

Functional relevance based on the continuous Shapley value

Pedro Delicado and Cristian Pachón-García
 Departament d'Estadística i Investigació Operativa
 Universitat Politècnica de Catalunya

June 6, 2025

Abstract

The presence of artificial intelligence (AI) in our society is increasing, which brings with it the need to understand the behavior of AI mechanisms, including machine learning predictive algorithms fed with tabular data, text or images, among others. This work focuses on interpretability of predictive models based on functional data. Designing interpretability methods for functional data models implies working with a set of features whose size is infinite. In the context of scalar on function regression, we propose an interpretability method based on the Shapley value for continuous games, a mathematical formulation that allows for the fair distribution of a global payoff among a continuous set of players. The method is illustrated through a set of experiments with simulated and real data sets. The open source Python package **ShapleyFDA** is also presented.

Keywords: interpretability, explainability, functional data analysis, continuous game theory, machine learning.

1 Introduction

Technological advances in recent years have affected data analysis significantly. For instance, modern smart devices are able to monitor health indicators sampling data several times per minute. This way of obtaining data leads to treat them as if they were functions depending on a continuous argument (time, wavelength, etc.). *Functional data analysis* (Ramsay and Silverman, 2005), or FDA for short, is a branch of statistics whose theoretical foundations come from functional analysis (see also Ferraty and Vieu, 2006; Horváth and Kokoszka, 2012; Kokoszka and Reimherr, 2017; Crainiceanu et al., 2024). In particular, prediction (understood in a broad sense) is a relevant problem in FDA. To fix ideas, given a functional explanatory variable (or functional feature) $\mathcal{X}(t)$, $t \in I = [a, b] \subset \mathbb{R}$, and a scalar response variable (or target) Y , one aims to find the best mapping between the functional feature and the target. This problem is known as *scalar-on-function regression*, and it has been addressed not only with statistical prediction methods (see, for example, Reiss et al., 2017; Gertheiss et al., 2024), but also with machine learning algorithms (see, for instance, Yao et al., 2021; Rao and Reimherr, 2023; Gertheiss et al., 2024). We focus on regression (Y continuous) but classification (Y binary or categorical) is handled in a similar way.

Machine learning models, even in the standard case of multivariate features, have enormous flexibility to encode the relationship between the explanatory variables and the response. This typically leads to difficulties in understanding exactly how machine learning algorithms work, justifying the use of the term “black box” to describe them. In the last two decades, a powerful research line (known as interpretable machine learning) has been developed to provide interpretability tools to algorithmic models. There is a considerable number of review papers (see, for instance, Barredo Arrieta et al., 2020) and three monographs on this topic: Biecek and Burzykowski (2021), Masís (2021), and Molnar (2022).

In this work we present a methodology, based on game theory, which provides interpretability to scalar-on-function prediction models. Our proposal can explain any kind of model (it is *agnostic* to the underlying prediction model) and it provides a *global* explanation (it shows how important each value of the functional explanatory variable is in the prediction process, taking into account all the observed data). To the best of our knowledge, this is the first work that addresses global agnostic interpretability for prediction models in the context of FDA. In fact, it is noteworthy that an examination of the intersection between functional data analysis and interpretable machine learning reveals a limited corpus of existing literature: only two works can be found, James et al. (2009), which is not agnostic to the model, and Carrizosa et al. (2024), which is not global.

The structure of this paper is as follows. Section 2 shows the connection between the Shapley value for finite games and interpretability for multiple regression, and introduces the Shapley value for continuous games. Based on it, our interpretability framework is detailed in Section 3, while Section 4 introduces the accompanying open source Python package (Python Core Team, 2021). In order to study and analyze this proposal, we conduct a set of simulations in Section 5, which additionally includes a real data example. Conclusions are provided in Section 6. Some appendixes are available as supplementary material: Appendix A offers a brief introduction to interpretable machine learning, and Appendix B includes extra outputs from simulations.

2 Interpretability based on game theory

This section establishes the connection between the Shapley value for finite games, a concept from game theory, and interpretability for multiple regression. We also introduce the Shapley when the game is continuous, the framework for our proposal in Section 3.

2.1 Finite games

In this Section we follow Winter (2002) to present the Shapley value. The seminal work was published by Shapley (1953). Given a set of players $N = \{1, \dots, n\}$, a *game* is a function $\nu : 2^N \rightarrow \mathbb{R}^+$, where 2^N is the set of all subsets of N and $\nu(\emptyset) = 0$. The mapping ν is known as the *payoff function* and $\nu(S)$ is interpreted as the payoff the coalition S receives for having played that game in a cooperative way. When all the players cooperate, the total payoff is $\nu(N)$. The relevant question in cooperative game theory is to find a fair distribution of $\nu(N)$ among the n players, or rephrased differently, to determine the relevance (the value) of each player in the overall coalition.

Let \mathcal{Q}_N be the set of all games ν with n players. A *value* is an operator $\varphi : \mathcal{Q}_N \rightarrow (\mathbb{R}^+)^n$ that assigns to each game a vector of length n , $\varphi(\nu) = \boldsymbol{\varphi}_\nu \in (\mathbb{R}^+)^n$, where the i -th component of the vector, $\varphi_{\nu,i}$ (or φ_i whenever the game can be omitted), represents the value of the player i when playing game ν . Shapley (1953) defines a set of axioms (or desirable properties) for values:

Efficiency. The sum of all values must equal the total payoff: $\sum_{i=1}^n \varphi_i = \nu(N)$.

Symmetry. Players $i, i' \in N$ are *symmetric* with respect to the game ν if for each $S \subset N$ such that $i, i' \notin S$, $\nu(S \cup \{i\}) = \nu(S \cup \{i'\})$. If this is the case then $\varphi_i = \varphi_{i'}$.

Dummy players. Player i is a *dummy player* (with respect to the game ν) if for every $S \subset N$, $\nu(S \cup \{i\}) - \nu(S) = 0$. If player i is a dummy player, then $\varphi_i = 0$.

Additivity. $\varphi(\nu + \omega) = \varphi(\nu) + \varphi(\omega)$, where ν and ω are games, and game $\nu + \omega$ is defined as $(\nu + \omega)(S) = \nu(S) + \omega(S)$.

Shapley (1953) proves that there is a unique value (now known as the *Shapley value* of the game) which satisfies the previous axioms, and that it is given by

$$\varphi_{\nu,i} = \frac{1}{n!} \sum_{\pi \in \Pi} [\nu(p_{\pi}^i \cup \{i\}) - \nu(p_{\pi}^i)], \quad (1)$$

where Π is the set of all permutations of the set N , π is a permutation and $p_{\pi}^i = \{j: \pi(i) > \pi(j)\}$ is the set of players preceding player i in permutation π . The quantity $[\nu(p_{\pi}^i \cup \{i\}) - \nu(p_{\pi}^i)]$ is the marginal contribution of player i to the coalition p_{π}^i and its Shapley value, $\varphi_{\nu,i}$, is the average of these marginal contributions over the possible permutations of N .

2.2 Interpretability based on finite games

Consider a multiple linear regression model with m individuals and p explanatory variables. Let $\mathbf{x}_1, \dots, \mathbf{x}_p$ and \mathbf{y} be, respectively, the vectors of predictors' values and responses, all in \mathbb{R}^m . Let $\bar{y} = 1/m \sum_{j=1}^m y_j$ and let \hat{y}_j be the j -th fitted value by ordinary least squares (OLS). The coefficient of determination, $R^2 = 1 - \sum_{j=1}^m (y_j - \hat{y}_j)^2 / \sum_{j=1}^m (y_j - \bar{y})^2$, is commonly used to measure the overall quality of the estimated model. When the p explanatory variables are uncorrelated, $R^2 = \sum_{i=1}^p R_i^2$, where R_i^2 is the coefficient of determination in the simple linear regression of \mathbf{y} against the i -th explanatory variable \mathbf{x}_i fitted by OLS. Therefore, R_i^2 is the contribution of \mathbf{x}_i to the global quality measure R^2 , and it is a good measure of the relevance of \mathbf{x}_i in the model. See, for instance, Grömping (2009).

The preceding decomposition of R^2 is not applicable when the explanatory variables are correlated. Lipovetsky and Conklin (2001) propose an alternative decomposition based on the Shapley value. Specifically, the authors propose to consider the p explanatory variables as the set of players and the game ν as the coefficient of determination R^2 . Consequently, $\nu(S)$ is the coefficient of determination $R^2(S)$ in the regression of \mathbf{y} against the variables belonging to S fitted by OLS. Therefore, the Shapley value of this game is a fair distribution of the total R^2 among the p predictors: $R^2 = \sum_{i=1}^p \varphi_{\nu,i}$, and $\varphi_{\nu,i}$ measures the importance of the i -th regressor in the model. Given that the exact computation of Shapley values is quite time intensive, the authors suggest to average over a moderate number of random permutations of the explanatory variables. Feldman (2005) extends the use of the Shapley value to parametric statistical or econometric models, as a measure of the relative importance of each variable. Cohen et al. (2007) propose to use the Shapley value as global measure of variable relevance in classification problems using any prediction model (or algorithm). They propose to use the accuracy in a test set as the game ν .

2.3 Continuous games

The primary objective of this work is to incorporate interpretability within the context of FDA. Therefore, it is considered to extend the previous relevance measure based on the Shapley value to prediction models with scalar response and functional regressor. In this context the set of players $\mathcal{X}(t)$, $t \in I$, is infinite, or, equivalently, the game becomes continuous. Thus, the Shapley value must be defined for *continuous games* (or *games with a continuum of players*: Shapley, 1961; Aumann, 1964). There exists a collection of publications devoted to this topic (Kannai, 1966; Aumann and Shapley, 1974; Hart and Neyman, 1988; Neyman, 1994; among others). Obtaining the Shapley value for a continuous game requires a more sophisticated mathematical framework than for a finite game.

Let $I = [0, 1]$ and $\mathcal{B} = \mathcal{B}(I)$ be the associated Borel σ -algebra. A (continuous) *game* is a function $\nu: \mathcal{B} \rightarrow \mathbb{R}^+$ such that $\nu(\emptyset) = 0$. I is called *the set of players*, \mathcal{B} *the set of coalitions*, (I, \mathcal{B}) *the space of players* and ν *the payoff function*. Although we work with $I = [0, 1]$, the forthcoming development is valid for any interval $[a, b] \subset \mathbb{R}$.

A game is said *monotonic* if $T \subset S$ implies $\nu(T) \leq \nu(S)$. Let Q be a collection of monotonic games in (I, \mathcal{B}) . Q is a *linear space* over (I, \mathcal{B}) if $\nu_1 + \nu_2 \in Q$ for any ν_1 and ν_2 , both in Q . An *automorphism* in (I, \mathcal{B}) is a one-to-one measurable function θ from I onto I with θ^{-1} measurable. Let \mathcal{G} be the group of automorphisms of (I, \mathcal{B}) . Given a game ν , an automorphism $\theta \in \mathcal{G}$ defines a new game from ν : $(\theta_*\nu)(S) = \nu(\theta(S))$ for all $S \in \mathcal{B}$. Q is *symmetric* if $\theta_*\nu \in Q$ for all $\nu \in Q$ and all $\theta \in \mathcal{G}$. A measure μ on (I, \mathcal{B}) is a monotonic game with the property of σ -additivity: for all countable collections $\{E_j\}_{j \geq 1}$ of pairwise disjoint sets in \mathcal{B} , $\mu(\cup_{j \geq 1} E_j) = \sum_{j \geq 1} \mu(E_j)$. Let \mathcal{M} be the set of measures defined on (I, \mathcal{B}) . Note that \mathcal{M} itself is a symmetric and linear subset of monotonic games.

A *value* on a linear symmetric space of monotonic games Q is a mapping $\psi : Q \rightarrow \mathcal{M}$ satisfying the following properties (Aumann and Shapley, 1974, page 16):

Efficiency. For all $\nu \in Q$, $[\psi(\nu)](I) = \nu(I)$.

Symmetry. For all $\nu \in Q$ and for all $\theta \in \mathcal{G}$, $\theta_*\psi(\nu) = \psi(\theta_*\nu)$.

Linearity. For all $\nu_1, \nu_2 \in Q$ and for all $\alpha, \beta \in \mathbb{R}^+$, $\psi(\alpha\nu_1 + \beta\nu_2) = \alpha\psi(\nu_1) + \beta\psi(\nu_2)$.

While the concepts of efficiency and linearity are relatively straightforward to interpret, the notion of symmetry is more abstract. According to Aumann and Shapley (1974), it means that “the value does not depend on how the players are named”. In essence, this property states that the value of ν is preserved by automorphisms.

In Aumann and Shapley (1974), the authors prove that, under certain assumptions on the linear symmetric space Q of monotonic games defined on (I, \mathcal{B}) , there exists a unique value ψ satisfying the efficiency, symmetry and linearity properties. Nevertheless, deriving a formula in the continuous games context is not as straightforward as it is in the finite case. The literature referenced at the beginning of this section presents two distinct methodologies for addressing this issue. The first one, known as *the axiomatic approach*, considers only a very specific subset of games for which a closed formula can be found. The so called *vector measure games* are an example of this situation. Let $\tau = (\tau_1, \dots, \tau_K)$ be a vector of K non-atomic measures defined on \mathcal{B} , with $\tau_k(I) \neq 0$. Let ρ be a real valued continuously differentiable function defined on \mathbb{R}^K with $\rho(0, \dots, 0) = 0$. A vector measure game has the form $\nu = \rho \circ \tau$. That is, $\nu(S) = \rho(\tau_1(S), \dots, \tau_K(S))$, so the payoff of coalition S only depends on a finite number K of measures of S . It can be proved that there exist an efficient, symmetric, and linear value function ψ for these games given by

$$\psi(\rho \circ \tau)(S) = \sum_{k=1}^K \tau_k(S) \int_0^1 \frac{\partial \rho}{\partial x_k}(t\tau_1(I), \dots, t\tau_K(I)) dt$$

which is called *the diagonal formula*. Further details are provided in Section 3.2 of Neyman (1994). This approach, however, is not generally applicable in the context of the present paper, as it is shown in Section 3.3.

The second approach, *the asymptotic approach*, considers a given continuous game as the limit of a sequence of finite games. This, in turns, allows computing the sequence of (finite) Shapley values. Therefore, the limit value for that sequence is the Shapley value for the continuous game. We consider to use the asymptotic approach because it is not as restrictive as the axiomatic approach. Our development is based on Kannai (1966) and Neyman (1994). Let $\nu : \mathcal{B} \rightarrow \mathbb{R}^+$ be a game, let $\mathcal{I} = \{I_1, \dots, I_q\} \subset \mathcal{B}$ be a partition of I , let \mathcal{P} be the σ -algebra generated by \mathcal{I} , and let $\nu_{\mathcal{P}}$ be the game restricted to \mathcal{P} , that is, $\nu_{\mathcal{P}} : \mathcal{P} \rightarrow \mathbb{R}^+$ and $\nu_{\mathcal{P}}(S) = \nu(S)$ for all $S \in \mathcal{P}$. As $\nu_{\mathcal{P}}$ is a finite game, the Shapley value described in Section 2.1, $\varphi(\nu_{\mathcal{P}})$, can be obtained.

In order to define the limit value, we need to consider a sequence of finer and finer partitions. Let $S \in \mathcal{B}$, a *S-admissible* sequence is an increasing sequence of subalgebras $(\mathcal{P}_j)_{j \in \mathbb{N}}$ with $S \in \mathcal{P}_1 \subset \mathcal{P}_2 \subset \dots \subset \mathcal{P}_k \subset \dots$ such that $\cup_{j \in \mathbb{N}} \mathcal{P}_j$ generates \mathcal{B} . $\psi(\nu)$ is *the asymptotic value of ν* if and

only if for every $S \in \mathcal{B}$ and any S -admissible sequence $(\mathcal{P}_j)_{j \in \mathbb{N}}$, $\lim_{j \rightarrow \infty} [\varphi(\nu_{\mathcal{P}_j})](S) = [\psi(\nu)](S)$. Under certain conditions, it can be proved that the limit exists, it is unique and it satisfies the properties previously described (Neyman, 1994). Therefore, under those conditions, the value for a continuous game can be obtained as the limit of the Shapley value for finite games.

3 Relevance based on the continuous Shapley value

In this section, we present a framework for defining a relevance function for prediction models with scalar response Y and a functional predictor \mathcal{X} with trajectories in $L^2(I) = \{g : I \rightarrow \mathbb{R} : \int_I g(t)^2 dt < \infty\}$, where $I = [0, 1]$ (for simplicity, we use $[0, 1]$ instead a generic interval $[a, b]$). The goal is to rank the points t in $I = [0, 1]$ according to their importance when predicting the response variable using a particular prediction model. We make use of the asymptotic approach described in Section 2.3.

Let \mathcal{X} and Y have a certain joint probability distribution, from which a test data set is available: (\mathcal{X}_j, y_j) , $j \in \{1, \dots, m\}$, independently drawn from (\mathcal{X}, Y) . Additionally, it is assumed that one has access to an already trained prediction model, $f : L^2(I) \rightarrow \mathbb{R}$, and that the data used to train such a model are independent of the test data.

One of the key elements is how to define the game $\nu : \mathcal{B} \rightarrow \mathbb{R}^+$ which will allow us to find the *Shapley value relevance function*. For $S \in \mathcal{B}$, roughly speaking $\nu(S)$ should be the proportion of variability of $\mathbf{y} = (y_1, \dots, y_m)^\top$ in the test sample explained by the prediction model f when only the information of S is considered.

A first attempt to compute $\nu(S)$ could be to retrain the model only considering the points $t \in S$, but this strategy would eventually be computationally very expensive as it would need to be performed for each $S \in \mathcal{B}$. This approach has certain similarities to the *leave-one-covariate-out* (LOCO) method for feature relevance in prediction models with a finite number of predictors (for details, see Appendix A in the supplementary material). Delicado and Peña (2023) propose a computationally cheaper alternative to LOCO: instead of leaving one covariate out, they replace it by its *ghost variable*, which is its conditional expectation given the rest of covariates. A related approach is employed in this work.

Specifically, we propose the creation of a new functional data set in which the data from $t \in S$ are retained while the remaining data are inferred from S . The new data set is created using the conditional expectation, that is,

$$\tilde{\mathcal{X}}_j^S(t) = \mathcal{X}_j(t) \cdot \mathbb{1}_S(t) + \dot{\mathcal{X}}_j(t) \cdot \mathbb{1}_{S^c}(t), \quad (2)$$

with

$$\dot{\mathcal{X}}_j(t) = \mathbb{E}(\mathcal{X}(t) \mid \{\mathcal{X}(u) = \mathcal{X}_j(u) : u \in S\}),$$

where $S^c = I \setminus S$ and $j \in \{1, \dots, m\}$. This way, the functions $\tilde{\mathcal{X}}_j^S(t)$ are evaluated in all $t \in I$ and they can be used as arguments of the already trained prediction model f . Conditional expectations are estimated under the assumption that \mathcal{X} is a Gaussian process. Details are given in Section 3.1, along with a discussion of its parallels with the approach of Kneip and Liebl (2020) to optimal reconstruction of partially observed functional data.

Next, we define $\tilde{y}_j^S = f(\tilde{\mathcal{X}}_j^S)$, $j \in \{1, \dots, m\}$, and $\nu(S)$ is defined as the coefficient of determination, computed as

$$\nu(S) = \tilde{R}^2(S) = 1 - \frac{\sum_{j=1}^m (y_j - \tilde{y}_j^S)^2}{\sum_{j=1}^m (y_j - \bar{y})^2}, \quad (3)$$

where \bar{y} is the average of the target (or response) in the test set. Observe that $\tilde{R}^2(I)$ coincides with the standard $R^2(I)$ because $\tilde{\mathcal{X}}_j^I = \mathcal{X}_j$ in Equation (2).

Figure 1 shows an example of how the new data set is obtained. In the left panel, there is a data set with 100 functions $\mathcal{X}_j(t)$ $j \in \{1, \dots, m = 100\}$. Details on how this data set was

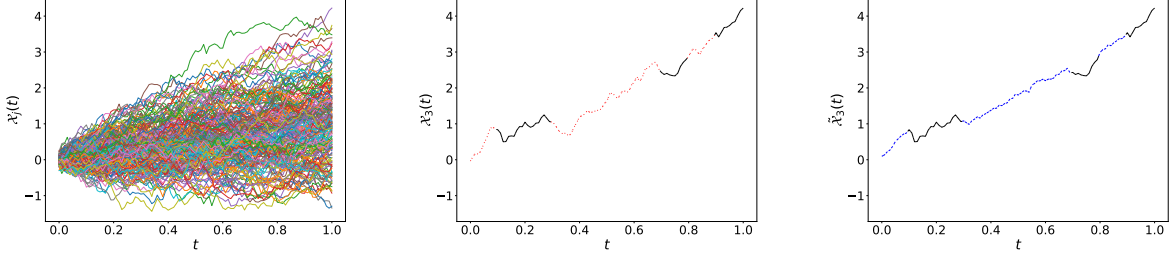


Figure 1: Example of a reconstructed function. (Left) Functional data set. (Middle) One functional data. Solid lines indicate intervals in S where the functional data is known, while dotted lines correspond to S^c where the functional data is unknown. (Right) Reconstructed function. The values of the functional data for the intervals in S^c (dashed lines) are computed as their conditional expectation given the values at the intervals in S (solid lines).

generated are given in Section 5.1. The middle plot is one of these functions. The intervals that are part of S are shown in solid lines, while those that are part of S^c are shown in dotted lines. The right side of the figure shows the reconstructed function $\tilde{\mathcal{X}}_j^S(t)$, where the dashed lines indicates those intervals for which conditional expectation was used.

Recall that our goal is to make use of the asymptotic approach to compute the Shapley value of the game ν . Therefore, let $\mathcal{I} = \{I_1, \dots, I_n\}$ be a partition of $I = [0, 1]$, where $I_i = [a_i, b_i)$, for $i \in \{1, \dots, n-1\}$, $I_n = [a_n, b_n]$, $a_1 = 0$, $b_n = 1$ and $b_{i-1} = a_i < b_i$ for $i \in \{2, \dots, n\}$. Let \mathcal{P} be the σ -algebra generated by \mathcal{I} . In the context of game theory, \mathcal{I} (or, equivalently, $N = \{1, \dots, n\}$) can be thought of as the set of players. Therefore, we define the finite game $\nu_{\mathcal{P}} : \mathcal{P} \rightarrow \mathbb{R}^+$ as the previous game ν restricted to \mathcal{P} .

The objective is to compute the Shapley value for a given subset I_i (player i) using the formula stated in Equation (1). Let Π the set of all permutations of \mathcal{I} , let $\pi \in \Pi$ and let p_{π}^i the set of players preceding i in π . By using Equation (2) it is not necessary to retrain a model for each different permutation.

In this context, the role of S is assumed by p_{π}^i (and then S^c is $\mathcal{I} \setminus p_{\pi}^i$). This, in turn, allows computing $\nu_{\mathcal{P}}(p_{\pi}^i)$. The same line of reasoning can be applied to $p_{\pi}^i \cup I_i$. Using the Shapley value for finite games, the relevance of the interval I_i is given by

$$\varphi_{\nu,i} = \frac{1}{n!} \sum_{\pi \in \Pi} [\nu_{\mathcal{P}}(p_{\pi}^i \cup I_i) - \nu_{\mathcal{P}}(p_{\pi}^i)] = \frac{1}{n!} \sum_{\pi \in \Pi} [\tilde{R}^2(p_{\pi}^i \cup I_i) - \tilde{R}^2(p_{\pi}^i)]. \quad (4)$$

In practice, as the size of the set of all permutations Π could be extremely large, a random subset of permutation $\Pi_0 \subset \Pi$ is employed.

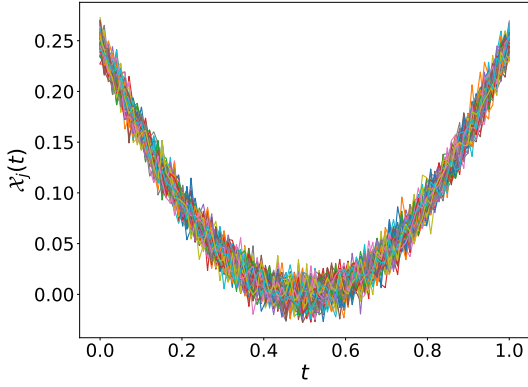
Applying the previous formula to each interval leads to obtain an n dimensional vector $\boldsymbol{\varphi} = (\varphi_{\nu,1}, \dots, \varphi_{\nu,n})^T$. So the *Shapley value relevance function* is the density function of the histogram-type measure defined by vector $\boldsymbol{\varphi}$:

$$\mathcal{R}_f(t) = \sum_{i=1}^n \frac{\varphi_{\nu,i}}{b_i - a_i} \mathbb{1}_{I_i}(t).$$

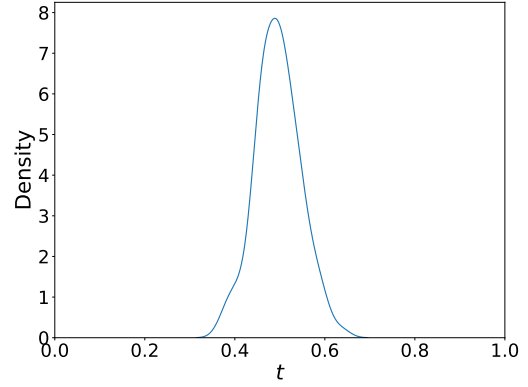
The Shapley value relevance function can also be reported as the polygonal function defined by the midpoints of the steps in $\mathcal{R}_f(t)$ and its boundary values.

An illustrative example is shown in Figure 2. On the top left panel, there is a functional data set with 100 curves. The interval considered is $I = [0, 1]$ and each function is generated as $\mathcal{X}(t) = (t - 0.5)^2 + \varepsilon(t)$ for t in a fine grid of points, where $\varepsilon(t) \sim N(\mu = 0, \sigma = 0.01)$ and $\varepsilon(t_1)$ is independent of $\varepsilon(t_2)$ if $t_1 \neq t_2$. The target variable is generated as $Y = \Upsilon(\mathcal{X}) = \min_{t \in I} [\mathcal{X}(t)]$. The top right panel of Figure 2 shows the density of $\text{argmin}[\mathcal{X}(t)]$.

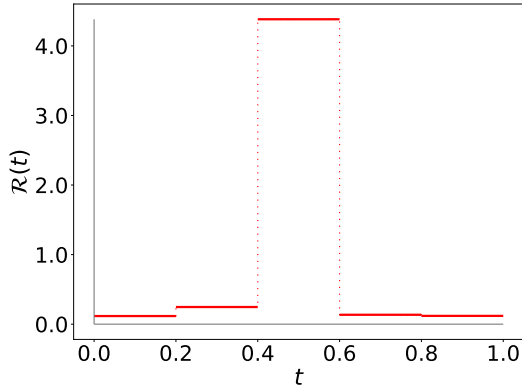
Let us suppose that I is divided into 5 parts (of the same length), that is, the players considered are I_1, \dots, I_5 . Since most of the density of $\text{argmin}[\mathcal{X}(t)]$ is in the interval $[0.4, 0.6]$,



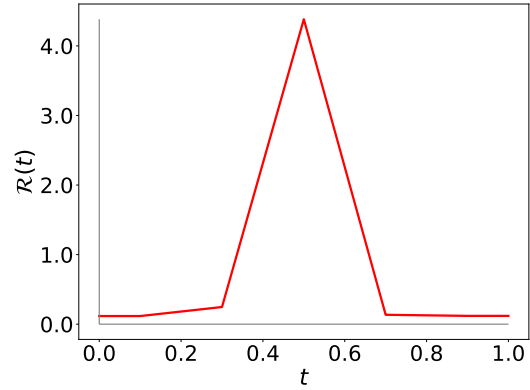
(a) Functional data set that is generated according to $\mathcal{X}(t) = (t - 0.5)^2 + \varepsilon(t)$.



(b) Estimated density of $\text{argmin}[\mathcal{X}(t)]$. The target variable is generated as $Y = \min_{t \in I} [\mathcal{X}(t)]$.



(c) Histogram-type Shapley value relevance function.



(d) Polygonal-type Shapley value relevance function.

Figure 2: Illustrative example of the Shapley value relevance function, which is able to detect that $[0.4, 0.6]$ is the most relevant interval.

when using a prediction model with a good predictive capacity, it is expected that the Shapley value relevance function assigns the majority of the relevance to player $I_3 = (0.4, 0.6]$. Let us use the true regression function as predictive function: $f(\mathcal{X}) = \Upsilon(\mathcal{X})$. The bottom left and right panels of Figure 2 depict respectively the histogram-type and the polygonal-type Shapley value relevance functions, which have the expected behavior.

3.1 Estimating conditional expectations

In this section we describe how $\dot{\mathcal{X}}_j(t) = \mathbb{E}(\mathcal{X}(t) \mid \{\mathcal{X}(u) = \mathcal{X}_j(u) : u \in S\})$ is estimated, $1 \leq j \leq m$. For this purpose, it is convenient to assume that \mathcal{X} is a Gaussian process observed at the set of points $\mathcal{T} = \{t_1, \dots, t_T\}$. Then, $(\mathcal{X}(t_1), \dots, \mathcal{X}(t_T))$ follows a multivariate normal distribution (with mean $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$) from which there are m independent observations. Let \mathbf{X} be the resulting $m \times T$ data matrix.

Let S be the subset of I which is assumed to be available (or observed). Let $\mathcal{T}_S = S \cap \mathcal{T}$ and let \mathcal{J}_S be the corresponding indexes in $\{1, \dots, T\}$. We refer to \mathcal{J}_S^c as the complementary set of \mathcal{J}_S . For the sake of clarity, we use the labels “O” for the *observed* set \mathcal{J}_S and “M” for the *missing* set \mathcal{J}_S^c . The estimated variance covariance matrix from \mathbf{X} , $\hat{\boldsymbol{\Sigma}}$, can be rearrange as

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{MM} & \hat{\boldsymbol{\Sigma}}_{MO} \\ \hat{\boldsymbol{\Sigma}}_{OM} & \hat{\boldsymbol{\Sigma}}_{OO} \end{bmatrix}.$$

Let $\hat{\boldsymbol{\mu}}_O$ (respectively, $\hat{\boldsymbol{\mu}}_M$) be the estimation of the mean vector for those columns whose

indexes belong to \mathcal{J}_S (respectively, \mathcal{J}_S^c). Let \mathbf{x}_{jO} be the vector that results from selecting the j -th row of \mathbf{X} and those columns whose indexes belong to \mathcal{J}_S . Therefore, the standard theory of multivariate normal distributions states that

$$\{\hat{\mathcal{X}}_j(t_h) : h \in \mathcal{J}_S^c\} = \hat{\mathbb{E}}(\{\mathcal{X}(t_h) : h \in \mathcal{J}_S^c\} \mid \{\mathcal{X}(t_d) = \mathcal{X}_j(t_d) : d \in \mathcal{J}_S\}) = \hat{\boldsymbol{\mu}}_M + \hat{\boldsymbol{\Sigma}}_{MO} \hat{\boldsymbol{\Sigma}}_{OO}^{-1} (\mathbf{x}_{jO} - \hat{\boldsymbol{\mu}}_O). \quad (5)$$

For the cases where $\hat{\boldsymbol{\Sigma}}_{OO}$ is not invertible, instead of $\hat{\boldsymbol{\Sigma}}_{OO}^{-1}$ we use $\hat{\boldsymbol{\Sigma}}_{OO}^+$, the Moore–Penrose pseudo-inverse of $\hat{\boldsymbol{\Sigma}}_{OO}$ computed by diagonalizing it, replacing all non-zero eigenvalues with their inverses, and leaving the zero eigenvalues as they are. It is worth remembering that Equation (5) gives also the expression of the least squares linear prediction of $\mathcal{X}_j(t_h)$, $h \in \mathcal{J}_S^c$, given the values $\mathcal{X}_j(t_d)$, $d \in \mathcal{J}_S$, even if multivariate normality is not assumed.

In terms of execution time, it is desirable to provide an expression to compute $\{\hat{\mathcal{X}}_j(t_h) : h \in \mathcal{J}_S^c\}$ for all $1 \leq j \leq m$ simultaneously. The matrix-wise expression is given by

$$\mathbf{1}_m \hat{\boldsymbol{\mu}}_M^\top + (\mathbf{X}_O - \mathbf{1}_m \hat{\boldsymbol{\mu}}_O^\top) \hat{\boldsymbol{\Sigma}}_{OO}^{-1} \hat{\boldsymbol{\Sigma}}_{OM},$$

where $\mathbf{1}_m$ is a vector of length m full of 1's and \mathbf{X}_O is the matrix derived by selecting those columns whose indexes belong to \mathcal{J}_S from \mathbf{X} .

3.2 Reconstructing partially observed functional data

The definition of a complete function $\tilde{\mathcal{X}}_j^S$ from \mathcal{X}_j , when only the values $\mathcal{X}_j(u)$ for $u \in S$ are known, is closely related to the problem of reconstructing partially observed functional data, which has received certain attention in the FDA literature. See, for instance, Goldberg et al. (2014); Kraus (2015); Kneip and Liebl (2020). In this last paper, the authors give the theoretical expression of the optimal linear reconstruction operator, which minimizes the pointwise mean squared prediction error for $\mathcal{X}_j(u)$, for any $u \in S^c$:

$$\hat{\mathcal{X}}_j(u) = \mu(u) + \sum_{k=1}^{\infty} \xi_{jk}^O \frac{\langle \gamma_u, \phi_k^O \rangle}{\lambda_k^O}, \quad (6)$$

where $\mu(t) = \mathbb{E}(\mathcal{X}(t))$, $t \in I$, is assumed to be known, (λ_k^O, ϕ_k^O) , $k \geq 1$, are the pairs of non-zero eigenvalues and eigenfunctions of the covariance operator of the functional data restricted to the observed set S (corresponding to the covariance function $\gamma^O(t, s) = \text{Cov}(\mathcal{X}(t), \mathcal{X}(s))$, t and s in S), $\langle \cdot, \cdot \rangle$ denotes the inner product in $L^2(S)$, μ^O and \mathcal{X}_j^O are the restrictions of μ and \mathcal{X}_j on S , respectively, $\xi_{jk}^O = \langle \phi_k^O, \mathcal{X}_j^O - \mu^O \rangle$ is the score of \mathcal{X}_j^O in the k -th principal function computed from $\gamma^O(\cdot, \cdot)$, and $\gamma_u(t) = \text{Cov}(\mathcal{X}(u), \mathcal{X}(t))$, t in S .

Let us show that the completion formula given by Equation (5) is a feasible version of the optimal reconstruction operator (6). We call $(\hat{\lambda}_k^O, \hat{\phi}_k^O)$, $k = 1, \dots, r_O$, the pairs of non-zero eigenvalues and eigenvectors of the sampling covariance matrix $\hat{\boldsymbol{\Sigma}}_{OO}$. Then

$$\hat{\boldsymbol{\Sigma}}_{OO} = \sum_{k=1}^{r_O} \hat{\lambda}_k^O \hat{\phi}_k^O (\hat{\phi}_k^O)^T, \text{ and } \hat{\boldsymbol{\Sigma}}_{OO}^+ = \sum_{k=1}^{r_O} \frac{1}{\hat{\lambda}_k^O} \hat{\phi}_k^O (\hat{\phi}_k^O)^T.$$

Let $\hat{\xi}_{jk}^O = (\hat{\phi}_k^O)^T (\mathbf{x}_{jO} - \hat{\boldsymbol{\mu}}_O)$ be the score of \mathbf{x}_{jO} in the k -th principal component computed from $\hat{\boldsymbol{\Sigma}}_{OO}$. Consider an index $h \in \mathcal{J}_S^c$ and let $u = t_h$. Let $\hat{\gamma}_u^T$ be the corresponding row of $\hat{\boldsymbol{\Sigma}}_{MO}$, which contains estimations of $\text{Cov}(\mathcal{X}(u), \mathcal{X}(t_d))$, d in \mathcal{J}_S . The row of the expression in Equation (5) corresponding to $u = t_h$ is

$$\hat{\mathcal{X}}_j(u) = \hat{\mu}(u) + \hat{\gamma}_u^T \sum_{k=1}^{r_O} \frac{1}{\hat{\lambda}_k^O} \hat{\phi}_k^O (\hat{\phi}_k^O)^T (\mathbf{x}_{jO} - \hat{\boldsymbol{\mu}}_O) = \hat{\mu}(u) + \sum_{k=1}^{r_O} \hat{\xi}_{jk}^O \frac{\hat{\gamma}_u^T \hat{\phi}_k^O}{\hat{\lambda}_k^O},$$

which coincides with Equation (6), once the unknown elements are estimated.

3.3 Functional relevance is not based on a vector measure game

In this section we show that, even in quite simple situations, the continuous games used for defining functional relevance are not vector measure games. Therefore, we can not expect to find a generic closed form expression (as the diagonal formula mentioned in Section 2.3) for the relevance function. Assume that the observed data are (\mathcal{X}_j, y_j) , $j \in \{1, \dots, m\}$, where the responses y_j are generated by a linear functional model: $Y = \int_0^1 \beta(t) \mathcal{X}(t) dt + \varepsilon$, with $\mathbb{E}(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma^2$. We fit a linear scalar-on-function regression model to estimate the functional coefficient $\beta(t)$. Assume that the estimation is so precise that we can operate as if $\beta(t)$ is known.

Consider the payoff function $\nu(S)$ defined in Equation (3), which defines the continuous game used to find the relevance function for the fitted scalar-on-function regression model. Observe that $\nu(S)$ depends on S only through

$$\begin{aligned} \frac{1}{m} \sum_{j=1}^m (y_j - \tilde{y}_j^S)^2 &= \frac{1}{m} \sum_{j=1}^m \left(\int_0^1 \beta(t) \mathcal{X}_j(t) dt + \varepsilon_j - \int_0^1 \beta(t) \tilde{\mathcal{X}}_j^S(t) dt \right)^2 \\ &= \frac{1}{m} \sum_{j=1}^m \left(\int_{S^c} \beta(u) (\mathcal{X}_j(u) - \dot{\mathcal{X}}_j^S(u)) du + \varepsilon_j \right)^2 \\ &\approx \text{Var} \left(\int_{S^c} \beta(u) (\mathcal{X}(u) - \dot{\mathcal{X}}^S(u)) du + \varepsilon \right) \\ &= \text{Var} \left(\int_{S^c} \beta(u) \mathcal{E}^S(u) du \right) + \sigma^2 \\ &= \int_{S^c} \int_{S^c} c_{\mathcal{E}}^S(u, v) \beta(u) \beta(v) du dv + \sigma^2 \\ &= \int_{S^c} \int_{S^c} \sum_{k=1}^{\infty} \lambda_k^S \xi_k^S(u) \xi_k^S(v) \beta(u) \beta(v) du dv + \sigma^2 \\ &= \sum_{k=1}^{\infty} \lambda_k^S \left(\int_{S^c} \xi_k^S(u) \beta(u) du \right)^2 + \sigma^2, \end{aligned}$$

where $\mathcal{E}^S = \mathcal{X} - \dot{\mathcal{X}}^S$ is the reconstruction error (which is assumed to have zero mean; see Kneip and Liebl, 2020, Theorem 2.2), $c_{\mathcal{E}}^S(u, v)$ is the covariance function of \mathcal{E}^S , and (λ_k^S, ξ_k) , $k \geq 1$, are the eigenvalues and eigenfunctions of the covariance operator of \mathcal{E}^S .

Let τ_k^S , $k \geq 1$, be signed measures defined as $\tau_k^S(B) = \int_B \xi_k^S(u) \beta(u) du$ for any Borel set $B \subseteq I$, and let τ_k^{S+} and τ_k^{S-} be the positive and negative parts of τ_k^S . We have shown that $\nu(S)$ is approximately equal to a function of

$$\sum_{k=1}^{\infty} \lambda_k^S \tau_k^S(S^c)^2 = \sum_{k=1}^{\infty} \lambda_k^S (\tau_k^S(I) - \tau_k^S(S))^2.$$

Therefore $\nu(S) \approx \rho_S(\{\tau_k^{S+}(S), \tau_k^{S-}(S) : k \geq 1\})$ for a function ρ_S defined on $\mathbb{R}^{\mathbb{N}}$. We conclude that, in general, the payoff function ν used to define functional relevance does not correspond to a vector measure game because: (i) the function ρ^S depends on a countably infinite set of measure functions, and (ii) ρ^S , as well as the measures τ_k^{S+} and τ_k^{S-} , depend on the set S on which ν is evaluated.

4 The ShapleyFDA package

An open source Python package, **ShapleyFDA**, has been released to PyPI to compute the Shapley value relevance function according to the methodology explained in this work. Given that the

calculations are inherently matrix-intensive, the fundamental component of **ShapleyFDA** is based on the **NumPy** package (Harris et al., 2020).

As a large volume of computations must be considered, our implementation stores the intermediate results in memory for future reference. For instance, given permutations $\pi_1 = (I_3, I_4, I_1, I_2)$ and $\pi_2 = (I_3, I_4, I_2, I_1)$ the set of players preceding player 1 in π_1 is the same as the set of players preceding 2 in π_2 . Upon the initial computation of the value for $p_{\pi_1}^1 = \{I_3, I_4\}$, it is stored. This approach guarantees that the same value will not be calculated twice, but rather used as many times as necessary.

ShapleyFDA assumes that the prediction model has already been trained. Furthermore, the package is designed so that it can handle multiple prediction models simultaneously to obtain the Shapley value relevance function for each of them. Consequently, the same set of random permutations is used when obtaining multiple Shapley value relevance functions. Moreover, the input test data set must be provided in matrix format. The package incorporates functionalities to display both figures, the histogram-type function and the polygonal-type function. Apart from the released version in PyPI, there is also a development version available at GitHub (<https://github.com/pachoning/ShapleyFDA>).

Now let us derive the complexity of our proposed algorithm. We consider Equation (4), where a random subset of permutations $\Pi_0 \subset \Pi$ is employed. First, we obtain the cost for a fixed permutation $\pi \in \Pi_0$, and then we consider the whole set of permutations. In order to compute $\tilde{R}^2(p_\pi^i \cup I_i)$, or $\tilde{R}^2(p_\pi^i)$, it is needed to first obtain $\tilde{\mathcal{X}}_j^S(t)$, as stated in Equation (2). All the details on how to compute $\tilde{\mathcal{X}}_j^S(t)$ are provided in Section 3.1. Of particular interest is the matrix-wise expression

$$\mathbf{1}_m \hat{\boldsymbol{\mu}}_M^\top + (\mathbf{X}_O - \mathbf{1}_m \hat{\boldsymbol{\mu}}_O^\top) \hat{\boldsymbol{\Sigma}}_{OO}^{-1} \hat{\boldsymbol{\Sigma}}_{OM}, \quad (7)$$

where subindex “O” refers to the T_O *observed* columns of \mathbf{X} , and subindex “M” refers to the remaining T_M *missing* columns. The computational complexity of Equation (7) is

$$\mathcal{O}(mT_O) + \mathcal{O}(mT_M) + \mathcal{O}(T_O^3) + \mathcal{O}(mT_O^2) + \mathcal{O}(mT_M T_O) = \mathcal{O}(T_O^3) + \mathcal{O}(mT_O^2) + \mathcal{O}(mT_M T_O),$$

where we use that the computational complexity of matrix multiplication of two matrices of sizes $s_1 \times s_2$ and $s_2 \times s_3$ is $\mathcal{O}(s_1 s_2 s_3)$, and that the computational complexity of matrix inversion of a matrix of size $s_4 \times s_4$ is $\mathcal{O}(s_4^3)$ (see, for instance, Farebrother, 1988, page 12). Observe that T_O and T_M are random variables (they depend on the random permutation π) and that $\mathbb{E}(T_O) = \mathbb{E}(T_M) = T/2$. Then, the expected computational complexity of Equation (7) is $\mathcal{O}(T^3) + \mathcal{O}(mT^2)$.

Let $\mathcal{O}_f(m, T)$ be the complexity of running the prediction algorithm for a matrix of size $m \times T$. Considering the n players and the $|\Pi_0|$ permutations, the global expected complexity of our proposal is $\mathcal{O}(T^3 |\Pi_0| n) + \mathcal{O}(mT^2 |\Pi_0| n) + \mathcal{O}_f(m, T) \mathcal{O}(|\Pi_0| n)$.

5 Experiments

Two types of experiments are conducted with the objective of showing the performance of our proposed methodology. In the first type, described in Section 5.1, we simulate data and then use three different prediction models to predict the response variable. Since we control the data generation process, we know in advance which points $t \in I$ are important. The second type, which is explained in Section 5.2, consists of exploring the Tecator data set (Borggaard and Thodberg, 1992). The code utilized in the experimental configuration is accessible via GitHub (github.com/pachoning/shapley_fda_experiments). Simulated data are also available at <https://bit.ly/4crYVt1>.

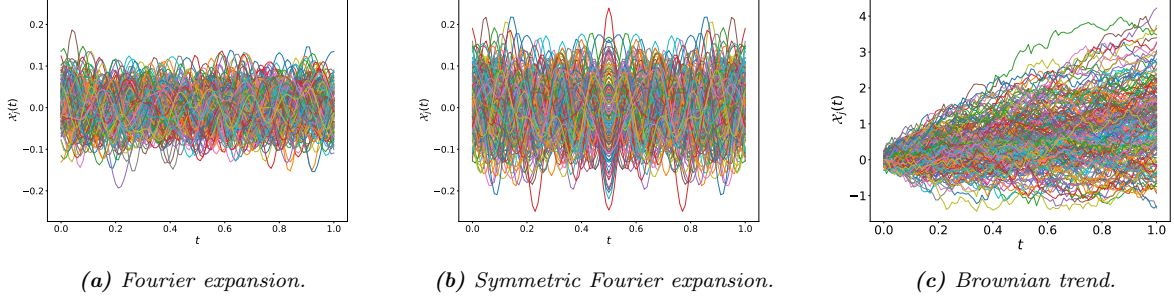


Figure 3: Functional data sets.

5.1 Simulated data

Regarding \mathcal{X} , we consider three different ways of generation, namely *Fourier expansion*, *symmetric Fourier expansion* and *Brownian trend* (short name for *Brownian motion with a trend*). The first one consists of linear combinations of the Fourier basis, $\mathcal{X}_F(t) = \sum_{s=0}^r Z_s \xi_s(t)$, where Z_s are independent draws of a standard normal distribution, while $\{\xi_s(t)\}_{s \geq 0}$ is the Fourier basis on I , and r is an even number. The goal of the second one is to obtain symmetric trajectories with respect to the middle point of I . This is achieved by building the functions as $\mathcal{X}_{sF}(t) = \mathcal{X}_F(t) + \mathcal{X}_F(1-t)$. When simulating data for the previous two cases, $\mathcal{X}_F(t)$ and $\mathcal{X}_{sF}(t)$, the value of r is fixed at 30. The third functional random variable taken into account is a Brownian motion with a trend. Let $\mathcal{B}(t)$ be a Brownian motion for $t \in I = [0, 1]$. Then, the functional random variable is $\mathcal{X}_B(t) = \mathcal{B}(t) + t$. Figure 3 illustrates the aforementioned functional random variables.

Every functional data set is stored in a matrix \mathbf{X} of size $m \times T$, where m takes two different values, 200 and 500, and T is equal to 101, being $t_1 = 0$ and $t_k = t_{k-1} + 0.01$, $k = 2, \dots, 101$. Regarding the partition $\mathcal{I} = \{I_1, \dots, I_n\}$, we consider $n = 20$ and all the intervals with the same length. In order to build the target variable Y , a transformation is applied to the functional random variable, $\Upsilon : L^2(I) \rightarrow \mathbb{R}$, leading to $Y = \Upsilon(\mathcal{X}) + \epsilon$, being $\epsilon \sim N(0, \sigma_\epsilon^2)$ independent of \mathcal{X} . We control the signal-to-noise ratio by using $\eta = \sigma_\epsilon^2 / \sigma_Y^2$, where $\sigma_Y^2 = \text{Var}(Y)$. We use two values for η : 0.05 and 0.25. With respect to Υ , 4 types of transformations are taken into account:

Linear unimodal: $\Upsilon_{lu}(\mathcal{X}) = \int_I \mathcal{X}(t) \beta_u(t) dt$, with $\beta_u(t)$ being the density function of a beta distribution with parameters 30 and 90. See left panel of Figure 4.

Linear bimodal: $\Upsilon_{lb}(\mathcal{X}) = \int_I \mathcal{X}(t) \beta_b(t) dt$, with $\beta_b(t) = (1/2)[\beta_u(t) + \beta_u(1-t)]$. See right panel of Figure 4.

Non-linear: $\Upsilon_{nl}(\mathcal{X}) = \max_{t \in I} \{|\beta_u(t) \mathcal{X}(t)|, |\beta_u(t) \mathcal{X}(1-t)|\}$.

Discrete: $\Upsilon_d(\mathcal{X}) = \mathcal{X}(0.15) + |\mathcal{X}(0.55)| + \mathcal{X}^2(0.35) \mathcal{X}(0.85)$.

In total, there are 48 different scenarios, the result of combining all the possibilities:

$$\{\mathcal{X}_F(t), \mathcal{X}_{sF}(t), \mathcal{X}_B(t)\} \times \underbrace{\{200, 500\}}_m \times \underbrace{\{0.05, 0.25\}}_\eta \times \{\Upsilon_{lu}, \Upsilon_{lb}, \Upsilon_{nl}, \Upsilon_d\}.$$

The relationship between \mathcal{X} and Y is modeled using three distinct prediction algorithms: a functional linear regression model (FLM for short; Ramsay and Silverman, 2005, Chapter 12), a functional k-nearest neighbor algorithm (FKNN for short; Ferraty and Vieu, 2006, Chapter 7) and a functional version of neural networks (FNN for short; Heinrichs et al., 2023). As for the

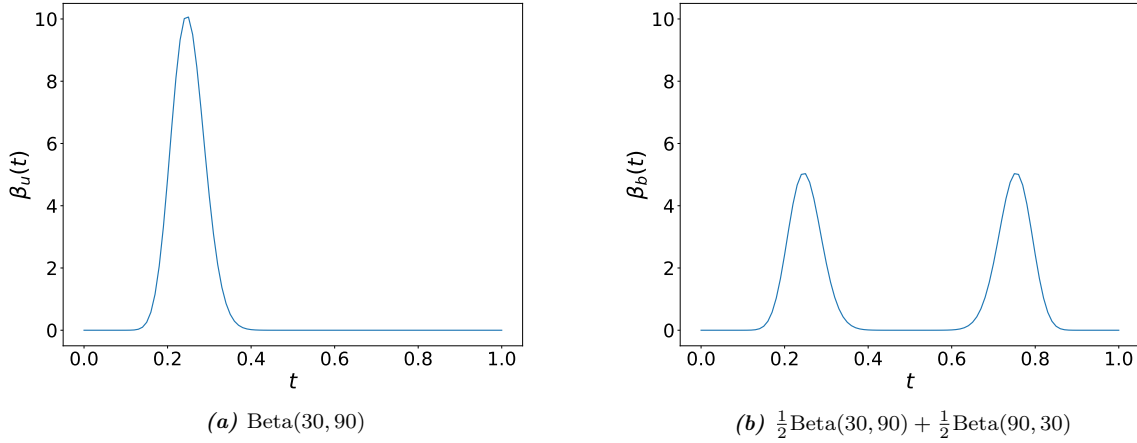


Figure 4: Beta density functions.

software used to train those models, **scikit-fda** (Ramos-Carreño et al., 2024) contains an implementation of FLM and FKNN. In addition, Florian Heinrichs has an open source implementation of FNN available at GitHub (github.com/FlorianHeinrichs/functional_neural_networks).

Every scenario is replicated 100 times. A replica is constituted by the generation of three independent data sets: training, validation and test. The first one is used to train the prediction models. As those algorithms require hyperparameter tuning, the second data set is used to do it. Finally, the goal of the third data set is to compute the Shapley value relevance function for each prediction algorithm. These three sets are of the same shape, $m \times T$. So, given a scenario, there are 100 sets of Shapley value relevance functions. A total of $|\Pi_0| = 1000$ random permutations of the set \mathcal{I} are generated.

In the context of basis representation, a preliminary analysis showed that the number of elements in the basis is more important than the type of basis itself. To this end, we decided to use splines as the basis for FLM and FKNN, and the default Legendre basis for FNN. The number of elements constitutes an hyperparameter for each of the models.

Shapley value relevance functions are obtained for each prediction algorithm and each scenario, as well as for each of the 100 simulations. These results are averaged across simulations. For the sake of brevity, we offer here the results corresponding to sample size 200 and signal-to-noise ratio given by $\eta = 0.05$ (12 scenarios out of 48; the remaining results are reported as Appendix B in the supplementary material). Figure 5 shows the mean Shapley value relevance functions, and Table 1 contains the mean and the standard deviation (in brackets) across the 100 simulations of $R^2(I)$, that is the coefficient of determination using all points $t \in [0, 1]$ evaluated on the test data sets.

First, we consider the scenarios with functional data generated as Fourier expansions (first column in Figure 5). When the target variable is derived from the linear unimodal or bimodal transformations (first and second row respectively), all the Shapley value relevance functions are able to identify the area with the highest relevance, which is the one formed by the points where the functional coefficients ($\beta_u(t)$ or $\beta_b(t)$) take their maximum values. Regarding the coefficients of determination $R^2(I)$, they indicate good fitting for both, FLM and FNN, and a poor performance of FKNN.

The third scenario (first column, third row in Figure 5), with target of non-linear type, shows that the Shapley value relevance functions may be different from each other. First, it can be seen that the Shapley value relevance function corresponding to the functional linear model is constantly equal to 0. According to it, no point is relevant for the model. As the relationship between \mathcal{X} and Y is non-linear, the estimated FLM is of very poor quality. This can be corroborated with the value of its $R^2(I)$, which mean value is close to 0 (see Table 1).

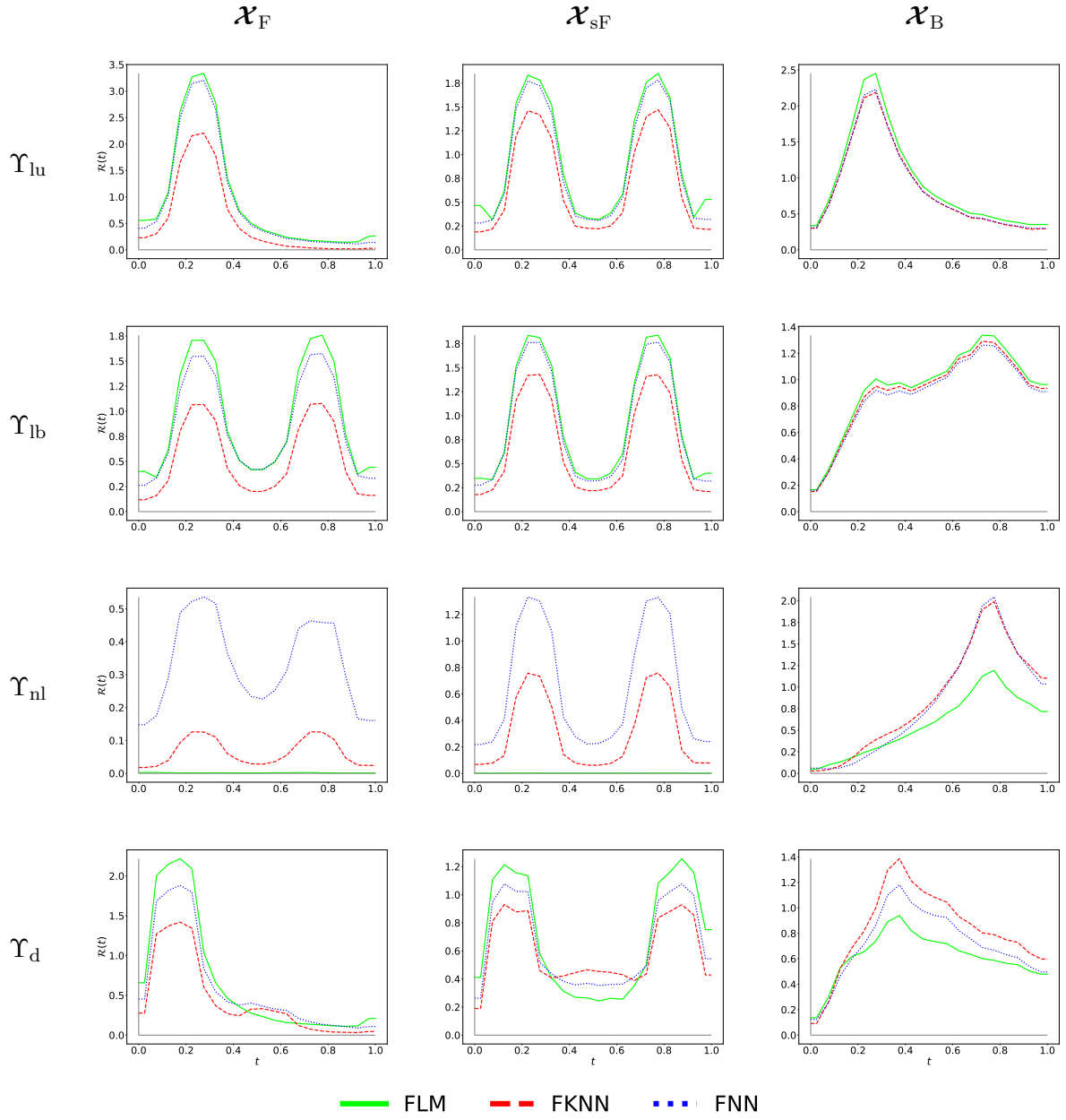


Figure 5: Mean Shapley value relevance functions for those scenarios with $m = 200$ and $\eta = 0.05$.

		\mathcal{X}_F	\mathcal{X}_{sF}	\mathcal{X}_B
Υ_{lu}	FLM	0.9468 (0.0054)	0.9483 (0.0053)	0.9467 (0.0058)
	FKNN	0.5399 (0.0392)	0.7048 (0.0259)	0.8541 (0.0191)
	FNN	0.8887 (0.0812)	0.8961 (0.0902)	0.8602 (0.1160)
Υ_{lb}	FLM	0.9464 (0.0062)	0.9470 (0.0053)	0.9479 (0.0051)
	FKNN	0.5377 (0.0370)	0.6962 (0.0273)	0.9149 (0.0091)
	FNN	0.8634 (0.0726)	0.9053 (0.0719)	0.8924 (0.0521)
Υ_{nl}	FLM	0.0289 (0.0331)	0.0264 (0.0326)	0.5718 (0.0753)
	FKNN	0.0490 (0.0804)	0.3101 (0.0668)	0.8637 (0.0163)
	FNN	0.3361 (0.1450)	0.6559 (0.1602)	0.8321 (0.0721)
Υ_d	FLM	0.6696 (0.0388)	0.6907 (0.0383)	0.6209 (0.0462)
	FKNN	0.4337 (0.0397)	0.5990 (0.0372)	0.8324 (0.0464)
	FNN	0.6070 (0.1568)	0.6492 (0.0909)	0.7196 (0.0644)

Table 1: Mean value (standard deviation) of $R^2(I)$ of FLM, FKNN and FNN for those scenarios with $m = 200$ and $\eta = 0.05$.

Therefore, it is expected not to find any relevant point $t \in I$ when using the FLM. However, FKNN and FNN models are able to detect the relevant points in prediction, with FNN showing a better performance than FKNN.

Consider now the fourth set of Shapley value relevance functions in the first column of Figure 5. It corresponds to using a discrete transformation when defining the target variable, $\Upsilon_d(\mathcal{X}) = \mathcal{X}(0.15) + |\mathcal{X}(0.55)| + \mathcal{X}^2(0.35)\mathcal{X}(0.85)$, which depends only on the values of \mathcal{X} at four points (0.15, 0.35, 0.55 and 0.85). The three Shapley value relevance functions indicate that the most relevant points t are those around 0.15, and that small to no relevance is assigned to the points 0.35, 0.55 or 0.85. Since the values taken by \mathcal{X} are close to 0, the result of $\mathcal{X}^2(0.35)\mathcal{X}(0.85)$ is a very small number. So, its effect over $\Upsilon_d(\mathcal{X})$ is expected to be negligible. As can be observed in the aforementioned figure, the values around $t \in \{0.35, 0.85\}$ have almost no impact, although there is a local maximum at $t = 0.85$. Regarding $t = 0.55$, as the absolute value is applied to $\mathcal{X}(0.55)$, the degree of variability of $\mathcal{X}(0.55)$ is limited to half. Therefore, the relevance of the point $t = 0.55$ is expected to be low. These conclusions are supported by the Shapley value relevance functions depicted in the aforementioned figure.

The second column of Figure 5 corresponds to the scenarios where data are generated using the symmetric Fourier expansion. The same insights derived for the first column can be applied in these 4 scenarios, with the exception that, as symmetric functions are employed, in all cases two relevant areas are obtained, which are symmetric with respect to the midpoint $t = 0.5$.

Finally, let us consider the last column of Figure 5, which corresponds to scenarios whose data are generated using a Brownian motion with a trend. The first row corresponds to a linear unimodal transformation. All Shapley value relevance functions consider the most relevant point to be the one that maximizes $\beta_u(t)$. In the case of the scenario where a linear bimodal is used (second row), two areas are particularly relevant in this regard: the first corresponds with points close to $t_1 = 0.25$ and the second with points close to $t_2 = 0.75$, being $\{t_1, t_2\} = \operatorname{argmax}_{t \in I} \beta_b(t)$. As the variance of the Brownian trend increases with t , there is more variability for points close to $t = 0.75$ than for points close to $t = 0.25$, and therefore, it is expected to obtain a higher relevance in the neighborhood of t_2 , a phenomenon that is indeed observed.

A similar argument explains why the highest relevance of the non-linear transformation (third row) is around $t = 0.75$: as the variance increases with t , $\max_{t \in I} \{|\beta_u(t)\mathcal{X}(t)|, |\beta_u(t)\mathcal{X}(1-t)|\}$ is around $t = 0.75$ instead of $t = 0.25$. With regard to the discrete transformation (fourth row), the most relevant point is $t = 0.35$, which is explained by (1) now the range of variability of

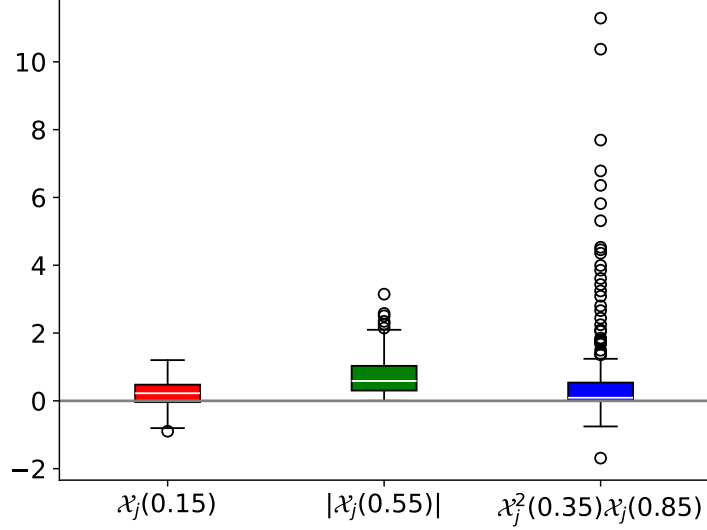


Figure 6: (Left) Boxplot of $\{\mathcal{X}_j(0.15)\}_{j=1}^m$. (Middle) Boxplot of $\{|\mathcal{X}_j(0.55)|\}_{j=1}^m$. (Right) Boxplot of $\{\mathcal{X}_j^2(0.35)\mathcal{X}_j(0.85)\}_{j=1}^m$ where, $\{\mathcal{X}_j\}_{j=1}^m$ corresponds to the Brownian trend data set shown in the rightmost panel of Figure 3.

$\mathcal{X}^2(0.35)\mathcal{X}(0.85)$ is larger than that of $\mathcal{X}(0.15)$ and $|\mathcal{X}(0.55)|$ (see Figure 6, corresponding to the Brownian trend data set) and (2) $\mathcal{X}(0.35)$ is more difficult to predict than $\mathcal{X}(0.85)$ using conditional expectations, because $t = 0.85$ is closer to the end of the path $\{\mathcal{X}(t) : t \in [0, 1]\}$ than $t = 0.35$.

A final remark on the execution times is in order. The main computational burden of the algorithm is the large number of random permutations required in order to approximate the Shapley values. For instance, a single simulation of those employed to generate the graphic of first row and first column of Figure 5 (which uses 1000 random permutations) takes 25 seconds to compute the FLM Shapley value relevance function in an iMac (2024) with an M4 chip (10 cores; four performance cores and six efficiency cores running up to 4.4 GHz and 2.9 GHz respectively) and 16 GB of RAM memory. To evaluate the effect of the number of random permutations, we repeat this experiment with 2000 and 5000 random permutations, resulting in computation times of 48 and 109 seconds respectively.

5.2 Real data

The Tecator data set has been widely used within the FDA literature (see, for instance, Ferraty and Vieu, 2006). This data set consists of 215 spectrometric curves of meat samples and it measures the near infrared (NIR) absorbance $\mathcal{A}(w)$ as a function of wavelength w (in nanometers). In addition, each curve is associated with three distinct quantities: the percentage of fat, water, and protein in the meat sample. The percentage of fat is used here as the target variable. Figure 7 depicts the set of spectrometric curves (left), their first derivatives (middle) and their second derivatives (right).

Following Ferraty and Vieu (2006, Section 7.2.1), we divide the data set into *learning sample* (curves 1 to 160; 74.4%) and *testing sample* (curves 161 to 215; 55 curves, 25.6%). Given that we differentiate between *training* and *validation* sets, we divide the learning sample randomly into *training set* (128 curves; 59.6%) and *validation set* (32 curves; 14.8%). The training set is employed for fitting the same prediction algorithms used in Section 5.1: FLM, FKNN and FNN. The validation set is used to perform hyperparameter tuning, and the test data set serves to compute the distinct Shapley value relevance functions.

The prediction algorithms are used to model the target variable as a function of the second derivatives of the spectrometric curves, as suggested in Ferraty and Vieu (2006, Section 7.2.2).

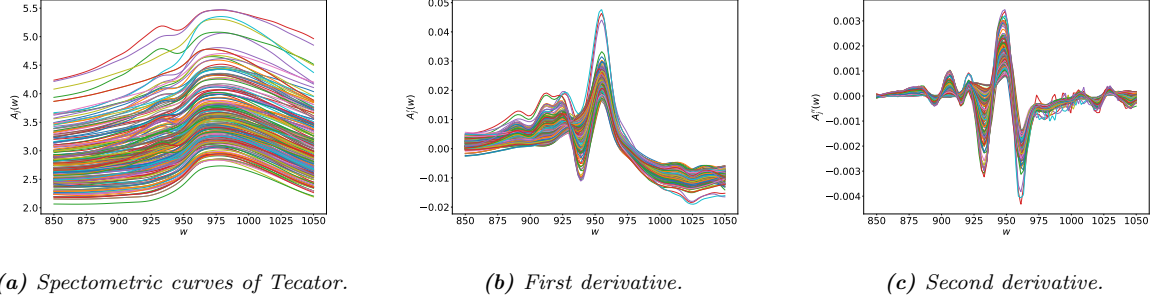


Figure 7: Tecator data set.

See also Boj et al. (2010). We proceed in the same way as in Section 5.1 to select the basis to represent the functional data. The $R^2(I)$ are 0.962 for FLM, 0.9808 for FKNN and 0.8196 for FNN. Regarding FLM, the top panel of Figure 8 shows the estimated function $\hat{\beta}(t)$. This function is very variable and difficult to interpret, since its shape does not seem to be related to that of the second derivatives (middle panel). Therefore, even FLM can benefit from the definition of a relevance function.

Next, we use our proposal to obtain interpretability for each of the prediction models, as in Section 5.1. The interval of interest, $I = [850, 1050]$, is divided into 20 parts, all of the same length, and 5000 permutations are performed. The bottom panel of Figure 8 depicts the Shapley value relevance functions, aligned with the set of second derivatives (middle panel). It should be recalled that the relevance is obtained at interval level. The three Shapley value relevance functions show their global maximum at $[1040, 1050]$. In this interval, the set of second derivatives manifests a certain degree of variability. On the other hand, all the Shapley value relevance functions consider the region within the interval $[930, 960]$ to be relevant. In this interval, the set of second derivatives exhibits rapid fluctuations between minimum and maximum, followed by another minimum. Finally, it is worth noting that all Shapley value relevance functions identify the interval $[970, 980]$ as the least relevant. This is likely due to the fact that the values of all second derivatives are close to each other at this interval.

6 Conclusions

This work represents a novel approach to addressing the global agnostic interpretability of prediction models within the field of functional data analysis. Our framework is based on game theory for games with a continuum of players, extending the Shapley value to the case of a set with infinite regressors. The central piece of this work is the Shapley value relevance function, that measures the relevance of all points of the interval where the functional data are observed. Alongside this manuscript, we present an open source Python package that implements this framework.

We have illustrated the performance of our work by means of a set of experiments. On the one hand, we use simulated data where the relevant points were known beforehand. The results have shown that these points have been successfully captured by the explainability method. On the other hand, we have explored the Tecator data set, where the second derivative has been used to model the data. According to our proposal, the most relevant intervals coincide with large variability areas of the second derivatives.

As future research, it would be interesting to extend this methodology in two directions: first, considering several functional regressors, and second, allowing a functional response and regressors that are either scalar or functional.

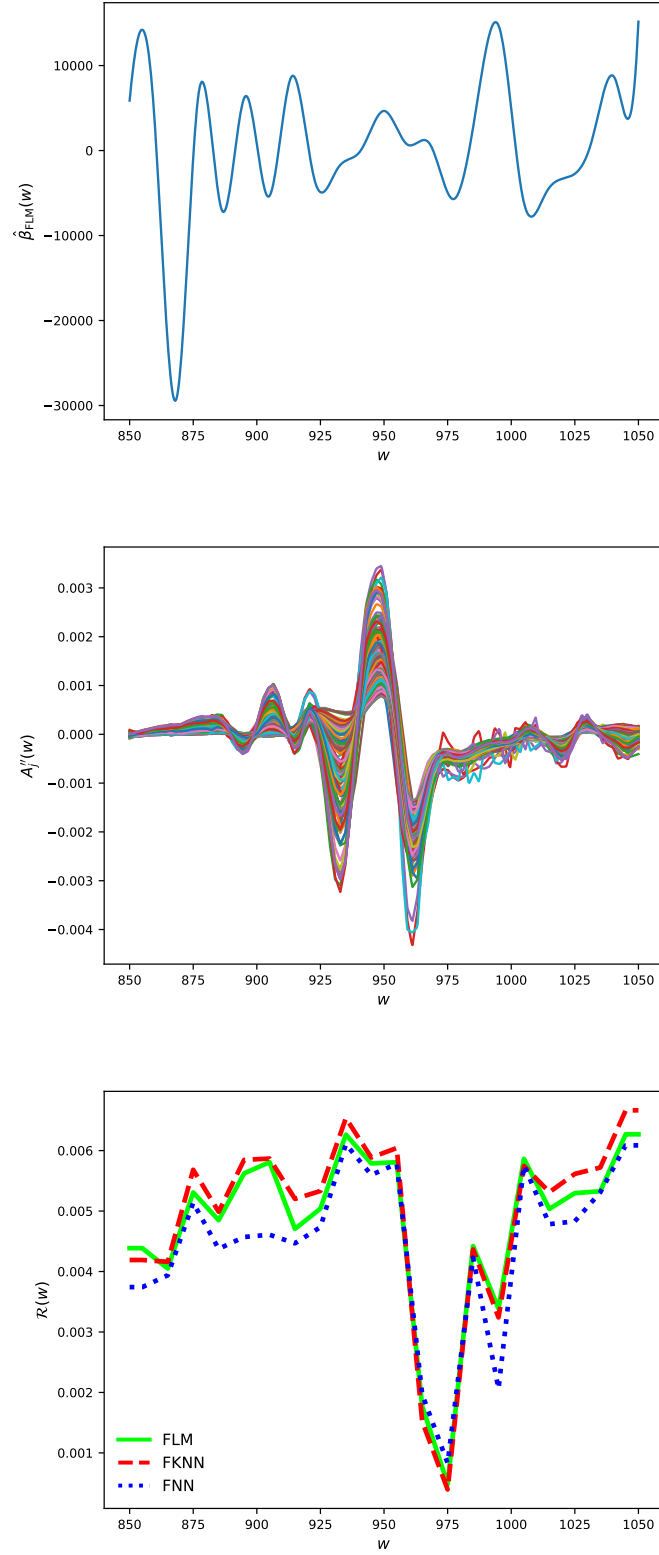


Figure 8: Shapley value analysis of Tecator data. (Top) Estimated beta function for the functional linear model using Tecator data set to predict the fat content. (Middle) Set of second derivatives. (Bottom) Shapley value relevance functions.

A Interpretable machine learning

This appendix summarizes the topic of interpretable prediction in the context of multiple regression. For a more detailed coverage see, for instance, Biecek and Burzykowski (2021) or Molnar (2022).

Breiman (2001) identifies two distinct cultures within the field of data analysis: the modeling culture, based on statistical inference, and the prediction culture, focused on machine learning techniques. In addition, Breiman (2001) points out a potential trade-off between predictive ability and interpretability of models. The predictive accuracy of machine learning algorithms, such as neural networks or random forests, often exceeds that of statistical models, such as linear or logistic regression. However, statistical models provide a simpler understanding of the relationship between the response variable and the input variables. In fact, machine learning algorithms are often referred to as “black boxes” because of their inability to provide understandable explanations of the reasons behind their predictions. Nevertheless, Breiman (2001) calls for the development of procedures that would allow for better interpretation of algorithmic models without compromising their predictive ability.

In interpretable machine learning, a distinction is done between *global* and *local* interpretability. On the one hand, global interpretability tools measure the importance or relevance of each explanatory variable in the prediction process over their whole support. On the other hand, local interpretability tools provide meaningful explanations of why the prediction model returns a certain estimated response for a given individual, identified with a particular combination of the values of the predictor variables.

It is also relevant to classify interpretability methods as *model-specific* or *model-agnostic*. The first category includes interpretability methods that are designed to interpret a particular prediction algorithm (such as random forests or neural networks), exploiting its internal structure for interpretation. In contrast, model-agnostic interpretability methods can be applied to any prediction model. They only need to evaluate the prediction model on data from the training or test sets, or perturbations of either, and do not have access to the internal structure of the prediction model.

In this paper, we are interested in global model-agnostic interpretability methods. In the usual framework of a prediction problem with p explanatory variables and one response to which a prediction model is fitted, global model-agnostic methods provide a relevance measure for each predictor. The most natural way to do this is *leave-one-covariate-out* (LOCO), and it has been used in multiple linear regression for decades. Two prediction models are fitted (one with all the predictors and the other with one predictor omitted) and their prediction errors are compared: the more different they are, the more important the omitted predictor is. The main drawback of LOCO is that it requires fitting p additional models, each with $(p - 1)$ predictors.

In Breiman (2001), a permutation-based approach is used to define variable importance in random forests, which is easily extended to any prediction model when a test sample is available. The average prediction error for the test sample is compared with the same measure when the test sample is modified randomly permuting the values of a specific explanatory variable. The larger that difference, the more important is the permuted variable. Despite its popularity, using random permutations for interpreting black box prediction algorithms has received numerous criticisms (see, for instance, Hooker et al., 2021; Delicado and Peña, 2023), mainly when the predictors are moderate or highly correlated.

Delicado and Peña (2023) propose an alternative approach. First, they fit the model with all the explanatory variables in the training sample. Then, they measure the individual relevance of each explanatory variable by comparing the predictions of the model in the test set with those obtained when an explanatory variable in the test set is replaced by its *ghost variable*, defined as the conditional expectation of that variable given the values of the other explanatory variables (using a linear model). Delicado and Peña (2023) show that, in linear models, the proposed measure gives results similar to LOCO and outperforms random permutations.

A different line of work in global model-agnostic relevance measure is based on the Shapley value, a concept coming from game theory. We devote Section 2 entirely to this approach, which is the one we follow in Section 3 to develop our proposal.

As mentioned in the main corpus of the paper, we only found two papers considering interpretability in functional data regression models. James et al. (2009) specifically consider the functional linear regression model, whose global interpretation is favored when the estimate of $\beta(t)$ is exactly zero over regions with no apparent relationship between the functional predictor and the scalar response. The authors propose an estimation method which combines the basis expansion of $\beta(t)$ and a lasso-type strategy for variable selection. Carrizosa et al. (2024) address the issue of local feature importance using counterfactual analysis for functional data. So, their goal is to explain (or rank) the features in a neighborhood of a given individual, with functional observations $\mathcal{X}_0(t)$. The method is presented in the context of (multiclass) classification. The work aims at obtaining a counterfactual explanation $\mathcal{X}(t)$, which is an artificial functional data, by combining existing instances in the dataset, called *prototypes*. To achieve it, an optimization problem is formulated, taking into account that the cost of perturbing $\mathcal{X}_0(t)$ to obtain $\mathcal{X}(t)$ must be minimal.

B Additional results of the simulation study

In this section we present the results of the remaining scenarios, which are:

- $m = 200$ and $\eta = 0.25$. See Figure 9 and Table 2.
- $m = 500$ and $\eta = 0.05$. See Figure 10 and Table 3.
- $m = 500$ and $\eta = 0.25$. See Figure 11 and Table 4.

		\mathcal{X}_F	\mathcal{X}_{sF}	\mathcal{X}_B
Υ_{lu}	FLM	0.7317 (0.0282)	0.7399 (0.0260)	0.7384 (0.0227)
	FKNN	0.3987 (0.0412)	0.5235 (0.0415)	0.6429 (0.0322)
	FNN	0.6936 (0.0570)	0.7137 (0.0418)	0.6884 (0.0655)
Υ_{lb}	FLM	0.7315 (0.0278)	0.7358 (0.0303)	0.7441 (0.0257)
	FKNN	0.3995 (0.0398)	0.5192 (0.0355)	0.7061 (0.0280)
	FNN	0.6542 (0.0720)	0.6945 (0.0683)	0.6919 (0.0586)
Υ_{nl}	FLM	0.0271 (0.0285)	0.0206 (0.0225)	0.4460 (0.0798)
	FKNN	0.0184 (0.0692)	0.2036 (0.0551)	0.6566 (0.0342)
	FNN	0.2147 (0.0952)	0.5086 (0.1176)	0.6271 (0.0864)
Υ_d	FLM	0.5010 (0.0556)	0.5248 (0.0488)	0.4911 (0.0490)
	FKNN	0.3161 (0.0513)	0.4414 (0.0395)	0.6274 (0.0530)
	FNN	0.4504 (0.1057)	0.4973 (0.0821)	0.5677 (0.0618)

Table 2: Mean value (standard deviation) of $R^2(I)$ of FLM, FKNN and FNN for those scenarios with $m = 200$ and $\eta = 0.25$.

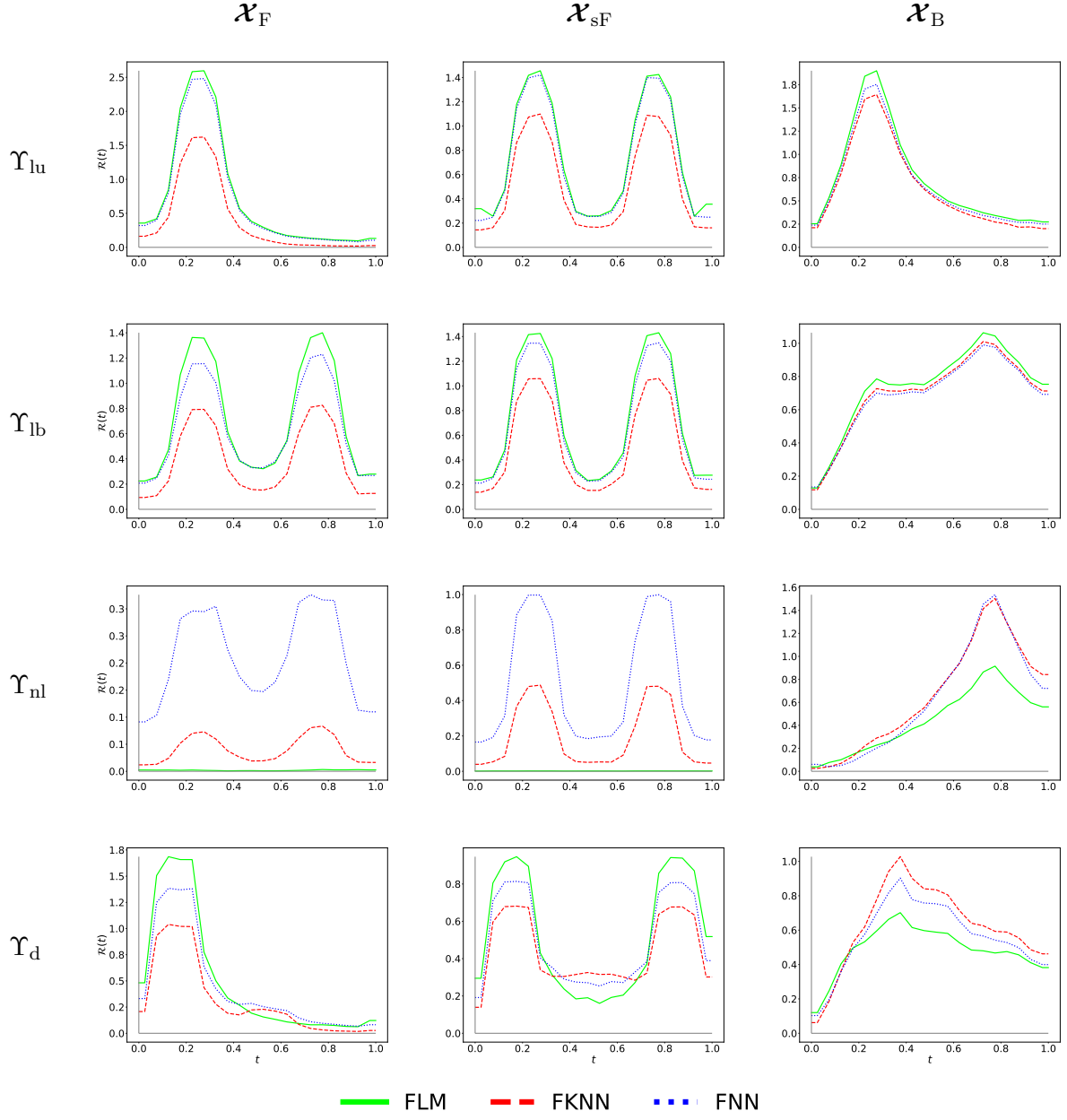


Figure 9: Mean Shapley value relevance functions for those scenarios with $m = 200$ and $\eta = 0.25$.

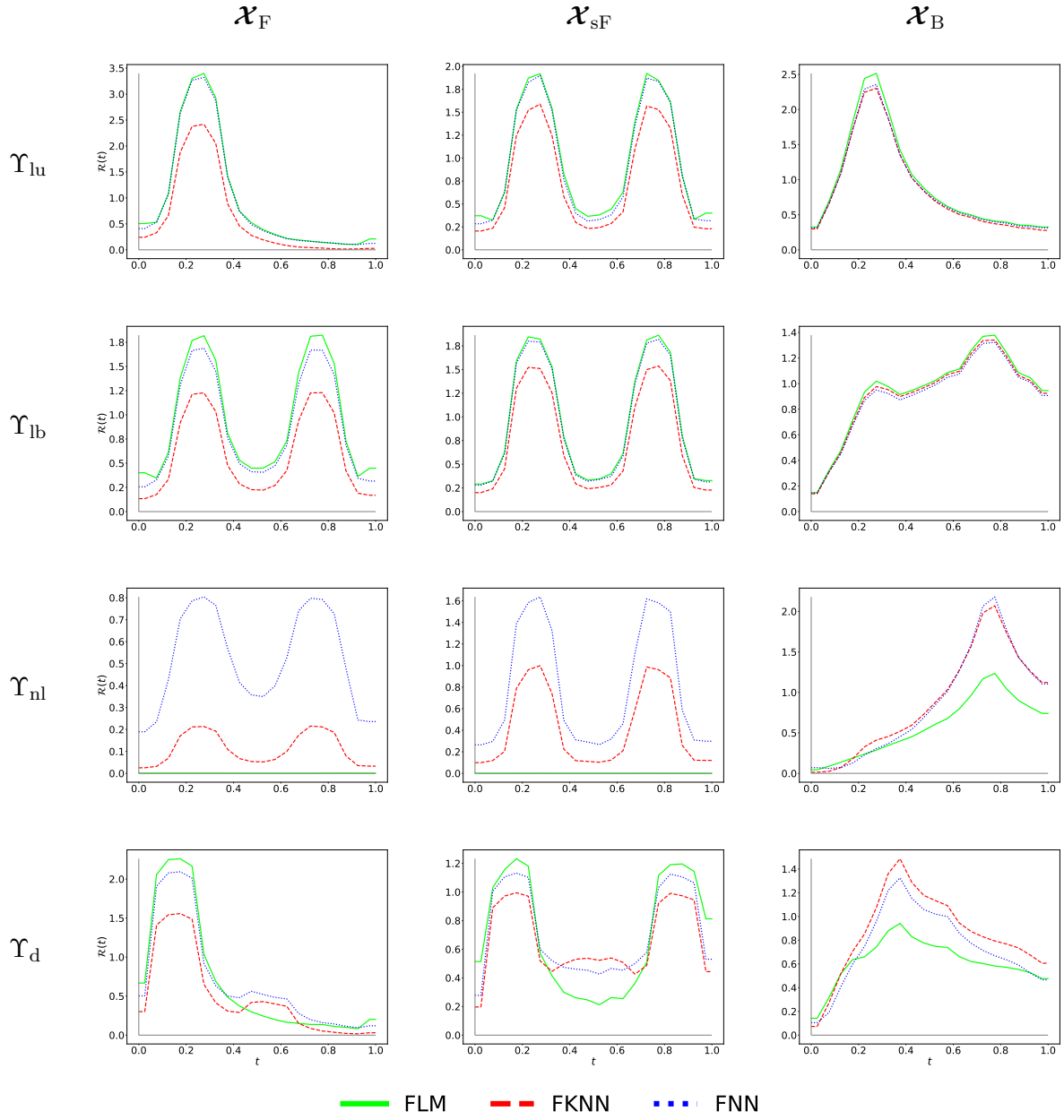


Figure 10: Mean Shapley value relevance functions for those scenarios with $m = 500$ and $\eta = 0.05$.

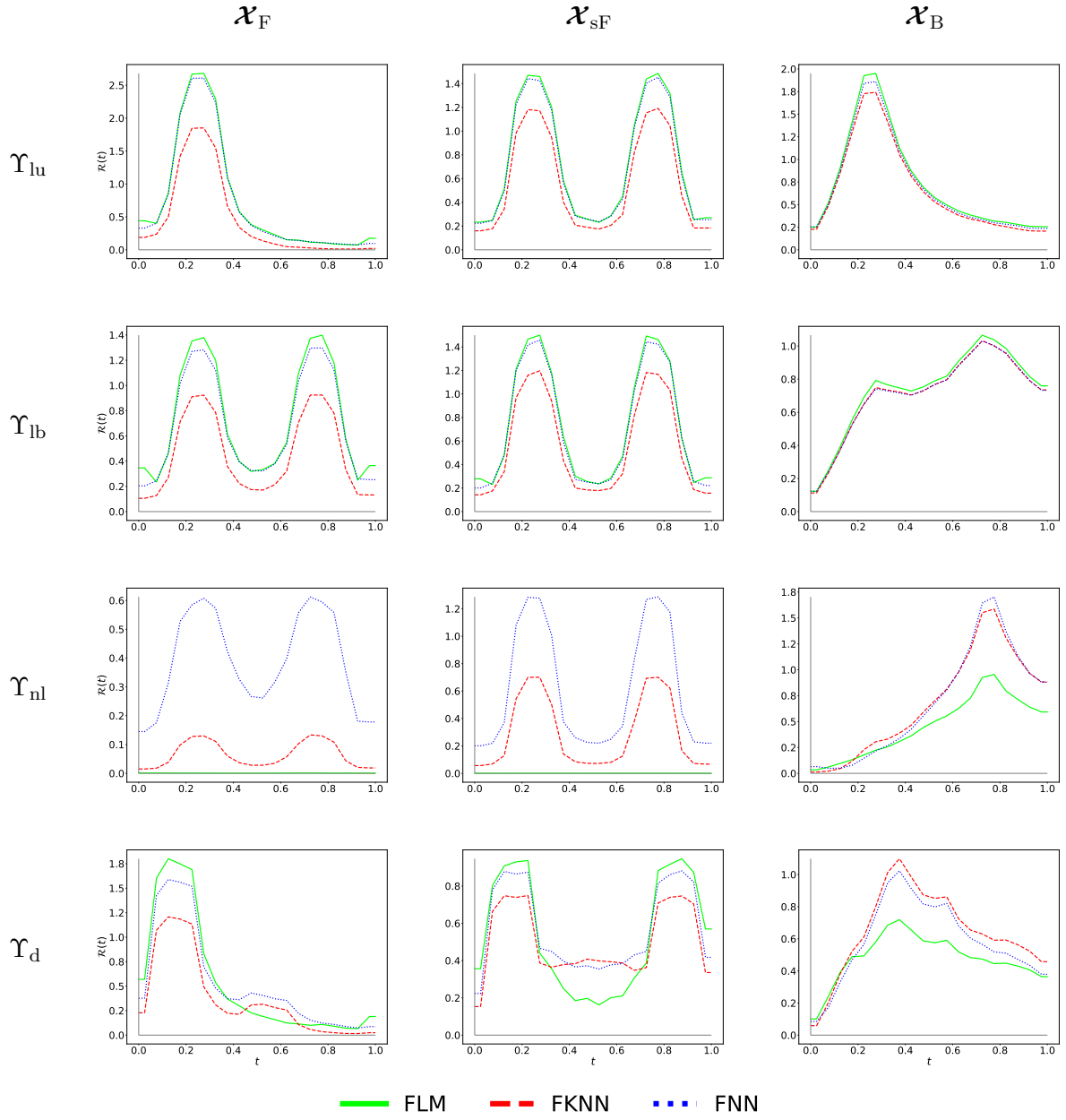


Figure 11: Mean Shapley value relevance functions for those scenarios with $m = 500$ and $\eta = 0.25$.

		\mathcal{X}_F	\mathcal{X}_{sF}	\mathcal{X}_B
Υ_{lu}	FLM	0.9488 (0.0034)	0.9491 (0.0034)	0.9484 (0.0032)
	FKNN	0.6044 (0.0274)	0.7550 (0.0165)	0.8805 (0.0078)
	FNN	0.9281 (0.0573)	0.9433 (0.0136)	0.9006 (0.0636)
Υ_{lb}	FLM	0.9486 (0.0031)	0.9491 (0.0035)	0.9492 (0.0034)
	FKNN	0.6046 (0.0203)	0.7556 (0.0134)	0.9258 (0.0052)
	FNN	0.8963 (0.0917)	0.9329 (0.0616)	0.9078 (0.0463)
Υ_{nl}	FLM	0.0111 (0.0113)	0.0108 (0.0116)	0.5820 (0.0531)
	FKNN	0.1108 (0.0437)	0.4303 (0.0330)	0.8828 (0.0093)
	FNN	0.5266 (0.0978)	0.8049 (0.0548)	0.8715 (0.0384)
Υ_d	FLM	0.6872 (0.0252)	0.6931 (0.0242)	0.6259 (0.0322)
	FKNN	0.4973 (0.0279)	0.6640 (0.0183)	0.8615 (0.0273)
	FNN	0.7136 (0.0841)	0.7197 (0.0705)	0.7506 (0.0477)

Table 3: Mean value (standard deviation) of $R^2(I)$ of FLM, FKNN and FNN for those scenarios with $m = 500$ and $\eta = 0.05$.

		\mathcal{X}_F	\mathcal{X}_{sF}	\mathcal{X}_B
Υ_{lu}	FLM	0.7432 (0.0180)	0.7476 (0.0134)	0.7435 (0.0151)
	FKNN	0.4581 (0.0221)	0.5741 (0.0195)	0.6728 (0.0191)
	FNN	0.7247 (0.0513)	0.7300 (0.0427)	0.7122 (0.0458)
Υ_{lb}	FLM	0.7415 (0.0185)	0.7442 (0.0172)	0.7430 (0.0157)
	FKNN	0.4613 (0.0252)	0.5719 (0.0225)	0.7164 (0.0168)
	FNN	0.6979 (0.0637)	0.7253 (0.0564)	0.7165 (0.0357)
Υ_{nl}	FLM	0.0102 (0.0106)	0.0107 (0.0114)	0.4560 (0.0457)
	FKNN	0.0641 (0.0463)	0.2988 (0.0350)	0.6799 (0.0176)
	FNN	0.3975 (0.0758)	0.6285 (0.0463)	0.6788 (0.0369)
Υ_d	FLM	0.5387 (0.0284)	0.5385 (0.0277)	0.4838 (0.0312)
	FKNN	0.3739 (0.0303)	0.5049 (0.0237)	0.6500 (0.0308)
	FNN	0.5400 (0.0697)	0.5731 (0.0503)	0.5929 (0.0391)

Table 4: Mean value (standard deviation) of $R^2(I)$ of FLM, FKNN and FNN for those scenarios with $m = 500$ and $\eta = 0.25$.

Acknowledgments

This research was supported by projects PID2020-116294GB-I00 and PID2023-148158OB-I00 (founded by MICIU/AEI/10.13039/501100011033 and European Union NextGeneratio-EU/PRTR), by AGAUR under grants 2020 FI SDUR 306 and 2021 SGR 00613, and by UPC under AGRUPS-2024.

References

- R. J. Aumann and L. S. Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974. ISBN 978-0-691-64546-9.
- Robert J. Aumann. Markets with a Continuum of Traders. *Econometrica*, 32:39–50, 1964. doi: 10.2307/1913732.
- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham

- Tabik, Alberto Barbado, Salvador Garcia, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. doi: 10.1016/j.inffus.2019.12.012.
- P Biecek and T Burzykowski. *Explanatory model analysis: Explore, explain and examine predictive models*. Chapman and Hall/CRC, 2021. ISBN 978-0-367-13559-1.
- Eva Boj, Pedro Delicado, and Josep Fortiana. Distance-based local linear regression for functional predictors. *Computational Statistics & Data Analysis*, 54(2):429–437, 2010. doi: 10.1016/j.csda.2009.09.010.
- Claus Borggaard and Hans Henrik Thodberg. Optimal minimal neural interpretation of spectra. *Analytical Chemistry*, 64(5):545–551, 1992. doi: 10.1021/ac00029a018.
- Leo Breiman. Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3):199–231, 2001. doi: 10.1214/ss/1009213726.
- Emilio Carrizosa, Jasone Ramírez-Ayerbe, and Dolores Romero Morales. A new model for counterfactual analysis for functional data. *Advances in Data Analysis and Classification*, 18(4):981–1000, 2024. doi: 10.1007/s11634-023-00563-5.
- Shay B. Cohen, Gideon Dror, and Eytan Ruppin. Feature Selection via Coalitional Game Theory. *Neural Computation*, 19(7):1939–1961, 2007. doi: 10.1162/neco.2007.19.7.1939.
- Ciprian M. Crainiceanu, Jeff Goldsmith, Andrew Leroux, and Erjia Cui. *Functional Data Analysis with R*. Chapman and Hall/CRC, 2024. ISBN 978-1-032-24472-3.
- Pedro Delicado and Daniel Peña. Understanding complex predictive models with ghost variables. *TEST*, 32(1):107–145, 2023. doi: 10.1007/s11749-022-00826-x.
- R. W. Farebrother. *Linear least squares computations*. Routledge, 1988. ISBN 978-0-824-77661-9. doi: 10.1201/9780203748923.
- Barry E. Feldman. Relative Importance and Value. Available at SSRN Electronic Journal, doi: 10.2139/ssrn.2255827, 2005.
- F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer, 2006. ISBN 978-0-387-30369-7.
- Jan Gertheiss, David Rügamer, Bernard X. W. Liew, and Sonja Greven. Functional Data Analysis: An Introduction and Recent Developments. *Biometrical Journal*, 66(7):e202300363, 2024. doi: 10.1002/bimj.202300363.
- Y. Goldberg, Y. Ritov, and A. Mandelbaum. Predicting the continuation of a function with applications to call center data. *Journal of Statistical Planning and Inference*, 147:53–65, 2014. doi: 10.1016/j.jspi.2013.11.006.
- Ulrike Grömping. Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *The American Statistician*, 63(4):308–319, 2009. doi: 10.1198/tast.2009.08199.
- Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.
- Sergiu Hart and Abraham Neyman. Values of non-atomic vector measure games: Are they linear combinations of the measures? *Journal of Mathematical Economics*, 17(1):31–40, 1988. doi: 10.1016/0304-4068(88)90025-0.

- Florian Heinrichs, Mavin Heim, and Corinna Weber. Functional neural networks: Shift invariant models for functional data with applications to EEG classification. In *Proceedings of the 40th International Conference on Machine Learning*, pages 12866–12881. PMLR, 2023.
- G. Hooker, L. Mentch, and S. Zhou. Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31(6):82, 2021. doi: 10.1007/s11222-021-10057-z.
- L. Horváth and P. Kokoszka. *Inference for Functional Data with Applications*. Springer, 2012. ISBN 978-1-4614-3654-6.
- Gareth M. James, Jing Wang, and Ji Zhu. Functional linear regression that’s interpretable. *The Annals of Statistics*, 37(5A):2083–2108, 2009. doi: 10.1214/08-AOS641.
- Yakar Kannai. Values of games with a continuum of players. *Israel Journal of Mathematics*, 4(1):54–58, 1966. doi: 10.1007/BF02760070.
- Alois Kneip and Dominik Liebl. On the optimal reconstruction of partially observed functional data. *The Annals of Statistics*, 48(3):1692–1717, 2020. doi: 10.1214/19-AOS1864.
- P. Kokoszka and M. Reimherr. *Introduction to Functional Data Analysis*. Chapman & Hall/CRC, 2017. ISBN 978-1-498-74634-2.
- David Kraus. Components and completion of partially observed functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4):777–801, 2015. doi: 10.1111/rssb.12087.
- Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001. doi: 10.1002/asmb.446.
- Serg Masís. *Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples*. Packt Publishing, 2021. ISBN 978-1800203907.
- Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Independently published, 2022. ISBN 979-8-411-46333-0.
- A. Neyman. *Value of Games with a Continuum of Players*, chapter VI, pages 67–79. Springer, 1994. ISBN 978-0-7923-3011-0. doi: 10.1007/978-94-017-1656-7_7.
- Python Core Team. *The Python Language Reference*. Python Software Foundation, 2021. URL <https://docs.python.org/3.10/reference/index.html>.
- Carlos Ramos-Carreño, José Luis Torrecilla, Miguel Carbajo-Berrocal, Pablo Marcos, and Alberto Suárez. scikit-fda: A Python Package for Functional Data Analysis. *Journal of Statistical Software*, 109(2):1–37, 2024. doi: 10.18637/jss.v109.i02.
- J. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer, 2005. ISBN 978-0-387-40080-8.
- Aniruddha Rajendra Rao and Matthew Reimherr. Nonlinear Functional Modeling Using Neural Networks. *Journal of Computational and Graphical Statistics*, 32(4):1248–1257, 2023. doi: 10.1080/10618600.2023.2165498.
- Philip T. Reiss, Jeff Goldsmith, Han Lin Shang, and R. Todd Ogden. Methods for Scalar-on-Function Regression. *International Statistical Review*, 85(2):228–249, 2017. doi: 10.1111/insr.12163.

- Lloyd S Shapley. A Value for n -Person Games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953. doi: 10.1515/9781400881970-018.
- Lloyd S. Shapley. Values of Games With Infinitely Many Players, 1961. *Research memorandum*. RAND Corporation, doi: 10.7249/RM2912.
- Eyal Winter. The Shapley Value. In Robert Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 3, chapter 53, pages 2025–2054. Elsevier, 2002. ISBN 978-0-444-89428-1. doi: 10.1016/S1574-0005(02)03016-3.
- Junwen Yao, Jonas Mueller, and Jane-Ling Wang. Deep learning for functional data analysis with adaptive basis layers. In *International Conference on Machine Learning*, pages 11898–11908. PMLR, 2021.