Received: 30 March 2022 | Revised: 5 July 2022 | Accepted: 19 July 2022

DOI: 10.1111/exsy.13118

ORIGINAL ARTICLE

Expert Systems WILEY

Subgroup mining for performance analysis of regression models

João Pimentel[1] | Paulo J. Azevedo[1,2] | Luís Torgo[1,3]

[1]Department of Informatics, University of Minho, Braga, Portugal

[2]INESCT TEC, Porto, Portugal

[3]Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada

Correspondence
Paulo J. Azevedo, INESCT TEC, Porto, 4050-190, Portugal.
Email: pja@di.uminho.pt

Abstract

Machine learning algorithms have shown several advantages compared to humans, namely in terms of the scale of data that can be analysed, delivering high speed and precision. However, it is not always possible to understand how algorithms work. As a result of the complexity of some algorithms, users started to feel the need to ask for explanations, boosting the relevance of Explainable Artificial Intelligence. This field aims to explain and interpret models with the use of specific analytical methods that usually analyse how their predicted values and/or errors behave. While prediction analysis is widely studied, performance analysis has limitations for regression models. This paper proposes a rule-based approach, Error Distribution Rules (EDRs), to uncover atypical error regions, while considering multivariate feature interactions without size restrictions. Extracting EDRs is a form of subgroup mining. EDRs are model agnostic and a drill-down technique to evaluate regression models, which consider multivariate interactions between predictors. EDRs uncover regions of the input space with deviating performance providing an interpretable description of these regions. They can be regarded as a complementary tool to the standard reporting of the expected average predictive performance. Moreover, by providing interpretable descriptions of these specific regions, EDRs allow end users to understand the dangers of using regression tools for some specific cases that fall on these regions, that is, they improve the accountability of models. The performance of several models from different problems was studied, showing that our proposal allows the analysis of many situations and direct model comparison. In order to facilitate the examination of rules, two visualization tools based on boxplots and density plots were implemented. A network visualization tool is also provided to rapidly check interactions of every feature condition. An additional tool is provided by using a grid of boxplots, where comparison between quartiles of every distribution with a reference is performed. Based on this comparison, an extrapolation of counterfactual examples to regression was also implemented. A set of examples is described, including a setting where regression models performance is compared in detail using EDRs. Specifically, the error difference between two models in a dataset is studied by deriving rules highlighting regions of the input space where model performance difference is

unexpected. The application of visual tools is illustrated using EDRs examples derived from public available datasets. Also, case studies illustrating the specialization of subgroups, identification of counter factual subgroups and detecting unanticipated complex models are presented. This paper extends the state of the art by providing a method to derive explanations for model performance instead of explanations for model predictions.

KEYWORDS
interpretability, machine learning, performance, regression

1 | INTRODUCTION

Compared to humans, machine learning (ML) models show many advantages that can be a key to solve specific problems. These advantages can be seen in terms of reproducibility, scaling, speed, and accuracy (Hall & Gill, 2019). However, recent ML approaches are, in most cases, extremely complex, making it difficult to understand their behaviour. These models are often denominated black boxes or opaque models, since one can neither understand their internals, nor the reasons behind certain results. Consequently, end users have been demanding for explainability, accountability and transparency, as these models are frequently used to make important and costly decisions in areas such as health and finance (Guidotti, Monreale, Ruggieri, Turini, et al., 2018). In fact, decision makers will most likely look forward for an explanation in order to fully trust the model, regardless of its predictive accuracy.

This need for explainability is an outcome of advanced ML algorithms tending to be non-interpretable, resulting in divergences between the interpretation of the model and the human level of reasoning. This leads to a need of choosing between a more complex model with better predictive performance, or a simpler model that is not capable of generating results as good as the first, but that is interpretable. Therefore, the interest in Explainable Artificial Intelligence (XAI) has been increasing. Specifically, according to the Defence Advanced Research Projects Agency (DARPA), XAI aims to "produce more explainable models, while maintaining a high level of learning performance (prediction accuracy); and enable human users to understand, appropriately, trust, and effectively manage the emerging generation of artificially intelligent partners" (Turek, 2018). It is also important to note that, due to the European Union (EU) General Data Protection Regulation (GDPR), an individual has the right to be given an explanation when decisions are made in an automated way (Parliament, 2020). As a consequence, the need for transparent and fair algorithms is urgent, being one of the greatest challenges in ML and Data Science.

In this paper, we propose a novel method to assist in explaining the performance of regression models, that is, models that make numeric predictions. This method is based on the analysis of multivariate interactions within the scope of the predictors, and provides a drill-down perspective of the performance of models. Contrary to most frequent approaches of XAI, we do not focus on explaining the predictions of models. Instead, we address another important aspect of XAI: accountability. More specifically, we focus on describing, in an interpretable fashion, the reasons for the errors of models. Our aim is to analyse areas where models show an anomalous performance when compared to the average performance. This perspective also allows end users to study the fairness of a model in terms of critical subgroups of data, for example, by highlighting poor model performance on sensitive subgroups.

The rest of the paper is organized as follows. Section 2 describes the state of the art in terms of methods to evaluate the error of prediction models. In Section 3, the details of the proposal are described and discussed. Section 4 demonstrates some use case applications. Finally, Section 5 contains the main conclusions and future work.

2 | RELATED WORK

There are many approaches to the problem of model interpretability, focusing on different aspects of the concept. The first big differentiating factor between techniques is if these are intrinsic or post-hoc. Intrinsic methods, also known as transparent boxes or ante-hoc methods, refer to models that were created to be interpretable on their own (Guidotti, Monreale, Ruggieri, Turini, et al., 2018). Oppositely, post-hoc methods refer to the application of tools to analyse pre-trained models, independently of their complexity (Du et al., 2019). As a result of focusing on tools that can used regardless of the algorithm, the emphasis of this paper is on well-established post-hoc methods that are used to interpret black box regression models, prioritizing model agnostic techniques. Usually, these model agnostic tools analyse the relationship between input and output, without being necessary to observe the inner components of the model (Hall & Gill, 2019). Inherently interpretable models, that is, white-box models, are not approached in this paper because these do not consist of post-hoc methods, and can be interpreted by analysing the model itself. Nonetheless, agnostic and post-hoc methods can be applied to this type of models to analyse different aspects of their responses.

Most of the existing techniques are centred around the analysis of the predicted values of models. There are global methods, which are used to describe how models behave from an overall perspective (Hall & Gill, 2019). Some evaluate the effect one or more features have on predictions, while identifying possible interactions between these predictors. Examples of this can be seen in Partial Dependence Plots (PDPs) (e.g., Friedman, 2001), Individual Conditional Expectation (ICE) plots (Goldstein et al., 2015), or Accumulated Local Effects (ALE) (Apley & Zhu, 2020). PDPs produce a plot showing the relation between the domain of a continuous variable and the predicted value of a model, based on the expected average effect of the predictor values on the output. These plots can be extended to bi-variate analysis, highlighting the interactions between two variable values and the output. On the other hand, ICE plots produce a similar result, but differ in the fact that the effect is shown for each example, meaning that the level of abstraction decreases and the individual detail of the analysis increases. However, PDPs and ICE plots can produce erroneous results due to highly correlated features. These methods require extrapolation of responses across feature values, possibly producing unrealistic combinations. ALE solve this problem by computing output alterations for data instances inside small windows of values of the variable of interest, not requiring the generation of new data and being less computationally expensive. Some other methods involve emulating the original model using a white-box, generating a relation between feature values and its outputs. For instance, some global surrogate models utilize decision trees to emulate the original models (Bastani et al., 2017; Craven, 1996; Evans et al., 2019), others general additive models or decision rules (Guidotti, Monreale, Ruggieri, Pedreschi, et al., 2018), among others.

The study and evaluation of a specific instance or a small group of similar instances is also crucial, hence, the use of local methods (Hall & Gill, 2019). Some of these are based on conditional situations, explaining how changes in the features alter the predicted value (e.g., Biecek, n.d.; Biecek, 2018). Others study the importance of feature values in relation to the output, possibly decomposing the prediction (e.g., Alvarez-Melis & Jaakkola, 2018; Guidotti, Monreale, Ruggieri, Pedreschi, et al., 2018; Lei et al., 2018; Ribeiro et al., 2016; Ribeiro et al., 2018; Staniak & Biecek, 2019). The analysis of interactions among features is also an important technique, explaining hidden patterns and confirming or disproving known ones (e.g., Lundberg & Lee, 2017; Shapley, 1953).

Proposed by Britton (2019), regional explanations describe behaviours that affect significant regions of the data. As a consequence, these are neither global, nor local methods, producing more general results than local methods and more specific than global techniques. Alongside this new paradigm, the author also proposed a model agnostic tool to evaluate algorithms designated Visual Interaction Effects (VINE). This tool utilizes modified ICE curves to produce detailed information about how feature interactions affect the predictions of models. Additionally, in order to identify the interactions, one must read the produced chart, which does requires any detailed statistical analysis.

Although understanding model predictions is fundamental to explain how a model processes data, studying prediction errors is also extremely important. It allows users to weigh the reliability of models, especially when the predictions may influence important decisions. Thus, by having an estimate of the expected error for a particular test case, users are empowered with a risk evaluation measure and can decide whether it is acceptable to rely on the model. Performance analysis is well studied within predictive analytics, with most existing tools focusing on providing scalar metrics of global performance, that is, reliable estimates of the expected average error of models. Some examples of these metrics are the error rate, mean absolute error, mean-squared error, among others. There are also graphical-based metrics, providing a different perspective of model performance, for example, Receiver Operating Characteristic (ROC) curves (e.g., Metz, 1978), Brier curves (Hernández-Orallo et al., 2011), among others. While these are used on classification tasks, similar methods have been proposed for regression, for example, Regression Receiver Operating Characteristic (RROC) space (Hernández-Orallo, 2013), Regression Error Characteristic (REC) plots (Bi & Bennett, 2003) or surfaces (Torgo, 2005).

Some other methods assess the importance a feature has on the performance of a model. Proposed by Casalicchio et al. (2019), Individual Conditional Importance (ICI) and Partial Importance (PI) are graphical tools to help in the visualization of how alterations in the values of a feature affect the performance of a model. These can be used to evaluate the model globally or for individual observations. Functionally, these are variants of PDPs and ICE plots, displaying the feature importance instead of the expected prediction. As a consequence, ICI and PI curves can be used to analyse and compare the feature importance across different subsets of data. Recently, Fisher et al. (2019) proposed Model Class Reliance (MCR), a tool to estimate the importance of a feature based on the prediction error when the values of the feature are permuted, that is, the order of the values is altered. An alternative to this would be the use of drop-column importance, which calculates the decrease in performance when a feature is removed from the model. This technique is more computationally and time demanding, as it requires the model to be retrained, excluding the predictor (Parr et al., 2018). Nonetheless, these aim to analyse how the errors behave and not how this behaviour is influenced by actual feature values, that is, do not provide an explanation to a specific situation.

Aiming to analyse such interactions, Areosa and Torgo (2020b) proposed Error Dependence Plots (EDPs). Their work aims at establishing a relation between the expected error and the values of a certain predictor variable. EDPs consist of a graphical representation of the expected error as a function of the domain of a predefined feature, using boxplots for each meaningful bin of feature values. Considering that EDPs ignore interactions among predictors and, consequently, possible scenarios that may have an impact on the performance of the model, the authors developed a variant of EDPs to evaluate up to three predictors at the same time. Conceptually, these are similar to standard EDPs, differing in the fact that these present the error distribution across all possible combinations of bins between the predictors, instead of a single variable (Areosa & Torgo, 2020b). Moreover, EDPs were extended in order to allow the comparison of multiple models, forming Multiple model Error Dependence Plots (MEDPs). These are particularly useful to identify whether the best performing model is outperformed in a certain range or category of

values and, consequently, to compare models presenting identical overall performance (Areosa & Torgo, 2019). Similarly to EDPs, MEDPs allow the visualization of interactions between features, specifically between two variables with the use of bi-variate MEDPs.

However, the limitations of EDPs when plotting multiple variables impair some usability, as most real world problems tend to have more than three predictors. Moreover, since the method can ignore some existing interactions, it can cause misleading conclusions (Areosa & Torgo, 2019). Areosa and Torgo (2019) also proposed Parallel Error Plots (PEPs), using a method similar to Parallel Coordinate Plots (Inselberg, 1985), in which each variable is shown on the *X*-axis and is represented by a vertical bar. This approach informs users about the dependency between the expected error and the various predictor variables, attempting to show the error profile across the set of predictors. In order to do so, PEPs divide the errors into very high errors, for example, the top 10% largest errors, and the remaining ones, colouring each line according to the respective category of error. Nonetheless, PEPs also have some limitations, namely in the presence of outliers, due to scaling effects, and in large datasets or datasets containing an extensive amount of predictors, as the visualization might become complex and hard to follow (Areosa & Torgo, 2019).

In this paper, we propose an alternative form of analysing the prediction error of models based on subgroup mining. This is not the first time that subgroup mining was used to characterize the performance of predictive models. Torgo et al. (2021) used categorical distribution rules to analyse errors and describe reasons for their occurrence in classification models. With this proposal, we focus on regression models instead, and provide different tools to visually analyse the discovered subgroups.

3 | ERROR DISTRIBUTION RULES

Our general goal is to find and describe, in an interpretable way, regions of the feature space where the performance of a regression model is abnormal. By abnormal we mean that it significantly deviates, positively or negatively, from its global overall average performance. Uncovering these regions can be of key importance as it, for instance, allows raising concerns on using the model when a certain test case is within a region where unusually bad performance is expected, that is, warns the end user that maybe the model should not be trusted in these cases. Finding these regions can be reduced to comparing error distributions, for example, the global error distribution versus the error distribution on a specific area of the feature space.

To find these regions with different error distributions, one can make use of a subgroup mining approach, using the global model performance as reference. A numeric property of interest is defined as the attribute of study. In our case, this property is the expected numeric prediction (regression) error. We have used the subgroup mining approach known as Distribution Rules (DR) (Jorge et al., 2006), where derived rules represent subgroups with distributions of a numeric property of interest that differ significantly from a given reference distribution. The significance of each distribution is measured using a goodness-of-fit statistical test between the distribution in study and another sample.

Formally, a DR has the form $A \Rightarrow y = D_{y|A}$. A represents the subgroup as a set of items, y is the target attribute and $D_{y|A}$ is an empirical distribution of the values of y where A is observed. One simple example of a DR is as follows: let the target attribute be the salary of each individual in dollars. The rule $gender = male \,\&\, age = young \Rightarrow$ {500/2, 504/4, 670/3, 811/1} states that, among young men, two receive a value of 500, four of 504, three of 670 and one of 811.

The procedure for deriving distribution rules is basically a modification of the classical association rules algorithm, being a rule-oriented algorithm instead of the original itemset-oriented. Subgroups are formed as itemsets and filtered using the classical minimal support threshold. The consequent is formed by the items that represent the distribution associated to a specific subgroup. Rules are derived if they pass the Kolgomorov-Smirnov test (KS-test), that is, if p value $\leq \alpha$ for a KS-test between $D_{y|\emptyset}$ (the reference distribution) and $D_{y|A}$. A typical value for α is 0.05.

We define Error Distribution Rules (EDR) as distribution rules derived from datasets describing the performance of regression models. Similar to Areosa and Torgo (2020b), a Cross-Validation (CV) procedure was followed to obtain more reliable error estimates for every available instance. Specifically, a 10-fold CV was used to calculate the prediction of a model for each instance. By comparing these predictions with the real values, we obtain a prediction error, which is a reliable estimate as it was obtained through CV. A dataset is created with the original feature values and with a new variable that contains these prediction errors. This dataset is then used to obtain the EDRs using the new variable as the property of interest.

In comparison to EDPs, EDRs generate less subgroups to be analysed than the plots of EDPs. In fact, if every interaction that contains up to three variables was to be analysed using EDPs, it would require the derivation of $N + \frac{N!}{2! \times (N-2)!} + \frac{N!}{3! \times (N-3)!}$ plots, where N represents the number of features in the dataset. Moreover, as DRs do not have any limitation regarding the number of features in a certain combination, EDRs can unveil interactions previously unattainable by EDPs, as these are limited to a maximum of three variables.

3.1 | Graphical analysis of EDRs

With the goal of facilitating the interpretation and exploration of EDRs, several graphical methods were implemented. Consider the rule in Listing 1.1. This rule represents the error values of a subgroup of data defined by NO_3 = [0.7–1.59] and *mxPH* = [7.75–8.22] and

Listing 1.1 **Example of an EDR and its metrics**

$NO_3 = [0.7–1.59]$, $mxPH = [7.75–8.22]$, $oPO_4 = [7.4–31.33]$:
Antecedent support: 0.0505050505050505,
$p -$ value: 0.00012841998681792,
Kurtosis: −2.19901515227728,
Skewness: 0.434551559705091,
Mean: 0.187197541228329,
Mode: 0.0681512165090887,
Median: 0.114042212638308,
Standard deviation: 0.111431482662837,
Values: {0.0681512165090887/1,0.102151015210666/1,...}

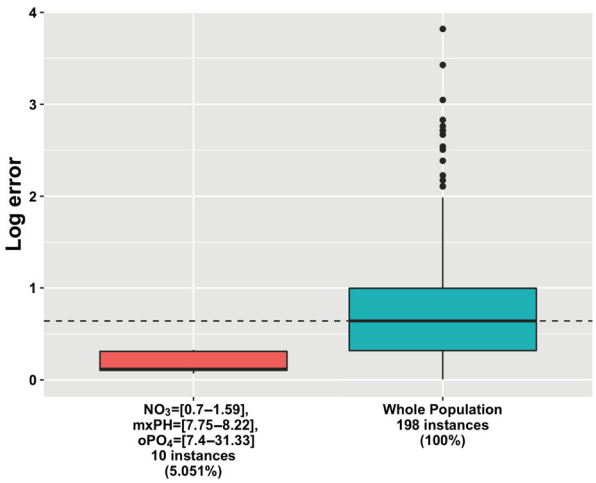


FIGURE 1 Boxplot visualization of an EDR from dataset A6 (see Table 1), trained with a GBM model (see Table 2)

$oPO_4 = [7.4–31.33]$ and was derived from the dataset representing the application of a GBM model to dataset A6. The subgroup that supports the rule is composed by 10 instances and the observed error values range from 0.06 up to 0.33 (the distribution is not completely shown in the listing).

The first method implemented to visualize EDRs uses boxplots like EDPs. These plots compactly represent a distribution by displaying summary statistics such as the median, the first (Q1) and third (Q3) quartiles. Figure 1 depicts the distribution of the variable of interest (the error) of the above rule in the form of a boxplot, and also the boxplot representing the distribution of the global performance of the model (GBM in the case of this rule). Here one can perceive that the subgroup presents much smaller error values to what was globally expected, that is, exhibits better performance. This means that if this rule is true for some new test case, the end user should take this as an indication that the model is expected to be very accurate and thus more trustworthy.

An alternative form of visualization was implemented using density plots, as these provide a different perspective on the distribution of errors. Such tool is illustrated in Figure 2, where we can compare the error distributions that were previously shown in Figure 1. Here, one can see that, while the global distribution is spread over the *X*-axis, the subgroup is concentrated on very low error values, having two major density peaks.

Although the number of derived subgroups is usually smaller than the number of EDPs for a given problem, this number can increase rapidly depending on data relations and/or looser filters. Thus, we have developed a simple and fast method to browse rules. This browsing tool is based on the construction of an interactive network using the *R* package *visNetwork* (Almende et al., 2019). The derived network contains elliptic- and diamond-shaped nodes. The first correspond to bins of features and the second represent rules. Nodes are colour labelled according to subgroup distribution, that is, a node is coloured blue when the median of the subgroup distribution is equal or smaller than the global median and red otherwise. This tool also allows the user to highlight a specific node and its direct connections, turning non-linked nodes grey. By doing so, or by hovering a node, its characteristics are displayed, that is, its designation and, in the case of diamond nodes, values such as mean, mode, or standard deviation of the error distribution. An example of such action can be seen in Figure 3. It corresponds to part of a network of rules derived from

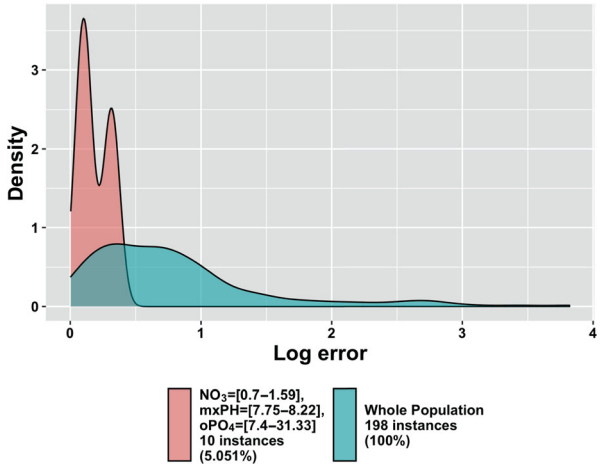


FIGURE 2 Density plot visualization of an EDR from dataset A6 (see Table 1), trained with a GBM model (see Table 2)

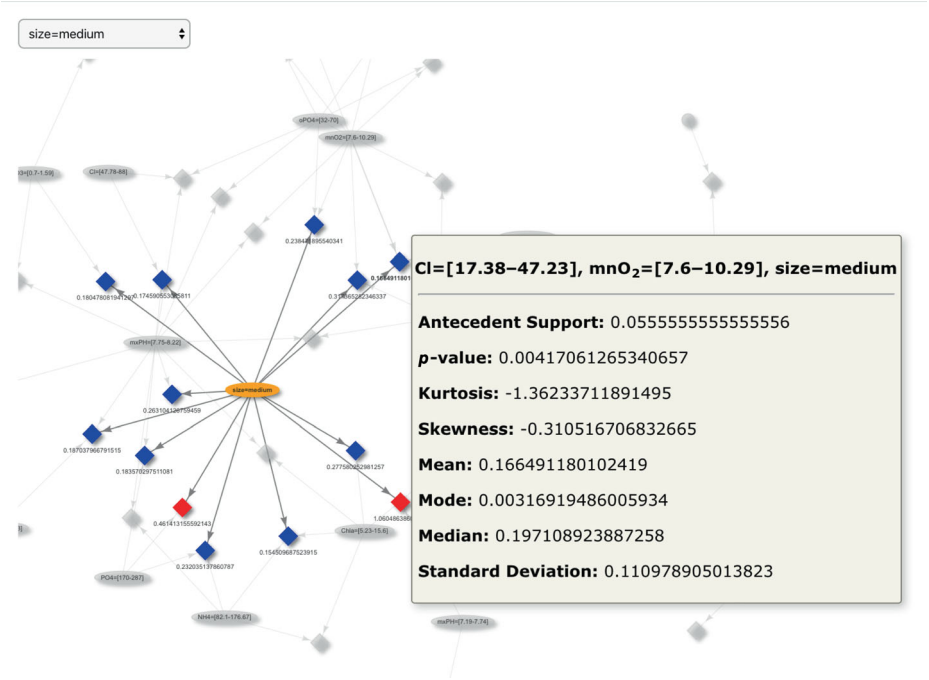


FIGURE 3 Network visualization of EDRs from dataset A7 (see Table 1), trained with a SVM model (see Table 2)

the application of a SVM model to dataset A7, where the condition (elliptic node) *size* = *medium* was selected and the linked subgroup (blue diamond node on the top right corner of the highlighted network) *Cl* = [17.38–47.23], *mn*O_2 = [7.6–10.29], *size* = *medium* hovered for details.

Using this exploration tool, the end user can quickly identify rules of interest that can then be further analysed by any of the other visualization methods we have described before.

When the goal is to compare multiple subgroups simultaneously, an alternative representation was derived. The idea consists of a grid of boxplots, where the summary points (upper and lower whiskers, Q1, median and Q3) of all subgroups are coloured based on how these compare to the same points of the reference distribution. There are three possible colours for each point: green, blue and red. Green means that the point has a lower value in comparison to the reference, blue equal and red higher. This allows end users to see, in a clear and simple way, how every subgroup compares to the reference and between themselves, as depicted in Figure 4.

Another interesting application of these boxplots is to use them with subgroups that share conditions, as seen in Figure 5. Here, one can see the comparison of subgroup PO_4 = [1–13.2] against all its specializations. In particular, this plot shows how the interaction of PO_4 = [1–13.2] with other feature values affects the performance of the model. For instance, by grouping PO_4 = [1–13.2] with *size* = *small* or *o*PO_4 = [1–7.3],

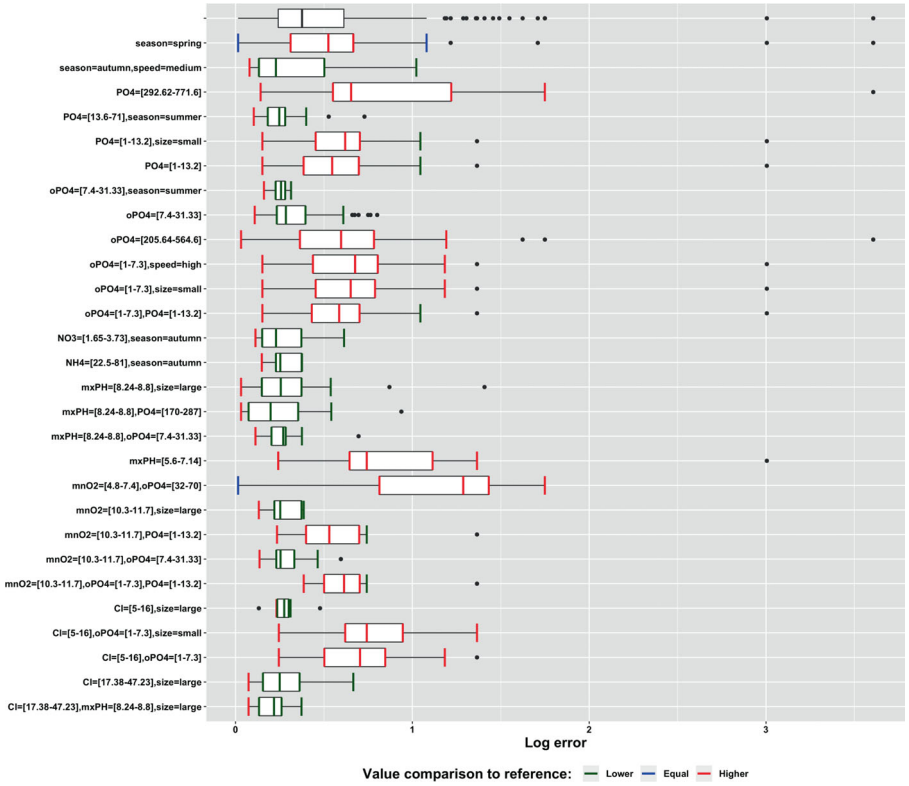


FIGURE 4 Boxplots from EDRs with coloured summary points from dataset A4 (see Table 1), trained with a GBM model (see Table 2)

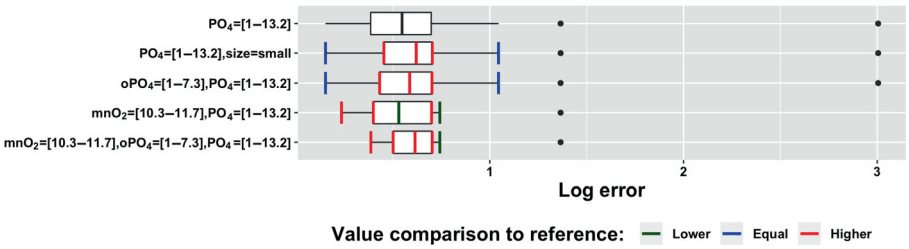


FIGURE 5 Boxplots from a filtered group of EDRs with coloured summary points based on feature PO4 from dataset A4 (see Table 1), trained with a GBM model (see Table 2)

the overall performance decreases, as described by the Q1, median and Q3 values. Actually, it is interesting to see that subgroup $PO_4 = [1–13.2]$, $mnO_2 = [10.3–11.7]$ yield higher values for lower-whisker, Q1 and Q3, but lower for both upper-whisker and median, indicating an intriguing performance.

3.2 | Counterfactual subgroups

Counterfactual analysis emerged as a tool to identify extreme situations in classification models. Here, we revisit the concept in a regression setting. Taking an initial subgroup, one wants to identify which features of the subgroup have to be updated to produce an extreme opposite output.

We implement counter-factual subgroups by considering the summary points of subgroups and focus on subgroups with the most interesting values. This approach reduces the number of cases for the users to analyse, focusing only on the ones that differ in some key aspect of the distribution.

The process starts by generating a dataset for each subgroup discovered containing replicas of the information of every other subgroup that shares some condition with the first. The constructed dataset can be seen as a list of key/value pairs (like in a hash table). Each key represents a subgroup and the value is a list of measures akin to that subgroup distribution, that is, the summary points. Each list is then ordered by the number

Listing 1.2 **Example of a subgroup and its list of similar subgroups, before filtering**

(1) $A = [1–2]$, $B = [3–4]$:
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Equal

(2) $A = [1–2]$, $B = [3–4]$, $C =$ "Tokyo":
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Equal

(3) $A = [1–2]$, $D = [5–7]$:
Lower-whisker: Lower
Q1: Higher
Median: Lower
Q3: Lower
Upper-whisker: Equal

of conditions shared with the subgroup in descending order. Hence, subgroups that have the most similarities to the discovered subgroup are closer to the top of the list. Then, every value of each pair in the dataset is compared to the reference distribution to obtain comparison references. Similar to the boxplots in Figures 4 and 5, there are three possible values: Higher, Equal or Lower. Finally, entries that do not show a contrasting behaviour are discarded.

Listing 1.2 contains an example of the creation of such dataset. We see that subgroups (2) and (3) are associated with subgroup (1), as all three share, at least, the condition $A = [1–2]$. Due to the fact that the performance of subgroup (2) is exactly the same as the one of subgroup (1), it is discarded, as in the context of possible counterfactual scenarios it is only interesting to analyse subgroup (3).

An example of some counter-factual subgroups can be seen in Listing 1.3, where the subgroups depicted were extracted from dataset A4. The base subgroup is defined by the conditions $size = large$, $Cl = [5–16]$ and the reference used was the global error distribution. As seen, this subgroup is characterized by having higher error values for the lower-whisker and lower for the remaining summary points. Moreover, after filtering non-interesting subgroups that share at least one condition with the base subgroup, only $oPO_4 = [1–7.3]$, $Cl = [5–16]$, $size = small$ and $oPO_4 = [1–7.3]$, $Cl = [5–16]$ were selected. Both subgroups are defined by the same behaviour in comparison to the global distribution, that is, lower values for the upper-whisker and higher for every other point. Consequently, their behaviour differs from $size = large$, $Cl = [5–16]$, making them counter-factual subgroups that lead to a differing performance regarding the expected error magnitudes.

4 | USE CASE ILLUSTRATIONS

The methodology used to validate the proposed tool and some illustrative examples of the use of EDRs to assess the performance of regression models are presented in this section.

4.1 | Methodology

In order to evaluate the developed tools, a considerable number of datasets was used, as seen in Table 1. Some datasets are only composed by a relatively small number of instances, for example, A1 or A7, while others are significantly larger, as CpuSm. The number of predictors also varies widely. To allow full reproducibility of the results, the original and transformed datasets are publicly available at Github. The repository contains

Listing 1.3 **Example of counter-factual subgroups from dataset A4 (see Table 1), trained with a GBM model (see Table 2)**

size = large, *Cl* = [5–16]:
Lower-whisker: Higher
Q1: Lower
Median: Lower
Q3: Lower
Upper-whisker: Lower

oPO_4 = [1–7.3], *Cl* = [5–16], size = small:
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Lower

oPO_4 = [1–7.3], *Cl* = [5–16]:
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Lower

TABLE 1 Datasets used for benchmarking. Mean and standard deviation (*SD*) values are rounded to three decimal places for visibility

Dataset	Source	Instances	Predictors			Target	
			Total	Numerical	Categorical	Mean	*SD*
A1	Areosa and Torgo (2020a)	198	11	8	3	16.996	21.422
A2	Areosa and Torgo (2020a)	198	11	8	3	7.471	11.065
A3	Areosa and Torgo (2020a)	198	11	8	3	4.334	6.977
A4	Areosa and Torgo (2020a)	198	11	8	3	1.997	4.439
A6	Areosa and Torgo (2020a)	198	11	8	3	6.005	11.711
A7	Areosa and Torgo (2020a)	198	11	8	3	2.487	5.182
Abalone	Areosa and Torgo (2020a)	4177	8	7	1	9.934	3.224
Acceleration	Areosa and Torgo (2020a)	1732	14	12	2	11.520	3.345
Airfoild	Areosa and Torgo (2020a)	1503	5	5	0	124.836	6.899
AvailPwr	Areosa and Torgo (2020a)	1802	15	8	7	105.597	52.357
Bank8FM	Areosa and Torgo (2020a)	4499	8	8	0	0.161	0.151
Boston	Areosa and Torgo (2020a)	506	13	13	0	22.533	9.197
ConcreteStrength	Areosa and Torgo (2020a)	1030	8	8	0	35.818	16.706
CpuSm	Areosa and Torgo (2020a)	8192	12	12	0	83.969	18.402
FuelCons	Areosa and Torgo (2020a)	1764	37	25	12	7.075	1.799
MachineCpu	Areosa and Torgo (2020a)	209	6	6	0	105.622	160.831
Masters	Acharya (2018)	500	7	6	1	0.722	0.141
MaxTorque	Areosa and Torgo (2020a)	1802	32	19	13	220.332	100.247
Servo	Areosa and Torgo (2020a)	167	4	2	2	1.390	1.560

TABLE 2 Regression algorithms and parameters used for benchmarking

Algorithm	Parameters	Package
ANN	Hidden Layer Neurons = 10, Maximum Iterations = 1000, Weight Decay = 0.1	nnet (Venables & Ripley, 2002)
GBM	Distribution = Gaussian, Total of Trees to Fit = 10,000, Maximum Depth of Each Tree = 1, Learning Rate = 0.001	gbm (Greenwell et al., 2020)
RF	Number of Trees to Grow = 500	randomForest (Liaw & Wiener, 2002)
SVM	Kernel = Radial, C = 1, Epsilon = 0.1, Gamma = 1/(Data Dimension)	e1071 (Meyer et al., 2020)

TABLE 3 Distribution details on datasets used for benchmarking. Values are rounded to three decimal places for visibility

Dataset	Target					Shape	
	Min.	Q1	Median	Q3	Max.	Kurtosis	Skewness
A1	0.000	1.525	6.950	24.800	89.800	1.274	1.452
A2	0.000	0.000	3.000	11.275	72.600	7.482	2.393
A3	0.000	0.000	1.550	4.975	42.800	6.817	2.449
A4	0.000	0.000	0.000	2.400	44.600	47.361	5.918
A6	0.000	0.000	0.000	6.975	77.600	11.610	3.118
A7	0.000	0.000	1.000	2.400	31.600	15.182	3.696
Abalone	1.000	8.000	9.000	11.000	29.000	2.324	1.113
Acceleration	3.900	9.300	11.200	13.500	31.700	2.254	0.769
Airfoild	103.400	120.200	125.700	130.000	141.000	−0.321	−0.418
AvailPwr	29.000	74.000	93.000	125.000	423.000	5.091	1.902
Bank8FM	0.000	0.033	0.117	0.249	0.802	0.617	1.082
Boston	5.000	17.020	21.200	25.000	50.000	1.451	1.102
ConcreteStrength	2.330	23.710	34.450	46.130	82.600	−0.323	0.416
CpuSm	0.000	81.000	89.000	94.000	99.000	12.713	−3.415
FuelCons	2.700	5.900	6.900	8.000	17.300	2.882	1.137
MachineCpu	6.000	27.000	50.000	113.000	1150.000	18.558	3.837
Masters	0.340	0.630	0.720	0.820	0.970	−0.472	−0.288
MaxTorque	65.000	150.000	195.500	270.000	875.000	4.526	1.632
Servo	0.131	0.503	0.731	1.259	7.100	1.916	1.759

three versions of each dataset: the original dataset (as publicly available), the original dataset containing the predictions for every learner in Table 2, and a fully prepared version for subgroup discovery with discretised numerical attributes and the numeric property of interest representing error measurements. For more information on each dataset, refer to Table 3.

As we wanted to avoid any model dependency bias within the experiments, four different algorithms were used, as seen in Table 2: an Artificial Neural Network (ANN), a Gradient Boosting Machine (GBM), a Random Forest (RF) and a Support Vector Machine (SVM). Each dataset mentioned in Table 1 was used as part of a regression task that consisted of applying these four predictive learning algorithms to the data. No hyperparameter optimisation was performed, as the goal was not to obtain the best possible results for each problem, but to evaluate how the proposed tool would perform on distinct scenarios. Similarly to the datasets, in order to allow full reproducibility of the experiments, the used parameters for each learner are displayed. We used the R programming language (Team, R.C, 2020) and open source implementations of these black-box algorithms.

After applying each algorithm to each dataset, the next step is to calculate error estimates. As described in Section 3, reliable error estimates for each combination of algorithm-dataset are calculated using a 10-fold CV process. A comprehensive step-by-step explanation of this task can be seen in Algorithm 1. This step ends with the creation of a new dataset composed of all the original feature values plus a new column with the calculated error estimates.

A constraint of the subgroup-mining algorithm used is the need to discretise numerical input features before the discovery process. As such, in this paper, we adopt a very simple unsupervised discretisation method. Similar to the default approach used by Areosa and Torgo (2020b)

Algorithm 1 **Method used to obtain reliable error estimates**

Input: Dataset D
Input: Algorithm A
Input: Fold Count K
Input: Error Metric Err
Output: Error Estimates E

$D' \leftarrow Permute(D)$ ▷Randomly permute the data
$P \leftarrow Partition(D', K)$ ▷Create K equal-size partitions
$E \leftarrow \emptyset$
for $p \in P$ **do**
  $M \leftarrow Train\left(A, D' \setminus D'_p\right)$ ▷Train A on all but the partition p cases
  $e_p \leftarrow \left\{Err(y', y) \mid < x, y > \in D'_p \wedge y' = Predict(M, x)\right\}$ ▷Calculate errors of the model predictions for the partition p cases
  $E \leftarrow E \cup e_p$
end for
return E

regarding EDPs, numerical attributes are discretised using the quartiles of the data, that is, [0%–10%] for extremely low values, [10%–35%] representing low values, [35%–65%] concerning central values, [65%–90%] regarding high values and [90%–100%] in respect to extremely high values. Ideally, this action should be performed using some level of domain knowledge about the dataset. The aim is to use a set of intervals for the range of that is meaningful for the end users of the dataset. If such expertise is not available, a possible alternative would be to make use of a discretisation process supervised by the property of interest.

After this action, the data is prepared to be used in the subgroup mining process. CAREN (Azevedo, 2001-2021; Azevedo, 2003; Azevedo, 2005; Jorge et al., 2006), a rule discovery engine, was used to derive all rules in our experiments, having the error estimate column obtained prior to attribute discretisation as the property of interest. Depending on the dataset, different support and pruning measures were used. The first is used to filter out lower supported subgroups, that is, each considered subgroup has to be covered by, at least, *X*% of the total number of instances in the dataset. The latter is a measure that defines how different a subgroup has to be from its generalizations, ignoring those not sufficiently distinct.

The discovery of rules ends with an output table composed of multiple columns, each defining a certain aspect of a rule. The most important ones for the contents of the paper are the list of feature conditions that define a subgroup and its distribution values. These are used to generate all the proposed analytical approaches. For example, a boxplot of a subgroup uses the distribution values graphically, and the feature conditions as its legend. Additionally, all plots were produced using an open-source package, *ggplot2* (Wickham, 2016).

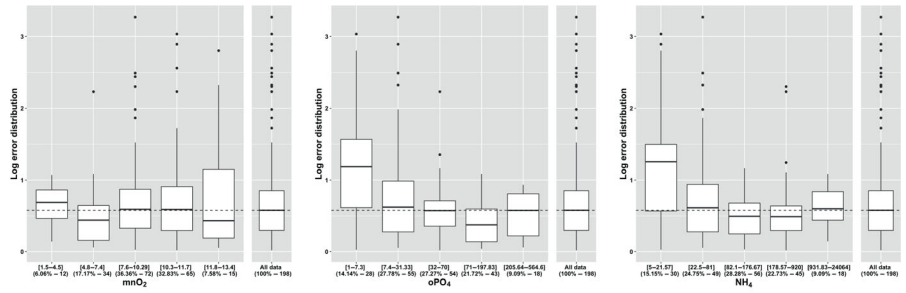
The last step of the used methodology is to analyse the discovered subgroups, comparing them between themselves and with one or more references. This step was initially performed by assessing the subgroups of a single model, followed by multi-model comparison.

4.2 | Application examples

In this section, a set of rules derived from the described datasets is presented to illustrate the potential of our proposals. The use of these rules allows users to uncover interactions between features with ease, specifically the ones that act differently from a given reference. Note that, for reproducibility purposes, all experiments presented in this section were performed using a minimum support of 5% and a pruning filter of 1×10^{-14}.

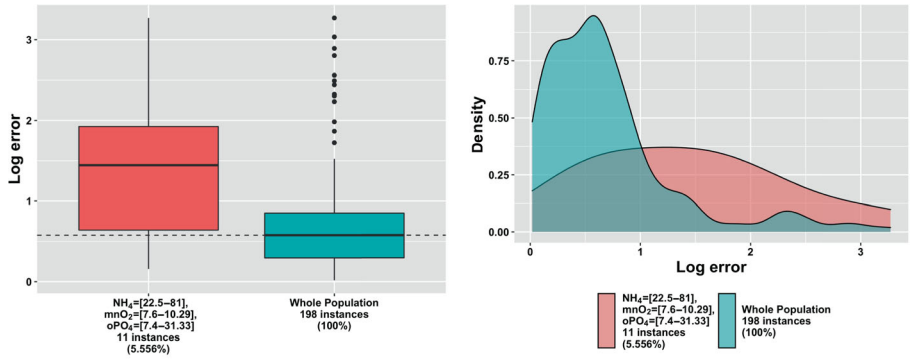
4.2.1 | Detection of unanticipated complex subgroups

The first example concerns the scenario where one can identify a set of subgroups that behave similarly, yet a more specific subgroup that combines all feature values behaves differently. This example was obtained by training a GBM model with dataset A1. By analysing the EDPs presented in Figure 6, one can observe some behavioural aspects about the performance of the model. For instance, the data bins characterized by



(a) Feature mnO2. (b) Feature oPO4. (c) Feature NH4.

FIGURE 6 EDPs from dataset A1 (see Table 1) to analyse features mnO2, oPO4 and NH4, trained with a GBM model (see Table 2)



(a) Boxplot. (b) Density plot.

FIGURE 7 EDR from dataset A1 (see Table 1) to analyse subgroup $NH4 = [22.5–81]$, $mnO2 = [7.6–10.29]$, $oPO4 = [7.4–31.33]$, trained with a GBM model (see Table 2)

$mnO_2 = [7.6–10.29]$ (Figure 6a), $oPO_4 = [7.4–31.33]$ (Figure 6b) and $NH_4 = [22.5–81]$ (Figure 6c) are defined by distributions similar to the global distribution.

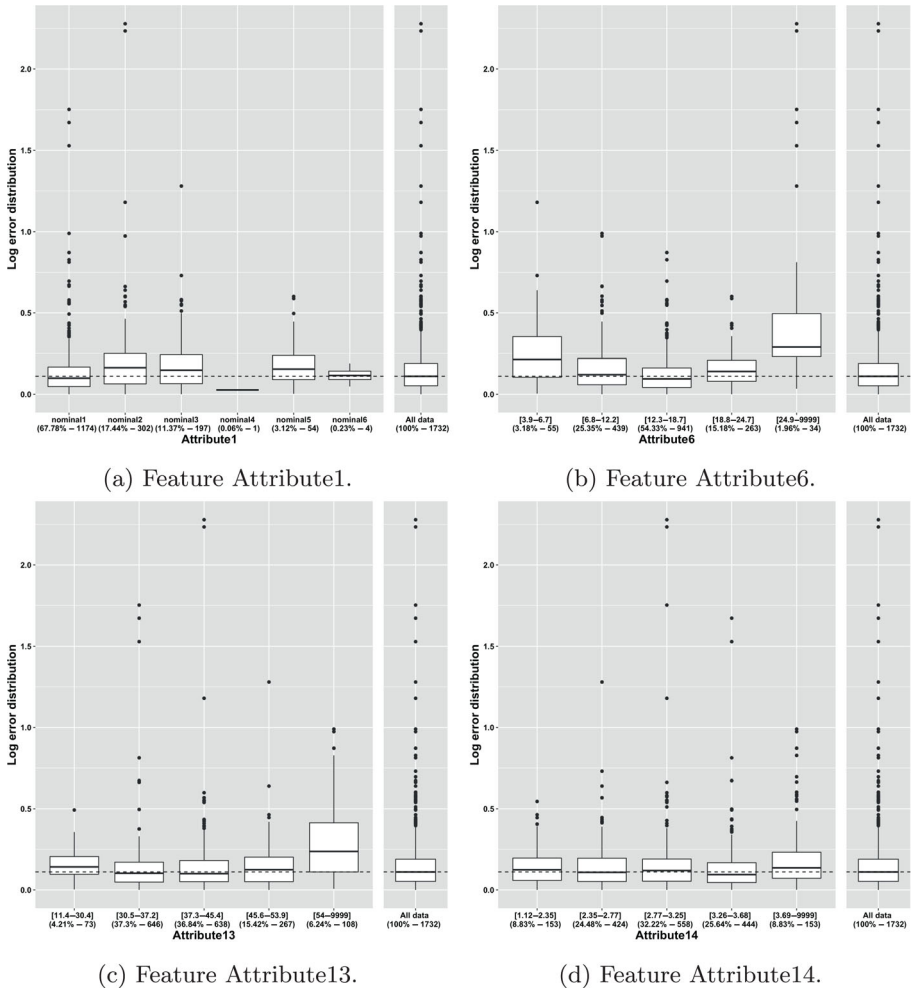
However, when considering a subgroup which includes all bins referred before using the EDR presented in Figure 7, we can see that the subgroup $NH_4 = [22.5–81]$, $mnO_2 = [7.6–10.29]$, $oPO_4 = [7.4–31.33]$ is characterized by having higher errors than expected. By analysing the boxplot in Figure 7a, it is clear that the error distribution of this subgroup has a higher value for all summary points when compared to the whole population. In fact, the Q1 value is even higher than the global median, displaying a significantly worse performance than one might initially expect. Moreover, the density plot in Figure 7b confirms that the majority of global error values are between 0 and 1. Contrarily, the error values of the subgroup are widely spread.

Comparing to EDPs, unless the multi-variate derivations were used, it would be difficult to predict the described behaviour for this subgroup, as the uni-variate EDPs do not display noticeably signs for such a differing performance.

4.2.2 | Detection of contradicting complex subgroups

Similarly to the previous case, a common scenario is when one can identify a set of subgroups that behave similarly and display some level of better/worse performance, but a specialization of such subgroups behaves in the opposing way. An example can be observed in the EDPs in Figure 8 derived from dataset acceleration. One can recognize that, for instance, $attribute1 = nominal1$ (Figure 8a), $attribute6 = [12.3–18.7]$ (Figure 8b), $attribute13 = [30.5–37.2]$ (Figure 8c) and $attribute14 = [3.26–3.68]$ (Figure 8d) have slightly better performance than globally expected, both in terms of median and Q3 values, having more compact distributions. Nonetheless, one could expect instances composed of these conditions to have similar errors to the ones expected, as the differences in the distributions are very slight.

With the use of EDRs, we see that this was not the case. A subgroup that connects the four conditions mentioned was detected, having smaller errors than globally expected. This behaviour is visible both in terms of median, as in Q1 and Q3 values, upper-whisker and the outliers of the distribution, as depicted in Figure 9. Actually, in Figure 9a we even see that the global median is greater than the Q3 value of this subgroup.



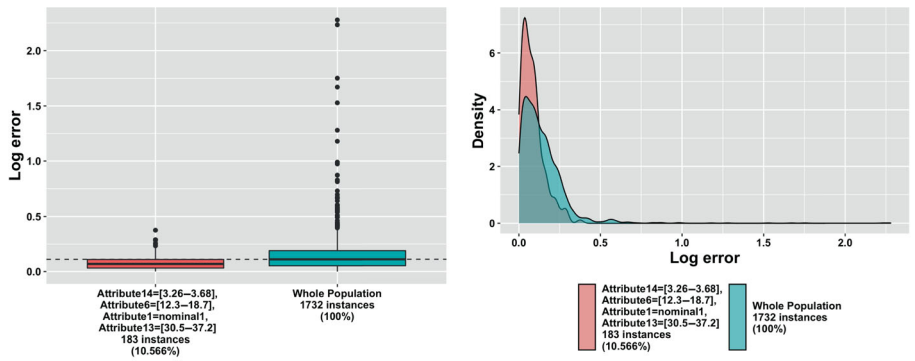
(a) Feature Attribute1.

(b) Feature Attribute6.

(c) Feature Attribute13.

(d) Feature Attribute14.

FIGURE 8 EDPs from dataset acceleration (see Table 1) to analyse features Attribute1, Attribute6, Attribute13 and Attribute14, trained with a SVM model (see Table 2)



(a) Boxplot.

(b) Density plot.

FIGURE 9 EDR from dataset acceleration (see Table 1) to analyse subgroup *attribute*14 = [3.26–3.68], *attribute*6 = [12.3–18.7], *attribute*1 = *nominal*1, *attribute*13 = [30.5–37.2], trained with a SVM model (see Table 2)

Moreover, it is important to emphasize that this is a subset of data with a considerable representation, corresponding to 10.6% of the total data (the distributions in terms of used attributes is 15.6% in *attribute*1 = *nominal*1, 19.5% in *attribute*6 = [12.3–18.7], 28.3% in *attribute*13 = [30.5–37.2] and a hefty 41.2% in *attribute*14 = [3.26–3.68]), being to some extent expected to show these signs in the EDPs.

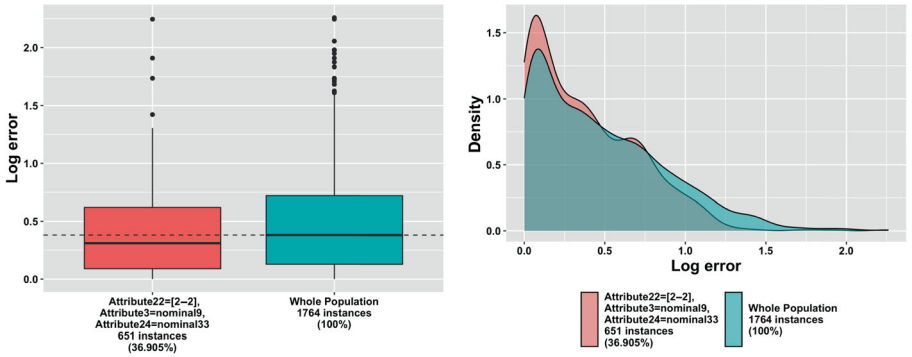
4.2.3 | Reduction in the number of subgroups to analyse

Another considerable advantage of the use of a method that uncovers interesting subgroups independently is the reduction in the number of situations to be analysed. Initially, the mining process detected such a large number of subgroups that we opted to increase the minimum support from 5% to 35% and the pruning filter from 1×10^{-14} to 0.001. By doing so, we intended to capture subgroups with higher frequency in this dataset, that is, with a larger representation. Nonetheless, close to 580 distinct subgroups were still uncovered. An example of these is the subgroup *attribute*22 = [2–2], *attribute*3 = *nominal*9, *attribute*24 = *nominal*33 presented in Figure 10. This subgroup is defined by smaller errors than expected and a representation of 36.9% of the total data instances.

It is important to note that, for this specific case, in order to find an interaction with three variables using EDPs and generate every plot of all interactions with three variables, it would be necessary to generate and analyse $\frac{37!}{3!\times(37-3)!} = 7770$ distinct plots. In comparison to the number of derived subgroups, these consist of only 7.5% of the 7770 plots that would require analysis.

4.2.4 | Comparison of multiple models

In most real world applications of ML, we need to perform some form of model comparison with the goal of selecting the best. EDRs can also be used with that purpose. Figure 11 depicts a subgroup defined by CGPA = [7.64–8.27] on dataset masters and shows how different models perform along this specific subgroup. This subgroup has a considerable representation, being composed by 145 instances, and corresponding to 29% of all examples. For all presented models, the subgroup has higher error values than expected. Comparing how models performed for the subgroup, it is visible that the ANN and SVM have more compact distributions, and the latter has the lowest values for all summary points, except



(a) Boxplot. (b) Density plot.

FIGURE 10 EDR from dataset FuelCons (see Table 1) to analyse subgroup *attribute*22 = [2–2], *attribute*3 = *nominal*9, *attribute*24 = *nominal*33, trained with an ANN model (see Table 2)

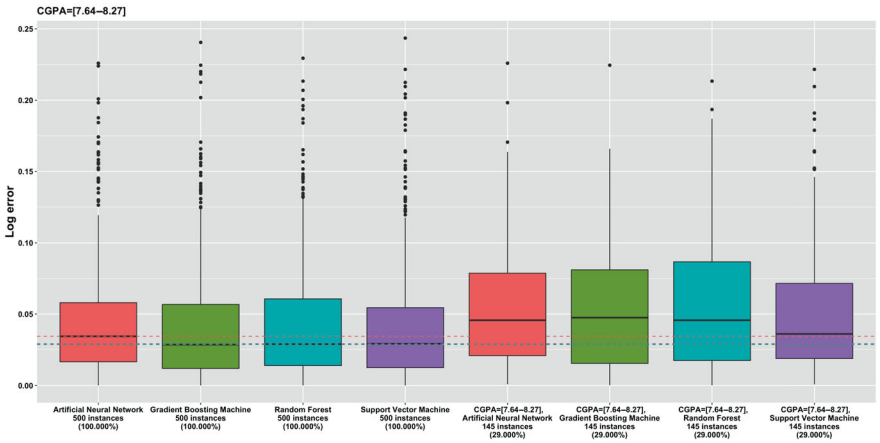


FIGURE 11 Boxplot EDRs from dataset masters (see Table 1) to analyse subgroup *CGPA* = [7.64–8.27], trained with all models (see Table 2)

Q1, which is smallest for the GBM. However, the SVM also has a considerable higher amount of outliers in comparison to the remaining models, something that could lead to further investigation in order to fully comprehend the behaviour of the model. Another interesting aspect of these plots is that the SVM has the closest median to the global distribution, being slightly higher than the global ANN model, contrarily to the other models that have considerably high values for this metric.

4.2.5 | Analysis of several subgroups

As important as the analysis of individual subgroups is, simultaneous analysis of several subgroups is extremely beneficial to have an overall idea of model behaviour. Also, one can check specific data items that have a differing behaviour than expected. Figure 12 displays the performance obtained from training a SVM model on dataset Masters as a set of boxplots with coloured summary points referring to how these relate to the global distribution. By analysing the plot, we see that there is an equilibrium of mostly higher and lower errors, as depicted by the amount of red and green coloured points, respectively. We also see that, for example, all subgroups that have the conditions *SOP* = [4–4.5] or *SOP* = [5–5] are characterized by smaller errors, contrarily to the ones with *SOP* = [1.5–2] that have higher error values. Another interesting aspect is to analyse how some subgroups, for example, *TOEFL.Score* = [95–101] and *TOEFL.Score* = [111–117], *University*. *Rating* = [5–5] have smaller outliers, being inside the area below the global upper-whisker. In addition, this plot allows the identification of some data items that compose more than one subgroup. It can be checked by the sample that is characterized by having the highest outlier for this model, being present in five different subgroups, among other analytical possibilities.

4.2.6 | Identification of counterfactual subgroups

On top of the analysis from the previous example, one can also identify some counter-factual subgroups. An example of such subgroups can be seen in the condition *Research* = 1, as portrayed in Listing 1.4. While the subgroups *University*. *Rating* = [2–2], *Research* = 1 and *University*. *Rating* = [2–2], *Research* = 1, *LOR* = [2.5–3.5] have worse overall performance in relation to the global distribution, *University*. *Rating* = [5–5], *Research* = 1 is defined by lower errors than expected. This contradicting behaviour is more noticeable in the values of Q1, median, Q3 and

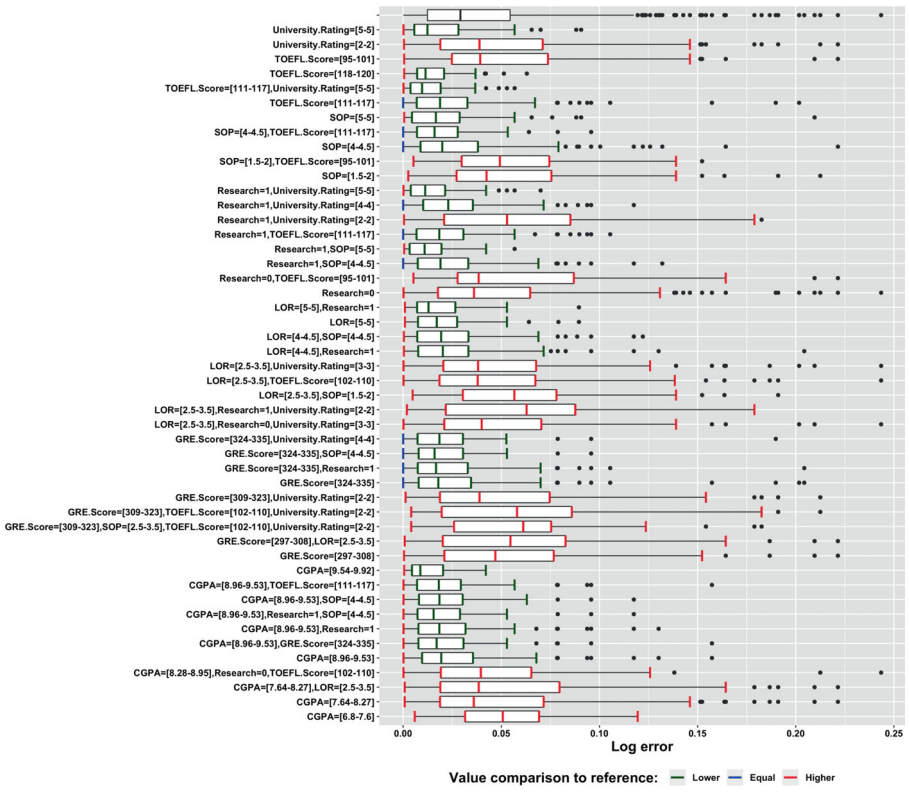


FIGURE 12 Global performance in dataset masters (see Table 1) using a SVM model (see Table 2)

Listing 1.4 Counter-factual examples for subgroup University. Rating = 5, Research = 1 of dataset Masters (see Table 1), trained with a SVM model (see Table 2)

University. Rating = [5–5], Research = 1:
Lower-whisker: Higher
Q1: Lower
Median: Lower.
Q3: Lower
Upper-whisker: Lower

University. Rating = [2–2], Research = 1, LOR = [2.5–3.5]:
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Higher

University. Rating = [2–2], Research = 1:
Lower-whisker: Higher
Q1: Higher
Median: Higher
Q3: Higher
Upper-whisker: Higher

upper-whisker of the subgroups. In fact, the latter subgroup has such a contradicting behaviour in relation to the other two, that its Q3 is very close to the Q1 of the remaining subgroups and its upper-whisker is even smaller than the others median.

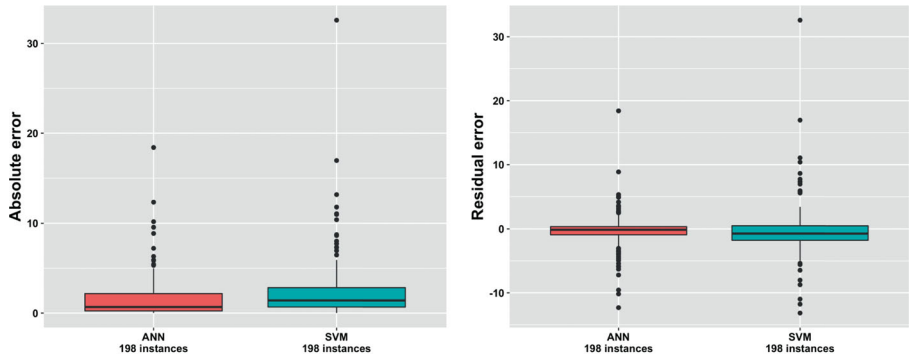
4.2.7 | Direct comparison between two models

As comparing multiple models by analysing plots can become complex over time due to models behaving differently for some subgroups, a final experiment was performed to demonstrate how EDRs can be used to directly compare two regression models. In order to do so, a new dataset was constructed where the property of interest concerns the difference between absolute errors of the two models for each test case. The core idea is to capture when the performance of a model surpasses the others, that is, when comparing M_1 and M_2, the property of interest is defined as $|E(M_1)| - |E(M_2)|$ and thus, when this value is positive, it means that M_1 has a higher error magnitude than M_2, and vice-versa.

As an example, we selected the ANN and SVM models trained on dataset A1. Figure 13 shows the differences between the absolute errors of both models in the form of absolute (Figure 13a) and residual (Figure 13b) values. We see that the ANN model appears to produce more accurate results. In the case of absolute errors, it has lower values for all summary points. Not only the distribution is closer to zero, but its median is approximately the same values as the Q1 of the SVM. By observing the residual values, we confirm this behaviour, as the distribution of the ANN is centred around zero (median is very close to zero), and it has a more compact distribution, that is, a smaller interval between the lower and upper whiskers. Nonetheless, the model has a considerably higher number of outliers in comparison to the SVM. Regarding the latter, due to having a negative median, this model appears to have a tendency to produce predictions that are higher than the real values and the ones predicted by the ANN.

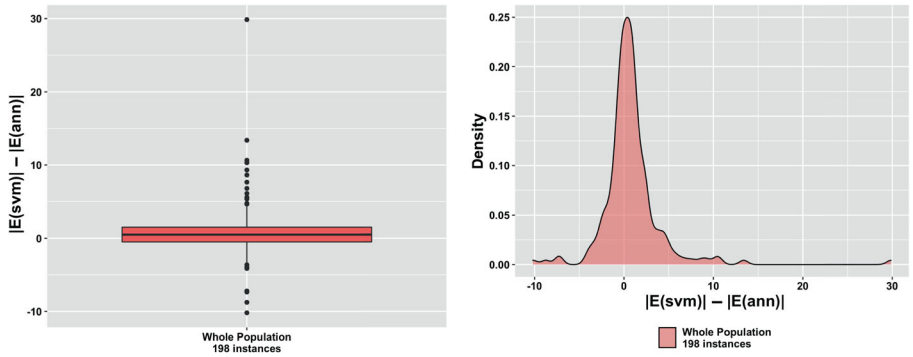
Figure 14 contains the graphical representation of the property of interest ($|E_{SVM}| - |E_{ANN}|$) in the form of a boxplot (Figure 14a) and a density plot (Figure 14b). We see that the distribution has a slight tendency to positive values, meaning that the SVM model produces errors with higher magnitude than the ones of the ANN model, as expected by analysing the previous figures.

However, this is not the case for all data instances, as some subgroups have a rather different behaviour. For instance, Figure 15 captures a situation where the performance of the models differs significantly from the global scenario. Clearly, the SVM outperforms the ANN model for subgroup $oPO_4 = [1–7.3]$, $mnO_2 = [7.6–10.29]$, as the distribution is mainly composed by negative values, indicating smaller magnitudes of error



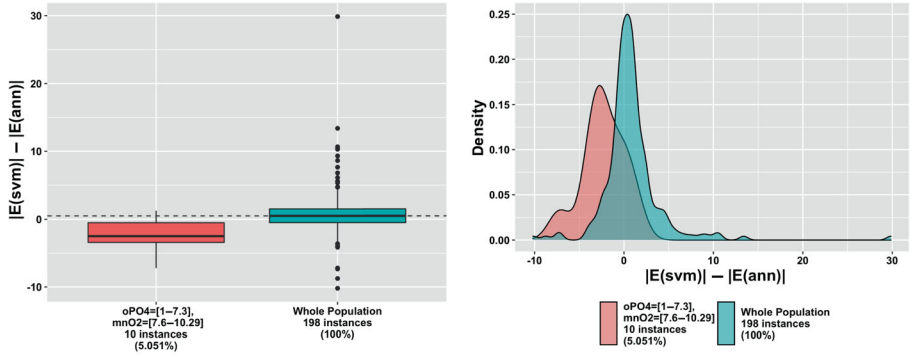
(a) Absolute. (b) Residual.

FIGURE 13 Absolute and residual errors from the application of ANN and SVM models (see Table 2) to dataset A1 (see Table 1)



(a) Boxplot. (b) Density plot.

FIGURE 14 Difference of absolute errors between SVM and ANN models (see Table 2) applied to dataset A1 (see Table 1)

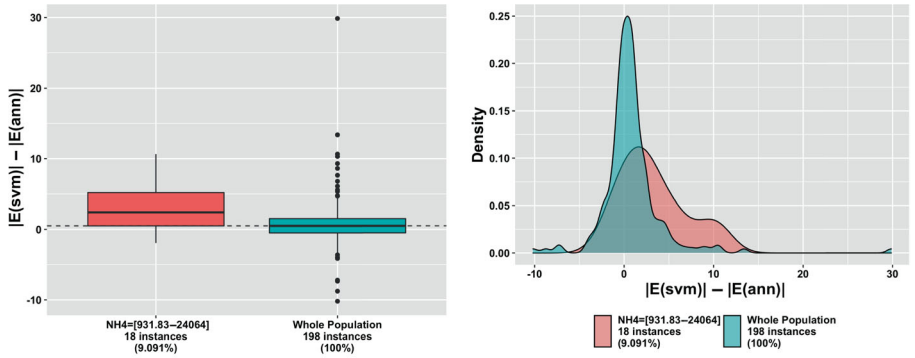


(a) Boxplot. (b) Density plot.

FIGURE 15 Graphical representation of an EDR representing the difference between absolute errors of SVM and ANN models (see Table 2), trained with dataset A1 (see Table 1), to analyse subgroup $oPO4 = [1–7.3]$, $mnO2 = [7.6–10.29]$

for the SVM model. In Figure 15b, the density plot substantiates this behaviour, as it shows a higher concentration of negative values for the subgroup.

Despite some contradicting subgroups in relation to the global expectation, there are conditions that behave similarly to what was expected, as seen in Figure 16. Here, the difference shows an even higher concentration of positive values comparing to the global distribution, manifesting lower magnitudes of error for the ANN model.



(a) Boxplot. (b) Density plot.

FIGURE 16 Graphical representation of an EDR representing the difference between absolute errors of SVM and ANN models (see Table 2), trained with dataset A1 (see Table 1), to analyse subgroup $NH4 = [931.83–24064]$

5 | CONCLUSIONS

In this paper, we proposed a novel approach to analyse the performance of regression models. This topic is of great importance due to the increase in the usage of ML to drive important day-to-day decisions. The proposed methods allow a more detailed and interpretable analysis of the performance of regression models, as opposed to the standard analysis through global performance metrics. Our methods allow end users to understand the risks associated with the predictions of the models, which has become a necessity for many organizations.

We proposed EDRs, a tool that can be seen as both an extension and a complement of EDPs, reducing the number of plots to analyse and allowing the finding of situations involving a large number of predictor variables, which is not easy in EDPs. These can be extremely important to highlight combinations of attributes whose best results are obtained with a model that is not the best overall. Moreover, it can be used to choose between models, or to produce ensembles to obtain the most accurate results, using the better performing models based on certain feature values.

In our illustrative examples of application, we were able to show many advantages of the use of a rule based method in the detailed performance analysis of regression models. Namely, to filter non-interesting subgroups and allow for individual or multi-exploration of distributions. In addition to that, rules allow the fast identification of unforeseen and contradicting subgroups. The first are important as the behaviour of a subgroup cannot always be detected by analysing less complex subgroups, and the second give end users an extra level of information on how the models behave for specific conditions that invert the expected performance.

The way discretisation of numerical attributes is approached in this proposal can be seen as a limitation, since no mechanical process is described. However, we rely on user domain knowledge to suggest the most adequate set of intervals to replace the range values of each numerical attribute. The derivation of EDRs lies on an association rule based algorithm, which are well known to behave exponentially on the number of frequent items (in our case, frequent pairs of attribute/value). Also, the need to specify threshold values for defining rarity (minimum support) and control on rules specialization (pruning) is strongly context dependent and can be regarded as a weakness.

A topic that we intend to explore in the future is the production of meta-models to generate ensembles by considering the distributions of these subgroups. The idea is to consider different weights attached to each model based on to the information on derived rules and associated subgroups.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in kaggle at https://www.kaggle.com/datasets/mohansacharya/graduate-admissions?select=Admission_Predict_Ver1.1.csv

ORCID

João Pimentel https://orcid.org/0000-0002-5794-3451
Paulo J. Azevedo https://orcid.org/0000-0002-0877-3070
Luís Torgo https://orcid.org/0000-0002-6892-8871

REFERENCES

Acharya, M.S. (2018). *Graduate admission 2—predicting admission from important parameters*. KAGGLE. https://www.kaggle.com/datasets/mohansacharya/graduate-admissions?select=Admission_Predict_Ver1.1.csv

Almende B.V., Thieurmel, B., Robert, T. (2019). *Visnetwork: Network Visualization using 'Vis.js' Library*. The R project. https://CRAN.R-project.org/package=visNetwork

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint*, arXiv:1806.08049.

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *82*(4), 1059–1086. https://doi.org/10.1111/rssb.12377

Areosa, I., & Torgo, L. (2019). Explaining the performance of black box regression models. In *2019 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 110–118). IEEE. https://doi.org/10.1109/DSAA.2019.00025

Areosa, I., Torgo, L.. (2020a). *Edps: A visual interpretation of regression error*. GitHub. https://github.com/ltorgo/ltorgo.github.io/blob/master/EDP/edp_code.zip

Areosa, I., & Torgo, L. (2020b). Visual interpretation of regression error. *Expert Systems*, *37*(6), e12621. https://doi.org/10.1111/exsy.12621

Azevedo, P.J. (2001-2021). *Caren—class project association rule engine*. Universidade do Minho. https://www.di.uminho.pt/∼pja/class/caren.html

Azevedo, P.J.. (2003). *Caren-a java based apriori implementation for classification purposes* (Technical Report). Université de Mons-Hainaut. Service de Science des Systèmes D'information. 1822/1982.

Azevedo, P.J.. (2005). *A data structure to represent association rules based classifiers* (Technical Report). Universidade do Minho, Departamento de Informática

Bastani, O., Kim, C., & Bastani, H. (2017). Interpretability via model extraction. *arXiv preprint*, arXiv:1706.09773.

Bi, J., & Bennett, K. P. (2003). Regression error characteristic curves. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 43–50). AAAI Press.

Biecek, P. (2018). Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, *19*(1), 3245–3249.

Biecek, P. n.d. Ceteris paribus plots. https://pbiecek.github.io/ceterisParibus/

Britton, M. (2019). Vine: Visualizing statistical interactions in black box models. *arXiv preprint*, arXiv:1904.00561.

Casalicchio, G., Molnar, C., & Bischl, B. (2019). Visualizing the feature importance for black box models. In *Machine learning and knowledge discovery in databases* (pp. 655–670). Springer International Publishing. https://doi.org/10.1007/978-3-030-10925-7_40

Craven, M. W. (1996). *Extracting comprehensible models from trained neural networks*. The University of Wisconsin-Madison.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77. https://doi.org/10.1145/3359786

Evans, B. P., Xue, B., & Zhang, M. (2019). What's inside the black-box? A genetic programming method for interpreting complex machine learning models. In *Proceedings of the genetic and evolutionary computation conference, GECCO'19, Association for Computing Machinery* (pp. 1012–1020). ACM Press. https://doi.org/10.1145/3321707.3321726

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, *24*(1), 44–65. https://doi.org/10.1080/10618600.2014.907095

Greenwell, B., Boehmke, B., Cunningham, J., Developers, G.. (2020). *gbm: Generalized Boosted Regression Models, r package version 2.1.8*. The R Project. https://CRAN.R-project.org/package=gbm

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., & Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *arXiv preprint*, *51*(5), arXiv:1805.10820.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, *51*, 1–42. https://doi.org/10.1145/3236009

Hall, P., & Gill, N. (2019). *An introduction to machine learning interpretability* (2nd ed.). O'Reilly Media.

Hernández-Orallo, J. (2013). Roc curves for regression. *Pattern Recognition*, *46*(12), 3395–3411. https://doi.org/10.1016/j.patcog.2013.06.014

Hernández-Orallo, J., Flach, P., & Ferri, C. (2011). Brier curves: A new cost-based visualisation of classifier performance. In *Proceedings of the 28th international conference on machine learning, ICML* (pp. 585–592). AAAI Press.

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, *1*, 69–91. https://doi.org/10.1007/BF01898350

Jorge, A. M., Azevedo, P. J., & Pereira, F. (2006). Distribution rules with numeric attributes of interest. In *Knowledge discovery in databases: PKDD, LNAI* (Vol. 4213, pp. 247–258). Springer. https://doi.org/10.1007/11871637_26

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., & Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, *113*(523), 1094–1111. https://doi.org/10.1080/01621459.2017.1307116

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Metz, C. E. (1978). Basic principles of roc analysis. *Seminars in Nuclear Medicine*, *8*(4), 283–298. https://doi.org/10.1016/S0001-2998(78)80014-2

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2020). *e1071: Misc functions of the department of statistics, probability theory group. (Formerly: E1071), r package version 1.7-4*. TU Wien. https://CRAN.R-project.org/package=e1071

Parliament of the European Union. (2020). C.: Recital 71 - profiling | general data protection regulation (gdpr). https://gdpr-info.eu/recitals/no-71/

Parr, T., Turgutlu, K., Csiszar, C., Howard, J.. (2018). *Beware default random forest importances*. Explained-AI. https://explained.ai/rf-importance/

Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD'16*. p. 1135–1144. Association for Computing Machinery (2016). https://doi.org/10.1145/2939672.2939778

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI conference on artificial intelligence*, *32*(1), 1–9. https://doi.org/10.1609/aaai.v32i1.11491

Shapley, L. S. (1953). A value for n-person games. Contributions to the theory of games. *Annals of Mathematics Studies*, *28*, 307–318.

Staniak, M., & Biecek, P. (2019). Explanations of model predictions with live and breakdown packages. *The R Journal*, *10*(2), 395–409. https://doi.org/10.32614/RJ-2018-072

Team, R.C. (2020). *R: A language and environment for statistical computing. R foundation for statistical computing*, The R Project. https://www.r-project.org/

Torgo, L. (2005). Regression error characteristic surfaces. In *Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, KDD'05* (pp. 697–702). Association for Computing Machinery. https://doi.org/10.1145/1081870.1081959

Torgo, L., Azevedo, P., & Areosa, I. (2021). Beyond average performance—exploring regions of deviating performance for black box classification models. *arXiv preprint*, arXiv:2109.08216.

Turek, M.. (2018). *Explainable artificial intelligence (xai)*. DARPA. https://www.darpa.mil/program/explainable-artificial-intelligence

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics* (fourth ed.). Springer. http://www.stats.ox.ac.uk/pub/MASS4

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag. https://ggplot2.tidyverse.org

AUTHOR BIOGRAPHIES

João Pimentel was an MSc student at Department of Informatics at University of Minho, Portugal, doing his master thesis on subgroup mining applied to performance analysis on regression models. This coming year, he will be a machine learning PhD candidate at Dalhousie University, Halifax, Nova Scotia, Canada.

Paulo J. Azevedo got his PhD from Imperial College at University of London (1995) and is now an Assistant Professor (with habilitation) at the Department of Informatics University of Minho. He is a member of the High-Assurance Software Lab (HASLab)—INESC Tec L.A. His research interests are Knowledge Discovery in Databases (data mining) and Machine Learning, in particular association rules, subgroup discovery, motif discovery in time series, graph mining, bioinformatics and recommendation systems.

Luís Torgo is a Canada Research Chair (Tier 1) on Spatiotemporal Ocean Data Analytics and a Professor at the Faculty of Computer Science of the Dalhousie University. He also holds appointments as an Associate Professor of the Faculty of Sciences of the University of Porto, Portugal and as an invited professor of the Stern Business School of the New York University, USA. At Dalhousie, he is the Deputy Director of the Institute for Big Data Analytics and the current Head of the Research Cluster on Big Data Analytics, AI and Machine Learning. He is the leader of the Ocean Data Analytics lab, a senior researcher of LIAAD / INESC Tec (Porto, Portugal), and an associate member of the Artificial Intelligence Institute of the University of Waikato (New Zealand).

How to cite this article: Pimentel, J., Azevedo, P. J., & Torgo, L. (2023). Subgroup mining for performance analysis of regression models. *Expert Systems*, *40*(1), e13118. https://doi.org/10.1111/exsy.13118