Applied Soft Computing Journal 180 (2025) 113300

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Metric tools for sensitivity analysis with applications to neural networks

Jaime Pizarroso *, David Alfaya [1], José Portela, Antonio Muñoz

Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas, Alberto Aguilera 23, 28015, Madrid, Spain

ARTICLE INFO

Keywords:
Sensitivity
Machine learning
Feature importance
Explainable AI
Regression
Feature engineering
Neural networks

ABSTRACT

As Machine Learning models are considered for autonomous decisions with significant social impact, the need to understand how these models work rises rapidly. Explainable Artificial Intelligence (XAI) aims to provide interpretations for predictions made by Machine Learning models, in order to make the model trustworthy and more transparent for the user. For example, selecting relevant input variables for the problem directly impacts the model's ability to learn and make accurate predictions. One of the main XAI techniques to obtain input variable importance is the sensitivity analysis based on partial derivatives. However, existing literature of this method provides no justification of the aggregation metrics used to retrieved information from the partial derivatives. In this paper, a theoretical framework is proposed to study sensitivities of ML models using metric techniques. From this metric interpretation, a complete family of new quantitative metrics called α-curves is extracted. These α-curves provide information with greater depth on the importance of the input variables for a machine learning model than existing XAI methods in the literature. We demonstrate the effectiveness of the α-curves using synthetic and real datasets, comparing the results against other XAI methods for variable importance and validating the analysis results with the ground truth or literature information.

1. Introduction

Artificial Intelligence (AI) has gained popularity in the last few years, exceeding expectations in a variety of fields. One example is in the field of predictive analytics, where Machine Learning (ML) models are used to make predictions about future events based on data patterns [1–4]. As data availability increases and more complex problems are tackled, models with a higher number of parameters are needed to accurately learn from data [5].

This increase in the complexity of the model is associated with a lack of interpretability and affects its credibility and trust. Explainable Artificial Intelligence (XAI) is a relatively recent field whose main objective is to make ML models trustworthy [6–8]. There is a growing interest in XAI, as it can help address some of the concerns around responsible AI. For example, if an AI system is used to make decisions that could have significant social impact (such as in healthcare or finance), then it is important that there is a way to understand how and why the system arrived at its decisions [9–11]. This would allow for accountability and transparency, two key components of responsible AI. Furthermore, XAI techniques are not only useful for validating a ML model, but can also be used to retrieve information from the dataset itself. This information can be used to corroborate the prior knowledge of a field, which is an important aspect for evaluating the quality of a model [12].

One important aspect of explainable models is the ability to understand and interpret the factors that drive their predictions. Variable importance metrics are a key tool in this endeavor, as they provide insights into which features of the input data are most important in determining the model's output. This can be useful for building trust in a model, as well as for identifying potential biases or errors in the model. On the one hand, some ML model topologies such as decision trees or linear regression are inherently transparent and provide variable importance measures based on model parameters. On the other hand, neural networks (NN) models are hard to interpret and additional methods must be used to calculate variable importances.

NN models have gained popularity in recent years due to their ability to learn complex patterns from data and make accurate predictions. Despite their high performance, NN models are not commonly used in critical applications due to their lack of interpretability [13]. Consequently, improving the methods to provide interpretability to NN models would unlock the potential of this type of models in applications where their adaptation capabilities provide a higher added value than traditional interpretable models [6,14].

* Corresponding author.
E-mail addresses: jpizarroso@comillas.edu (J. Pizarroso), dalfaya@comillas.edu (D. Alfaya), jportela@comillas.edu (J. Portela), amunoz@comillas.edu (A. Muñoz).

[1] David Alfaya was supported by grants PID2019-108936GB-C21 and PID2022-142024NB-I00 funded by MCIN/AEI/10.13039/501100011033.

https://doi.org/10.1016/j.asoc.2025.113300
Received 19 June 2024; Received in revised form 3 February 2025; Accepted 7 May 2025
Available online 31 May 2025

Some techniques are already available to estimate variable importance of NN models. The most commonly used are Input Permutation [15], which consists in perturbing the input data and observing the effect on the model's output and SHAP (SHapley Additive exPlanations) [16], which assigns an importance value to each feature by averaging over all possible coalition of features. These techniques have notable advantages: they do not depend on the topology of the ML model being analyzed (model-agnostic method) and provide quantifiable information of variable importance [17,18]. In addition, some model-specific methods have been developed for feature importance in NN, such as Garson's feature importance [19] and Olden's feature importance [20].

However, all of these techniques provide a global variable importance without analyzing the distribution of the local importance along the input space. This lack of information can make it difficult to understand how a feature is impacting the model's predictions in different regions of the input space, which can be critical for many applications. A potential consequence of not addressing this question is to consider a variable which only affects the output in a specific region of the input space as less important compared to a variable with a wider-range impact, even if its individual effect is smaller. This could lead to flawed decision making and incorrect conclusions about the data, potentially neglecting a variable with a notable effect on the output of the model for some cases. This highlights the need for further research on developing variable importance metrics that can provide a more comprehensive understanding of feature importance. As far as the authors are aware, there is currently no XAI method that quantitatively addresses this question.

In this paper, a method to obtain information about local and global importances of input variables is presented. The developed method combines the sensitivity analysis of ML models using partial derivatives [21–25] with a mathematical study of certain metric spaces and operator metrics which can be associated to the model and the dataset. These partial derivatives, also called sensitivities, measure the degree to which the output of the model is affected by changes in the input variables. These sensitivities can be calculated for the samples in the input dataset, obtaining a distribution of sensitivity values for each of the input variables. Information of the model can be retrieved from these distributions, such as which variable has the greatest impact on the predictions. This sensitivity analysis method is being applied in numerous fields such as traffic crash modeling [26], chemical industry [27], meteorological modeling [28] and social studies [29,30]. However, it only provides global-level feature importance information, often overlooking local non-linearities that can significantly impact the model's behavior in specific regions of the input space.

A global theoretical metric interpretation of the sensitivity of a ML model which takes into account the whole dataset at once is presented and metrics which aggregate the local sensitivities across the dataset are derived from it as pointwise Lipschitz constants of a certain variation operator associated to the model and the dataset. The pointwise Lipschitz constant of a function between two metric spaces is a mathematical tool used precisely to quantify to what extent a variation of an input affects its output (see, for instance, [31]). Computing local and global Lipschitz constants of the NN model itself has been used in the literature for other purposes as measures of NN robustness [32,33] and deep NN stability [34], however, to the authors' knowledge, analyzing sensitivity from the perspective of this mathematical tool is a new approach. Moreover, the novel proposed metric framework allows us to obtain novel sensitivity measures for general ML models with relevant applications to obtaining variable importance metrics.

Hence, a complete family of new quantitative metrics called α-curves is extracted. These α-curves provide information with greater depth of the variation of the output of a ML model with respect to a specific variable throughout the entire input space. This information not only allows to determine which are the most important input variables for the model, but also detects if there are regions in the input space where a variable is specially relevant.

Our contributions are summarized as follows:

- We propose the novel α-curves XAI method, which integrates a metric interpretation of partial derivatives into a unified framework for sensitivity analysis of machine learning models whose jacobian can be computed. This is specially useful for neural networks, as their jacobian can be computed using automatic differentiation methods.
- The α-curves approach simultaneously captures global variable importance and uncovers detailed local variations, enabling a transition from overall influence to region-specific sensitivities. Unlike traditional techniques such as SHAP or Input Permutation, our method does not assume linear relationships between inputs and outputs, thus preserving and revealing complex non-linear behaviors and interactions.
- By analyzing the evolution of aggregated sensitivities over varying α values, α-curves provide deeper insights into how feature effects vary across the input space, identifying regions where variables have an exceptionally high impact—information that existing XAI methods often overlook.
- The computational complexity of α-curves is comparable to that of standard partial derivative-based sensitivity analysis, scaling linearly with the number of samples and features, which makes it highly efficient for large-scale models.

The rest of the paper is organized as follows. Section 2 collects a state-of-the-art review of XAI techniques applied to obtaining variable importance metrics. In Section 3, the theoretical framework and the α-curves quantitative metrics are presented. Section 4 provides a methodology to use α-curves as sensitivity analysis method of ML models. Section 5 presents examples of this methodology applied to both synthetic and real datasets, demonstrating that the method presented in this paper is able to retrieve information from ML models with greater detail than other XAI techniques. Section 7 concludes the article and presents future research lines.

2. State of the art

At the time of writing, XAI (Explainable Artificial Intelligence) techniques can be categorized based on three characteristics as outlined by Molnar (2022). Firstly, techniques can be intrinsic or post-hoc; the model's simple structure may provide an intrinsic explanation for its decisions, or a separate method may be applied post-training to extract insights. Secondly, techniques may be model-specific or model-agnostic; some methods are tailored for specific machine learning models, while others can be applied universally. Lastly, techniques can be global or local in their scope; some methods aim to explain the behavior of the model across the entire input space, whereas others focus on individual data points.

It is out of the scope of this paper to provide an in-depth state-of-the-art review of XAI techniques, so we only gather a review of post-hoc regression techniques focused on variable importance metrics. For a broader review of XAI methods, we refer the reader to [9,35–38]. Apart from the intrinsic explainable models, such as linear regression and the family of decision trees [39], the main variable importance XAI methods are:

1. Input permutation [15,22]. The technique involves shuffling the values of one input feature and observing the effect on the model's prediction. The resulting change in a chosen error metric for each Input Permutation represents the relative importance of each input variable.

2. Input perturbation [22,40]. Similar to Input Permutation, it consists in adding a small perturbation to each input variable while maintaining the other inputs at a constant value. The resulting change in a chosen error metric for each input perturbation represents the relative importance of each input variable.
3. Partial derivatives method for sensitivity analysis [21–23,41–43]. It performs a sensitivity analysis by computing the partial derivatives of the model output with regard to the input neurons evaluated on the samples of the training dataset (or an analogous dataset).
4. Shapley values [16,44]: originated in game theory, the shapley value is essentially the average expected marginal contribution of one variable after all possible input variable combinations have been considered. An evolution of this method is the SHapley Additive exPlanation (SHAP) [45] method, where the Shapley Values for an ML model are calculated based on LIME (instead of calculating all combinations), reducing the computational resources.
5. Garson's method for variable importance [19]. It consists of summing the product of the absolute value of the weights connecting the input variable to the response variable through the hidden layer. Afterwards, the result is scaled relative to all other input variables. The relative importance of each input variable is given as a value from zero to one.
6. Olden's method for variable importance [20]. This method is similar to Garson's, but it uses the real value instead of the absolute value of the connection weights and it does not scale the result.

The following section presents a brief explanation of the sensitivity analysis based on partial derivatives, which is the basis for developing the $\alpha-$curves methodology.

2.1. Sensitivity analysis based on partial derivatives

Given a Neural Network model fitted for a certain dataset, the sensitivity of the kth output of the model with respect to the jth input variable evaluated in the sample $\bar{x}_i$ is given by:

$$s_{jk}\Big|\bar{x}_i = \frac{\partial y_k}{\partial X_j}\left(\bar{x}_i\right) .$$

Feature importance is taken as the mean squared sensitivity of the output with regard to the input variable:

$$S_j^{sq} = \sqrt{\frac{\sum_{i=1}^{N}\left(s_j\Big|\bar{x}_i\right)^2}{N}},$$

where j is the index of the feature whose importance we want to calculate, N is the number of samples in the dataset we are using for the sensitivity analysis.

Two other sensitivity-based measures are presented in [46]: mean and standard deviation of sensitivities:

$$S_j^{avg} = \frac{\sum_{i=1}^{N} s_j\Big|\bar{x}_i}{N}$$

$$S_j^{sd} = \sigma\left(s_j\Big|\bar{x}_i\right);\ \ i \in \{1,\dots,N\}.$$

Based on these measures, the following information can be obtained from a ML model:

- Input variable j has a non-linear relationship with the output if S_j^{sd} is far from 0.
- Input variable j has a linear relationship with the output if $S_j^{sd} \approx 0$ and $S_j^{avg} \neq 0$.
- Input variable j has no relationship with the output if standard deviation $S_j^{sd} \approx 0$ and $S_j^{avg} \approx 0$.

These are useful measures to retrieve information from a ML model. In [46] a comparison of sensitivity analysis using partial derivatives with most of the other methods is performed. The main advantage of this method is that it provides feature importance measures together with information about the relationship between the output and the input, requiring less computational resources compared to other techniques.

However, the feature importance measures give few information about the sensitivity distribution along the input space. An input variable that has low sensitivity in most of the input space but high sensitivity in specific regions could be incorrectly deemed unimportant, misleading the user. The $\alpha-$curves method is an evolution of sensitivity analysis, providing a metric interpretation of the partial derivatives distribution. This provides extra information by not only giving the same feature importance information, but also provides information about how the sensitivity with respect to a variable is distributed in the input space.

It must be noted that exists other XAI techniques based on using partial derivatives for analyzing neural networks, particularly in the context of image data. For instance, Saliency Maps highlight the most important pixels in an image for a neural network's prediction by computing the gradient of the output with respect to the input image [47]. Grad-CAM (Gradient-weighted Class Activation Mapping) uses gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting important regions in the image [48]. Additionally, SmoothGrad enhances gradient-based sensitivity maps by adding noise to the input and averaging the resulting gradients, leading to more visually coherent and stable saliency maps [49]. These techniques are specifically designed for image data and visual explanations. In contrast, the proposed method in this paper focuses on developing a comprehensive metric framework for analyzing sensitivities and variable importance in machine learning models applied to regression tasks, providing both local and global insights across the input space.

3. A metric interpretation of sensitivity

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be a differentiable function. Let $\mathcal{X} = \{\bar{x}_i\}_{i=1}^N$ be a dataset with $\bar{x}_i = (x_{i,1},\dots,x_{i,n}) \in \mathbb{R}^n$ for each $i = 1,\dots,N$. We propose the following metric framework for analyzing the sensitivity of the function $f(x_1,\dots,x_n)$ with respect to variable X_j over the dataset $\mathcal{X}$. We will analyze variations of the values of f at the points of $\mathcal{X}$ when we perturb each point $\bar{x}_i$ with a variation in the jth component of the point in the following way. We will measure the total variation of the values

$$f(x_{i,1},\dots,x_{i,j}+h_i,\dots,x_{i,n})$$

when we introduce small perturbations $h_1,\dots,h_N \in \mathbb{R}$ on the variable X_j of each point $\bar{x}_1,\dots,\bar{x}_N \in \mathcal{X}$ respectively.

In order to make this precise, we need to fix a way to measure the total variation of f across the dataset $\mathcal{X}$ and a way to measure the perturbation $(h_1,\dots,h_N)$. Let us fix metrics $\|-\|_H$ and $\|-\|_Y$ on $H := \mathbb{R}^N$ and $Y := \mathrm{Fun}(\mathcal{X},\mathbb{R}^m) \cong \mathbb{R}^{mN}$ respectively. Then we define the total variation of f over $\mathcal{X}$ by a perturbation $\bar{h} = (h_1,\dots,h_N) \in H$ on variable X_j as

$$v_{\mathcal{X},j}(f,\bar{h}) = \|\left(f(x_{i,1},\dots,x_{i,j}+h_i,\dots,x_{i,n}) - f(x_{i,1},\dots,x_{i,n})\right)_{i=1}^N\|_Y.$$

We define the *sensitivity of f with respect to variable X_j over the dataset $\mathcal{X}$ for the metrics $\|-\|_H$ and $\|-\|_Y$* as the maximum variation $v_{\mathcal{X},j}(f,\bar{h})$ relative to the size of small perturbations $\bar{h}$.

$$s_{\mathcal{X},j}(f) := \lim_{\varepsilon\to 0}\frac{\sup_{\|\bar{h}\|_H=\varepsilon} v_{\mathcal{X},j}(f,\bar{h})}{\varepsilon}.$$

A natural setup for this metric analysis is to choose the involved metrics to be L^p norms. Recall that for each $p \in [1, \infty)$, we define the L^p norm as

$$\|(x_1, \dots, x_M)\|_p = \left(\sum_{i=1}^{M} |x_i|^p \right)^{1/p}.$$

Taking the limit when $p \to \infty$, we also have

$$\|(x_1, \dots, x_M)\|_\infty = \max\{|x_i|\}.$$

In this case, explicit formulas for $s_{\mathcal{X},j}(f)$ can be computed in terms of the differential of f at each point in $\mathcal{X}$.

Let $d_j f$ denote the differential of f with respect to variable X_j, i.e., if $f = (f_1, \dots, f_m)$, then

$$d_j f = \left(\frac{\partial f_1}{\partial X_j} dX_j, \dots, \frac{\partial f_m}{\partial X_j} dX_j \right).$$

Then the following theorem (whose complete proof can be found at Appendix A) enables an efficient and simple computation of the metric sensitivity invariant $s_{\mathcal{X},j}$ when the chosen metrics are L^p metrics.

Theorem 3.1. *Let $f = (f_1, \dots, f_m) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be a C^2 function. Assume that $\|-\|_H = \|-\|_p$ and $\|-\|_Y = \|-\|_q$.*

- *If $p \le q$ then*

$$s_{\mathcal{X},j}(f) = \max_i \left\{ \left\| d_j f(\bar{x}_i) \right\|_q \right\}.$$

- *Otherwise, if $p > q$ then*

$$s_{\mathcal{X},j}(f) = \left(\sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{\frac{p}{p-q}} \right)^{\frac{p-q}{pq}}.$$

Remark 3.2. Observe that when we take L^p and L^q norms on H and Y respectively, then $s_{\mathcal{X},j}(f)$ is a norm.

3.1. Sensitivity α-curves associated to a real function

Some interesting additional analysis can be derived from Theorem 3.1 when the function f is scalar. Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a scalar function and let $\mathcal{X} = \{\bar{x}_i\}_{i=1}^N$ be a dataset with $\bar{x}_i \in \mathbb{R}^n$. The previous Theorem allows us to compute explicitly the sensitivity $s_{\mathcal{X},j}(f)$ for each choice of L^p norms on the perturbation and the target values. We observe, however, that when the target of the function is $\mathbb{R}$, some of the sensitivities agree for different choices of (p, q), resulting in the fact that all the L^p norm choices can be summarized on a 1-parametric set of metrics which can then be rewritten in terms of the α-mean of the values $\left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right|$ when $\bar{x}_i$ runs through the dataset $\mathcal{X}$ and $1 \le \alpha \le \infty$.

Corollary 3.3. *If $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a C^2 function, $\|-\|_H = \|-\|_p$ and $\|-\|_Y = \|-\|_q$ with $p > q$, then*

$$s_{\mathcal{X},j}(f) = N^{1/\alpha} M_\alpha \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\},$$

where $\alpha = \frac{pq}{p-q}$ and

$$M_\alpha\{t_1, \dots, t_N\} = \left(\frac{\sum_{i=1}^N t_i^\alpha}{N} \right)^{1/\alpha}$$

is the generalized α-mean of the values. When $p \le q$ then

$$s_{\mathcal{X},j}(f) = M_\infty \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\} = \max_i \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\}.$$

This motivates the following definition. Let us define the *α-mean sensitivity of f with respect to variable X_j on the dataset $\mathcal{X}$* as

$$\mathrm{ms}^\alpha_{\mathcal{X},j}(f) := M_\alpha \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\}.$$

Then, define the sensitivity α-curve as the map

$$\begin{aligned} \mathrm{ms}_{\mathcal{X},j}(f) : [1, \infty] &\longrightarrow [0, \infty) \\ \alpha &\mapsto \mathrm{ms}^\alpha_{\mathcal{X},j}(f). \end{aligned}$$

On the other hand, observe that the Generalized Mean Inequality implies that for each $0 \le \alpha < \beta \le \infty$ we have

$$M_\alpha \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\} \le M_\beta \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\}$$

and we know that

$$M_\infty \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\} = \lim_{\alpha \to \infty} M_\alpha \left\{ \left| \frac{\partial f}{\partial X_j}(\bar{x}_i) \right| \right\}$$

so we conclude that $\mathrm{ms}_{\mathcal{X},j}(f)$ is an increasing bounded curve whose limit when $\alpha \to \infty$ is $\mathrm{ms}^\infty_{\mathcal{X},j}(f)$. In virtue of the metrical interpretation of the sensitivity from the previous section, a representation of this curve, together with the asymptotic value $\mathrm{ms}^\infty_{\mathcal{X},j}(f)$, yields an interesting visualization of the whole sensitivity analysis which is independent on the choice of the L^p norms on the perturbation and target spaces. We will call this representation the *sensitivity α-curve associated to f with respect to variable X_j over the dataset $\mathcal{X}$*. See Section 4.1 for plotting details and Figs. 1, 4(b) or 8(b) for examples.

3.2. Relation with distributions of partial derivatives

The α-curves from the previous section can be given an alternative description in terms of the distribution of partial derivatives [46] mentioned in the introduction. This duality reinforces the usefulness of the α-mean sensitivities and the α-curve as quantitative tools for performing deep meaningful sensitivity analysis.

Assume that the points of the dataset $\mathcal{X}$ have been drawn randomly uniformly and that, therefore, they inherit a uniform discrete distribution on them (with probability $1/N$ over each point). Let us consider the function $g_j(x) = \left| \frac{\partial f}{\partial X_j}(x) \right|$ representing the (local) sensitivity of f with respect to X_j at the point x. Then a direct computation shows that for each $\alpha \in [1, \infty)$:

$$\mathbb{E}[g_j^\alpha] = \left(\mathrm{ms}^\alpha_{\mathcal{X},j}(f) \right)^\alpha.$$

As a consequence, all moment maps of the distributions of partial derivatives g_j can be computed as polynomials in the α-mean sensitivities. In particular, this proofs that the α-mean sensitivities encode exactly as much information as the moment maps of the distributions of partial derivatives across the dataset.

This dual interpretation has interesting theoretical ramifications on the interpretation and validity of several sensitivity analysis methodologies. On the one hand, it proves that any qualitative analysis on the distributions of local sensitivities can be aggregated as an analysis of the corresponding α-curve instead, showing that α-curves are at least as informative as the distribution of local sensitivities. Nevertheless, we will provide some experimental evidences which prove that an analysis on α-curves allows an easier detection of certain qualitative properties of the dependence of a function with respect to a variable than the analysis on moment maps of the distributions of partial derivatives. For instance, the quantitative variation in the tail and higher moment maps of a distribution associated to the existence of regions of the space where a variable is locally relevant is much more subtle than the variations of the corresponding α-mean sensitivities for high α. This implies that it is easier to detect these patterns by observing the corresponding α-curves than by observing the derivative distributions.

On the other hand, this allows to provide an additional metric interpretation (and, thus, additional theoretical support) for the usage of the moments of derivative distributions as sensitivity measures, as used in [42,46,50].

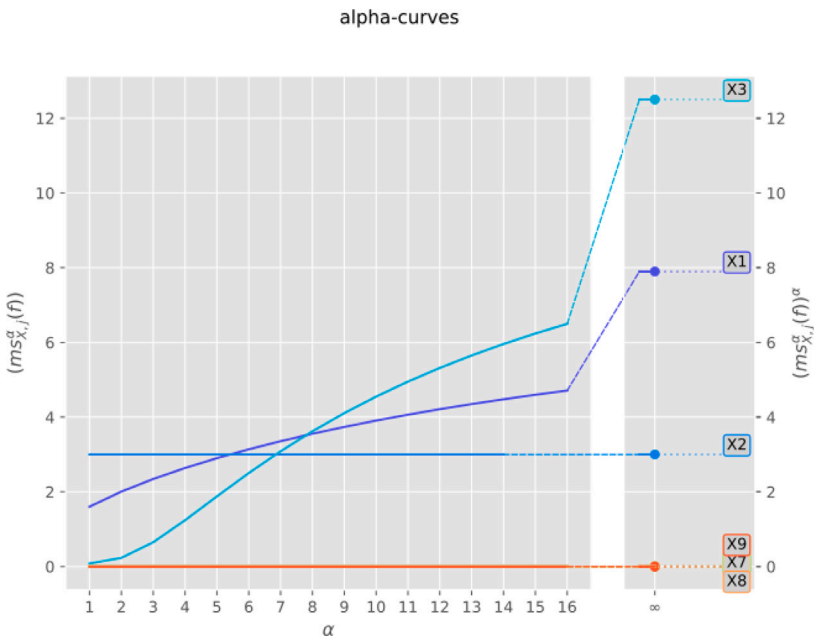


Fig. 1. α-curves of the cubic root synthetic dataset (see Section 5.1 for details). This plot shows, on the horizontal axis, the parameter α ranging from 1 to 16, and then a break at $\alpha = \infty$. Each line corresponds to one input variable $(X_1, \ldots, X_9)$, and the vertical axis measures $\mathrm{ms}^{\alpha}_{\mathcal{X},j}(f)$, i.e., the α-mean sensitivity of the output with respect to that variable. As α increases, the metric increasingly emphasizes regions where the variable has very high local sensitivity. Variable X_3 shows a high localized sensitivity, as indicated by their sharp increase at higher α values and their high asymptotic sensitivity at $\alpha = \infty$. In contrast, variables like X_7, X_8, and X_9 remain constant and close to zero, indicating irrelevance across the dataset. The steady increase in X_1 shows a general non-linear relationship with the output, and the horizontal line of X_2 shows a linear relationship.

4. Methodology of the α-curves analysis

As a consequence of the previous theoretical analysis, we have built a family of metrics capable of quantifying the sensitivity of any model f with respect to a variable X_j. In this section, we will explain some methodologies capable of exploiting this new theoretical framework to detect some patterns in the roles that each variable play in the model. Contrary to other sensitivity analysis methodologies, our proposed method will be able to capture high sensitivity behaviors of a variable which may only occur in a certain local region of the phase space, even if that variable presents generally a low sensitivity across the rest of the dataset.

For simplicity, we will focus the analysis on scalar regression problems. In this case, we have proven that computing the α-curves of the function for each input variable allows a complete and simultaneous study of all the possible L^p metric interpretations of sensitivity for each variable across the whole dataset. Each α-mean sensitivity can be used individually as a theoretically sound sensitivity metric, but analyzing the differences between the values of the α-sensitivities $\mathrm{ms}^{\alpha}_{\mathcal{X},j}$ for different variables X_j and different choices of α opens new deeper layers to the sensitivity analysis and can be used to detect further properties and interactions between the variables of a model than other methods used in the literature. We proceed to describe a methodology for retrieving some of the properties of the variables of a model from an α-curve plot.

We would like to clarify that this is not an exhaustive description of the types of analysis that can be done within the α-curve framework. For instance, we believe that this family of sensitivity metrics and the theoretical framework which supports them can be incorporated as part of more complex XAI analysis. It is just a showcase of some of its basic properties and further exploration of this methodology will be subject of future work.

4.1. Model sensitivity analysis and α-curve plots

The α-curve analysis presented in this work is a tool designed to study scalar regression data models (in particular, as stated in Theorem 3.1 and Corollary 3.3, $\mathcal{C}^2$ data models). Thus, in order to analyze raw data with this method, the first step is to build an appropriate representative data model f. An important example is to choose f to be an AI system of some type trained over the data. The α-sensitivity analysis is always meant to study properties of the chosen model f and not necessarily on the data which generated it (eg., it will analyze the way a trained AI model interacts with its input variables, not the data which was used to train it).

In order to have proper comparisons between variables, it is important that the variables are normalized or have comparable magnitudes before constructing the model f and performing the α-sensitivity analysis. Otherwise, renormalizations may need to be taken into account when performing the study (the α-curve equation is homogeneous of degree 1 on linear scaling of the functions and variables).

The α-curves are visualized in a two-dimensional plot where the x-axis represents the varying α values in the interval $[1, \infty]$ and the y-axis corresponds to the computed α-mean sensitivities. For each input variable X_j, its α-curve is plotted by computing $\mathrm{ms}^{\alpha}_{\mathcal{X},j}(f)$ across a range of α values (see an example in Fig. 1). By the Generalized Mean Inequality, we know that each α-curve is increasing and bounded, and that the limit for $\alpha = \infty$ coincides with the maximum sensitivity of f with respect to variable X_j reached at some point in the dataset. In our experiments, we found that evaluating α up to order 16 captures most of the informative changes, as further increases provide negligible additional insight until reaching the asymptotic value at $\alpha = \infty$, which we have found to be of high utility for the analysis. Thus, values between the chosen maximum plotted α value (e.g., $\alpha = 16$) and $\alpha = \infty$ are omitted from the visualization to avoid unnecessary computation.

To effectively display the asymptotic behavior, the x-axis is broken to include a second plot specifically dedicated to showing the α-mean sensitivity at $\alpha = \infty$, which corresponds to the maximum sensitivity value $\mathrm{ms}^{\infty}_{\mathcal{X},j}(f)$. Each α-curve is linked to a label identifying its corresponding feature via a pointed line extending toward the right-hand plot, where the asymptotic values are marked. This visual linkage aids in associating each curve with its feature identity, especially when multiple curves converge or exhibit similar trends for lower α values.

4.2. Comparison of variables for a fixed α

For each value of α, the set of values $\mathrm{ms}^{\alpha}_{\mathcal{X},j}(f)$ provides a sound measure of the sensitivity of the model f with respect to each variable X_j. Thus, it can be used to compare which input variables are more relevant for the output in the model.

In the literature, the metrics obtained from $\alpha = 1$ and $\alpha = 2$ (or, derived ones, like the variance of the distribution of partial derivatives, which can be computed from these two values, cf. Section 3.2), have been used as sensitivity metrics [42,46,50] and utilized, for instance, for variable pruning [51,52]. As a consequence of the discussion from Section 3.2, the mean sensitivity $\mathrm{ms}^{1}_{\mathcal{X},j}(f)$ for $\alpha = 1$ represents the average of the pointwise absolute sensitivities of the model with respect to variable X_j across the dataset. Similarly, $\mathrm{ms}^{2}_{\mathcal{X},j}(f)$ for $\alpha = 2$ computes second standard moment of the distribution of punctual sensitivities of the model with respect to variable X_j across the dataset.

Corollary 3.3 now provides additional sound theoretical framework which supports mathematically the choice of any of these aggregation functions as a way to derive a total sensitivity metric from the values of the derivatives of the function across the dataset.

On the other hand, each vertical cut α to the α-curve plot can be used to compare the variables and draw quantitative and qualitative conclusions about their relative relevance to the model. If there are model-driven reasons to fix a certain metric on the input and perturbation spaces (see 3.1), then the corresponding α-value should be chosen for the comparison. Otherwise, any value of α could, theoretically, be used for the sensitivity comparison task independently. Nevertheless, analyzing the whole picture across all different α allows a deeper understanding of the behavior of the model.

Due to the properties of α-means, as α increases, the average value $\mathrm{ms}^{\alpha}_{\mathcal{X},j}(f)$ takes more into account the existence of regions in the dataset where the sensitivity with respect to the variable is higher than average ("exceptionally sensitive" or "localized high sensitivity" behavior). Lower values of α focus instead on the "average behavior" of the function with respect to the variable.

The analysis of high values of α may be crucial for certain tasks like variable pruning. It is possible that a variable has almost no impact on a regression problem if one looks at a generic point in the phase space, but that there exists a mode change in the model making the variable very relevant for the analysis when the inputs move inside a certain critical region (think, for instance, in the case where there exist "activation" variables or states, which enable a different variable to influence the result but otherwise disable it). A general pruning analysis with $\alpha = 1$ or $\alpha = 2$ could "discard" the variable as irrelevant for the model, whereas it might be the most relevant variable for high α metrics (see, for instance, Fig. 1).

The limit values $\mathrm{ms}^{\infty}_{\mathcal{X},j}(f)$ included in the plot help identify the extremal cases. They measure the maximum sensitivity of f with respect to the input variable X_j that can be found at any point in the dataset.

4.3. Analysis of the variation of an α-curve

Due to the aforementioned properties of the α-means (consequence of the convexity of the power functions for exponents at least 1), studying the variation of the α-sensitivity when α changes can give a lot of information on the dynamics of the variables of the model f. Let us study some examples.

4.3.1. Linearity analysis

By the Generalized Mean Inequality, the α-curve of variable X_j is constant if and only if f is of the form

$$f(X_1, \dots, X_n) = g(X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n) + CX_j$$

for some function g depending only on the rest of the variables. By extension, the closer an α-curve is to be flat, the closer the dependence of f with respect to X_j is to a linear dependence. For instance, when an α-curve starts almost flat and then starts increasing more starting at some alpha, this can mean that the derivative $\frac{\partial f}{\partial X_j}$ has a low variation through the majority of the dataset, but that there are one or more regions of the phase space where it changes more, either due to its own non-linear behavior (like an activation function, or function where the derivative increases close to a point, like a C^2-approximation of a square root), or due to an interaction with other variables.

4.3.2. Irrelevant variables

As a particular case of the previous analysis, f does not depend on a variable X_j if and only if the α-curve is constantly zero. The closer a curve is to 0, the less important the variable is for the model.

If a curve starts very constant and close to 0 but increases afterwards, this indicates that the output of the model has, in general, a low dependence on the variable, but that there exists a region in the phase space in which the variable is indeed relevant for the model.

These properties can be used to improve the specificity of variable pruning methodologies. If a variable presents a low value of $\mathrm{ms}^{\infty}_{\mathcal{X},j}(f)$ (and, thus, the whole α-curve is low) then it is not important for the model and it can be safely removed. On the contrary, a variable presenting higher values of the curve for some α (and thus, a higher $\mathrm{ms}^{\infty}_{\mathcal{X},j}(f)$) should not be pruned without a deeper analysis.

4.3.3. Detection of local regions with high sensitivity

As outlined before, it is possible for a variable to have low sensitivity for low α but high sensitivity in higher α. This makes comparing its α-sensitivity with the α-sensitivity of other variables depend heavily on α. When this happens and a variable is not sensitive for low α but it becomes highly sensitive for high α, two things can happen.

- The variable shows a non-linear behavior on X_j which makes the partial derivative $\frac{\partial f}{\partial X_j}$ increase only on certain values of X_j.
- There exists an interaction between the variable and a combination of other variables which makes the derivative become high in a certain region of the phase space.

The higher the variation of the α-curve and the earlier this variation appears, the stronger and more generalized the interaction or non-linear effect is across the dataset. If the α-curve starts flat and then there is a sudden increase, it is more probable that the interaction or non-linear input effect on the output is relevant only in certain bounded areas of the dataset.

A limitation of this method (see Section 7) is that it is difficult to distinguish between the increase in sensitivity produced by an interaction between variables (eg. when they are equally distributed) and a non-linear input effect (which can be thought of as a self-interaction of the variable). We expect to solve this limitation in future work through the usage of complementary interaction analysis methodologies.

4.4. Example of qualitative analysis through the graphical representation of α-curves

Fig. 1 illustrates a practical example of such an α-curve plot. The details on the synthetic dataset and model used to obtain the plot will be further described in the following Section 5.1. In this example we can deduce the following qualitative information from the α-curve plot:

- Variables X_4 to X_{10} exhibit no relationship with the output, resulting in flat horizontal lines at zero across all α values.
- Variable X_2 shows a linear relationship with the output, depicted by a flat horizontal line at a constant value different from zero throughout the α-range.
- Variable X_1 has a non-linear relationship with the output, reflected in an α-curve that slowly rises from low α values, indicating an absence of regions with exceptionally high sensitivity.

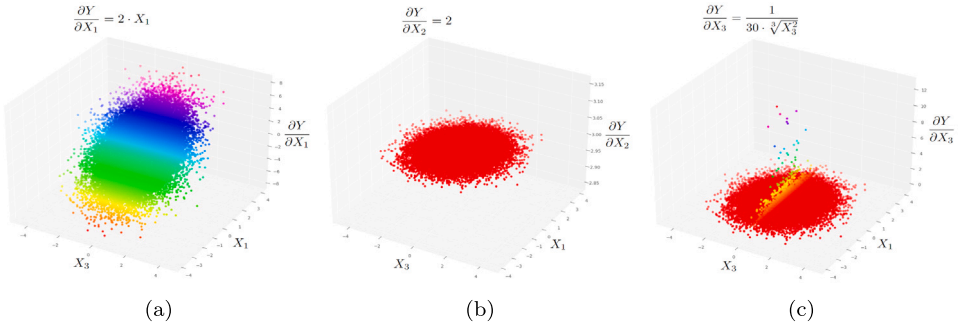


Fig. 2. 3D plots of partial derivatives of output Y with respect to inputs X_1, X_2 and X_3 ((a), (b) and (c) respectively) for square root synthetic dataset. X-axis follows X_1 and y-axis follows X_3 in the three plots. X_2 is not used as plot axis due to the irrelevance of this variable on the derivative plots. Figure shows a non-linear relationship of the output with input variable X_1, a linear relationship with X_2 and a non-linear relationship in a localized region with X_3.

- Variable X_3 also has a non-linear relationship, but its α-curve displays a rapid increase at higher α values, indicating the presence of local regions with very high sensitivity compared to its global behavior.

This plotting strategy allows analysts to visually compare the sensitivity profiles of different variables, identifying both their average impact and localized effects within the model.

5. Experimental results

This section contains XAI analysis performed on various synthetic and real datasets using sensitivity analysis based on partial derivatives, SHAP, Input Permutation and the α-curves method.

5.1. Synthetic dataset

A synthetic dataset with known derivatives is used to illustrate the usefulness of the α-curves to retrieve information about how the model uses the input variables to predict the output variable. The dataset is composed by 8 input variables $[X_1, \dots, X_9]$ and one output variable $Y \in \mathbb{R}$ created as a function of the input variables, i.e., $Y = f(\mathbf{X})$. Input variables $\mathbf{X}$ are 50 000 samples drawn from a normal distribution with $\mu = 0$ and $\sigma = 1$. Partial derivatives and SHAP values are calculated analytically from the output expression for each dataset to avoid inherent modeling error which might obfuscate relationships between inputs and outputs.

In this case, the output follows the next expression:

$$Y = (X_1)^2 + 2 \cdot X_2 + \frac{1}{10} \cdot \sqrt[3]{X_3}. \tag{1}$$

From Fig. 2, we can conclude that X_2 has a linear relationship with Y as $\frac{\partial Y}{\partial X_2}$ is constant and different from zero for all samples. X_1 and X_3 have a non-linear relationship with Y, as $\frac{\partial Y}{\partial X_1}$ and $\frac{\partial Y}{\partial X_3}$ are not constant for all samples. Furthermore, Fig. 2(c) shows that $\frac{\partial Y}{\partial X_3} = 0$ for most samples, except for the samples where X_3 is close to 0. In these samples, sensitivities with respect of X_3 are far higher than for the other input variables, so changes of X_3 in this region of the input space shall provoke large changes on Y. This can be understood as a local importance of X_3, and it shall be detected by XAI methods.

Results of XAI analysis performed on Eq. (1) are presented in Fig. 3. Fig. 3(a) shows the sensitivity plots as introduced in [46]. First plot shows two sensitivity metrics: mean (x-axis) and standard deviation (y-axis). Second plot of Fig. 3(a) shows the mean squared sensitivity for each of the input variables, which could be used as a variable importance metric. A broader explanation of these metrics can be found in Section 2. According to this metrics, the following information can be retrieved from Fig. 3(a):

- X_2 variable has a linear relationship with the output.
- X_1 variable has a non-linear relationship with the output.
- X_3 is almost irrelevant to predict the output, with much lower importance than X_1 and X_2, but greater than $X_4 - X_8$.
- The remaining variables have no relationship with the output.

The same information is obtained using SHAP from Figs. 3(b) and 3(c). In Fig. 3(b), linear relationship of Y and input X_2 can be seen in the perfect correlation between the values of X_2 and its impact on Y. Non-linear relationships of Y and inputs X_1 is also easily detected, as there is no correlation between the values of X_1 and its impact on Y. Fig. 3(c) is analogous to second plot of Fig. 3(a), but showing mean of the absolute SHAP values instead of mean squared sensitivity as variable importance measure.

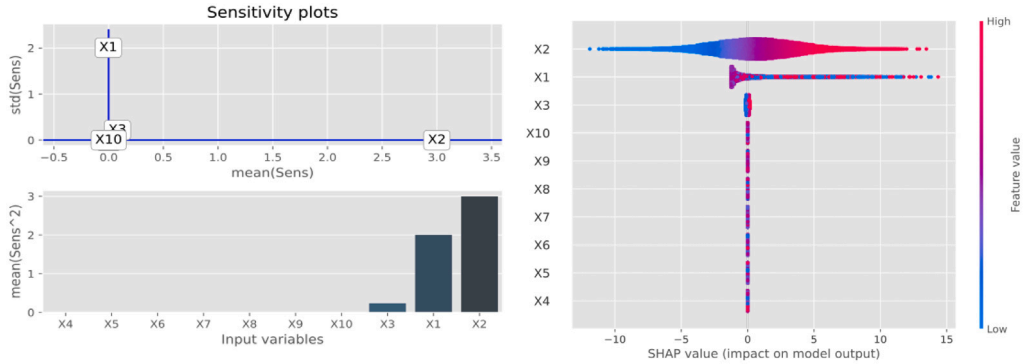
Fig. 3(d) shows the importance metrics assigned to the input variable with the Input Permutation technique. These metrics are almost identical to SHAP importance metrics presented in Fig. 3(c), with similar relative importances between input variables. This technique does not provide information about the type of relationship between output and input variables, but it is notably less computationally expensive than the others.

Using the $\alpha-$curves methodology described in Section 4, the information obtained from the other XAI methods can also be obtained from Fig. 1. However, it also shows that, apart from the non-linearities presented in X_1 and X_3, there are regions where output Y is far more sensitive to X_3 than to X_2. In fact, peak sensitivities in some samples are detected, as can be seen by the rapid increase of $\mathrm{ms}^{\alpha}_{X_3}(f)$ for $\alpha > 4$. This information could not be retrieved using the other two methods, which assigned little importance to X_3, due to the aggregation techniques used to calculate importance of the input variables.

5.2. Real datasets

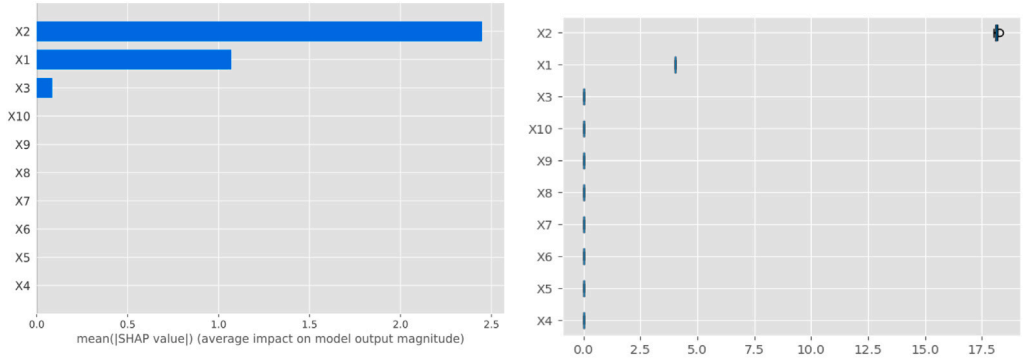
Based on the analysis performed in the previous section, similar analysis can be conducted on datasets from real sources. In this section, sensitivity analysis is performed on the California housing dataset [53] and the Parkinson's Disease regression dataset [54]. These datasets were selected because they both represent regression tasks with varying numbers of input features, allowing for an evaluation of the interpretability and scalability of the proposed methodology. While other larger datasets were considered, they were dismissed to ensure the resulting plots remained comprehensible when analyzed with sensitivity and explainable AI (XAI) methods. A scalability study to demonstrate the methodology's performance on larger datasets is included in Section 5.3.

To model the relationship between input and output variables, a Multi-Layer Perceptron (MLP) with one hidden layer is trained for each dataset. The model hyperparameters, including the number of neurons in the hidden layer, activation functions, and learning rate, were optimized using 10-fold cross-validation to ensure robust performance while avoiding overfitting.



(a) Sensitivity plots of cubic root synthetic dataset. The results confirm that X_2 has a linear relationship with the output and X_1 a non-linear relationship. X_3 seems to have little to no relationship with the output.

(b) The SHAP value summary plot showcases the impact of input variables on the output. The linear correlation between X_2 and Y is evident, as well as the non-linear relationship for X_1.



(c) SHAP Importance values of cubic root synthetic dataset. It states X_2 as the most important variable, with X_1 also being important and X_3 having little to no importance.

(d) Input permutation importance values of cubic root synthetic dataset. It shows that only X_2 and X_1 are important.

Fig. 3. XAI analysis of the cubic root synthetic dataset. Sensitivity metrics, SHAP values, and Input Permutation metrics illustrate the relationships between the model output and input variables. The plots highlight the importance and type of relationship (linear or non-linear) of each input variable (X_1, X_2) with the output, while demonstrating the irrelevance of the remaining variables (X_4 to X_9), and a doubtful relationship with X_3.

The first partial derivatives of the trained MLP models were calculated using the `neuralsens` package [46], which enables efficient computation of derivatives for sensitivity analysis. It should be noted that the α-curves methodology directly applies to these calculated partial derivatives, meaning that any user-preferred package capable of computing partial derivatives of a machine learning model can be employed. For comparison, SHAP values were calculated and analyzed for each experiment using the homonymous `shap` package [45] and Input Permutation Importance was calculated using the `dalex` package [55].

5.2.1. *California housing*

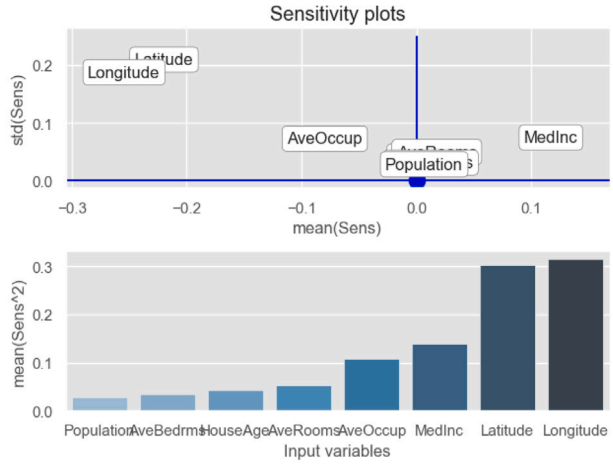
This dataset was derived from the 1990 U.S. census, using one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3000 people). The target variable is the median house value for California districts, expressed in hundreds of thousands of dollars ($100,000) [56].

This dataset is composed of the following variables:

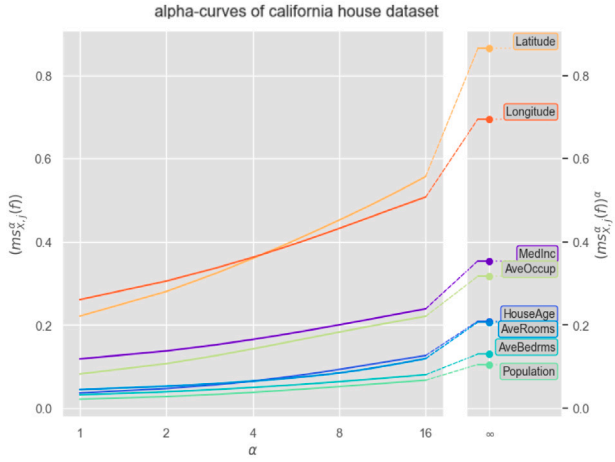
- MedInc: median income in block group
- HouseAge: median house age in block group
- AveRooms: average number of rooms per household
- AveBedrms: average number of bedrooms per household
- Population: block group population
- AveOccup: average number of household members
- Latitude: block group latitude
- Longitude: block group longitude
- MedHouseVal: median price of block group

To model the relationship between these input variables and the target variable, a Multi-Layer Perceptron (MLP) was employed. The MLP obtained using 10-fold cross validation consisted of a single hidden layer with 5 neurons and a sigmoid activation function. It was trained with a learning rate of 0.1 for 150 epochs.

Following the methodology for sensitivity analysis, Fig. 4(a) shows that longitude and latitude of the block group are the most important variables with a highly non-linear relationship with the output, followed by the median income of the families in block group and the average number of household members with a more linear relationship. The rest of the variables have almost no relationship with the output and may be discarded from the model. This result seems intuitive because location is typically one of the key factors affecting house value. Additionally, the number and type of neighbors in the area often indicate a block's overall economic status, where fewer people with higher income might indicate exclusive villas and more people with lower income might indicate residential blocks. The remaining variables correlate with factors we would expect to influence house prices, this is, size of the house ($AveRooms$) and how up-to-date house features are ($HouseAge$) is more important than population of the block. One might object that $AveBedrms$ also correlates with the size



(a) Sensitivity plots of California housing dataset. Location-related variables (`Latitude` and `Longitude`) emerge as the most significant predictors of house prices, exhibiting a highly non-linear relationship with the target. Socioeconomic indicators (`MedInc` and `AveOccup`) show less important relationships.



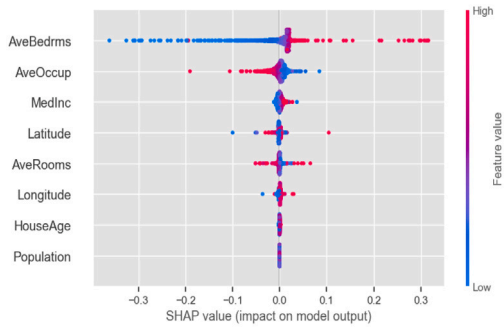
(b) α-curves of california housing dataset. The curves reveal the dominant influence of location-related variables, as indicated by $\mathrm{ms}^{\alpha}_{\mathrm{Longitude},j}(f)$ and $\mathrm{ms}^{\alpha}_{\mathrm{Latitude},j}(f)$ being higher than for the rest of the variables for all values of α. While `Longitude` has consistently high sensitivity, `Latitude` exhibits localized importance in specific regions, potentially corresponding to areas near the coastline or high-value districts. The curves also highlight the relevance of `HouseAge` and `AveRooms` for certain subsets of the data.

Fig. 4. Sensitivity and α-curve analyses of the California housing dataset. These analyses reveal that location-related variables (`Latitude` and `Longitude`) play a dominant role in predicting house prices, with `Longitude` showing consistently high importance and `Latitude` demonstrating localized impacts in specific regions, such as coastal areas. The α-curves highlight the nuanced relevance of variables like `HouseAge` and `AveRooms` in specific subsets of the data.

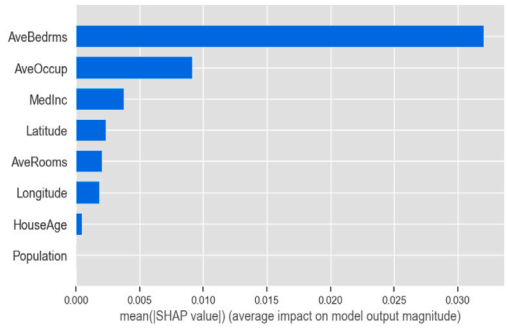
of the house and shall be assigned a higher importance. However, size of the house can be determined with *AveRooms*, so the information provided by *AveBedrms* may only be used to distinguish between same size houses with different number of bedrooms. This information might not influence house price as much as the house size, so consequently the *AveBedrms* variable is not as important as *AveRooms*.

More information can be retrieved using the $\alpha-$curves methodology. The location of the house is still the most important information to predict the price, but Fig. 4(b) shows that, although *Latitude* have a lower mean sensitivity, there are regions of the input dataset where the *Latitude* sensitivity is the highest. This might indicate a region where a small change in latitude switches between two blocks with significantly different prices, maybe between first and second beach line blocks. Rest of the variables shows similar information than the retrieved from Fig. 4(a), although it can be seen than the maximum sensitivity of *HouseAge* is similar to the maximum sensitivity of *AveRooms*. This might indicate that, although the mean effect of the *HouseAge* is not as important as the *AveRooms* variable, the house age might influence the price of the house as much as the size if the house. This makes sense, as an older house usually needs house renovations and this might decrease the price.

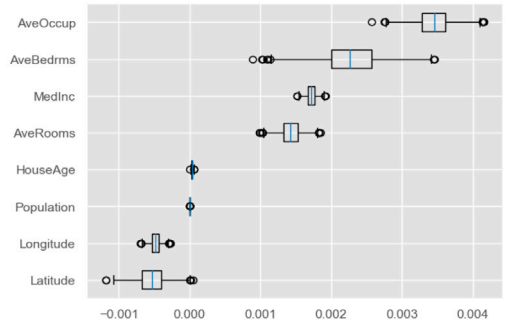
It must be noted that it was not computationally feasible to analyze all the samples of the dataset using SHAP, so only 1000 random samples were analyzed. Figs. 5(a) and 5(b) shows that the most important variable is the average number of bedrooms per household followed by the average number of household members. Relationship between these inputs and the output seems linear, where a large number of bedrooms and fewer households corresponds to a higher house price. This correlates with the idea of luxury villas and residential blocks stated earlier. However, it appears counter-intuitive that the location



(a) SHAP values of California housing dataset. SHAP analysis assigns the highest importance to `AveBedrms` and `AveOccup`, suggesting a more linear relationship between these variables and house prices. Interestingly, location variables (`Longitude` and `Latitude`) appear less important.



(b) SHAP Importance values of California housing dataset. SHAP importance values highlight `AveBedrms` and `AveOccup` as the most impactful variables, with location variables ranked lower.



(c) Input permutation importance values of California housing dataset. This analysis suggests that house size and occupancy-related variables (`AveOccup`, `AveRooms`) are the most important.

Fig. 5. SHAP and Input Permutation analyses of the California housing dataset. SHAP values emphasize the importance of socioeconomic variables such as `AveBedrms` and `AveOccup`, while assigning less importance to location variables (`Latitude` and `Longitude`), which contrasts with the sensitivity and α-curve analyses. Input permutation metrics identify house size and occupancy-related variables (`AveRooms` and `AveOccup`) as the most significant predictors.

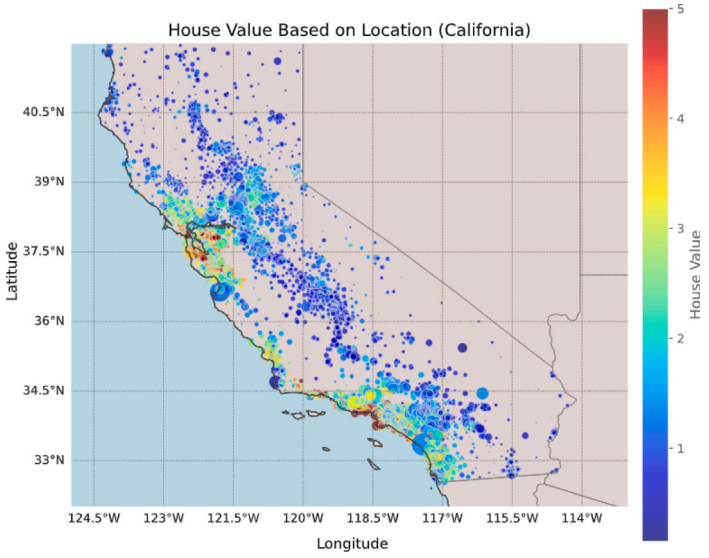


Fig. 6. Heatmap showing the spatial distribution of median house prices in California based on latitude and longitude. The color scale represents the median house price, while the size of the circles indicates the block group population. The figure highlights that house prices inversely correlate with the distance to the shoreline, which is reflected in the interaction between latitude and longitude variables in the dataset.

of the block given by the *Latitude* and *Longitude* variable is barely important compared with the rest of variables.

Considering the Input Permutation Importances shown in Fig. 5(c), it can be seen how the importances assigned to the input variables vary depending on the permutation performed on the variable. In this case, age of the house *HouseAge* and population of the block *Population* does not influence on the performance of the model (importance of these variables being zero is related to no change in the error metric when this variables are permutted), coinciding with the rest of the XAI techniques. House size occupation related variables are the most important according to this technique, assigning the lowest importances to the location related variables (*Longitude* and *Latitude* variables). Moreover, this technique assigns a negative importance to these variables, implying that permutting *Longitude* and *Latitude* results in a more accurate model. Again, this appears counterintuitive and may be misleading, possibly because of the way the variables were permuted.

Fig. 6 shows the distribution of prices in California state, where we can see that based on latitude and longitude great differences in house price can be distinguished. As expected, the highest house prices are in the population centers on the beachfront (in this case, San Francisco and Los Angeles). This corroborates the information provided by the $\alpha-$curves method, where location related variables are the most important to predict house prices, which contradicts the explanation provided by SHAP.

5.2.2. Parkinson's disease

This dataset contains biomedical voice measurements from 42 individuals with early-stage Parkinson's disease who were part of a six-month trial utilizing a telemonitoring device for remote symptom progression monitoring [54]. This dataset is composed by the following variables:

- **Jitter (%):** The percentage of jitter, which represents the short-term variability of the fundamental frequency.
- **Jitter(Abs):** Absolute jitter, representing the absolute differences in consecutive periods, measured in seconds.
- **Jitter:RAP:** Relative Amplitude Perturbation, a measure of the variability in the amplitude of vocal fold vibration.
- **Jitter:PPQ5:** Five-point Period Perturbation Quotient, a measure of the variability in pitch period size over five pitch periods.
- **Jitter:DDP:** Dimensionless Drift Parameter, a composite measure derived from RAP.
- **Shimmer:** A measure of the amplitude variability of the vocal fold vibration.
- **Shimmer(dB):** The logarithmic measure of shimmer, expressed in decibels.
- **Shimmer:APQ3:** Three-point Amplitude Perturbation Quotient, a measure of the variability in amplitude over three pitch periods.
- **Shimmer:APQ5:** Five-point Amplitude Perturbation Quotient, a measure of the variability in amplitude over five pitch periods.
- **Shimmer:APQ11:** Eleven-point Amplitude Perturbation Quotient, a measure of the variability in amplitude over eleven pitch periods.
- **Shimmer:DDA:** A composite measure derived from APQ measures.
- **NHR:** Noise-to-Harmonics Ratio, a measure of the ratio of noise to tonal components in the voice signal.
- **HNR:** Harmonics-to-Noise Ratio, a measure of the ratio of tonal components to noise components in the voice signal.
- **RPDE:** Recurrence Period Density Entropy, a non-linear measure that quantifies the predictability and complexity of a signal.
- **DFA:** Detrended Fluctuation Analysis, a method for determining the statistical self-affinity of a signal.
- **PPE:** Pitch Period Entropy, a measure of the regularity and stability of the pitch.
- **Sex:** The gender of the individual.

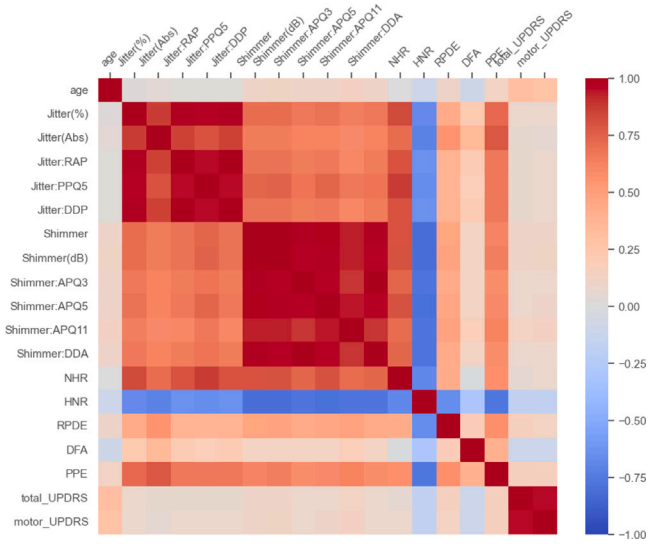


Fig. 7. Heatmap of the correlation between the input features in the Parkinson's Disease Voice dataset. This figure illustrates the relationships among the various biomedical voice measurements. The heatmap reveals significant collinearity within groups of variables, such as the Jitter and Shimmer families, which could result in multicollinearity challenges in modeling. For instance, variables like `Jitter(%)` and `Jitter(Abs)`, as well as `Shimmer` and `Shimmer(dB)`, exhibit strong positive correlations. To address this, representative variables (`Jitter(Abs)` and `Shimmer`) were selected for further analysis, following insights from the literature that highlight their relevance in Parkinson's Disease diagnosis.

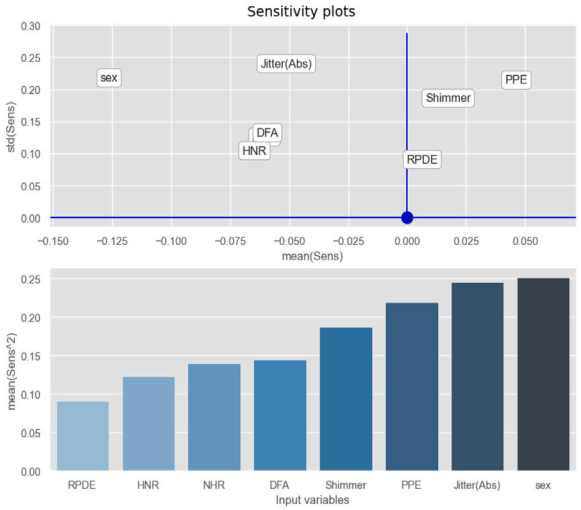
- **Age:** The age of the individual (not used in this analysis).
- **Motor UPDRS:** Unified Parkinson's Disease Rating Scale; Motor section score, a measure of motor function (not used in this analysis).
- **Total UPDRS:** Unified Parkinson's Disease Rating Scale; Total score, a comprehensive measure of disease progression (used as the target variable).

The target variable, in this case, is the Total UPDRS, a quantitative measure of disease progression. Predicting this output based on the selected input variables is an important task for proactive healthcare and disease management, enabling timely interventions that could potentially slow down the disease progression or manage the symptoms more effectively.

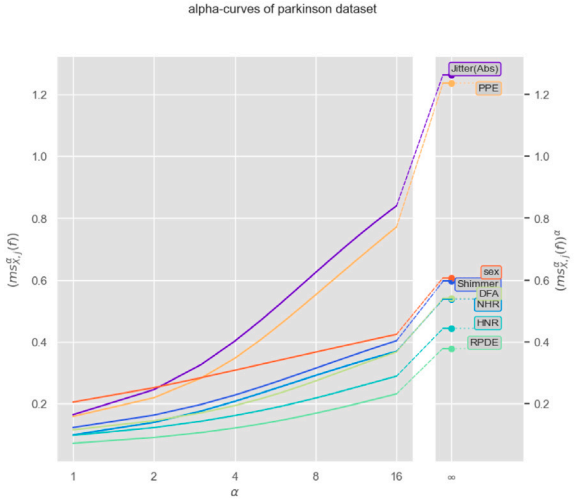
Before training the model, it is crucial to understand the interrelationships among the input variables. A widely used tool for this purpose is the correlation matrix, which provides a numerical and visual representation of how variables interact with each other. The correlation matrix was computed for all the variables present in the Parkinson's Disease Voice Dataset.

Fig. 7 showcases the correlation matrix of the dataset variables. It shows that some variables present high collinearity, which may lead to multicollinearity issues in the model. For example, the sets of Jitter and Shimmer variables were highly correlated among themselves. To mitigate this, only one representative from each set, namely `Jitter(Abs)` and `Shimmer`, were retained for the analysis. These variables were selected based on existing literature, where Chiaramonte and Bonfiglio [57] presents a meta-study of PD diagnosis based on voice measurements. This meta-study suggested that the variables `Shimmer` and `Jitter(Abs)` were the most affected compared to the other Shimmer and Jitter related variables in the revised studies, underscoring their significance in analyzing voice disorders related to Parkinson's Disease.

Additionally, the `Age` variable was excluded from the model as it acted as an identifier for the patients: its values remained constant



(a) Sensitivity plots of Parkinson's Disease dataset. Variables such as `sex`, `Jitter(Abs)`, and `PPE` show significant influence, with `DFA` and noise-related variables (`NHR`, `HNR`, `RPDE`) having minimal impact on the model output.



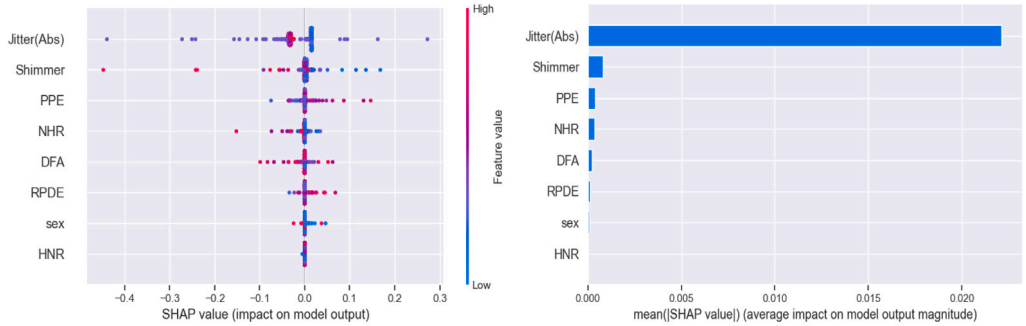
(b) α-curves of Parkinson's Disease dataset. These curves show the importance of variables like `Jitter(Abs)` and `PPE` across patient cohorts, highlighting localized relevance not captured by other methods. The consistent curve for `sex` indicates its stable contribution to the output.

Fig. 8. Comparative sensitivity and α-curve analyses of the Parkinson's Disease dataset. These plots highlight the importance of key variables like `Jitter(Abs)`, PPE, and `sex` in predicting Parkinson's Disease progression. Sensitivity analysis provides a global view of variable relevance, while α-curves reveal localized patterns of variable importance, offering insights into cohort-specific impacts not captured by other methods.

during the trials, hence providing information to the model that is not expected to be present in unseen data.

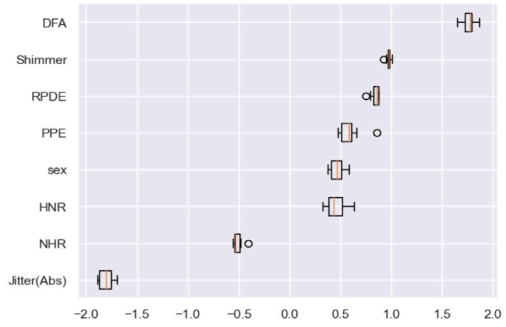
To model the relationship between these input variables and the target variable, a Multi-Layer Perceptron (MLP) was employed. The MLP obtained using 10-fold cross validation consisted of a single hidden layer with 15 neurons and a sigmoid activation function. It was trained with a learning rate of 0.001 for 500 epochs.

Following the methodology for sensitivity analysis, Fig. 8(a) reveals a non-linear correlation among all variables in predicting Parkinson's Disease (PD) progression. The most crucial variables identified are the patient's sex (`sex`), absolute jitter (`Jitter (Abs)`), and Pitch Period Entropy (PPE), followed by `Shimmer`. Conversely, `DFA` and the noise-related variables `NHR`, `HNR` and `RPDE` are deemed less significant, with `DFA` being the most significant by a little margin. According to literature, Azadi et al. [58] determines that jitter emerged as a pivotal parameter for differentiating PD patients due to its capability to measure frequency changes from cycle to cycle in speech signals. It also highlights the sex of the individual (`sex`) as a crucial variable in this analysis. They found substantial differences in speech characteristics between men and women, thus necessitating a segregated analysis for male-only and female-only populations. It was observed that the values of the extracted jitter and shimmer features varied distinctly between male and female subjects when compared between PD patients and healthy individuals. This implies that the sex of the individuals significantly impacts the acoustic parameters being studied, hence influencing the diagnostic accuracy. Vizza et al. [59] found that the amplitude variability of the vocal fold vibration, represented by `Shimmer`, exhibit significant variations in PD patients compared to healthy controls. This correlates with the importance in the `Shimmer` variable to diagnose PD based on voice recordings. Regarding PPE, Little et al. [60] found that this measure provides a nuanced assessment of abnormal pitch variations, distinguishing PD-induced dysphonic variations from natural pitch variations. By analyzing pitch on a perceptually-relevant,



(a) SHAP values of Parkinson's Disease dataset. This plot emphasizes non-linear relationships between input variables and the output. `Jitter(Abs)` and `PPE` appear as the most impactful features, with variability in their contributions across patients.

(b) SHAP Importance values of Parkinson's Disease dataset. The average SHAP values confirm the high importance of `Jitter(Abs)`, with `PPE` and `Shimmer` also contributing significantly. Other variables show negligible impact.



(c) Input permutation importance values of Parkinson's Disease dataset. This analysis assigns a high negative importance to `Jitter(Abs)`, contrasting with its significance in the literature. While `DFA` shows the highest importance, the results differ from sensitivity and SHAP analyses.

Fig. 9. SHAP and Input Permutation analyses of the Parkinson's Disease dataset. SHAP value and importance metrics emphasize the non-linear contributions of variables like `Jitter(Abs)`, PPE, and `Shimmer` to the model's predictions. Input permutation analysis highlights discrepancies in variable rankings, including counterintuitive negative importances for some features.

logarithmic scale, PPE more accurately captures the non-Gaussian fluctuations in pitch period variation associated with PD-related dysphonia. This methodological shift offers a more precise tool for analyzing voice disorders in PD, enhancing the diagnosis of the disease. Regarding HNR, NHR and RPDE variables, Lahmiri [61] defends the potential of discerning PD progression through noise-associated variables. Yet, newer studies [62,63] suggests that these metrics are not as significantly influenced by the disease compared to the Jitter, Shimmer, or PPE measures derived from the patients' voice recordings. In the case of DFA, Minamisawa et al. [64] and Kirchner et al. [65] found DFA to be a suitable indicator of PD, although Miranda et al. [66] found this variable not as important as `Jitter` regarding model performance.

Although results from sensitivity analysis are coherent with the literature reviewed, more information can be retrieved using the α–curves methodology presented in this paper. Fig. 8(b) shows that, on a global scale, `sex` is the variable with the highest importance. Nonetheless, the variables `Jitter (Abs)` and PPE have nearly as much relevance as `sex` globally, yet they are more important for specific cohorts of patients to determine the PD progression. Notably, by $\alpha = 4$, they are already the most influential variables, indicating their notable impact across the majority of patients. This trend might be associated with different subtypes of PD, as elucidated in Tsanas and Arora [67], where distinct PD subtypes exhibited differential impacts on a patient's voice frequency. At a local level, both `Jitter (Abs)` and PPE markedly surpass the rest of the variables in importance. Here, the α–curves could be highlighting those patients for whom the PD subtype significantly influences the PPE and `Jitter (Abs)` variables. On the contrary, the small slope of the α-curve of the `sex` variable denotes a comparatively consistent impact of the patient's gender on how PD progression affects the patient's voice capabilities. This is corroborated by Azadi et al. [58], who found that even amidst the presence of gender-specific PD symptoms, voice alterations attributable to PD maintained a consistency within each gender group. With respect to `Shimmer`, its ascending curve reflects a low global-level relevance of the variable, with a subgroup of patients where the importance of this variable is similar to the `sex` variable. This potentially correlates once again with PD subtypes, where an specific group of patients presents deeper symptoms of voice amplitude variability than others. Regarding the rest of the variables, DFA, NHR, HNR, and RPDE all exhibit a lesser impact on the analysis. Among these, the average impact of DFA, NHR, and HNR is similar to that of `Shimmer`, albeit slightly lower, while RPDE demonstrates a significantly lower impact. Interestingly, DFA and NHR show a degree of relevance in certain portions of the dataset, nearly as much as `Shimmer` or `sex`. The curves of these two variables are almost identical and almost parallel to that of `Shimmer`, albeit lying below it, suggesting that the magnitude of the regions (i.e., types of patients) where these three variables are relevant for PD diagnosis might be similar. Conversely, HNR and RPDE are less relevant than the others at any level of analysis, being the two least influential variables in the dataset for PD diagnosis.

Analyzing SHAP results presented in Figs. 9(a) and 9(b), it emphasizes the importance of the `Jitter (Abs)` variable, with a substantial drop in importance for the `Shimmer` variable, and virtually no significance attributed to the remaining variables. However, the summary plot 9(a) mainly showcases a non-linear relationship between all variables and the output, without providing much more information. It does hint at a diverse level of variable contributions across different patients, suggesting that certain variables may have a higher impact on the output for some individuals. It shall be noted that the SHAP analysis operates under the assumption of input variables' independence and local linearity for each sample, which might overlook potential interactions among variables . As a result, this could lead to a scenario where some variables appear irrelevant, whereas their effects may only be discernible when considered in conjunction with other variables.

Fig. 9(c) shows the Input Permutation results. Among these, the high negative importance associated with `Jitter(Abs)` is particularly noteworthy as it contrasts with the reviewed literature, which often underscores the significant positive role of jitter in differentiating Parkinson's Disease (PD) patients. Similarly, `DFA` showcasing the highest importance is not consistent with the revised literature [66], where this variable is often described as having lesser or varied importance across different studies. On the other hand, the positive importance of `Shimmer` aligns well with literature [59], reaffirming its relevance in PD progression analysis. The lower importances of PPE and `sex` hint at a possible oversight of their interactions with other variables or their nuanced influence in PD progression which might not be fully captured in the permutation importance analysis.

Based on the information retrieved from the previous methods, the only common information is the relevance of the `Shimmer` variable to predict PD progression. Importance assigned for the rest of the variables depends on the method used to analyze the model, being sensitivity analysis and the $\alpha-$curves method the most coherent with the reviewed literature.

5.3. Computational cost

Using the NeuralSens package [46], the computational cost of obtaining the Jacobian of a feed forward neural network at a set of samples involves the same number of matrix multiplications as an evaluation of the network at the samples. If f is a model with M features and the dataset has N samples, computing the α-mean sensitivity $\mathrm{ms}^{\alpha}_{\mathcal{X},j}(f)$ of a model f from the set of derivatives of f at the samples has only cost $\mathcal{O}(N)$ for each α, so the cost of computing the α-means of all variables is of the same order as $\mathcal{O}(N)$ model evaluations. It is particularly relevant that this cost is linear in the number of samples and only grows on the number of features at the same rate as the cost of a model evaluation. This is much less than the computational cost of methods like SHAP, which has a computational complexity of $\mathcal{O}(N2^M)$ model evaluations and Input Permutation Importance, with a computational complexity of $\mathcal{O}(M2^N)$ model evaluations.

Furthermore, to evaluate the computational efficiency of SHAP, Input Permutation Importance, and the α-curves method, we conducted a comprehensive experiment involving various configurations of a MLP model with different configurations. To train this MLP model, we have created a synthetic regression dataset with varying number of input features and samples. Specifically, we varied the number of features from 5 to 100 in steps of 5, the number of samples from 100 to 5100 in steps of 100, and the number of neurons in the hidden layer of the MLP model from 10 to 200 in steps of 10. For each configuration, we measured the execution time required to compute the explainability metrics using the three methods. The results of this experiment are shown in Fig. 10.

The experiment was conducted on a laptop equipped with an Intel Core i7-13700H processor, 32 GB of RAM, and running on a 64-bit Windows 11 Enterprise operating system. During the execution of the experiment, it was observed that the computation of SHAP values for more than 20 features was not feasible due to excessive memory usage and processing time, making it impractical for larger feature sets. In contrast, the $alpha-$curves method demonstrated superior scalability with the number of features and consistently outperformed both SHAP and Input Permutation Importance in terms of execution speed. Our method not only scaled efficiently as the number of features increased but also proved to be the fastest among the three, highlighting its effectiveness for large-scale sensitivity analysis in machine learning models.

5.4. Advantages and limitations of α-curves analysis

The experimental comparisons of the α-curves method against established XAI approaches, such as SHAP and Input Permutation Importance, reveal a number of notable advantages and some limitations.

Among its key advantages, α-curves provide both global and local insights in a single framework, allowing practitioners to detect input-space regions where a specific variable may have an unusually high impact, information that pure aggregation approaches might overlook. Furthermore, by examining how sensitivities evolve as α increases, the approach captures non-linear relationships and potential variable interactions, revealing localized high-sensitivity regions that average-based techniques might miss. The method also rests on a robust mathematical foundation by interpreting sensitivities through the lens of metric spaces and differential operator norms, ensuring theoretically sound metrics that directly connect to the distribution of partial derivatives. In terms of computational efficiency, α-curves exhibit a linear complexity with respect to the number of samples and features, making the technique substantially more scalable than alternatives like SHAP whose computational time grows exponentially with dimensionality, becoming unfeasible for large datasets. Finally, because it is a natural extension of derivative-based sensitivity analysis, α-curves can be integrated seamlessly with prior derivative-focused methods, preserving established workflows while adding richer diagnostic capabilities.

Nonetheless, α-curves have certain limitations. First, they are currently designed for scalar regression scenarios with differentiable functions, which constrains their direct applicability to classification problems, non-differentiable models, or settings where analytical derivatives are inaccessible. This limitation would be addressed in future developments of the method where other suitable metrics for discrete input and output variables are used. Second, the observed steep rise in an α-curve can signify either a pronounced non-linear dependence on one variable or an interaction among multiple variables, and the method does not inherently distinguish between these possibilities. Other methods that provide information about interaction between variables may be needed for clarity. Finally, the focus on partial derivatives implies a vulnerability to large derivative values or outliers, which can disproportionately influence the higher end of the α range and thus skew sensitivity results. In general, this outlier effect would be marginal and affect mostly in very high values of alpha, but should be taken into account when analyzing the α-curve shape.

6. Method availability and applicability

The developed α-curves method has been implemented and is readily available in the NeuralSens package [46] for both Python and R programming languages. The code is available for inspection at https://github.com/JaiPizGon/NeuralSens.

In this paper, the utility of the α-curves method has been demonstrated using Multi-Layer Perceptron (MLP) models, showcasing its effectiveness in providing detailed insights into feature importance within neural network-based regression models. However, the utility of this method extends beyond MLP models. It can be applied to any regression model, provided that the partial derivatives of the output with respect to the inputs can be computed. These partial derivatives can be obtained through the use of automatic differentiation tools such as the autograd package from PyTorch [68]. By leveraging these tools, users can apply the α-curves method to gain deeper insights into the behavior and sensitivity of their regression models.

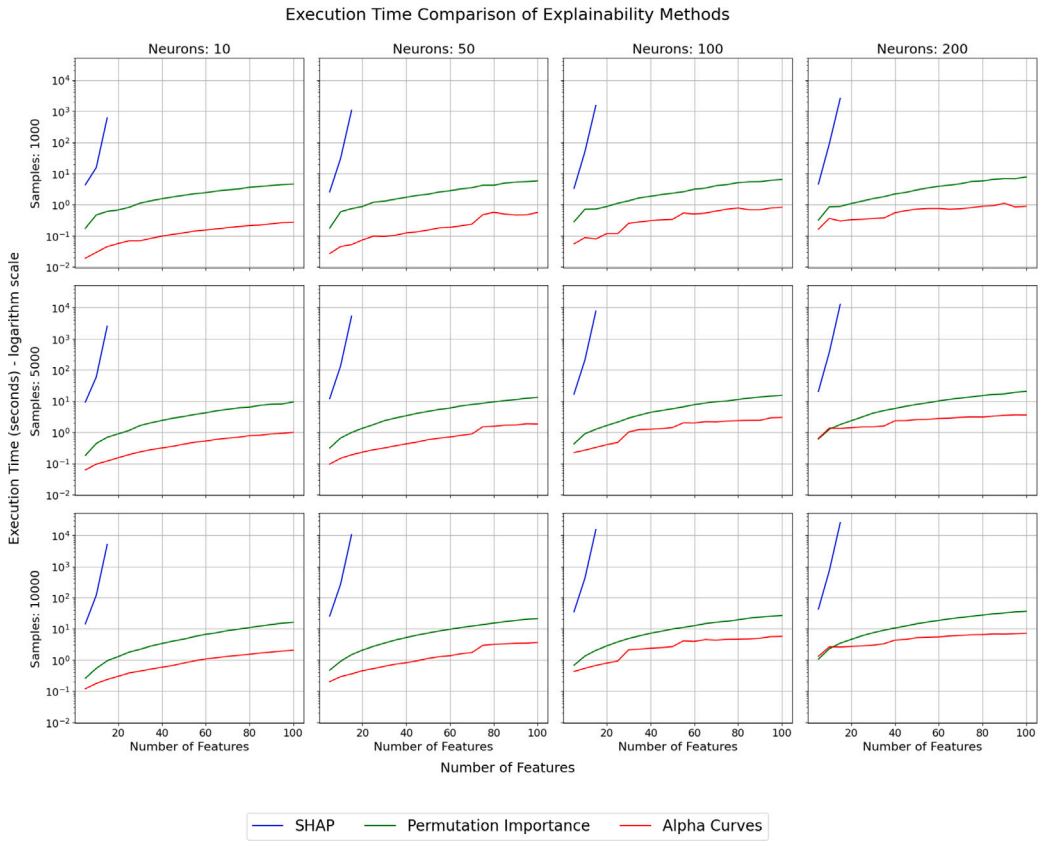


Fig. 10. Execution time comparison between SHAP, Input Permutation Importance, and α-curves XAI methods. This plot evaluates the scalability of each explainability method with respect to the number of features, number of samples, and number of neurons in the model. The y-axis is displayed on a logarithmic scale to better capture the differences in execution times across methods. SHAP exhibits the highest execution time across all scenarios, particularly as the number of features increases, making it unfeasible for high-dimensional datasets. Input permutation importance demonstrates moderate scalability but still shows significant time increases with higher feature counts. In contrast, the α-curves method maintains consistent and significantly lower execution times across varying model complexities and dataset sizes, demonstrating its computational efficiency and suitability for large-scale problems.

7. Conclusions and future work

In this paper, we have proposed a novel XAI method to interpret ML models based on a metric interpretation of partial derivatives. Given a fitted ML model, the sensitivities of the output variable with respect to the inputs provide a relevant measure of the significance of the features in the problem analyzed [50]. However, obtaining a meaningful metric to quantify feature importance based on the sensitivities remained an open issue.

This paper presents one major advantage with respect to previous works in this field. It provides theoretical proof and practical use of the $\alpha-$curves methodology, a novel technique to obtain feature importance information by aggregating the effect of sensitivities across the whole input space. While many existing XAI techniques tend to focus either on global or local explanations, our proposed method bridges the gap between these two levels of interpretation. By providing a coherent framework that seamlessly transitions from global to local interpretations, it ensures that users gain a holistic understanding of feature importance at both levels. This dual-level insight makes it easier for practitioners to trust and validate the model's decisions, fostering a more robust and transparent ML application in real-world scenarios.

In this paper, the $\alpha-$curves methodology is applied to the analysis of Neural Networks. Nevertheless, it can be applied to any model whose partial derivatives can be calculated, allowing for a higher flexibility when facing a ML problem.

The comparative analysis performed against other commonly used XAI methods (SHAP and Input Permutation Importance) in synthetic and real datasets sharply demonstrate the effectiveness of the method to detect relevant features in the datasets. On the one hand, it provides information about the type of relationship between inputs and the output (linear, nonlinear). On the other hand, the method can detect if there exists regions in the input space where certain variables have a greater effect on the output than others. Detecting these regions is crucial, for example, to avoid removing features considered irrelevant when they only have a strong impact on a reduced area of the input space.

Finally, it is worth remarking that the proposed theoretical framework and the α-curves methodology can be applied to both numerical and categorical features, provided that the data model embeds those variables in some $\mathbb{R}^n$. For instance, these methods can be used to study neural networks trained over a dataset containing discrete variables, as they analyze the continuous model (the neural network, which receives real inputs) and not the dataset itself, whose categorical variables are embedded in the real inputs of the network. Nonetheless, using this theoretical framework as a starting point, more intrinsic methodologies for explaining categorical inputs or outputs could be examined in detail in future works by replacing the L^p metrics by other suitable information metrics.

CRediT authorship contribution statement

Jaime Pizarroso: Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation. **David Alfaya:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **José Portela:** Writing – original draft, Validation, Supervision, Resources, Project administration, Investigation. **Antonio Muñoz:** Writing – original draft, Validation, Supervision, Resources, Project administration, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of the main theorem

This section contains the full mathematical proof of the main Theorem 3.1.

Let us start the analysis with some useful notations. Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be a function and let $\mathcal{X} = \{\bar{x}_i\}_{i=1}^N$ be a set of points. Given a perturbation vector $\bar{h} = (h_1, \dots, h_N) \in \mathbb{R}^N$ with the same size as the dataset, denote

$$\Delta_{\bar{h},j}\bar{x}_i = (0, \dots, \overset{j}{\widetilde{h_i}}, \dots, 0) \in \mathbb{R}^n$$

so that for each i

$$(x_{i,1}, \dots, x_{i,j} + h_i, \dots, x_{i,n}) = \bar{x}_i + \Delta_{\bar{h},j}\bar{x}_i.$$

Recall that $d_j f$ denotes the differential of f with respect to variable X_j, i.e., if $f = (f_1, \dots, f_m)$, then $d_j f$ is the $\mathbb{R}^m$-valued differential form

$$d_j f = \left(\frac{\partial f_1}{\partial X_j} dX_j, \dots, \frac{\partial f_m}{\partial X_j} dX_j \right).$$

Finally, let $D_{\mathcal{X},j} f : \mathbb{R}^N \longrightarrow \mathbb{R}^{Nm}$ be the linear operator

$$D_{\mathcal{X},j} = \bigoplus_{i=1}^N d_j f(\bar{x}_i).$$

It is then the operator that for each $\bar{h} \in \mathbb{R}^N$ yields

$$D_{\mathcal{X},j} f(\bar{h}) := \left(d_j f(\bar{x}_1) h_1, \dots, d_j f(\bar{x}_N) h_N \right).$$

We can state the following estimate. Recall that given a linear map between finite dimensional normed spaces $A : (E, \|-\|_E) \longrightarrow (F, \|-\|_F)$ we define the norm of A with respect to the norms of E and F as

$$\|A\|_{\|-\|_E, \|-\|_F} = \max_{\bar{x} \neq 0} \frac{\|A\bar{x}\|_F}{\|\bar{x}\|_E} = \max_{\|\bar{x}\|_E = 1} \|A\bar{x}\|_F.$$

Recall that $\|-\|_H$ denotes a norm on $\mathbb{R}^N$ and $\|-\|_Y$ is a norm on $\mathbb{R}^{mN}$.

Theorem A.1. *Let f be a C^2 function. Then*

$$s_{\mathcal{X},j}(f) = \|D_{\mathcal{X},j} f\|_{\|-\|_H, \|-\|_Y}.$$

Proof. By Taylor's Theorem, at each $\bar{x}_i \in \mathcal{X}$

$$f(\bar{x}_i + \Delta_{\bar{h},j}\bar{x}_i) - f(\bar{x}_i) = d_j f(\bar{x}_i) h_i + r_i(h_i),$$

where $r_i(h_i) = o(|h_i|)$. Thus, we have

$$v_{\mathcal{X},j}(f, \bar{h}) = \|(d_j f(\bar{x}_i) h_i)_{i=1}^N + (r(h_i))_{i=1}^N\|_Y = \|D_{\mathcal{X},j} f(\bar{h}) + r(\bar{h})\|_Y,$$

where $r(\bar{h}) = (r_1(h_1), \dots, r_N(h_N))$. By triangular inequality, we have

$$\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y - \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y \leq \sup_{\|\bar{h}\|_H = \varepsilon} v_{\mathcal{X},j}(f, \bar{h}) \leq \sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y + \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y$$

so

$$\lim_{\varepsilon \to 0} \frac{\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y - \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y}{\varepsilon} \leq s_{\mathcal{X},j}(f) \leq \lim_{\varepsilon \to 0} \frac{\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y + \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y}{\varepsilon}.$$

Observe that, by linearity of $D_{\mathcal{X},j} f(\bar{h})$,

$$\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y = \varepsilon \|D_{\mathcal{X},j} f\|_{\|-\|_H, \|-\|_Y}.$$

On the other hand, as all norms on $Y = \mathbb{R}^{mN}$ are equivalent, we have that $r(\bar{h}) = o(\|\bar{h}\|_Y)$, so

$$\lim_{\varepsilon \to 0} \frac{\sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y}{\varepsilon} = 0$$

and, therefore, we obtain that

$$\lim_{\varepsilon \to 0} \frac{\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y - \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y}{\varepsilon} = \lim_{\varepsilon \to 0} \frac{\sup_{\|\bar{h}\|_H = \varepsilon} \|D_{\mathcal{X},j} f(\bar{h})\|_Y + \sup_{\|\bar{h}\|_H = \varepsilon} \|r(\bar{h})\|_Y}{\varepsilon} = \|D_{\mathcal{X},j} f\|_{\|-\|_H, \|-\|_Y}$$

and the Theorem follows. □

Using this operator norm framework allows us to analyze explicitly the cases where the norms $\|-\|_H$ and $\|-\|_Y$ are L^p-norms and to prove the main Theorem 3.1. We will split the proof depending on which norm corresponds to an L^p norm with the highest value of p.

From this point on, we will assume that $\|-\|_H$ is an L^p-norm and that $\|-\|_Y$ is an L^q norm.

Proof of the main theorem when $p = q$

Theorem A.2. *Let f be a C^2 function. If $\|-\|_H$ and $\|-\|_Y$ are the L^p norms, then*

$$s_{\mathcal{X},j}(f) = \max_i \left\{ \left\| d_j f(\bar{x}_i) \right\|_p \right\}.$$

Proof. By Theorem A.1, we have

$$s_{\mathcal{X},j}(f) = \|D_{\mathcal{X},j} f\|_{p,p}.$$

Thus, we have to compute

$$\max_{\|\bar{h}\|_p = 1} \|D_{\mathcal{X},j} f\|_p = \max_{\|\bar{h}\|_p = 1} \left(\sum_{i=1}^N \sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p |h_i|^p \right)^{1/p}$$

under the constraint $\sum_i |h_i|^p = 1$. Changing the variable $H_i = |h_i|^p$ and rising to power p the optimized function yields the following linear optimization problem.

$$\max_{\substack{\sum_i H_i = 1 \\ H_i \geq 0 \, \forall i}} \sum_{i=1}^N \left(\sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p \right) H_i$$

which is attained by taking H_i to be 1 when its coefficient is the biggest possible and 0 in any other case. Thus

$$\max_{\substack{\sum_i H_i = 1 \\ H_i \geq 0 \, \forall i}} \sum_{i=1}^N \left(\sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p \right) H_i = \max_i \sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p$$

and, therefore,

$$\max_{\|\bar{h}\|_p = 1} \|D_{\mathcal{X},j} f\|_p = \left(\max_i \sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p \right)^{1/p} = \max_i \left(\sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^p \right)^{1/p} = \max_i \|d_j f(\bar{x}_i)\|_p.$$ □

Proof of the main theorem when $p > q$

Theorem A.3. *Let f be a C^2 function. If $\|-\|_H = \|-\|_p$ and $\|-\|_Y = \|-\|_q$ with $p > q$, then*

$$s_{\mathcal{X},j}(f) = \left(\sum_{i=1}^N \left(\sum_{k=1}^m \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{\frac{p}{p-q}} \right)^{\frac{p-q}{pq}}.$$

Proof. As before, by Theorem A.1, we have

$$s_{\mathcal{X},j}(f) = \|D_{\mathcal{X},j} f\|_{p,q}.$$

We have to compute

$$\|D_{\mathcal{X},j} f\|_{p,q} = \left(\max_{\|\bar{h}\|_p=1} \sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right) |h_i|^q \right)^{1/q}.$$

Changing the variable $H_i = |h_i|^q$, setting $\bar{H} = (H_1, \dots, H_N)$ and rising to power q the optimized function, we need to find the maximum of

$$\sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right) H_i$$

under the constraint $\|\bar{H}\|_{p/q} = 1$. Taking $p' = \frac{p}{p-q}$ and $q' = p/q$ (so that $\frac{1}{p'} + \frac{1}{q'} = 1$) and applying Hölder's Inequality with norms $L^{p'}$ and $L^{q'}$ yields

$$\sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right) H_i \leq \left\| \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)_{i=1}^{N} \right\|_{p'} \|\bar{H}\|_{q'}$$

$$= \left\| \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)_{i=1}^{N} \right\|_{p'}$$

with equality when $h_i \propto \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{p'/q'}$. Finally, taking the qth root of this quantity yields

$$\|D_{\mathcal{X},j} f\|_{p,q} = \left(\left\| \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)_{i=1}^{N} \right\|_{p'} \right)^{1/q}$$

$$= \left(\sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{p'} \right)^{q/p'}$$

obtaining the desired result. □

Proof of the main theorem when $p < q$

Theorem A.4. *Let f be a C^2 function. If $\|-\|_H = \|-\|_p$ and $\|-\|_Y = \|-\|_q$ with $p < q$, then*

$$s_{\mathcal{X},j}(f) = \max_i \left\{ \left\| d_j f(\bar{x}_i) \right\|_q \right\}.$$

Proof. By Theorem A.1, we have

$$s_{\mathcal{X},j}(f) = \|D_{\mathcal{X},j} f\|_{p,q}.$$

Thus, we have to compute

$$\|D_{\mathcal{X},j} f\|_{p,q} = \left(\max_{\|\bar{h}\|_p=1} \sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right) |h_i|^q \right)^{1/q}.$$

Changing the variable $H_i = |h_i|^p$, setting $\bar{H} = (H_1, \dots, H_N)$ and rising to power q the optimized function, we need to find the maximum of

$$\sum_{i=1}^{N} \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right) H_i^{q/p}$$

under the constraint $\sum_i H_i = 1$, with $H_i \geq 0$ for all i. As $q/p > 1$, the objective functional is a convex function on H_i and, thus, it attains its maximum value at one of the vertices of the domain. It is then clear that the maximum is attained taking $H_i = 1$ precisely where the coefficient $\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q$ attains its maximum value and $H_i = 0$ for the rest of the values, so

$$\max_{\|\bar{h}\|_p=1} \|D_{\mathcal{X},j} f\|_q = \left(\max_i \sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{1/q}$$

$$= \max_i \left(\sum_{k=1}^{m} \left| \frac{\partial f_k}{\partial X_j}(\bar{x}_i) \right|^q \right)^{1/q} = \max_i \|d_j f(\bar{x}_i)\|_q. \quad \square$$

Appendix B. Pseudocode for α-curves analysis

Below is the pseudocode for performing the α-curves analysis on pre-computed derivatives to quantify the feature importances from both global and local perspectives. These derivatives could be from any machine learning model. Package `neuralsens`, which includes functions to perform the α-curves method analysis, directly computes these partial derivatives for multi layer perceptron (MLP) models.

Algorithm 1 ALPHA_SENS_CURVES: Compute α-Curves

Require: Jacobian $\in \mathbb{R}^{N \times n}$: Jacobian of machine learning model.
Require: `max_`α: maximum alpha to analyze. Defaults to 16.
Require: `step_`α: increment between consecutive alphas. Defaults to 1.

$A \leftarrow \{1,\ 1 + \texttt{step\_}\alpha,\ 1 + 2 \cdot \texttt{step\_}\alpha, \dots,\ \texttt{max\_}\alpha\}$
α_curves $\leftarrow$ empty 2D array of shape $(|A|,\ n)$
for $j \leftarrow 0$ to $n - 1$ **do**
  for $t \leftarrow 0$ to $|A| - 1$ **do**
    $\alpha \leftarrow A[t]$
    α_mean $\leftarrow \left(\frac{1}{N} \sum_{i=1}^{N} \left| \text{Jacobian}[i, j] \right|^{\alpha} \right)^{\frac{1}{\alpha}}$
    α_curves$[t, j] \leftarrow \alpha$_mean
  end for
end for
return α_curves

Explanation of the steps:.

- *Input Data.* The matrix Jacobian $\in \mathbb{R}^{N \times n}$ contains pre-computed partial derivatives of the model's output with respect to each input variable. Specifically, Jacobian$[i, j]$ is $\frac{\partial \hat{y}_i}{\partial X_j} = \frac{\partial f}{\partial X_j}(\bar{x}_i)$, for sample index i and feature index j.
- *Alpha Range.* An array A is constructed to hold the discrete α values from 1 to `max_`α, spaced by `step_`α. Each $\alpha \in A$ corresponds to a particular way of aggregating partial derivatives (α-mean).
- *Initialize α-Curves.* An empty 2D array, α_curves, of shape $(|A|, n)$ is allocated to store, for each feature j (columns) and each α (rows), the aggregated sensitivity metric.
- *Double Loop.*
  - Outer loop: for$j \leftarrow 0$ to $n - 1$. Iterates over each feature j.
  - Inner loop: for$t \leftarrow 0$ to $|A| - 1$. Iterates over each α value in A.
  - Inside the inner loop, the α-mean quantity is computed as:

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left| \text{Jacobian}[i, j] \right|^{\alpha} \right)^{\frac{1}{\alpha}}$$

  - The α-mean is stored in the corresponding $[t, j]$ element of the α_curves object.
- *Return.* The function returns α_curves, a matrix whose rows represent different values of α and whose columns represent different features. This can be plotted to visualize how feature sensitivities evolve from global (small α) to local (large α).

Data availability

Data will be made available on request.

References

[1] P.S. Kohli, S. Arora, Application of machine learning in disease prediction, in: 2018 4th International Conference on Computing Communication and Automation, ICCCA, 2018, pp. 1–4, http://dx.doi.org/10.1109/CCAA.2018.8777449.

[2] F. Shamout, T. Zhu, D.A. Clifton, Machine learning for clinical outcome prediction, IEEE Rev. Biomed. Eng. 14 (2020) 116–126, http://dx.doi.org/10.1109/RBME.2020.3007816.

[3] M. Rashid, B.S. Bari, Y. Yusup, M.A. Kamaruddin, N. Khan, A comprehensive review of crop yield prediction using machine learning approaches with special emphasis on palm oil yield prediction, IEEE Access 9 (2021) 63406–63439, http://dx.doi.org/10.1109/ACCESS.2021.3075159.

[4] R. Zhang, L. Wang, S. Cheng, S. Song, MLP-based classification of COVID-19 and skin diseases, Expert Syst. Appl. 228 (2023) 120389, http://dx.doi.org/10.1016/j.eswa.2023.120389.

[5] J. Dean, Google research: Themes from 2021 and beyond, 2022, URL https://ai.googleblog.com/2022/01/google-research-themes-from-2021-and.html.

[6] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.-R. Muller, Explaining deep neural networks and beyond: A review of methods and applications, Proc. IEEE Inst. Electr. Electron. Eng. 109 (3) (2021) 247–278.

[7] W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Vol. 11700, Springer Nature, 2019.

[8] F. Cabitza, A. Campagner, G. Malgieri, C. Natali, D. Schneeberger, K. Stoeger, A. Holzinger, Quod erat demonstrandum? - Towards a typology of the concept of explanation for the design of explainable AI, Expert Syst. Appl. 213 (2023) 118888, http://dx.doi.org/10.1016/j.eswa.2022.118888.

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

[10] R. Benjamins, A. Barbado, D. Sierra, Responsible AI by design in practice, 2019, http://dx.doi.org/10.48550/ARXIV.1909.12838, arXiv preprint arXiv:1909.12838. URL https://arxiv.org/abs/1909.12838.

[11] A. Preece, D. Harborne, D. Braines, R. Tomsett, S. Chakraborty, Stakeholders in explainable AI, 2018, http://dx.doi.org/10.48550/ARXIV.1810.00184, arXiv preprint arXiv:1810.00184.

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

[13] C.-H. Cheng, F. Diehl, G. Hinz, Y. Hamza, G. Nuehrenberg, M. Rickert, H. Ruess, M. Truong-Le, Neural networks for safety-critical applications — challenges, experiments and perspectives, in: 2018 Design, Automation and Test in Europe Conference and Exhibition, 2018, pp. 1005–1006, http://dx.doi.org/10.23919/DATE.2018.8342158.

[14] A. Galli, M.S. Piscitelli, V. Moscato, A. Capozzoli, Bridging the gap between complexity and interpretability of a data analytics-based process for benchmarking energy performance of buildings, Expert Syst. Appl. 206 (2022) 117649, http://dx.doi.org/10.1016/j.eswa.2022.117649.

[15] D. Pawade, A. Dalvi, J. Gopani, C. Kachaliya, H. Shah, H. Shah, XAI—An approach for understanding decisions made by neural network, in: Recent Trends in Communication and Intelligent Systems, Springer Singapore, Singapore, 2021, pp. 155–165.

[16] L.S. Shapley, A value for n-person games, Princet. Univ. Press (1953) 307–318.

[17] J. Tritscher, M. Ring, D. Schlr, L. Hettinger, A. Hotho, Evaluation of post-hoc XAI approaches through synthetic tabular data, in: Foundations of Intelligent Systems, Springer International Publishing, Cham, 2020, pp. 422–430.

[18] S. Hariharan, R.R. Rejimol Robinson, R.R. Prasad, C. Thomas, N. Balakrishnan, XAI for intrusion detection system: comparing explanations based on global and local scope, J. Comput. Virol. Hacking Tech. (2022) http://dx.doi.org/10.1007/s11416-022-00441-2.

[19] G.D. Garson, Interpreting neural-network connection weights, AI Expert 6 (4) (1991) 46–51, http://dx.doi.org/10.5555/129449.129452.

[20] J.D. Olden, M.K. Joy, R.G. Death, An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data, Ecol. Model. 178 (3–4) (2004) 389–397, http://dx.doi.org/10.1016/j.ecolmodel.2004.03.013.

[21] I. Dimopoulos, P. Bourret, S. Lek, Use of some sensitivity criteria for choosing networks with good generalization ability, Neural Process. Lett. 2 (1995) 1–4, http://dx.doi.org/10.1007/bf02309007.

[22] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecol. Model. 160 (3) (2003) 249–264, http://dx.doi.org/10.1016/S0304-3800(02)00257-0.

[23] M. Gevrey, I. Dimopoulos, S. Lek, Two-way interaction of input variables in the sensitivity analysis of neural network models, Ecol. Model. 195 (1) (2006) 43–50, http://dx.doi.org/10.1016/j.ecolmodel.2005.11.008.

[24] G.S. Lumacad, J.V.C. Damasing, S.B.M. Tacastacas, A.R.T. Quipanes, Analyzing sensitive factors affecting online academic performance in the new normal: A machine learning perspective, in: 2022 XVII Latin American Conference on Learning Technologies, LACLO, 2022, pp. 01–07, http://dx.doi.org/10.1109/LACLO56648.2022.10013373.

[25] K.K. Gelaye, F. Zehetner, C. Stumpp, E.G. Dagnew, A. Klik, Application of artificial neural networks and partial least squares regression to predict irrigated land soil salinity in the Rift Valley Region, ethiopia, J. Hydrol.: Reg. Stud. 46 (2023) 101354, http://dx.doi.org/10.1016/j.ejrh.2023.101354.

[26] X. Wen, Y. Xie, L. Jiang, Z. Pu, T. Ge, Applications of machine learning methods in traffic crash severity modelling: current status and future directions, Transp. Rev. 41 (6) (2021) 855–879.

[27] S. Rajabi, Z. Derakhshan, A. Nasiri, M. Feilizadeh, A. Mohammadpour, M. Salmani, S.H. Kochaki, H. Shouhanian, H. Hashemi, Synergistic degradation of metronidazole and penicillin g in aqueous solutions using AgZnFe2O4@ chitosan nano-photocatalyst under UV/persulfate activation, Environ. Technol. Innov. 35 (2024) 103724.

[28] C. Yang, J. Xu, Tropical cyclone wind field reconstruction for hazard estimation via Bayesian hierarchical modeling with neural network, Earth Space Sci. 11 (12) (2024) e2024EA003678.

[29] J.L. Arroyo-Barrigüete, C. Escudero-Guirado, B. Minguela-Rata, Factors influencing the social perception of entrepreneurs in Spain: A quantitative analysis from secondary data, PLoS One 18 (12) (2023) e0296095.

[30] J.M. Ortiz-Lozano, P. Aparicio-Chueca, X.M. Triadó-Ivern, J.L. Arroyo-Barrigüete, Early dropout predictors in social sciences and management degree students, Stud. High. Educ. 49 (8) (2024) 1303–1316.

[31] E. Durand-Cartagena, J. Jaramillo, Pointwise Lipschitz functions on metric spaces, J. Math. Anal. Appl. 363 (2) (2010) 525–548, http://dx.doi.org/10.1016/j.jmaa.2009.09.039.

[32] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2014, arXiv preprint arXiv:1312.6199.

[33] A. Polo-Molina, D. Alfaya, J. Portela, A mathematical certification for positivity conditions in neural networks with applications to partial monotonicity and ethical AI, 2024, arXiv:2406.08525. URL https://arxiv.org/abs/2406.08525.

[34] T. Fel, D. Vigouroux, R. Cadène, T. Serre, How good is your explanation? Algorithmic stability measures to assess the quality of explanations for deep neural networks, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 720–730.

[35] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller, G. Montavon, Toward explainable artificial intelligence for regression models: A methodological perspective, IEEE Signal Process. Mag. 39 (4) (2022) 40–58, http://dx.doi.org/10.1109/MSP.2022.3153277.

[36] D. Minh, H.X. Wang, Y.F. Li, T.N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artif. Intell. Rev. 55 (5) (2022) 3503–3568, http://dx.doi.org/10.1007/s10462-021-10088-y.

[37] T. Speith, A review of taxonomies of explainable artificial intelligence (XAI) methods, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2239–2250, http://dx.doi.org/10.1145/3531146.3534639.

[38] K. Bykov, M.M.C. Höhne, K.-R. Müller, S. Nakajima, M. Kloft, How much can I trust you? – Quantifying uncertainties in explaining neural networks, 2020, arXiv preprint arXiv:2006.09000.

[39] C. Kästner, Interpretability and explainability, 2021, URL https://ckaestne.medium.com/interpretability-and-explainability-a80131467856.

[40] M. Scardi, L.W. Harding, Developing an empirical model of phytoplankton primary production: A neural network case study, Ecol. Model. 120 (2–3) (1999) 213–223, http://dx.doi.org/10.1016/s0304-3800(99)00103-9.

[41] I. Dimopoulos, J. Chronopoulos, A. Chronopoulou-Sereli, S. Lek, Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens City (Greece), Ecol. Model. 120 (2–3) (1999) 157–165, http://dx.doi.org/10.1016/s0304-3800(99)00099-x.

[42] A. Muñoz, T. Czernichow, Variable selection using feedforward and recurrent neural networks, Eng. Intell. Syst. Electr. Eng. Commun. 6 (2) (1998) 91–102.

[43] H. White, J. Racine, Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates, IEEE Trans. Neural Netw. 12 (4) (2001) 657–673, http://dx.doi.org/10.1109/72.935080.

[44] E. Strumbelj, I. Kononenko, An efficient explanation of individual classifications using game theory, J. Mach. Learn. Res. 11 (2010) 1–18.

[45] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774.

[46] J. Pizarroso, J. Portela, A. Muñoz, NeuralSens: Sensitivity analysis of neural networks, J. Stat. Softw. 102 (7) (2022) 1–36, http://dx.doi.org/10.18637/jss.v102.i07.

[47] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, arXiv preprint arXiv:1312.6034.

[48] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[49] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, 2017, arXiv preprint arXiv:1706.03825.

[50] H. White, J. Racine, Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates, IEEE Trans. Neural Netw. 12 (4) (2001) 657–673.

[51] I.-C. Yeh, W.-L. Cheng, First and second order sensitivity analysis of MLP, Neurocomputing 73 (10–12) (2010) 2225–2233.

[52] X. Zeng, Z. Zhen, J. He, L. Han, A feature selection approach based on sensitivity of RBFNNs, Neurocomputing (2017).

[53] I. Learning Lab, The california housing dataset, 2022, URL https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html.

[54] A. Tsanas, M. Little, Parkinsons Telemonitoring, 2009, UCI Machine Learning Repository.

[55] H. Baniecki, W. Kretowicz, P. Piatyszek, J. Wisniewski, P. Biecek, Dalex: Responsible machine learning with interactive explainability and fairness in python, J. Mach. Learn. Res. 22 (214) (2021) 1–7, URL http://jmlr.org/papers/v22/20-1473.html.

[56] R. Kelley Pace, R. Barry, Sparse spatial autoregressions, Statist. Probab. Lett. 33 (3) (1997) 291–297, http://dx.doi.org/10.1016/S0167-7152(96)00140-X.

[57] R. Chiaramonte, M. Bonfiglio, Acoustic analysis of voice in Parkinson's disease: a systematic review of voice disability and meta-analysis of studies., Rev. de Neurol. 70 (11) (2020) 393–405.

[58] H. Azadi, M.-R. Akbarzadeh-T, A. Shoeibi, H.R. Kobravi, Evaluating the effect of Parkinson's disease on jitter and shimmer speech features, Adv. Biomed. Res. 10 (2021).

[59] P. Vizza, G. Tradigo, D. Mirarchi, R.B. Bossio, N. Lombardo, G. Arabia, A. Quattrone, P. Veltri, Methodologies of speech analysis for neurodegenerative diseases evaluation, Int. J. Med. Inform. 122 (2019) 45–54.

[60] M. Little, P. McSharry, E. Hunter, J. Spielman, L. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease, Nat. Preced. (2008) 1–1.

[61] S. Lahmiri, Parkinson's disease detection based on dysphonia measurements, Phys. A 471 (2017) 98–105.

[62] S.S. Upadhya, A. Cheeran, Investigation of pitch and noise features extracted from voice samples of healthy and Parkinson affected people using statistical tests, J. Eng. Sci. Technol. 13 (9) (2018) 2792–2804.

[63] T. Romero Arias, I. Redondo Cortés, A. Pérez Del Olmo, Biomechanical parameters of voice in Parkinson's disease patients, Folia Phoniatr. et Logop.: Off. Organ the Assoc. Logop. Phoniatr. (IALP) 10 (2023) 000533289.

[64] T. Minamisawa, K. Takakura, T. Yamaguchi, Detrended fluctuation analysis of temporal variation of the center of pressure (COP) during quiet standing in Parkinsonian patients, J. Phys. Ther. Sci. 21 (3) (2009) 287–292.

[65] M. Kirchner, P. Schubert, M. Liebherr, C.T. Haas, Detrended fluctuation analysis and adaptive fractal analysis of stride time data in Parkinson's disease: stitching together short gait trials, PLoS One 9 (1) (2014) e85787.

[66] J.A.Z. Miranda, J.E.C. Pillajo, F.L.G. Arévalo, J.D.V. Sánchez, Selección de funciones de voz mediante algoritmos genéticos para la detección de la enfermedad de Parkinson, Rev. de Investigación Tecnolog´ i As de la Información 10 (21) (2022) 140–150.

[67] A. Tsanas, S. Arora, Data-driven subtyping of Parkinson's using acoustic analysis of sustained vowels and cluster analysis: Findings in the Parkinson's voice initiative study, SN Comput. Sci. 3 (3) (2022) 232.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, 2019, arXiv:1912.01703.