Implementing local-explainability in Gradient Boosting Trees: Feature Contribution

Ángel Delgado-Panadero[a], Beatriz Hernández-Lorca[a], María Teresa García-Ordás[b,*], José Alberto Benítez-Andrades[c]

[a]*Machine Learning Engineer at Paradigma Digital, S.L.*
[b]*SECOMUCI Research Group, Escuela de Ingenierías Industrial e Informática, Universidad de León, Campus de Vegazana s/n, C.P. 24071 León, Spain*
[c]*SALBIS Research Group, Department of Electric, Systems and Automatics Engineering, Universidad de León, Campus of Vegazana s/n, León, 24071, León, Spain*

Abstract

Gradient Boost Decision Trees (GBDT) is a powerful additive model based on tree ensembles. Its nature makes GBDT a black-box model even though there are multiple explainable artificial intelligence (XAI) models obtaining information by reinterpreting the model globally and locally. Each tree of the ensemble is a transparent model itself but the final outcome is the result of a sum of these trees and it is not easy to clarify.

In this paper, a feature contribution method for GBDT is developed. The proposed method takes advantage of the GBDT architecture to calculate the contribution of each feature using the residue of each node. This algorithm allows to calculate the sequence of node decisions given a prediction.

Theoretical proofs and multiple experiments have been carried out to demonstrate the performance of our method which is not only a local explicability model for the GBDT algorithm but also a unique option that reflects GBDTs internal behavior. The proposal is aligned to the contribution of characteristics having impact in some artificial intelligence problems such as ethical analysis of Artificial Intelligence (AI) and comply with the new European laws such as the General Data Protection Regulation (GDPR) about the right to explain and nondiscrimination.

*Corresponding author

Email addresses: delgadopanadero@gmail.com (Ángel Delgado-Panadero), beahernandezlorca@gmail.com (Beatriz Hernández-Lorca), mgaro@unileon.es (María Teresa García-Ordás), jbena@unileon.es (José Alberto Benítez-Andrades)

Keywords: XAI, Gradient Boosting Trees, Explainable Artificial Intelligence

1. Introduction

In this paper, we focus on Explainable Artificial Intelligence (XAI) of Gradient Boosting models. Explainability methods can be divided into two big branches based on the scope of the method: global and local methods [1].

Global explanations refer to the trained model. They answer questions such as *How good is the model?*, *Which variables are important to the model?*, and *How does a variable affect the average prediction?* [2]. Quite a few global explainability techniques have been published in recent years ([3, 4, 5]). For example in the work developed by Agarwal et al. [6], Neural Additive Models (NAMs) are proposed. These networks combine some of the expressivity of Deep Neural Networks (DNNs) with the inherent intelligibility of generalized additive models. Both networks are trained jointly and can learn arbitrarily complex relationships between their input feature and the output. Returning to GBDTs, its global explanation is described by Gini importance or more commonly, Feature importance [7]. In the same way, in [8], the authors propose a method that augments global explanations with the proportion of samples that each attribution best explains and specifies which samples are described by each attribute.

The other branch is local explainability. In this work we focus on this type of algorithm. They have the ability to understand predictions by answering questions like *Which variables contribute to the selected prediction?*, *How does a variable affect the prediction?*, and *Does the model fit well around the prediction?* [2]. In recent years, there have been many methods that attempt to explain neural network decisions, but few focus on techniques such as GBDT [9, 10, 11, 12, 13].

Based on the usage, XAI methods can be divided into Post-hoc techniques and intrinsic techniques [14, 15]. Current XAI methods for GBDT are Post-hoc, this means that it is necessary to apply a transparent model that locally approximates the black-box model in a vecinity of the prediction point, and these are very useful and very used in different researchs [3, 16, 8, 17, 18]. They are model-agnostic methods, meaning they can be used in many artificial intelligence (AI) algorithms using Local Interpretable Model-Agnostic

Explanation (LIME) and Shapley values (SHAP) [1]. This also means that they are a reinterpretation of the model in the vicinity around the prediction, which translates into loss of model complexity, and accuracy in predictions.

On the other hand, intrinsic explainable methods [19, 20, 21] have interpretable elements baked into them. Intrinsic methods of explanations are inherently model-specific. This means that the explainer depends on the model architecture which cannot be re-used for other classifier architectures without designing the explanation algorithm specifically for the new architecture [15].

Gradient Boosted Decision Trees (GBDT) is a specific implementation of Boosting Machines [22] and one of the most powerful algorithms in Machine Learning. It is widely used in many fields and its rate of success is very high: healthcare [23, 24], education [25], energy [26], economics [27], etc. It has one big caveat: it is considered a black-box model. A black-box model describes a system in which the algorithm and predictions are not understood by a human just by looking at its parameters and features. Black-box models could suffer from lack of trust even if their performance is more than acceptable and they make it difficult to comply with General Data Protection Regulation (GDPR) articles about the right to explain and nondiscrimination [28].

Because of the importance of explicability in machine learning models and the success of GBDT as a predictive model, SHAP explicability have been used many applications and studies. However the SHAP explicability over the GBDT seem to have lack of accuracy and stability in their explicability values [29, 30, 31].

The rest of the paper is organized as follows: Section 2 presents the methodology including the proposed algorithm, and the mathematical proof. Experiments and results are detailed in section 3. Finally, section 4 shows the conclusions of our proposal.

2. Methodology

2.1. Background and motivation

In this paper we introduce **Decision Contribution**, a new algorithm for GBDT. This new method allows the exact sequence path of decisions of the trees to be calculated from the ensemble for a selected prediction. Roughly speaking, Decision Contribution reinterprets node values as the difference of each node value with its father value (or residues), so each tree result

is not the leaf node value but the sum of all the node residues from the sequence path. Each residue is an estimation of the decision influence in the final result. Any node has a decision related to a feature and a threshold therefore we can link a residue to a specific feature.

After that, **Feature Contribution** is defined as the sum of a feature residues. Given the nature of decision trees we can also retrieve information about thresholds and features involved in each node. This gives the whole picture of the path and decisions taken to get to a prediction. Reinterpreting nodes has little to no cost in the calculation of predictions.

To be able to explain and comprehend GBDT predictions is an important step towards model explainability. **Feature Contribution** is not only a local explainability technique. This method would also make GBDT locally explainable, ergo, it would not be a black-box algorithm anymore. It differs from XAI methods as it is an intrinsic algorithm linked to the GBDT architecture and the result is the exact contribution for each feature for a selected prediction. While model-agnostic XAI techniques cause loss of information and only give a reinterpretation of the contribution of features, Decision Contribution allows us to know the train of decisions of each prediction. This means we know the path chosen for the prediction in each tree of the ensemble, together with the feature and threshold.

2.2. Mathematical Proof

Below it is the theoretical proof of Decision Contribution. As mentioned in the introduction, this algorithm tries to estimate the influence of each decision in the final result. Thus it is not odd to prove the goodness of the estimation in terms of *decisions*.

Given a pair of random variables (Y, X) from an unknown probability distribution we define Y^k and X^k as a $i.i.d.$ samples from the probability distribution. The values of X can be geometrically represented as vector points in the n-dimensional feature vector space $x \in \mathbb{R}^n$.

The set of decisions of s_j is traditionally defined as an ordered sequence of evaluations for each node, checking whether the component x_j of a point x is greater or lesser than a certain threshold, th_j. We can also define each decision as intervals as follows:

$$s_j(\theta)|_{\theta=(k_j,th_j)} = \begin{cases} x \quad / \quad x_{k_j} \in (-\infty, th_j] & \text{if } x_{k_j} \leq th_j \\ x \quad / \quad x_{k_j} \in (th_j, \infty) & \text{if } x_{k_j} > th_j. \end{cases} \tag{1}$$

where s_j is the decision made from the father, Q_{j-1}, to create its son node. The sequence of s_j for $j = 0, 1, 2, .., i$ are all the decisions in order made by the tree to achieve the node Q_i. The index here does not represent the node index in the tree but the sequence of decisions followed to achieve that node. Each tree node, Q_i, is a region of space defined as follows

$$Q_i(\theta) = \bigcap_{j=0}^{i} s_j(\theta_j), \tag{2}$$

Given a X^k, the tree performs an ordered set of decisions to get the value of leaf node where X^k falls in. The value of the leaf node is computed during the training as the expected value of Y given X.

$$h(x;\theta) = \mathbb{E}(Y|X = x;\theta)|_{\theta/x\in Q_i(\theta)}. \tag{3}$$

The tree behaves as a predictor function $h(x)$, from a family of functions $h(x;\theta)$, which tries to estimate the distribution of $P(Y|X = x)$.

$$h(x;\theta_t) \quad / \quad \theta_t = \underset{\theta}{\operatorname{argmin}}\mathbb{E}(\mathcal{L}(Y, h(x;\theta))) \qquad x \in Q_i. \tag{4}$$

Following this idea and having the values of all the tree nodes, we can define for each node, Q_i an estimator, $h_{Qi}(x)$, of Y given a point x in the region of the space defined by Q_i, from a family of functions $h_{Qi}(x;\theta_i)$,

$$h_{Qi}(x;\theta_t) \quad / \quad \theta_t = \underset{\theta}{argmin}\mathbb{E}(\mathcal{L}(Y, h_{Qi}(x;\theta)) \qquad x \in Q_i. \tag{5}$$

where $\mathcal{L}(u, v)$ is a loss function and $h_{Qi}(x)$ is the predictor of the node Q_i. The function $h_{Qi}(x)$ is the expected value of Y too, but in the domain of x defined by the node Q_i. The difference between the prediction of Q_n from Q_m when evaluating a point x is

$$\mathbb{E}(Y|X = x')|_{x'\in Q_n} = \mathbb{E}(Y|X = x')|_{x'\in Q_m\cap s_n} \quad \forall x \in Q_n. \tag{6}$$

so

$$\begin{aligned} h_{Q_n}(x) - h_{Q_m}(x) = \\ \mathbb{E}_m(Y(\mathcal{H})|\mathcal{H} = s_n) - \mathbb{E}_m(Y) = \\ \mathbb{E}_m(Y(\mathcal{H}) - Y|\mathcal{H} = s_n) \qquad \forall x \in Q_n. \end{aligned} \tag{7}$$

where $\mathbb{E}_m(Y)$ is the expected value in the domain of Q_m and $Y(\mathcal{H})$ is the marginal distribution of Y given the condition $\mathcal{H}$. This expression is analogous to the information gain due to a split, which measures the split quality as the difference between the entropy, $S(Y)$, of the distribution before and after the split.

$$IG(s_n) = S(Y) - S(Y(\mathcal{H})|\mathcal{H} = s_n). \tag{8}$$

Following this idea we can conclude that the expected difference between the value of a node and its father is caused by the decision s_n. Given a father node, Q_m, and its decision s_n to achieve its son Q_n, we define the contribution of the decision to the final prediction as

$$g(s_n(x)) \coloneqq \mathbb{E}(Y(\mathcal{H}) - Y|\mathcal{H} = s_n)|_{x \in Q_m}, \tag{9}$$

where $g(s_n)$ is a function that returns the contribution in units of Y due to the decision s_n. The previous is only true in the domain of Q_m, however, assuming that features, X_i, are statistically independent, the margin distribution should be equal to the original distribution, so the conclusion is true in the whole feature space. This can be written mathematically as:

$$\mathbb{E}(Y(\mathcal{H}) - Y|\mathcal{H} = s_i) = \mathbb{E}(Y(\mathcal{H}', \mathcal{H}) - Y(\mathcal{H}')|\mathcal{H} = s_i) \quad \forall \mathcal{H}'. \tag{10}$$

So, we can conclude that the decision contribution of a certain decision is equal to the difference between the marginal distribution of Y before and after the current decision, independently of the previous decisions. This is:

$$g(s_i) = \mathbb{E}(Y(\mathcal{H}) - Y|\mathcal{H} = s_i). \tag{11}$$

2.2.1. Reinterpreting Gradient Boosting Decision Trees

The GBDT algorithm is defined iteratively as follows

$$F^t(x) = F^{t-1}(x) + \alpha h^t(x). \tag{12}$$

where $F^t(x)$ is the model at the training iteration t, α is the learning rate and $h^t(x)$ is the model trained from a base family model, $h(x;\theta)$, (in our case CART) trained to predict the residual differences of the previous model

$$h^t(x) = h(x;\theta_t) \quad / \quad \theta_t = \arg\min_{\theta} \mathbb{E}(\mathcal{L}([Y - F^{t-1}(x)], h(x;\theta))). \tag{13}$$

The function $F^0(x)$ has multiple definitions depending on the implementation, but commonly it is set as the prior probability or mean value of the distribution Y. The previous definition is very convenient in understanding the learning process of $h^t(x)$, nevertheless, it is exactly the same to

$$F^t(x) = \sum_{l=0}^{t} \alpha h^l(x). \tag{14}$$

Predictions made by a decision tree are defined as follows

$$h(x;\theta) = \mathbb{E}(Y|X = x;\theta)|_{\theta/x\in Q_i(\theta)}. \tag{15}$$

Without any change in the final result, we can overwrite predictions as follows

$$h^l(x) = \sum_{j=0}^{i(l)} g^l(s_j) \quad / \quad x \in s_j, \tag{16}$$

where l is the index of the tree from the GBDT ensemble, $i(l)$ is the number of nodes used by the tree l in the decision of x and j is the index of the sequence of decisions in the prediction for x. So, the prediction is not expressed as final node value but as the addition of all deviations of each node from the previous node from the first node to the latest. This does not change any tree result, but it does allow the prediction to be written as a sum.

With the definition of the CART function as the sum of decision contributions and the definition of the GBDT as the addition of models, we can express the model as follows

$$F^t(x) = \sum_{l=0}^{t} \sum_{j=0}^{i(l)} \alpha g^l(s_j) \quad / \quad x \in s_j, \tag{17}$$

where $g^l(s_j)$ is the contribution of the decision s_j made by the tree h^l. This proves that the GBDT algorithm can be expressed as the sum terms in which each one is the deviation on the expected value of the margin from the previous expected value as the node decision is the margin condition.

2.3. Implementation

The proposed Decision Contribution procedure consist of a non-parametric algorithm which estimates, for every decision from every tree of a trained GBDT, how much influence it has on the final prediction.

Algorithm 1: GBDT Decision Contribution

Input : trainedGBDT, X_i
Output: decisions, contributions

$decisions = [\ ]$
$contributions = [\ ]$
for *each tree* $\in$ *trainedGBDT* **do**
    $nodes = tree.$`decisionPath`(X_i)
    $currentNode =$ `getRootNode`$(nodes)$
    $previousValue = 0$
    while `isNotLeaf`$(currentNode)$ **do**
        `// Add node information`
        $decisions.append(currentNode.$`decision`$)$
        $contribution.append(currentNode.$`value`$ - previousValue)$

        `// Update loop variables`
        $previousValue = currentNode.$`value`
        $currentNode =$ `getSonNode`$(currentNode, X_i)$
    end while
end for
return *decisions, contributions*

Following the terminology from CART [7] trees are built by a set of hierarchic binary splits in which each split is made using a certain feature which results in a node. The split is made by trying to optimize a certain loss function of the dependent variable. The nodes at which the tree stop splitting are called leaf nodes. Consequently, for every node we have a value that estimates the independent variable in the node and, if it is not a leaf node, the split will lead the two son nodes (left and right nodes).

An example of a node split can be shown in figure 1.

The Decision Contribution procedure has two main steps. The first step is to extract from every tree, all the nodes through which a given sample passes. Depending on the feature sample values, the sample falls on one side

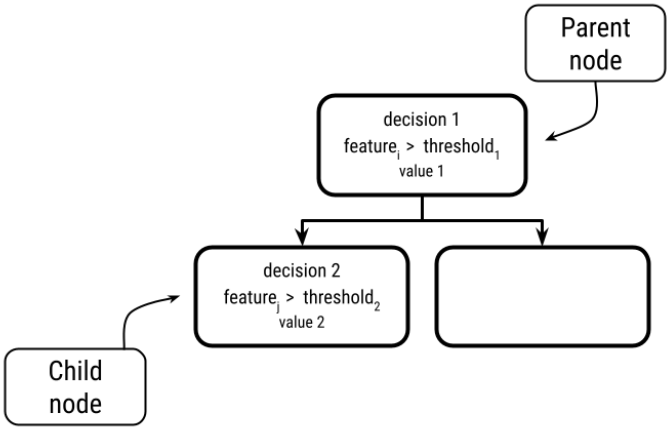

Figure 1: Example of a node split in the tree structure. Every node has an average value of the target and a decision based on a certain feature and threshold that splits the feature space into two son nodes.

of the split or the other and this will lead to only one of the son nodes. The set of nodes through which a sample passes results in a prediction defining a path and the decisions from those nodes are the only decisions that convert the data into a certain result.

Having all the nodes of the decision path through which the sample passes for every tree, the second step is to recover the decision that defined this node (i.e. the father node split) and the values of each node from the decision path and of its father. With this, for each decision we define a contribution value to the final result which is the difference between the node value and its father's node value.

The Decision Contribution procedure consists of assigning a contribution value for every decision which measures the effect that it has in the final result. To summarize this information, it can be aggregated in two different ways:

2.3.1. Decision Space

Beforehand, the contributions for a sample prediction only gives information about that prediction, moreover,in the set of decisions made by a tree to achieve a final result, many of them can be redundant because one decision

can be a subset of a future decision from the tree.

One option to aggregate this information is to compute the intersection spaces defined by all the splits from the nodes of the decision path. This will remove the redundant decision splits and will define a region in the feature space. This region is the region assigned to a leaf of the tree. Without any loss of information, we can assigning the set of contribution not only to the sample prediction but also to all the samples that fall in the same region.

2.3.2. Feature Contribution

The other option is to total the contribution from all of the decisions which use the same feature in the splitting. From this approach we can measure the influence that each feature has on the final result for a certain sample.

This approach is pretty similar to the Gini importance algorithm proposed for tree ensembles to give global explicability which assigns the global contribution of each feature which totals the loss gained in each split made by each feature. The difference from the Gini importance algorithm is that it tries to answer the question "Which feature split better the data?" while in a local explicability method our goal is to answer "How much each feature contributes to the final result?".

For that reason we require to calculate how much the prediction is updated after each decision over a certain feature. That is exactly how we have defined each decision contribution. Consequently, the sum of all of the contributions over the decisions from the same feature can be interpreted as the measure of the influence of the influence of that feature in the final result.

3. Experiments and results

3.1. Datasets

The following two datasets has been used in this paper in order to test the proposed method:

3.1.1. Diabetes dataset

The diabetes dataset was presented in [32] Least Angle Regression paper. This dataset contains ten baseline variables obtained for each of the n = 442 diabetes patients, as well as the response of interest, a quantitative measure

of disease progression one year after the baseline[1]. Attribute information [2] can be found in table 3.1.1 .

Table 1: Attribute information

Acronym	**Description**
age	age in years
sex	sex
BMI	Body Mass Index
bp	Average Blood Pressure
s1	tc, Total Serum Cholesterol
s2	ldl, Low-Density Lipoproteins
s3	hdl, High-Density Lipoproteins
s4	tch, Total Cholesterol / HDL
s5	ltg, Possibly Log of Serum Triglycerides Level
s6	glu, Blood Sugar Level

3.1.2. Concrete dataset

The Concrete dataset has used in [33] for modeling concrete strength with seven factors: water/cement ratio, water, cement, fine aggregate, coarse aggregate, maximum grain size, and age of testing.

It consist on n = 1030 samples with eight numerical independent features of the concrete and a dependent column with a numerical measure of the strength of the concrete sample.

3.2. Experimental Setup

In this section, two sets of experiments have been introducted to demonstrate that the method proposed works as an explainer of the decision made by the tree during the inference at a feature level. To do so we are going to focus on the Feature Contribution.

Both sets of experiments are tested over the datasets described in section 3.1, the diabetes and the concrete dataset.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html
[2] https://scikit-learn.org/stable/datasets/toy_dataset.html#diabetes-dataset

The first set of experiments tests model contribution proposed in this research in different situations (correlated features and noise) and checks its behaviour when bias is introduced to the training dataset. The second group of experiments compare model contribution proposed in this research with other XAI methods, specifically SHAP values [34] and Lime [35] to understand their similarities as well as their differences. It is important to note that every XAI intrinsic technique applied to an algorithm depends on the proper architecture of the algorithm itself. There's no other intrinsic method for GBDT so comparing our proposed method with other intrinsic algorithms is out of scope for this paper. Our experiments compare our proposal with other extrinsic methods due to this drawback.

For all these tests scikit-learn library implementation of the Gradient Boosting has been used. The training parameters, when not specified, have been set as default using the implementation from scikit-learn[3]. A splitting over each dataset is performed, the training set contains 90% of the samples. The aim of these experiments is not accuracy, but to show the feature contribution behaviour on predictions over the test set.

Each result is shown in a plot. The x axis shows the experiment observation and the y axis shows the contribution measured by the proposed algorithm Feature Contribution. The total contribution of a feature is the sum its residues.

3.3. Model Consistency Test

In the first lot of experiments, behaviour under correlation and behaviour under noise are evaluated.

Main goal of this research with these experiments is to prove the Feature Contribution is in line with the model behaviour. Changing the training set, the trained model will change and the feature contribution will be expected to change too. This situation is due to the definition of decision contribution which is intrinsic to the model.

3.3.1. Behaviour under correlation

In this experiment, it has been possible to see the behaviour of the proposed feature contribution model on test data when the tree structure

[3]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

changes because of a leak and noise in the training data. The expected behaviour is that the proposed feature contribution model reflects the changes in the trees, whatever they are or even if these changes are biased.

To do so, a new synthetic, highly-correlated feature is created as the result of a random factor multiplication and constant addition of another feature (referred as base feature). Then a comparison of the new contribution versus the original one is carried out. To demonstrate the dependence of this result with the intrinsic randomness of the trees, this experiment is carried out for five different random states initializations. This shows how the contribution over the test set changes.

To create the new feature, the variable with the greatest impact on the result is used as base feature: (*BMI* for the diabetes dataset and *Age* for the concrete dataset). This impact has been measured with the Feature Importance [7] implemented in the scikit-learn library.

This experiment has been carried out by doing 5 different tests using a Gradient Boosting model built with 10 trees (n_estimators=10^4) with different random states. Then those are compared with the same model and the original dataset. Each experiment is represented in a subplot in Fig. 3 and Fig. 2.

Figures below show the final contribution of each feature in the predictions over the test set, i.e., the sum of residues of each feature affecting predictions. The feature contribution can be either positive or negative, depending how the feature in case affects the final prediction.

Fig. 2 and Fig. 3 show the expected behaviour: the sum contributions of the new feature with the base feature is the same as the original one meaning these two variables are the same. In terms of trees, this is because both variables perform the same splitting performance so the splitting criterion chooses any of them exactly when it used the original variable and obtains the same result.

On the other hand, figures show that the contribution of both, base feature and correlated variable is not exactly the same. This may seem unreasonable however it is because the random splitting nature of the trees. As both features split the data equally, the tree chooses one randomly.

[4]The model works equally good with more trees, however we have just used 10 trees because the effect of the feature contribution in these experiments, is more difficult to see with more trees due to the appearance of other effects such as overfitting, model bias,...

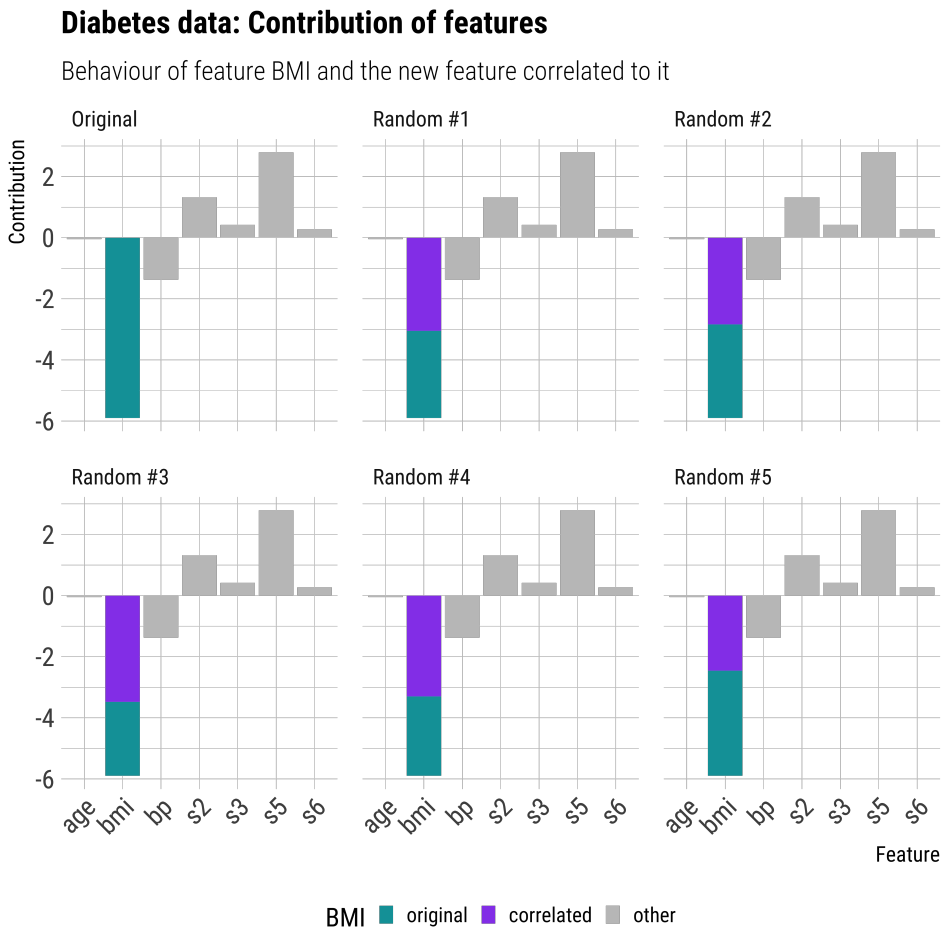

Figure 2: Contribution of the new feature, *correlated*, vs contribution of the original base feature, *BMI*. Only showing features which contribute to the final prediction.

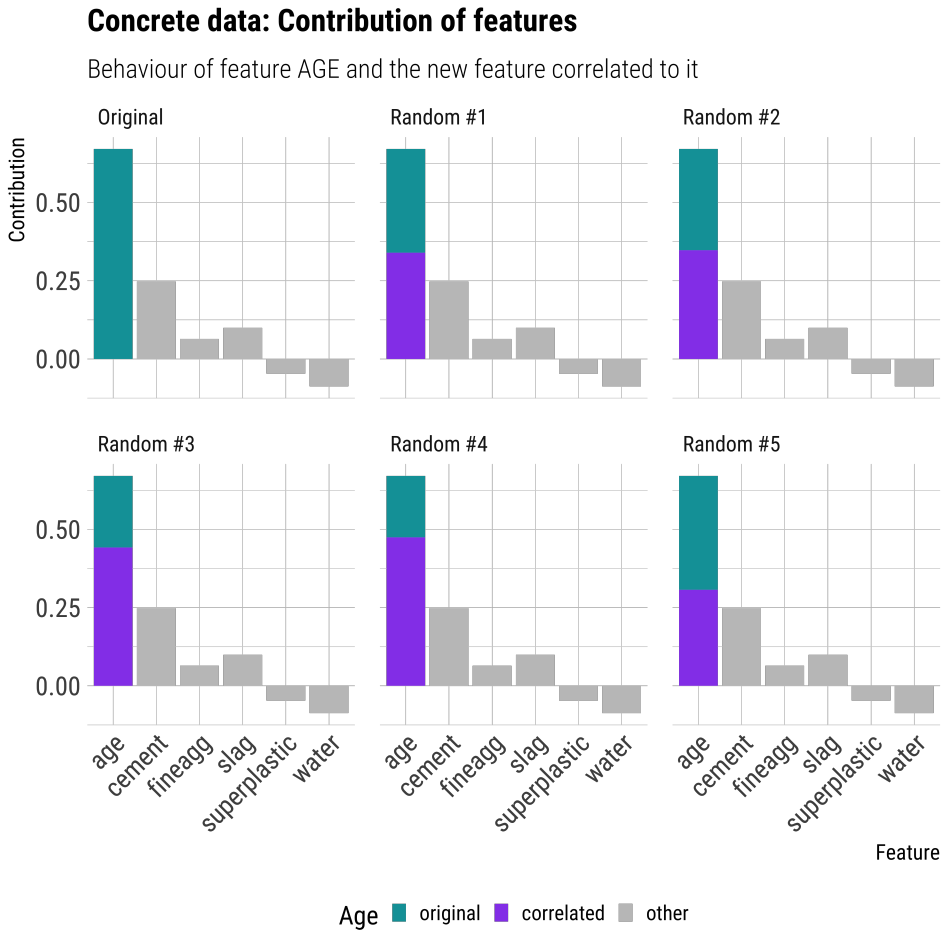

Figure 3: Contribution of the new feature, *correlated*, vs contribution of the original base feature, *age*. Only showing features which contribute to the final prediction.

This randomness when choosing a feature during the model training has direct implications during the model inference. If the distributions of the variables stopped being so highly correlated during the inference, the trees would have a random bias due to which one of these two variables is chosen in each split. This can be measured with the proposed method.

3.3.2. Behaviour under noise

In a consistent explicability model, the impact of noise should be noticed only in the loss of contribution and only in the feature in which the noise is introduced. However in this experiment, it is demonstrated that noise has an impact on the contribution of other features. This behaviour makes sense because of the nature of tree models. The goal of this experiment is to show the inherent behaviour to the GBDT model architecture.

To prove it, we will measure how the contribution of a feature degrades during the inference when random Gaussian noise is added to that feature. The mean of noise distribution is zero and the variance a percentage of the original distribution variance from that feature. Feature contribution is measured in the same inference predictions for different percentages of variance noise. Figures show how much the feature contribution changes from one percentage to another.

Following the same criteria as from the previous section, *BMI* and *age* features (for Diabetes and Concrete datasets respectively) are selected to add noise to it (for the same reason as in the previous experiment).

The training hyperparameters for the model are 10 trees and a depth of 2 (n_estimators=10, max_depth=2). The reason behind this decision is that with higher depth the results are less clear because of the overfitting over the noise added.

Figures below show the mean contribution of each feature in the predictions over the test set, as the previous experiment. The feature contribution can be either positive or negative, depending how the feature in case affects the final prediction. Each subplot shows the contribution of a feature, first when no noise is added and the rest the changes on the contribution after noise is added to the chosen feature for each dataset.

In Fig. 4 and Fig. 5 features which have contributed to predictions with different levels of noise (100%, 200%, 300% and 400%) are shown in subplots. Each column is the percentage of standard deviation from the original feature used as standard deviation of the noise distribution. It can be seen that feature contribution tends to decrease on the base feature as

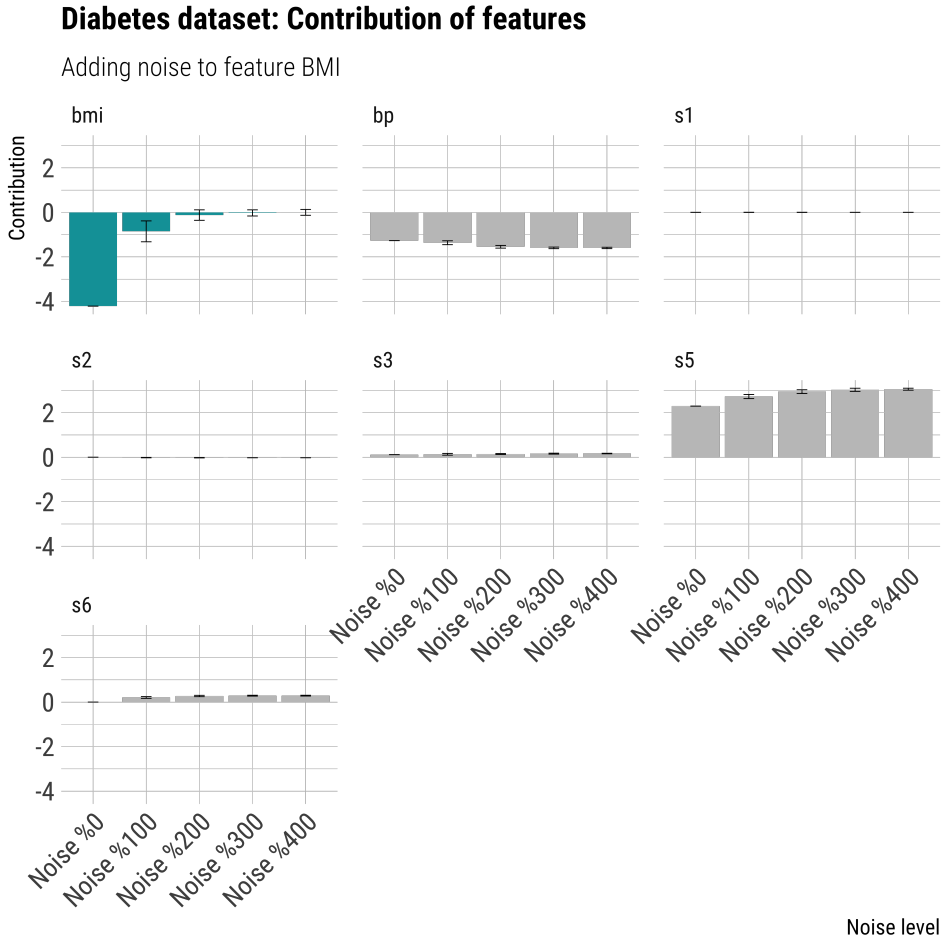

Figure 4: Feature contribution representation under different noise levels (100%, 200%, 300% and 400%) induced to BMI. As BMI looses influence on prediction the rest of the variables contribute more.

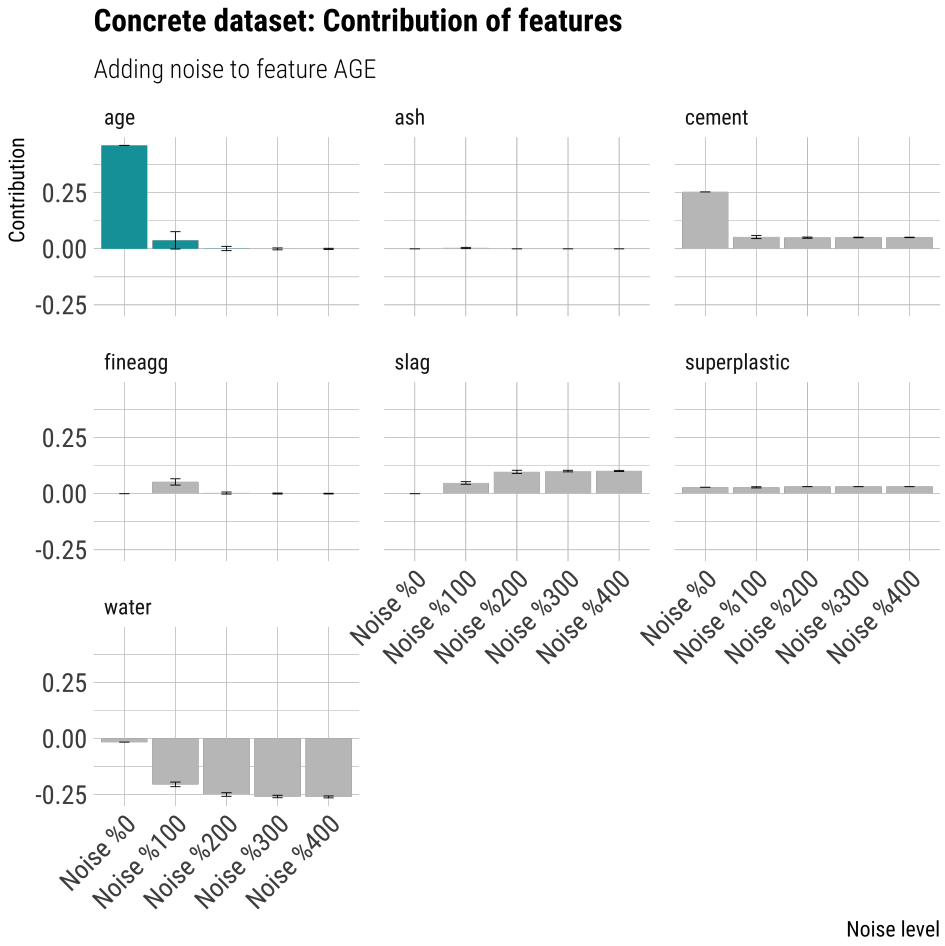

Figure 5: Feature contribution representation under different noise levels (100%, 200%, 300% and 400%) induced to age. As age looses influence on prediction the rest of the variables contribute more.

the level of noise increases, however, this decrease is not constant. Moreover, the noise does not only affect the variable to which noise is added, but also the others. They change not only its contribution value but also the order or influence and even choosing new features.

This effect can be understood directly by noting that under certain levels of noise we begin to see a non-zero contribution from features that were not even considered in other levels of noise. This can be only because the tree has not considered them during the training, so we are measuring a direct effect from the tree.

One of the reasons for this behaviour, is that the random noise can have a greater effect on different nodes of the tree and according to which node it happens in, it will use a new feature or not.

3.4. Comparative test

In this second set of experiments the proposed contribution method has been evaluated in relation to Lime [35] and SHAP [34] XAI algorithms applied to the Gradient Boosting. These methods have been chosen because of its relevance and use in local explicability.

In the 3.4.1 experiment, the goal is to see similarities and differences between methods, taken into account that SHAP is a reference in explicability. In 3.4.2, a prediction of an outlier is introduced and results are discussed.

3.4.1. Explicability comparison

In the experiment, the proposal of this research is compared with the results given by two of the most widely used algorithm for explicability: SHAP and Lime. First, an analysis of how similar they are overall when the test sample is carried out. The test has been made using 10 trees (n_estimators=10)[5]. The choice of this depth is because higher depth tend to overfit in this dataset.

The y axis shows the contribution of each feature, the SHAP method shows how much each feature is pushing the model output from the base value, which is essentially the same as the residues do in each node decision.

Fig. 6 and Fig. 7 show that the proposal gives similar results to the SHAP model and disagrees with the Lime model. This is because the assumption of

[5]The model works equally good with more trees, however we have just used 10 trees because the effect of the feature contribution in these experiments, is more difficult to see with more trees due to the appearance of other effects such as overfitting, model bias,...

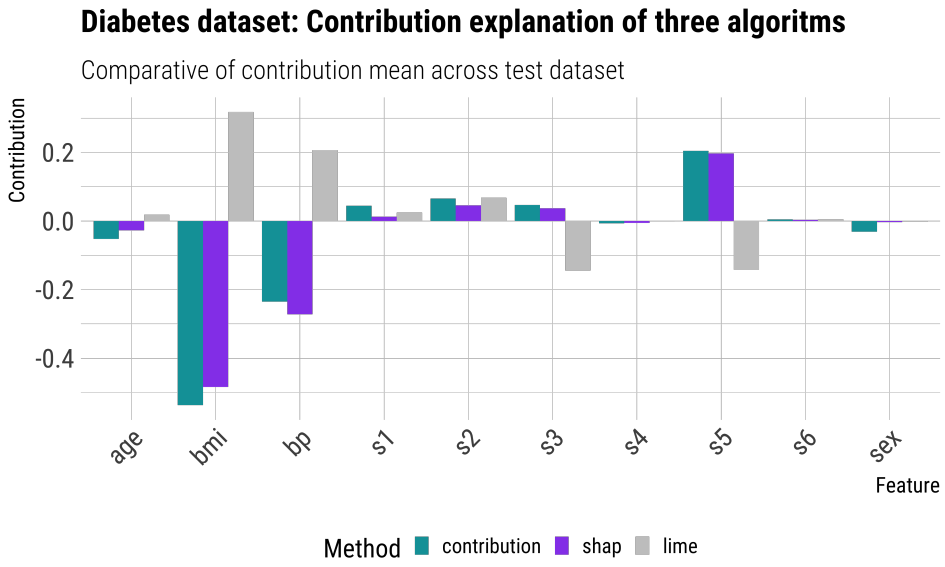

Figure 6: Comparative results between the proposal of this research (Feature Contribution), SHAP and Lime algorithms. Image shows similarities between the feature explanation of SHAP and the method proposed.

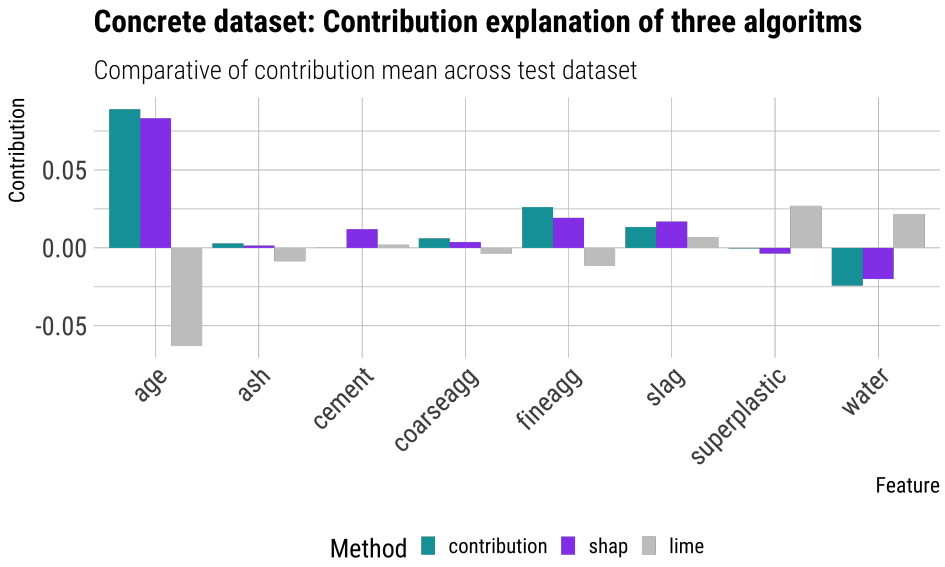

Figure 7: Comparative results between the proposal of this research (Feature Contribution), SHAP and Lime algorithms. Image shows similarities between the feature explanation of SHAP and the method proposed.

linearity made by Lime is not generally true and contributes bias especially in areas of the feature space in which there is a high variance. On the other hand, the contribution model and the SHAP model are expected to behave properly in areas of the feature space in which the linear assumption is not correct.

3.4.2. Outlier performance

As demonstrated in the previous experiment, the contribution model works quite similar to the SHAP model on a representative sample so the outlier performance will be only over the Feature Contribution and SHAP methods. In this experiment, the comparison is made over non-representative samples (outliers)to see the capability of explanation of both algorithms. We expect the contribution model to make a better explanation than SHAP because it follows just the decision process of the outliers sample rather than taking all the tree paths into account as SHAP.

For this experiment, an artificial sample register is created (X_fake and y_fake). This is intended to be a normal register except for a manipulation in a certain feature, specifically the first feature of each dataset (age and cement for diabetes and concrete datasets respectively), which seem to be symmetrically distributed, with finite variance and the features with the least impact in the final result.

This register, X_fake, consist of the mean of each feature from the training set except for the first feature, which is updated to have a value of one standard deviation greater than the greater value of this feature from the training set. The label, y_fake, is set to be equal to the greater value of the training sample of the target feature plus its standard deviation.

In this case, we have also used 10 trees (n_estimator=10) and deeper trees (max_deep=15) because in a non-deep tree, the explicability models tend to agree between themselves.

Both models seem similar and both fail to assign the contribution of the value to the specific features (age and cement respectively), which are the only ones different from the mean (see Fig. 8 and Fig. 9). The reason is that because of the residual nature of the GBDT ensemble, not even with an obvious separation of an outlier, all the trees performs this split at first.

However, the proposed model performs better than the SHAP value, because the contribution in the mentioned features is clearly higher and all the other feature contributions are near to zero except those which contribute more and even so the contribution is less from SHAP model too. With this

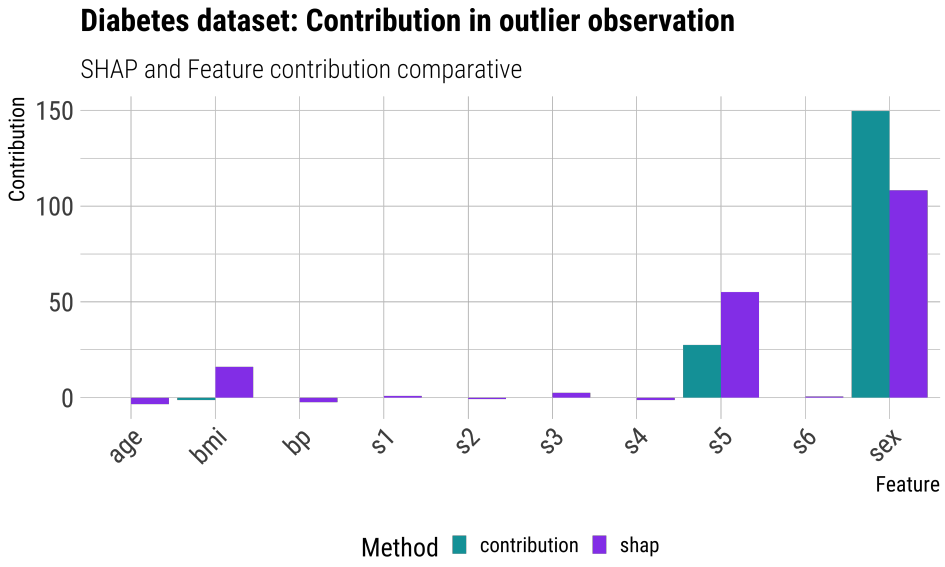

Figure 8: SHAP and Feature contribution comparison over non representative samples.

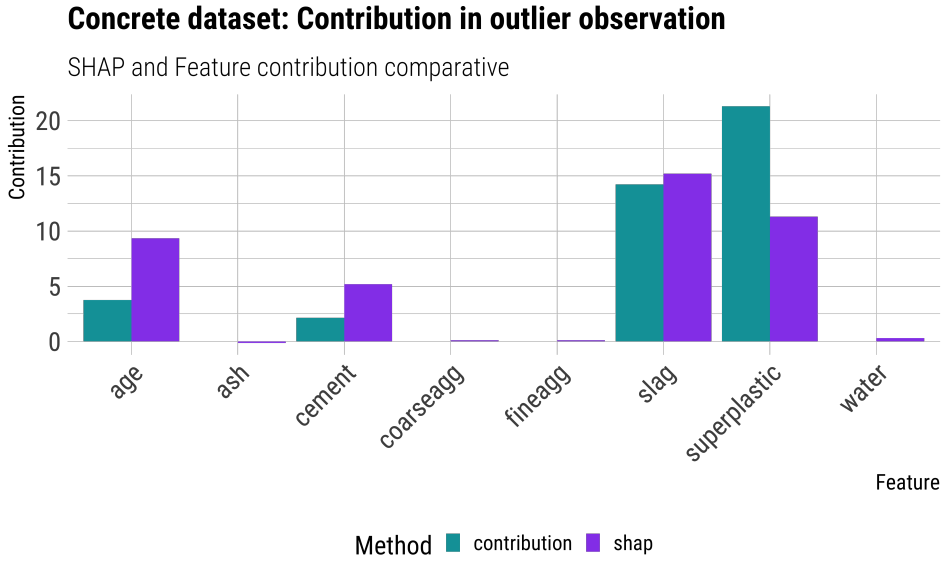

Figure 9: SHAP and Feature contribution comparison over non representative samples.

we can conclude that our contribution model not only reflects better the nature of the trees, but also, performs better showing exceptional rules due to outliers.

3.5. Experiments results

Tests have shown that the proposed explainability algorithm behaves as would be expected from an explainability function. In the consistency test the feature contribution values decrease their absolute value when adding noise to the feature or creating a correlated feature in the dataset. In the comparative tests we have seen the feature contribution values are similar to SHAP values in the general case but feature contribution values reflect better the contribution of outlier features than the SHAP values.

It is importance to understand that the feature contribution values does not reflect the real importance of a feature to the target but rather they just reflect what have the model learned in the trained process for a given trained model. In fact, the feature contribution values are related to the Gini importance values, but can be different in some scenarios because the Gini importance values reflect the loss decrease during the training process meanwhile the feature contribution values reflect the decision process of the model trained.

The feature contribution algorithm shows that it is possible to extract reasoning insights from the hierarchical structure of the trees and their node values. This opens new researching lines about adding explicability to other three based algorithms: Decision Trees, Random Forests,... and even to see the tree models outputs not just as the leaf values but rather as a sequence of connected nodes with addictive structure of their values.

4. Conclusions

In this paper we have proposed a local explicability interpretation for the GBDT based on expressing the tree prediction as the sum of contributions from each node. We have demonstrated theoretically that this novel approach is true under some assumptions which are also made by the GBDT themselves when computing the loss gain of each decision from the CART tree and the global explicability using the Gini importance from the tree ensembles algorithm.

We have also show empirically that this approach reflects better the nature of the GBDT than any general explainable model. All the other local

explicability models tend to be affected by decisions that do not actually affect the desired prediction, meanwhile the Decision Contribution algorithm only takes the decisions that really affect the sample into account. We have also seen that the only method that captures the influence of the stochastic nature of the trees in the final prediction at a feature level is also our proposed Feature Contribution algorithm.

Because of the theoretical demonstration as well as the empirical results we can conclude that our proposal is not only a local explicability model for the GBDT algorithm but also the only option that reflects the internal behavior.

Data availability

The source code that support the findings of this study are available in GitHub with the identifier https://doi.org/10.5281/zenodo.5566814

Acknowledgements

Special thanks to Alberto Rodríguez y Sara San Luís for giving us feedback after reading this manuscript from its initial stage, allowing us to improve different aspects of it.

References

[1] A. V. Konstantinov, L. V. Utkin, Interpretable machine learning with an ensemble of gradient boosting machines, arXiv 222 (2020) 106993. arXiv:2010.07388, doi:10.1016/j.knosys.2021.106993.
URL `https://doi.org/10.1016/j.knosys.2021.106993`

[2] P. Biecek, T. Burzykowski, Explanatory Model Analysis, Chapman and Hall/CRC, New York, 2021.
URL `https://pbiecek.github.io/ema/`

[3] Y. Goyal, A. Feder, U. Shalit, B. Kim, Explaining Classifiers with Causal Concept Effect (CaCE), arXiv (jul 2019). arXiv:1907.07165.
URL `http://arxiv.org/abs/1907.07165`

[4] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K. R. Müller, Unmasking Clever Hans predictors and assessing what

machines really learn, Nature Communications 10 (1) (2019) 1–8. arXiv:1902.10178, doi:10.1038/s41467-019-08987-4.
URL https://doi.org/10.1038/s41467-019-08987-4

[5] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), 35th International Conference on Machine Learning, ICML 2018 6 (2017) 4186–4195. arXiv:1711.11279.
URL http://arxiv.org/abs/1711.11279

[6] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, G. E. Hinton, Neural Additive Models: Interpretable Machine Learning with Neural Nets, arXiv (apr 2020). arXiv:2004.13912.
URL http://arxiv.org/abs/2004.13912

[7] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Wadsworth International Group, 1984.

[8] M. Ibrahim, M. Louie, C. Modarres, J. Paisley, Global Explanations of Neural Networks: Mapping the Landscape of Predictions, AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (2019) 279–287arXiv:1902.02384.
URL http://arxiv.org/abs/1902.02384

[9] V. Petsiuk, A. Das, K. Saenko, RISE: Randomized Input Sampling for Explanation of Black-box Models, arXiv (jun 2018). arXiv:1806.07421.
URL http://arxiv.org/abs/1806.07421

[10] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, S. Dähne, Learning how to explain neural networks: PatternNet and PatternAttribution, arXiv (may 2017). arXiv:1705.05598.
URL http://arxiv.org/abs/1705.05598

[11] A. Chattopadhay, A. Sarkar, P. Howlader, V. N. Balasubramanian, Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks, in: Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Vol. 2018-Janua, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 839–847. doi:10.1109/WACV.2018.00097.

[12] M. Sundararajan, A. Taly, Q. Yan, Axiomatic Attribution for Deep Networks, 34th International Conference on Machine Learning, ICML 2017 7 (2017) 5109–5118. arXiv:1703.01365.
URL http://arxiv.org/abs/1703.01365

[13] M. Ancona, E. Ceolini, C. Öztireli, M. Gross, Towards better understanding of gradient-based attribution methods for Deep Neural Networks, arXiv (nov 2017). arXiv:1711.06104.
URL http://arxiv.org/abs/1711.06104

[14] J. van der Waa, E. Nieuwburg, A. Cremers, M. Neerincx, Evaluating XAI: A comparison of rule-based and example-based explanations, Artificial Intelligence 291 (2021) 103404. doi:10.1016/j.artint.2020.103404.
URL https://doi.org/10.1016/j.artint.2020.103404

[15] A. Das, G. Student Member, P. Rad, S. Member, Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey, Tech. rep. arXiv:2006.11371v2.

[16] A. Ghorbani, J. Wexler Google Brain, J. Zou, B. Kim Google Brain, Towards Automatic Concept-based Explanations, Tech. rep. (2019).
URL https://github.com/amiratag/ACE

[17] C. Burns, J. Thomason, W. Tansey, Interpreting Black Box Models via Hypothesis Testing, FODS 2020 - Proceedings of the 2020 ACM-IMS Foundations of Data Science Conference (2019) 47–57arXiv:1904.00045, doi:10.1145/3412815.3416889.
URL http://arxiv.org/abs/1904.00045 http://dx.doi.org/10.1145/3412815.3416889

[18] H. Li, Y. Tian, K. Mueller, X. Chen, Beyond saliency: Understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation, Image and Vision Computing 83-84 (2019) 70–86. arXiv:1712.08268, doi:10.1016/j.imavis.2019.02.005.
URL https://github.com/Hey1Li/Salient-Relevance-Propagation.

[19] R. Caruana, Y. Lou, J. G. Microsoft, P. Koch, M. Sturm, N. Elhadad, 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,

ACM, New York, NY, USA.
URL http://dx.doi.org/10.1145/2783258.2788613

[20] V. Schetinin, J. E. Fieldsend, D. Partridge, T. J. Coats, W. J. Krzanowski, R. M. Everson, T. C. Bailey, A. Hernandez, Confident interpretation of Bayesian decision tree ensembles for clinical applications, IEEE Transactions on Information Technology in Biomedicine 11 (3) (2007) 312–319. doi:10.1109/TITB.2006.880553.

[21] L. Grosenick, S. Greer, B. Knutson, Interpretable Classifiers for fMRI Improve Prediction of Purchases, IEEE Transactions on Neural Systems and Rehabilitation Engineering 16 (6) (2008) 539–548. doi:10.1109/TNSRE.2008.926701.
URL https://pubmed.ncbi.nlm.nih.gov/19144586/

[22] J. H. Friedman, Greedy function approximation: A gradient boosting machine, Annals of Statistics 29 (2000) 1189–1232.

[23] M. A. Morid, K. Kawamoto, T. Ault, J. Dorius, S. Abdelrahman, Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation, AMIA ... Annual Symposium proceedings. AMIA Symposium 2017 (2018) 1312–1321.
URL https://pubmed.ncbi.nlm.nih.gov/29854200 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977561/

[24] H. Yang, Y. Luo, X. Ren, M. Wu, X. He, B. Peng, K. Deng, D. Yan, H. Tang, H. Lin, Risk Prediction of Diabetes: Big data mining with fusion of multifarious physical examination indicators, Information Fusion (2021). doi:https://doi.org/10.1016/j.inffus.2021.02.015.
URL https://www.sciencedirect.com/science/article/pii/S1566253521000397

[25] K. F. Hew, X. Hu, C. Qiao, Y. Tang, What predicts student satisfaction with moocs: A gradient boosting trees supervised machine learning and sentiment analysis approach, Computers & Education 145 (2020) 103724. doi:https://doi.org/10.1016/j.compedu.2019.103724.
URL https://www.sciencedirect.com/science/article/pii/S0360131519302775

[26] H. Lu, F. Cheng, X. Ma, G. Hu, Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower, Energy 203 (2020) 117756.

doi:https://doi.org/10.1016/j.energy.2020.117756.
URL https://www.sciencedirect.com/science/article/pii/S036054422030863X

[27] P. Carmona, F. Climent, A. Momparler, Predicting failure in the U.S. banking sector: An extreme gradient boosting approach, International Review of Economics& Finance 61 (2019) 304–323. doi:https://doi.org/10.1016/j.iref.2018.03.008.
URL https://www.sciencedirect.com/science/article/pii/S1059056017306950

[28] B. Goodman, S. Flaxman, European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation", AI Magazine 38 (3) (2017) 50–57. doi:10.1609/aimag.v38i3.2741.
URL https://ojs.aaai.org/index.php/aimagazine/article/view/2741

[29] A. Yasodhara, A. Asgarian, D. Huang, P. Sobhani, On the trustworthiness of tree ensemble explainability methods, Machine Learning and Knowledge Extraction (2021) 293–308doi:10.1007/978-3-030-84060-0_19.
URL http://dx.doi.org/10.1007/978-3-030-84060-0_19

[30] A. S. Bakouregui, H. M. Mohamed, A. Yahia, B. Benmokrane, Explainable extreme gradient boosting tree-based prediction of load-carrying capacity of frp-rc columns, Engineering Structures 245 (2021) 112836. doi:https://doi.org/10.1016/j.engstruct.2021.112836.
URL https://www.sciencedirect.com/science/article/pii/S014102962100986X

[31] G. Alicioglu, B. Sun, A survey of visual analytics for explainable artificial intelligence methods, Computers & Graphics (2021). doi:https://doi.org/10.1016/j.cag.2021.09.002.
URL https://www.sciencedirect.com/science/article/pii/S0097849321001886

[32] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of Statistics 32 (2) (2004) 407–451.

[33] I.-C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cement and Concrete Research 28 (12) (1998) 1797–1808. doi:https://doi.org/10.1016/S0008-8846(98)00165-3.
URL https://www.sciencedirect.com/science/article/pii/S0008884698001653

[34] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations

to global understanding with explainable AI for trees, Nature Machine Intelligence 2 (1) (2020) 56–67. doi:10.1038/s42256-019-0138-9.
URL http://www.nature.com/articles/s42256-019-0138-9

[35] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should i trust you?" Explaining the predictions of any classifier, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 13-17-August-2016, Association for Computing Machinery, 2016, pp. 1135–1144. arXiv:1602.04938, doi:10.1145/2939672.2939778.
URL http://dx.doi.org/10.1145/2939672.2939778