by the constrained representation of ITEA, having a search space that tends to favor this SR method. Fig. 7b shows a Venn diagram of the intersection of the data sets that both methods obtained the generating function. Only 7% were mutually found by both methods, meaning that by merging the results of both metho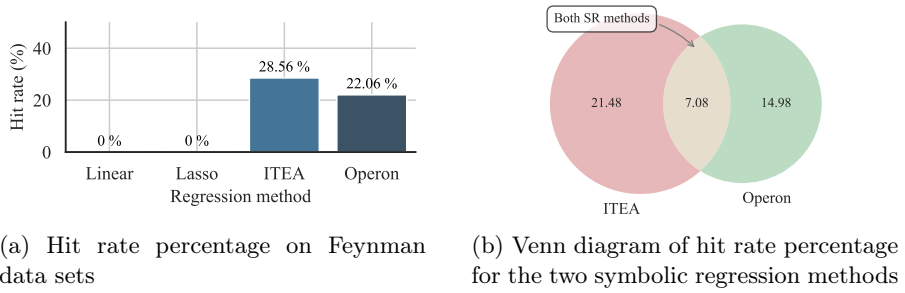ds, SR obtained a hit rate of about 42% on this benchmark. These results highlight the possibility of using SR as an innate interpretable model without using a *post-hoc* explanatory method.

7.1.2 Explanation computational cost

Table 5 shows the median and IQR of the execution time for each explainer when combined with different models. Regarding the global methods, Partial Effect, Permutation Importance, and Morris Sensitivity have a significantly smaller computation run-time than SHAP and SAGE. This is expected since the former methods only require the evaluation of a well-defined function to every point in the data set, while the latter methods require the sampling of different subsets of features. We should notice that it is possible to reduce the run-time of both SHAP and SAGE with the cost of reducing the explanation quality.

For the local explanations, Partial Effects are the fastest method since it becomes just a function applied to a given instance after calculating the symbolic derivative. The other methods are comparable regarding run-time but with some combinations that stand out: SHAP and IG with Random Forest and IG with ITEA require a significantly longer time when compared to other regressors using the same explanation method. These methods require a

Table 5: Median and IQR of execution time (in seconds) each explainer took to generate the explanations.

Global Explanation	Partial Effects	SHAP	SAGE	Permutation Importance	Morris Sensitivity
KNN	—	21.07 ± 19.71	1225.18 ± 5217.46	0.70 ± 0.82	0.30 ± 0.31
Linear	0.00 ± 0.00	14.47 ± 14.91	27.21 ± 110.25	0.06 ± 0.06	0.36 ± 0.38
Lasso	0.00 ± 0.00	14.05 ± 11.00	21.45 ± 107.16	0.06 ± 0.06	0.31 ± 0.29
Decision Tree	—	14.30 ± 14.07	38.83 ± 100.23	0.06 ± 0.08	0.30 ± 0.31
RF	—	111.42 ± 94.86	3116.55 ± 9144.32	6.94 ± 6.23	0.42 ± 0.34
MLP	—	40.21 ± 19.14	549.54 ± 3613.16	1.87 ± 1.59	0.44 ± 0.33
SVM	—	18.74 ± 26.52	2077.81 ± 10092.26	1.28 ± 3.55	0.33 ± 0.32
XGB	—	18.14 ± 19.20	824.13 ± 3644.58	0.78 ± 0.70	0.29 ± 0.28
Operon	0.66 ± 0.46	12.37 ± 9.77	25.57 ± 159.81	0.06 ± 0.04	0.21 ± 0.09
ITEA	0.14 ± 0.13	25.41 ± 22.89	486.43 ± 1869.14	1.55 ± 1.01	0.41 ± 0.28

Local Explanation	Partial Effects	SHAP	Integrated Gradients	LIME	ELA
KNN	—	2.12 ± 1.87	4.53 ± 3.82	3.73 ± 2.99	—
Linear	0.00 ± 0.00	1.37 ± 1.52	0.92 ± 0.69	4.25 ± 3.47	1.83 ± 1.68
Lasso	0.00 ± 0.00	1.40 ± 1.16	0.86 ± 0.73	3.65 ± 2.75	1.75 ± 1.49
Decision Tree	—	1.26 ± 1.21	0.94 ± 0.84	3.70 ± 3.19	—
RF	—	11.41 ± 10.15	167.02 ± 154.94	6.93 ± 6.95	—
MLP	—	3.98 ± 1.71	11.41 ± 8.51	6.52 ± 4.76	—
SVM	—	1.98 ± 2.91	1.02 ± 0.73	3.30 ± 2.63	—
XGB	—	1.70 ± 1.86	2.36 ± 2.24	3.76 ± 3.69	—
ITEA	0.13 ± 0.14	2.71 ± 2.24	17.67 ± 16.99	4.02 ± 2.68	2.16 ± 1.71
Operon	0.06 ± 0.04	1.09 ± 0.89	0.75 ± 0.48	2.93 ± 1.48	1.25 ± 0.49

higher number of evaluations to generate the feature importance, and so they are more affected by prediction model size. Specifically for SR methods, we used the model agnostic version of IG that approximate derivatives to a fair performance comparison.

7.1.3 Local Explanations

Fig. 8 show the heatmap plot of different robustness measures for the combinations of regressors and explanation models for the local explanations. The Jaccard Stability (Fig. 8a) is a very sensitive measure since we are calculating based only on the top feature, so either the explanation gets everything right or everything wrong in a single explanation. We can see that, despite this difficulty, Partial Effects, ELA and SHAP obtained the highest score for every prediction model. This means that the top feature does not change around a small neighborhood. Integrated Gradient presented stability issues under this measure for the ensemble methods and KNN. LIME also presented worse results for every regressor but still higher than the Random Importance.

Regarding Infidelity (Fig. 8b), ELA was the only explanation method that obtained worse results than Random Importance. This means that ELA importance rank is sensitive to a slight change in the value of the feature. This result is due to the dependency on having appropriate neighborhood points in the data set, and this impact should be reduced when using larger data sets.

Finally, the Integrated Gradient obtained worse results than Random Importance under the Stability measure (Fig. 8c) for the ensemble methods and kNN. We notice from these results that the SR methods obtained very close results to the reference ground-truth (Feynman), meaning that they are as robust as the true expression.

Fig. 9 shows the heatmap plots for the Cosine similarity and the NMSE of the local explanations calculated using the ground-truth as a reference point. Considering the cosine similarity, which measures whether the sign of the feature importance is the same as the ground-truth, we can see that almost every combination obtained a value closer to 1.0, meaning that even an inaccurate model can at least indicate correctly whether a feature has a positive or negative influence to the target feature. The SR methods obtained the best possible results with every explanation method except LIME. When also considering the magnitude of the feature importance, we can see in Fig. 9b the SHAP explainer is capable of returning explanations with smaller errors for black-box models. On the other hand, LIME presented a very similar performance for every regressor. Viewing from the regressor side, the SR methods returned the best explanations for every method, except LIME. These results indicate that not only an accurate model is essential for a correct explanation but also that is must be a flexible model capable of capturing the different properties of the generating function.

7.1.4 Global Explanations

Fig. 10 shows the heatmaps for the global explanations. The cosine similarity for the global models is very similar to the local explanations, and most combinations of regressors and explanation models can indicate the correct sign of the effect of each feature. On the other hand, we observed a higher increase in NMSE when comparing SR models against simpler models like Linear and Lasso. The SR methods outperformed other methods with every explainer. Regarding the explainers, Morris Sensitivity is more sensitive to model accuracy, and every other explanation model maintained a small error except for the linear models.

7.2 Selected benchmark

This subsection reports the obtained results with the selected data sets focused on the SR models. These data sets are known to be challenging to any regression model, and, as such, we expect to observe a disagreement in the explanation of the SR models compared to the ground-truth.

7.2.1 Model accuracy

Table 6 shows the accuracy of ITEA, Operon and MLP models for the selected data sets. We chose MLP as a black-box representative since it

	KNN	Linear	Lasso	Decision Tree	RF	MLP	SVM	XGB	Operon	ITEA	Feynman
Partial Effects		**1.0** ±0.0	**1.0** ±0.0						0.99 ±0.01	0.99 ±0.02	**1.0** ±0.01
ELA		**1.0** ±0.02	**1.0** ±0.01						0.98 ±0.05	**1.0** ±0.01	**1.0** ±0.01
SHAP	0.98 ±0.02	0.99 ±0.02	0.99 ±0.02	0.98 ±0.02	0.98 ±0.02	0.98 ±0.01	0.98 ±0.02	0.97 ±0.03	0.98 ±0.02	0.98 ±0.02	0.98 ±0.02
Integrated Gradients	0.79 ±0.13	0.97 ±0.03	0.97 ±0.03	0.99 ±0.01	0.45 ±0.21	0.96 ±0.04	0.96 ±0.04	0.64 ±0.14	0.96 ±0.03	0.96 ±0.04	0.96 ±0.03
LIME	0.65 ±0.21	0.67 ±0.18	0.69 ±0.15	0.64 ±0.21	0.64 ±0.18	0.64 ±0.2	0.65 ±0.22	0.64 ±0.19	0.65 ±0.2	0.64 ±0.21	0.78 ±0.17
Random Importance	0.32 ±0.11	0.31 ±0.11	0.31 ±0.1	0.33 ±0.09	0.33 ±0.09	0.33 ±0.09	0.31 ±0.1	0.33 ±0.09	0.33 ±0.09	0.33 ±0.09	0.31 ±0.11

(a) Jaccard Stability (greater is better) *heatmap* on the Feynman data sets.
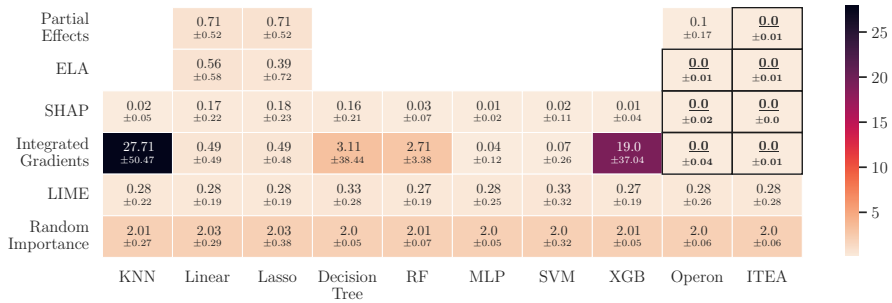
	KNN	Linear	Lasso	Decision Tree	RF	MLP	SVM	XGB	Operon	ITEA	Feynman
Partial Effects		**0.0** ±0.0	**0.0** ±0.0						**0.0** ±0.0	**0.0** ±0.0	**0.0** ±0.0
ELA		3.95 ±4.28	4.44 ±4.14						6.58 ±5.49	4.82 ±3.7	4.87 ±3.46
SHAP	0.04 ±0.48	0.01 ±0.14	0.01 ±0.14	0.09 ±1.54	0.02 ±0.29	0.01 ±0.2	0.01 ±0.19	0.04 ±0.57	0.02 ±0.28	0.02 ±0.24	0.02 ±0.23
Integrated Gradients	0.18 ±2.13	**0.0** ±0.05	**0.0** ±0.05	0.09 ±2.39	0.04 ±0.44	0.01 ±0.12	0.01 ±0.13	0.08 ±1.51	0.01 ±0.16	0.02 ±0.15	0.01 ±0.14
LIME	0.04 ±0.53	0.01 ±0.13	0.01 ±0.13	0.01 ±1.52	0.01 ±0.25	0.01 ±0.2	0.01 ±0.22	0.03 ±0.6	0.01 ±0.29	0.02 ±0.23	0.01 ±0.24
Random Importance	0.07 ±0.44	0.03 ±0.06	0.03 ±0.07	0.15 ±1.65	0.05 ±0.25	0.04 ±0.14	0.04 ±0.13	0.08 ±0.52	0.05 ±0.18	0.05 ±0.16	0.04 ±0.15

(b) Infidelity (smaller is better) *heatmap* on the Feynman data sets.

	KNN	Linear	Lasso	Decision Tree	RF	MLP	SVM	XGB	Operon	ITEA	Feynman
Partial Effects		**0.0** ±0.0	**0.0** ±0.0						0.02 ±0.53	0.02 ±0.32	0.01 ±0.15
ELA		0.29 ±0.26	0.32 ±0.29						0.39 ±0.35	0.32 ±0.17	0.33 ±0.18
SHAP	0.01 ±0.17	**0.0** ±0.06	**0.0** ±0.05	0.06 ±1.21	0.01 ±0.15	**0.0** ±0.09	**0.0** ±0.1	0.03 ±0.39	0.02 ±0.18	0.02 ±0.19	0.01 ±0.11
Integrated Gradients	61.77 ±1096.62	**0.0** ±0.05	**0.0** ±0.05	39.62 ±385.14	6.5 ±234.52	**0.0** ±0.11	**0.0** ±0.09	35.74 ±596.4	0.02 ±0.37	0.01 ±0.24	0.01 ±0.12
LIME	1.32 ±27.82	0.81 ±20.33	0.77 ±23.1	1.16 ±28.73	1.13 ±21.79	1.35 ±31.03	1.14 ±30.78	1.21 ±25.77	1.36 ±31.91	1.59 ±36.05	1.01 ±14.79
Random Importance	4.15 ±6.24	4.15 ±6.3	4.18 ±6.33	4.05 ±6.05	4.05 ±6.04	4.04 ±6.06	4.21 ±6.34	4.05 ±6.08	4.04 ±6.06	4.05 ±6.06	4.11 ±6.4

(c) Stability (smaller is better) *heatmap* on the Feynman data sets.

Fig. 8: Local explanation robustness metrics *heatmaps* showing the median and IQR for each explainer-regressor combination. The best values cells are highlighted with a black edge.

obtained the next best results after SR models. We can see from this table that the SR models are still the most accurate among the tested models, but, except for Pagie-1, all algorithms created inaccurate models, as can be seen from the high values for MAE and NMSE. The former metric is more robust

	KNN	Linear	Lasso	Decision Tree	RF	MLP	SVM	XGB	Operon	ITEA
Partial Effects		0.96 ±0.12	0.96 ±0.12						0.97 ±0.12	**1.0** ±0.0
ELA		0.93 ±0.15	0.96 ±0.14						**1.0** ±0.0	**1.0** ±0.0
SHAP	**1.0** ±0.01	0.99 ±0.05	0.99 ±0.05	0.96 ±0.11	0.99 ±0.02	**1.0** ±0.0	**1.0** ±0.01	**1.0** ±0.01	**1.0** ±0.01	**1.0** ±0.0
Integrated Gradients	0.0 ±0.28	0.99 ±0.03	0.99 ±0.04	0.0 ±0.0	0.81 ±0.22	**1.0** ±0.02	**1.0** ±0.02	0.36 ±0.8	**1.0** ±0.01	**1.0** ±0.0
LIME	0.9 ±0.24	0.93 ±0.2	0.89 ±0.24	0.83 ±0.25	0.85 ±0.21	0.86 ±0.2	0.89 ±0.22	0.85 ±0.2	0.86 ±0.23	0.87 ±0.22
Random Importance	0.87 ±0.14	0.85 ±0.14	0.82 ±0.16	0.85 ±0.04	0.85 ±0.04	0.84 ±0.04	0.8 ±0.04	0.85 ±0.04	0.84 ±0.06	0.85 ±0.04

(a) Median Cosine Similarity (greater is better) *heatmap* between local explanations obtained over the regressor and the original Feynman equation.

	KNN	Linear	Lasso	Decision Tree	RF	MLP	SVM	XGB	Operon	ITEA
Partial Effects		0.71 ±0.52	0.71 ±0.52						0.1 ±0.17	**0.0** ±0.01
ELA		0.56 ±0.58	0.39 ±0.72						**0.0** ±0.01	**0.0** ±0.01
SHAP	0.02 ±0.05	0.17 ±0.22	0.18 ±0.23	0.16 ±0.21	0.03 ±0.07	0.01 ±0.02	0.02 ±0.11	0.01 ±0.04	**0.0** ±0.02	**0.0** ±0.0
Integrated Gradients	27.71 ±50.47	0.49 ±0.49	0.49 ±0.48	3.11 ±38.44	2.71 ±3.38	0.04 ±0.12	0.07 ±0.26	19.0 ±37.04	**0.0** ±0.04	**0.0** ±0.01
LIME	0.28 ±0.22	0.28 ±0.19	0.28 ±0.28	0.33 ±0.28	0.27 ±0.19	0.28 ±0.25	0.33 ±0.32	0.27 ±0.19	0.28 ±0.26	0.28 ±0.28
Random Importance	2.01 ±0.27	2.03 ±0.29	2.03 ±0.38	2.0 ±0.05	2.01 ±0.07	2.0 ±0.05	2.0 ±0.32	2.01 ±0.05	2.0 ±0.06	2.0 ±0.06

(b) Median NMSE (smaller is better) *heatmap* between local explanations obtained over the regressor and the original Feynman equation.

Fig. 9: Local explanation quality metrics *heatmaps* showing the median and standard deviation for each explainer-regression combination. The best values are highlighted with a black border around it, and the color maps always attribute lighter colors to better results.
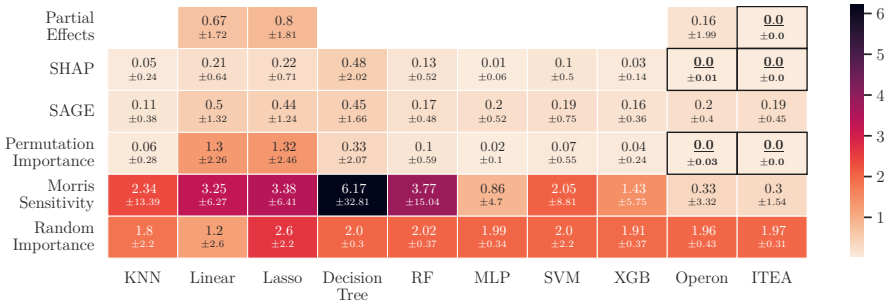
to outliers — as it uses the absolute values—, and the latter tends to alleviate the prediction errors smaller than 1.0, while punishing prediction errors higher than this value — since it squares the errors. The SR methods were outperformed by the best black-box model only using the MAE metric on the Korns-12 data set.

In practice, those models would be discarded as they do not capture the relationship of the observation features with the target value. Notice that this creates a situation where we have an unreliable model but will still extract the feature importance information.

The difference in results between ITEA and Operon can be explained by the expression size as reported in Table 7. We can see that ITEA generated expressions 3 to 4 times larger than Operon. Also, as expected, none of the SR algorithms could find a perfect hit in any run.

(a) Median Cosine Similarity (greater is better) *heatmap* between global explanations obtained over the regressor and the original Feynman equation.

(b) NMSE (smaller is better) *heatmap* between global explanations obtained over the regressor and the original Feynman equation.

Fig. 10: Global explanation quality metrics *heatmaps* showing the median and standard deviation for each explainer-regression combination. The best values are highlighted with a black border around it, and the color maps always attribute lighter colors to better results.

Table 6: Median of MAE and NMSE for the two symbolic regression method and the best performing black-box method for the selected data sets from the literature. The best value for each data within each metric set is highlighted in bold.

	MAE			NMSE		
	ITEA	Operon	MLP	ITEA	Operon	MLP
Korns-11	7.67	**7.26**	7.65	**15.43**	32.76	262.80
Korns-12	0.92	0.88	**0.87**	**17.43**	30.03	30.07
Vladislav.	1.41	**0.19**	0.31	**1.18**	2.97	1.19
Pagie-1	**0.00**	0.01	0.06	**0.00**	**0.00**	0.01

Table 7: Median expression size and hit rate (expressed as a percentage since some regressors are non-deterministic) for methods that return mathematical expressions.

| | Median expression size | | | | # hits | |
	Linear	Lasso	ITEA	Operon	ITEA	Operon
Korns-11	21.0	1.0	181.0	42.0	0%	0%
Korns-12	16.0	1.0	180.0	56.5	0%	0%
Vladislav.	17.0	1.0	110.5	72.0	0%	0%
Pagie-1	1.0	1.0	92.0	24.0	0%	0%

Table 8: Median of the quality measures for the selected **local** explanatory methods with ITEA and Operon.

| ITEA | Cosine similarity | | | NMSE | | |
	IG	PE	SHAP	IG	PE	SHAP
Korns-11	**1.00**	-0.07	-0.48	50.28	$9.00e+14$	**1.64**
Korns-12	**1.00**	0.00	-0.34	$1.34e+7$	$5.61e+17$	**3.87**
Vladslav.	**1.00**	0.87	0.91	0.51^*	0.45^*	$\mathbf{0.14}^*$
Pagie-1	**1.00**	**1.00**	-0.61	0.47^*	$\mathbf{0.00}^*$	0.24^*

| Operon | Cosine similarity | | | NMSE | | |
	IG	PE	SHAP	IG	PE	SHAP
Korns-11	0.25	**0.05**	0.07	**1.25**	**1.25**	1.33
Korns-12	**1.00**	**0.00**	-0.07	126.71	3.20	**2.99**
Vladislav.	**1.00**	0.270	0.93	0.03^*	$\mathbf{0.00}^*$	0.01^*
Pagie-1	**1.00**	0.85	**1.00**	0.98^*	$\mathbf{0.00}^*$	0.06^*

7.2.2 Local Explanations

Table 8 shows a detailed analysis of the performance of each local explainer with ITEA and Operon, focusing on the best performing explanatory methods observed in the heatmap in Fig. 9: Integrated Gradients (IG), Partial Effects (PE) and SHAP. We report the two explanation quality measures cosine similarity and the MSE normalized by the variance of the expected explanation (NMSE). Results marked with an * were not normalized by the original feature importance explanation since every feature has the same importance making the variance zero.

With the bad prediction accuracy for these problems, we notice a degradation in the feature importance values generated by the explanation models. Both ITEA and Operon found inconsistent directions for their feature importance when using PE and SHAP on Korns-11 and Korns-12 data sets regarding the cosine similarity. Using IG, they found that the sign of importances were partially correlated with the ground-truth for Korns-12. With the

Vladislavleva-4 data set, the directions obtained by IG and SHAP were more consistent with the ground-truth, even though this model was also inaccurate. Pagie-1 did not get a perfect score even though they returned perfectly accurate models. This means that, even though the regression models are perfect w.r.t. the validation set, they may not correspond to the exact behavior of the true model.

Partial Effects had a better performance with Operon rather than with ITEA. To investigate this problem, we selected the expressions from ITEA and Operon that had an explanation error close to the median reported in Table 8 for the Korns-11 data set. The selected ITEA model is a sum of cosines given by:

$$
\begin{aligned}
\text{ITExpr}(x, y, z, v, w) = &- 0.308 \cdot \cos\left(\frac{xw^4}{yz^3v^4}\right) + 0.133 \cdot \cos\left(\frac{y^2v^3}{xz^3}\right) \\
&- 0.089 \cdot \cos\left(\frac{z^4v^3w^4}{x^2y^2}\right) + 0.137 \cdot \cos\left(\frac{yz^2w^4}{v}\right) \\
&- 0.121 \cdot \cos\left(\frac{v^4w^2}{x^4z}\right) + 0.175 \cdot \cos\left(\frac{zw^4}{xy^3v^4}\right) \\
&+ 0.104 \cdot \cos\left(\frac{x^3yzw}{v^4}\right) - 0.098 \cdot \cos\left(\frac{z^4v^4}{x^2y^3}\right) \\
&+ 0.091 \cdot \cos\left(\frac{xz^3w^4}{yv^4}\right) + 2.118.
\end{aligned}
$$

The true importance value for this data set for any given point is $[C, 0, 0, 0, 0]$ with C representing the importance value of x at any given point, and this value is usually with a magnitude of 10^4. We can notice that the ITEA expression uses every feature of the data set, not only the single meaningful variable x. This means that it can attribute non-zero importance values to noisy variables. At some points, we observed that the partial effect assigned zero importance to x and a large value for one or two other variables, explaining this large error observed in our results. On the other hand, Operon was more competent in selecting the important features for its models, a selected example for this data set is:

$$
\begin{aligned}
\text{OPExpr}(x, y, z, v, w) = {}& 82.931 \\
&- 81.160 \cdot \sin(\tanh(-3.433x)^{2^{2^{2^{2^2}}}} + \cos(\tanh(\tanh(\tanh(-1.182z)))^2)),
\end{aligned}
$$

which significantly reduces the average error as there is only one noisy variable in this expression.

8 Discussion

The obtained results bring forward some aspects worth considering whenever we seek to extract additional information from the prediction models. These results show additional evidence favoring Symbolic Regression as a graybox model that can either be interpreted without additional support or using external models that extract the information of interest.

8.1 Symbolic Regression can have a good trade-off between accuracy and interpretability

As we reported in Section 7.1.1, the Symbolic Regression models obtained more accurate models than the black-box models for the Feynman benchmark. Also, about 40% of the ground-truth models were successfully retrieved by at least one of the SR methods. This indicates the possibility of using the symbolic models explicitly when understanding the model behavior. Also, even for the models that do not correspond to the ground-truth, the symbolic models still allowed the extraction of near-perfect feature importance using different explanatory models, as reported in Section 7.1.3. Additionally, their explanations were as robust as those returned by MLP and SVM. Unlike these models, Random Forest and Gradient Boosting had lower quality explanations and were less robust due to their internal discretization of the feature space. Besides these advantages, the symbolic model also allows us, under appropriate conditions, to generate the symbolic partial derivatives of the model. This, in turn, makes it possible to calculate another feature importance measure: the partial effect. With the partial derivative expressions, we can also reduce the estimation error of the Integrated Gradient.

We also found some counterpoints to the above conclusion in Section 7.2 where we tested a set of challenging benchmarks. None of the regression models could generate a reasonable predictor for three out of the four data sets. As an implication of these models, we observed a higher error in the feature importance estimation. The main problem was that all of the predictors could not select the correct features for the model, thus attributing non-zero importance to noisy features. This leads to our next point of discussion.

When comparing the SR methods with the linear methods (Linear and Lasso) when using the regression-specific explainer Partial Effects, it is crucial to notice that linear methods will always hold equal or more stable explanations since the models do not perform any transformation or interaction over the variables; thus small changes will always have a direct and linear effect in the output. The SR methods, however, could approximate the behavior of linear models for explanation robustness, as we can see from the similar reported results for the robustness metrics in Fig. 8.

8.2 The explanation quality correlates with the model quality

Following the previous observation, we found that most explanatory models could still return a reasonable explanation even when the prediction models were not perfect (consider, for example, the accuracy of SVM explanations). Nevertheless, a practitioner should ensure that the prediction model has a reasonable quality and be aware of the possible inaccuracies when extracting an explanation. At the bare minimum, the practitioner should follow the standard Machine Learning pipeline of data cleaning, feature engineering, and hyper-parameters optimization for high-stakes prediction tasks before trusting the explanation extracted from the models.

We noticed that we did not perform an extensive grid search in the challenging benchmarks due to computational and time constraints. Having said that, a more careful experiment could improve the SR models quality for these challenging problems. Even when the model is accurate, we still have to be careful about the explanations because some regression models do not comply with the *desiderata* of explanatory methods.

8.3 Trusting an explanation requires evaluating its robustness and quality w.r.t. the prediction model

Even when the prediction model was accurate, we observed some instability in the explanations as depicted in Fig. 8 of Section 7.1.3. We noticed that the explanations produced by Integrated Gradient, when coupled with decision tree-based models, were unstable since these models generate a step function to approximate non-linear relationships. The explanation at specific points may suffer an enormous change when evaluated at neighborhood points. The practitioner should be aware of these limitations when choosing the right prediction and explanatory methods.

9 Conclusions

This paper proposed a benchmark of explanatory methods for regression models using the Feynman data set as a proxy to evaluate these models using a ground-truth as a reference. The main objective was to compare the performance of Symbolic Regression models when coupled with different explainers in contrast with other linear and non-linear regression models coupled with the same explainers.

For this purpose, we have implemented the *iirsBenchmark* as an open-source framework that contains wrapper methods to different regression models, including two symbolic regression and many global and local explanatory methods that return the importance of each feature for the prediction. This framework evaluates each pair regressor-explainer following the robustness and quality criteria. The robustness criteria tests how sensitive an

explainer is with slight variations in the data. The *desideratum* is that an explainer will not abruptly change the explanation with small perturbations. The quality of the explanations is evaluated as to whether the direction of the feature importance agrees with the ground-truth and how close they are to the true explanation.

We have done extensive experiments to evaluate how well Symbolic Regression models fare in this setting. We have found that, specifically for the Feynman data sets, SR was capable of returning accurate models that made it possible to extract the correct explanations even when the returned expression did not correspond to the ground-truth. Compared to the other regressors, SR performance was among the most stable explanations and closer to the ground-truth.

In conclusion, we have found evidence that SR models can return accurate models that correspond to the expected properties and behavior of the true model as captured by the explanatory methods.

For future work, we intend to expand the coverage for more symbolic regression methods, making it possible to compare different representations for symbolic regression, different coefficient optimization methods, and different evolutionary heuristics. One shortcoming of our experimental methodology is that we only tested problems of the same domain (i.e., physics). We will also add more regression problems from different domains in future work.

We also plan to investigate the role of dimensionality in feature importance explanations. We expect that dimensionality will have a compromise with explanation quality and robustness since larger dimensionalities add more degree of freedom when creating feature importance explanations.

Declarations

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

[1] Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law **28**(2), 237–266 (2020). https://doi.org/10.1007/s10506-019-09255-y

[2] Winter, G.: Machine learning in healthcare: A review. British Journal of Health Care Management **25**(2), 100–101 (2019). https://doi.org/10.12968/bjhc.2019.25.2.100

[3] Roscher, R., Bohn, B., Duarte, M.F., Garcke, J.: Explainable machine learning for scientific insights and discoveries. IEEE Access **8**, 42200–42216 (2020) https://arxiv.org/abs/1905.08883. https://doi.org/10.1109/ACCESS.2020.2976199

[4] Modarres, C., Ibrahim, M., Louie, M., Paisley, J.: Towards explainable deep learning for credit lending: A case study, 1–8 (2018) https://arxiv.org/abs/1811.06471

[5] Yoo, S., Xie, X., Kuo, F.-C., Chen, T.Y., Harman, M.: Human competitiveness of genetic programming in spectrum-based fault localisation: Theoretical and empirical analysis. ACM Trans. Softw. Eng. Methodol. **26**(1) (2017). https://doi.org/10.1145/3078840

[6] Lones, M.A., Alty, J.E., Cosgrove, J., Duggan-Carter, P., Jamieson, S., Naylor, R.F., Turner, A.J., Smith, S.L.: A new evolutionary algorithm-based home monitoring device for parkinson's dyskinesia. Journal of Medical Systems **41**(11) (2017). https://doi.org/10.1007/s10916-017-0811-7

[7] Lynch, D., Fenton, M., Fagan, D., Kucera, S., Claussen, H., O'Neill, M.: Automated self-optimization in heterogeneous wireless communications networks. IEEE/ACM Transactions on Networking **27**(1), 419–432 (2019). https://doi.org/10.1109/TNET.2018.2890547

[8] Izzo, D., Simões, L.F., Märtens, M., de Croon, G.C.H.E., Heritier, A., Yam, C.H.: Search for a grand tour of the jupiter galilean moons. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation. GECCO '13, pp. 1301–1308. Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2463372.2463524. https://doi.org/10.1145/2463372.2463524

[9] Semet, Y., Berthelot, B., Glais, T., Isbérie, C., Varest, A.: Expert competitive traffic light optimization with evolutionary algorithms. In: VEHITS, pp. 199–210 (2019)

[10] Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research **15**(90), 3133–3181 (2014)

[11] Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems **34**(6), 14–23 (2019). https://doi.org/10.1109/MIS.2019.2957223

[12] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 4768–4777. Curran

Associates Inc., Red Hook, NY, USA (2017)

[13] Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access **6**, 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052

[14] Aldeia, G.S.I., de França, F.O.: Measuring feature importance of symbolic regression models using partial effects. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '21. ACM, New York, NY, USA (2021). https://doi.org/10.1145/3449639.3459302. https://doi.org/10.1145%2F3449639.3459302

[15] Orzechowski, P., Cava, W.L., Moore, J.H.: Where are we now?: A large benchmark study of recent symbolic regression methods. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '18, pp. 1183–1190. ACM, New York, NY, USA (2018). https://doi.org/10.1145/3205455.3205539. http://doi.acm.org/10.1145/3205455.3205539

[16] Cava, W.L., Orzechowski, P., Burlacu, B., de França, F.O., Virgolin, M., JIN, Y., Kommenda, M., Moore, J.H.: Contemporary symbolic regression methods and their relative performance. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021). https://openreview.net/forum?id=xVQMrDLyGst

[17] Kronberger, G., de França, F.O., Burlacu, B., Haider, C., Kommenda, M.: Shape-constrained symbolic regression–improving extrapolation with prior knowledge. Evolutionary Computation, 1–24

[18] Affenzeller, M., Winkler, S.M., Kronberger, G., Kommenda, M., Burlacu, B., Wagner, S.: Gaining deeper insights in symbolic regression. In: Riolo, R., Moore, J.H., Kotanchek, M. (eds.) Genetic Programming Theory and Practice XI, pp. 175–190. Springer, New York, NY (2014). https://doi.org/10.1007/978-1-4939-0375-7_10. https://doi.org/10.1007/978-1-4939-0375-7_10

[19] de França, F.O.: A greedy search tree heuristic for symbolic regression. Information Sciences **442-443**, 18–32 (2018). https://doi.org/10.1016/j.ins.2018.02.040

[20] Ferreira, L.A., Guimaraes, F.G., Silva, R.: Applying genetic programming to improve interpretability in machine learning models. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, New York (2020). https://doi.org/10.1109/cec48606.2020.9185620. https://doi.org/10.1109%2Fcec48606.2020.9185620

[21] de França, F.O., Aldeia, G.S.I.: Interaction–Transformation Evolutionary Algorithm for Symbolic Regression. Evolutionary Computation **29**(3), 367–390 (2021) https://arxiv.org/abs/https://direct.mit.edu/evco/article-pdf/29/3/367/1959462/evco_a_00285.pdf. https://doi.org/10.1162/evco_a_00285

[22] de França, F.O., de Lima, M.Z.: Interaction-transformation symbolic regression with extreme learning machine. Neurocomputing **423**, 609–619 (2021)

[23] Kantor, D., Von Zuben, F.J., de França, F.O.: Simulated annealing for symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 592–599 (2021)

[24] Filho, R.M., Lacerda, A., Pappa, G.L.: Explaining symbolic regression predictions. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, New York (2020). https://doi.org/10.1109/cec48606.2020.9185683. https://doi.org/10.1109%2Fcec48606.2020.9185683

[25] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Computing Surveys **51**(5), 1–45 (2018) https://arxiv.org/abs/1802.01933. https://doi.org/10.1145/3236009

[26] Ljung, L.: Perspectives on system identification. Annual Reviews in Control **34**(1), 1–12 (2010). https://doi.org/10.1016/j.arcontrol.2009.12.001

[27] Marcinkevičs, R., Vogt, J.E.: Interpretability and Explainability: A Machine Learning Zoo Mini-tour, 1–24 (2020) https://arxiv.org/abs/2012.01805

[28] Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019) https://arxiv.org/abs/1811.10154. https://doi.org/10.1038/s42256-019-0048-x

[29] Wu, Z.F., Li, J., Cai, M.Y., Lin, Y., Zhang, W.J.: On membership of black-box or white-box of artificial neural network models. In: 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), pp. 1400–1404 (2016). https://doi.org/10.1109/ICIEA.2016.7603804

[30] Loyola-González, O.: Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. IEEE Access **7**, 154096–154113 (2019). https://doi.org/10.1109/ACCESS.2019.2949286

[31] Julia Angwin, S.M. Jeff Larson, Lauren Kirchner, P.: Machine bias:

There's software used across the country to predict future criminals. and it's biased against blacks (2016)

[32] Datta, A., Tschantz, M.C., Datta, A.: Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. CoRR **abs/1408.6491** (2014) https://arxiv.org/abs/1408.6491

[33] Lipton, Z.C.: The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (2018). https://doi.org/10.1145/3236386.3241340

[34] Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine learning interpretability: A survey on methods and metrics. Electronics **8**(8), 832 (2019). https://doi.org/10.3390/electronics8080832

[35] Arrieta, A.B., Díaz-Rodríguez, N., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012

[36] Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)

[37] Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018, 80–89 (2019) https://arxiv.org/abs/1806.00069. https://doi.org/10.1109/DSAA.2018.00018

[38] Sendak, M., Elish, M.C., Gao, M., Futoma, J., Ratliff, W., Nichols, M., Bedoya, A., Balu, S., O'Brien, C.: The human body is a black box. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. ACM, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3372827. https://doi.org/10.1145/3351095.3372827

[39] Ghassemi, M., Oakden-Rayner, L., Beam, A.L.: The false hope of current approaches to explainable artificial intelligence in health care. The Lancet Digital Health **3**(11), 745–750 (2021). https://doi.org/10.1016/s2589-7500(21)00208-9

[40] Banerjee, I., Bhimireddy, A.R., Burns, J.L., Celi, L.A., Chen, L.-C., Correa, R., Dullerud, N., Ghassemi, M., Huang, S.-C., Kuo, P.-C., Lungren, M.P., Palmer, L., Price, B.J., Purkayastha, S., Pyrros, A., Oakden-Rayner, L., Okechukwu, C., Seyyed-Kalantari, L., Trivedi, H., Wang, R., Zaiman, Z., Zhang, H., Gichoya, J.W.: Reading Race: AI Recognises Patient's Racial Identity In Medical Images (2021)

[41] Yang, M., Kim, B.: Benchmarking Attribution Methods with Relative Feature Importance (2019)

[42] Camburu, O.-M., Giunchiglia, E., Foerster, J., Lukasiewicz, T., Blunsom, P.: The Struggles of Feature-Based Explanations: Shapley Values vs. Minimal Sufficient Subsets (2020)

[43] Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 2801–2807. International Joint Conferences on Artificial Intelligence Organization, California, USA (2019). https://doi.org/10.24963/ijcai.2019/388. https://doi.org/10.24963/ijcai.2019/388

[44] Camburu, O., Giunchiglia, E., Foerster, J., Lukasiewicz, T., Blunsom, P.: Can I trust the explainer? verifying post-hoc explanatory methods. CoRR **abs/1910.02065** (2019) https://arxiv.org/abs/1910.02065

[45] Alvarez-Melis, D., Jaakkola, T.S.: On the Robustness of Interpretability Methods (Whi) (2018) https://arxiv.org/abs/1806.08049

[46] Hooker, G., Mentch, L.: Please Stop Permuting Features: An Explanation and Alternatives, 1–15 (2019) https://arxiv.org/abs/1905.03151

[47] Orcun Yalcin, M., Fan, X.: On Evaluating Correctness of Explainable AI Algorithms: an Empirical Study on Local Explanations for Classification (April), 0–7 (2021)

[48] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '15, pp. 1721–1730. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2783258.2788613. https://doi.org/10.1145/2783258.2788613

[49] Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C.A., Casalicchio, G., Grosse-Wentrup, M., Bischl, B.: General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models (01) (2020) https://arxiv.org/abs/2007.04131

[50] Yang, M., Kim, B.: BIM: towards quantitative evaluation of interpretability methods with ground truth. CoRR **abs/1907.09701** (2019) https://arxiv.org/abs/1907.09701

[51] Guidotti, R.: Evaluating local explanation methods on ground truth. Artificial Intelligence **291**, 103428 (2021). https://doi.org/10.1016/j.artint.2020.103428

[52] Hooker, S., Erhan, D., Kindermans, P.-J., Kim, B.: A benchmark for interpretability methods in deep neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 9737–9748. Curran Associates, Inc., Red Hook, NY, USA (2019). https://proceedings.neurips.cc/paper/2019/file/fe4b8556000d0f0cae99daa5c5c5a410-Paper.pdf

[53] Vaughan, J.W., Wallach, H.: A human-centered agenda for intelligible machine learning. Machines We Trust: Getting Along with Artificial Intelligence (2020)

[54] White, D.R., McDermott, J., Castelli, M., Manzoni, L., Goldman, B.W., Kronberger, G., Jaśkowski, W., O'Reilly, U.-M., Luke, S.: Better GP benchmarks: community survey results and proposals. Genetic Programming and Evolvable Machines **14**(1), 3–29 (2012). https://doi.org/10.1007/s10710-012-9177-2

[55] McDermott, J., Jong, K.D., O'Reilly, U.-M., White, D.R., Luke, S., Manzoni, L., Castelli, M., Vanneschi, L., Jaskowski, W., Krawiec, K., Harper, R.: Genetic programming needs better benchmarks. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference - GECCO '12. ACM Press, New York, NY, USA (2012). https://doi.org/10.1145/2330163.2330273. https://doi.org/10.1145/2330163.2330273

[56] Udrescu, S.M., Tegmark, M.: AI Feynman: A physics-inspired method for symbolic regression. Science Advances **6**(16) (2020) https://arxiv.org/abs/1905.11481. https://doi.org/10.1126/sciadv.aay2631

[57] Udrescu, S.-M., Tan, A., Feng, J., Neto, O., Wu, T., Tegmark, M.: Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. Advances in Neural Information Processing Systems **33**, 4860–4871 (2020)

[58] Yasui, Y., Wang, X.: Statistical Learning from a Regression Perspective vol. 65, pp. 1309–1310 (2009). https://doi.org/10.1111/j.1541-0420.2009.01343_5.x

[59] Kuonen, D.: Regression Modeling Strategies: with Applications to Linear Models, Logistic Regression, and Survival Analysis vol. 13, pp. 415–416 (2004). https://doi.org/10.1177/096228020401300512

[60] Asadzadeh, M.Z., Gänser, H.-P., Mücke, M.: Symbolic regression based hybrid semiparametric modelling of processes: An example case of a bending process. Applications in Engineering Science **6**, 100049 (2021). https://doi.org/10.1016/j.apples.2021.100049

[61] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. A Bradford book. Bradford, Bradford, PA (1992). https://books.google.com.br/books?id=Bhtxo60BV0EC

[62] Kommenda, M., Burlacu, B., Kronberger, G., Affenzeller, M.: Parameter identification for symbolic regression using nonlinear least squares. Genet Program Evolvable Mach **21**(3), 471–501 (2019). https://doi.org/10.1007/s10710-019-09371-3

[63] Kommenda, M., Burlacu, B., Kronberger, G., Affenzeller, M.: Parameter identification for symbolic regression using nonlinear least squares. Genetic Programming and Evolvable Machines **21**(3), 471–501 (2020)

[64] Burlacu, B., Kronberger, G., Kommenda, M.: Operon c++: An efficient genetic programming framework for symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '20, pp. 1562–1570. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3377929.3398099. https://doi.org/10.1145/3377929.3398099

[65] Luke, S.: Two fast tree-creation algorithms for genetic programming. Trans. Evol. Comp **4**(3), 274–283 (2000). https://doi.org/10.1109/4235.873237

[66] Aldeia, G.S.I.: Avaliação da interpretabilidade em regressão simbólica. Master's thesis, Universide Federal do ABC, Santo André, SP (December 2021)

[67] Breiman, L.: Random forests **45**(1), 5–32 (2001). https://doi.org/10.1023/a:1010933404324

[68] Ribeiro, M.T., Singh, S., Guestrin, C.: "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. KDD '16, pp. 1135–1144. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778. https://doi.org/10.1145/2939672.2939778

[69] Miranda Filho, R., Lacerda, A., Pappa, G.L.: Explaining symbolic regression predictions. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2020). IEEE

[70] Covert, I., Lundberg, S., Lee, S.-I.: Understanding Global Feature Contributions With Additive Importance Measures (2020)

[71] Morris, M.D.: Factorial sampling plans for preliminary computational experiments. Technometrics **33**(2), 161–174 (1991)

[72] Nori, H., Jenkins, S., Koch, P., Caruana, R.: Interpretml: A unified framework for machine learning interpretability. CoRR **abs/1909.09223** (2019) https://arxiv.org/abs/1909.09223

[73] Sundararajan, M., Taly, A., Yan, Q.: Axiomatic Attribution for Deep Networks (2017)

[74] Aumann, R.J., Shapley, L.S.: Values of Non-atomic Games. Princeton University Press, Princeton, NJ, USA (2015)

[75] Lüdecke, D.: ggeffects: Tidy data frames of marginal effects from regression models. Journal of Open Source Software **3**(26), 772 (2018)

[76] Norton, E.C., Dowd, B.E., Maciejewski, M.L.: Marginal Effects—Quantifying the Effect of Changes in Risk Factors in Logistic Regression Models. JAMA **321**(13), 1304–1305 (2019) https://arxiv.org/abs/https://jamanetwork.com/journals/jama/articlepdf/2728169/jama_norton_2019_gm_190004.pdf. https://doi.org/10.1001/jama.2019.1954

[77] Long, J.S., Mustillo, S.A.: Using predictions and marginal effects to compare groups in regression models for binary outcomes **50**(3), 1284–1320 (2018). https://doi.org/10.1177/0049124118799374

[78] Mize, T.D., Doan, L., Long, J.S.: A general framework for comparing predictions and marginal effects across models. Sociological Methodology **49**(1), 152–189 (2019) https://arxiv.org/abs/https://doi.org/10.1177/0081175019852763. https://doi.org/10.1177/0081175019852763

[79] Onukwugha, E., Bergtold, J., Jain, R.: A primer on marginal effects—part i: Theory and formulae. PharmacoEconomics **33**(1), 25–30 (2015). https://doi.org/10.1007/s40273-014-0210-6

[80] Agresti, A., Tarantola, C.: Simple ways to interpret effects in modeling ordinal categorical data. Statistica Neerlandica **72**(3), 210–223 (2018) https://arxiv.org/abs/https://onlinelibrary.wiley.com/doi/pdf/10.1111/stan.12130. https://doi.org/10.1111/stan.12130

[81] Norton, E.C., Dowd, B.E., Maciejewski, M.L.: Marginal effects—quantifying the effect of changes in risk factors in logistic regression models. JAMA **321**(13), 1304 (2019). https:

//doi.org/10.1001/jama.2019.1954

[82] Plumb, G., Al-Shedivat, M., Xing, E.P., Talwalkar, A.: Regularizing black-box models for improved interpretability. CoRR **abs/1902.06787** (2019) https://arxiv.org/abs/1902.06787

[83] Alvarez-Melis, D., Jaakkola, T.S.: Towards Robust Interpretability with Self-Explaining Neural Networks (2018)

[84] Yeh, C.K., Hsieh, C.Y., Suggala, A.S., Inouye, D.I., Ravikumar, P.: On the (In)fidelity and sensitivity of explanations. Advances in Neural Information Processing Systems **32**(NeurIPS) (2019) https://arxiv.org/abs/1901.09392

[85] Zhou, Z., Hooker, G., Wang, F.: S-lime. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021). https://doi.org/10.1145/3447548.3467274

[86] Loh, W.-L., *et al.*: On latin hypercube sampling. Annals of statistics **24**(5), 2058–2080 (1996)

[87] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research **7**(1), 1–30 (2006)

[88] Lee, S., Lee, D.K.: What is the proper way to apply the multiple comparison test? Korean Journal of Anesthesiology **71**(5), 353–360 (2018). https://doi.org/10.4097/kja.d.18.00242