Entropic Variable Boosting for Explainability & Intepretability in Machine Learning

Abstract

In this paper, we present a new explainability formalism to make clear the impact of each variable on the predictions given by black-box decision rules. Our method consists in evaluating the decision rules on test samples generated in such a way that each variable is stressed incrementally while preserving the original distribution of the machine learning problem. We then propose a new computationally efficient algorithm to stress the variables, which only reweights the reference observations and predictions. This makes our methodology scalable to large datasets. Results obtained on standard machine learning datasets are presented and discussed.

1 Introduction

Machine learning algorithms build predictive models which are nowadays used for a large variety of tasks. They have become extremely popular in various applications such as finance, insurance risk, health-care, recommendation systems as well as industrial applications of all kinds including predictive maintenance, defect detection or industrial liability. Such algorithms are designed to assist human experts by giving access to valuable predictions and even tend to replace human decisions in many fields, achieving an extremely good performance.

The performance of machine learning models is usually quantified in terms of predictive accuracy. In many cases, the decision rules learned by machine learning models indeed minimize a prediction error measured on a pre-defined set of labeled examples, denoted the learning sample. The labels of new data are then predicted based on the learned decision rules.

Over the last decades, the complexity of such algorithms has grown, going from simple and interpretable prediction models based on regression rules to very complex models such as random forest, gradient boosting and models using deep neural networks. Such models are designed to maximize the accuracy of their predictions at the expense of the interpretability of the decision rule. Little is also known about how the information is processed in order to obtain a prediction, which explains why such models are widely considered as black-box models.

This lack of interpretability gives rise to several issues. When an empirical risk is minimized, the efficiency of a machine learning procedure highly depends on the nature of the optimization problem (e.g. convexity and unimodality). Challenging optimization problems may lead to decision rules that are unstable or highly dependent on the optimization procedure. Another subtle, though critical, issue is also that the optimal decision rules learned by a machine learning algorithm highly depend on the properties of the learning sample. If a learning sample presents a bias or unwanted trends, then the decision rules learned by the machine learning algorithm will reproduce the bias or trends, even if there is no intention of doing so. These flaws or misbehaviors will therefore be propagated in future predictions. This lack of explainability and the dangers that machine learning algorithms convey explain that many users express a lack of trust in these algorithms. The European Parliament even adopted a law called GDPR (General Data Protection Regulation) to protect citizens from decisions made without the possibility of explaining why they were taken, introducing a right for explanation in the civil code. We believe that a solution is not to abandon black-box models which have yet proven to be useful in many cases, but rather to improve the interpretability of machine learning algorithms. Hence, building intelligible models is nowadays an important research direction in data science.

Different methods have been proposed to make understandable the reasons leading to a prediction, each author using a different notion of explainability and interpretability of a decision rule. We mention early works by [12] for recommender systems, [5] for neural networks [9], or [16] for generalized additive models. Another generic solution has been described in [1] and [4] focused on medical applications. Recently, a spe-

Proceedings of the 22^{nd} International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

cial attention has also been given to deep neural systems. We refer for instance to [17], [20] and references therein. Clues for real-world applications are given in [11] and [19] recently proposed to locally mimic a black-box model and then to give a feature importance analysis of the variables at the core of the prediction rule. In [15], a discussion was recently opened to refine the discourse on interpretability. In [13] the authors finally proposed a strategy to understand black-box models, as we do, but in a parametric setting.

Inspired by sensitivity analysis of computer code experiments [14], we propose in this paper a sensitivity analysis strategy for machine learning algorithms. In the field of computer code experiments sensitivity analysis is an active research topic. In this context, sensitivity analysis allows to rank the relative importance of the input variables involved in an abstract input-output relationship modeling the computer code under study. The much popular way to perform this analysis relies on the so-called Sobol' indices built on second order moments, as developed for instance in the pioneering work of [21]. Note that this index is related to the *Mean Decrease in Accuracy* (MDA) score produced by the random forest algorithm [10].
In this paper, we use the idea developed in [14] consisting in reweighing the observations by stressing the mean of one of the explanatory variables. Then, we quantify the stress impact through variations of a pre-defined quality index. We apply this method to machine learning methods in order to understand the effect of each variable after having learned black-box decision rules that are potentially complex. As mentioned earlier, a learned relation between the input variables $X^1, \dots, X^p$ and the prediction $f_n(X^1, \dots, X^p)$ is not necessarily clear. For linear rules such as regression type prediction rules (*e.g.* regression, logit regression, linear SVM) or decision trees, there is an interpretation that enables to explain individually the effect of each variable. In most cases, the rules are however too complex and cannot be understood directly. For instance, Random Forests methods, boosted algorithms or deep learning algorithms are black-box strategies for which the role played by each variable is not clear. We refer to [22] for a description of all these methods.

Our conception of the notion of interpretability for machine learning algorithms is the ability to quantify the specific influence of each variable on the predictions. In order to make understandable how a black-box rule is constructed, our goal is therefore to quantify the particular effect of each of the p covariates. To achieve this, we study particular variations of each test variable X^j in order to understand how the predictions are impacted by such changes. By studying the effect of different modifications of each variable on the predictions, we quantify the effect and the causality of each variable with respect to the decision rule.
The main difficulty related to such modifications is to create test data $(X_i, Y_i), i = 1, \dots, n$ from the original test sample, with the constraint that the distribution of the new data is as close as possible to the underlying data distribution. The goal of this constraint is to ensure that the created test data correspond to realistic observations and do not create artificial outliers such that the PAC learning framework [23] still holds.
In addition to propose our explainability formalism, our second main contribution is then to define an algorithm to generate such datasets, with the least modification of the distribution function. This algorithm is based on an information theory framework using entropy projection with Kullback-Leibler information as developed in [6, 7] for instance. This method has the key advantages that it quickly reweights the original datasets and that it does not require to compute new predictions using the transformed test data. It therefore enables fast computations of the output distributions by only reweighing the empirical criteria.

The paper falls into the following parts. The proposed data reweighting method to perturb the original sample is explained in Section 2. Section 3 recasts the notions of interpretability for machine learning using perturbed data. In particular, we construct indicators explaining decision rules for three different cases: 2-class classification, multi-class classification and the regression case. Section 4 finally presents applications on real datasets out of [8].

2 Optimal perturbation of distributions under moment constraints

In order to experience and to explore the behavior of a predictive model, a natural idea is to study its response to stressed inputs. Given a probability distribution Q on an abstract measurable polish space $(E, \mathcal{B}(E))$ and ε in a neighborhood of 0, there are many ways to create a perturbed probability measure Q_ε close to Q. A natural way to build such Q_ε is to stress the mean value of a given variable. That is, to consider a Q-integrable real random variable Φ defined on $(E, \mathcal{B}(E))$ and to enforce its mean value to deviate a little bit from the original mean value while the distribution of the whole random variables remains close to Q. Then, an information theory point of view leads to the use of the Kullback-Leibler information to perform this task.

To begin with, let us recall the definition of the Kulback-Leibler information (also called mutual entropy). If P is a probability measure on $(E, \mathcal{B}(E))$, then the Kullback-Leibler information $KL(P,Q)$ is defined as

$$\begin{cases} \displaystyle\int_E \log \frac{\mathrm{d}P}{\mathrm{d}Q}\,\mathrm{d}P, \text{ if } P \ll Q \text{ and } \log \frac{\mathrm{d}P}{\mathrm{d}Q} \in L^1(P), \\ +\infty, \text{ otherwise.} \end{cases}$$

Our information theory based trick to perform a stressed probability of the inputs of the learning system consists in minimizing $KL(P,Q)$ over the probability measures P that satisfy

$$\int_E \Phi(x)\,\mathrm{d}P(x) = \phi(\varepsilon),$$

where ϕ is a given continuous function on a neighborhood of 0 with $\phi(0) = \int_E \Phi(x)\,\mathrm{d}Q(x)$. In other words, we consider the following optimization problem

$$(P)_{\Phi,\phi,\epsilon} \quad : \quad \inf KL(P,Q),\ P \in \mathbb{P}_{\Phi,\phi,\epsilon},$$

where $\mathbb{P}_{\Phi,\phi,\epsilon}$ is the set of probability measures on $(E, \mathcal{B}(E))$ such that

$$\int_E \Phi(x)\,\mathrm{d}P(x) = \phi(\varepsilon).$$

Hence $\phi(\varepsilon)$ represents the average amount of deformation, that has to be incorporated in the initial distribution Q without modifying too much the initial distribution so that the KL distance between the initial and the warped distribution remains small.

We also set $Q_\varepsilon := \arg\inf_{P \in \mathbb{P}_{\Phi,\phi,\epsilon}} KL(P,Q)$ whenever it exists. The following theorem due to Csiszár in [6] and [7] explains the solution of the previous optimization problem.

Theorem 2.1. *Assume that $\mathbb{P}_{\Phi,\phi,\epsilon}$ contains a probability measure that is mutually absolutely continuous with respect to Q. Then, Q_ε exists and is unique. Furthermore,*

$$Q_\varepsilon = \frac{\exp \lambda_\varepsilon \Phi}{Z(\lambda_\varepsilon)} Q. \qquad (1)$$

Here, $Z(\lambda) := \int_E e^{\lambda \Phi(x)}\,\mathrm{d}Q(x)$, $(\lambda \in \mathbb{R})$, and λ_ε is the unique minimizer of the strictly convex function

$$H(\lambda) := \log Z(\lambda) - \lambda \phi(\varepsilon),\ (\lambda \in \mathbb{R}).$$

We will mainly put in action the previous theorem in the following frame of discrete distributions.

Definition 2.2. *Let $x_1, \ldots, x_n$ be two by two distinct real numbers and let $m = (1/n)\sum_{i=1,\ldots,n} x_i$. Let $\epsilon \in \mathbb{R}$ be so that $\min_{i=1,\ldots,n} x_i < m + \epsilon < \max_{i=1,\ldots,n} x_i$. Let*

$$\psi_{x_1,\ldots,x_n}(\tau) = \log\left(\frac{1}{n}\sum_{j=1}^{n} \exp(\tau x_j)\right).$$

Let $\tau(m+\epsilon)$ be the unique minimizer of the strictly convex function

$$H_{m+\epsilon}(\tau) = \psi_{x_1,\ldots,x_n}(\tau) - \tau(m+\epsilon).$$

Then, let $\lambda_1^{(x_1,\ldots,x_n),\epsilon}, \ldots, \lambda_n^{(x_1,\ldots,x_n),\epsilon}$ be defined by, for $i = 1, \ldots, n$

$$\lambda_i^{(x_1,\ldots,x_n),\epsilon} = \exp\Big(\tau(m+\epsilon)x_i - \psi_{x_1,\ldots,x_n}(\tau(m+\epsilon))\Big).$$

In the following, we let $X = (X^1, \ldots, X^p) \in \mathbb{R}^p$ be an input variable, $Y \in \mathbb{R}$ be the corresponding label and $\hat{Y} = f_n(X)$ be the predicted label, with $f_n : \mathbb{R}^p \to \mathbb{R}$ being the black-box model. We consider a test base $(X_1, \hat{Y}_1, Y_1, \ldots, X_N, \hat{Y}_N, Y_N)$ with again $\hat{Y}_i = f_n(X_i)$. We let $Q_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_i,\hat{Y}_i,Y_i}$ be the original (unperturbed) distribution of the test base. Using Theorem 2.1, we modify the empirical distribution of the test sample as follows:

Definition 2.3. *For $i_0 \in \{1, \ldots, p\}$, let $m_{i_0} = (1/N)\sum_{i=1}^{N}(X_i)^{i_0}$. Let $\alpha \in (0, 1/2)$ and let $q_{i_0,\alpha}$ and $q_{i_0,1-\alpha}$ be the α and $1-\alpha$ quantiles of $\{(X_1)^{i_0}, \ldots, (X_N)^{i_0}\}$. Assume that α is small enough so that $q_{i_0,\alpha} < m_{i_0} < q_{i_0,1-\alpha}$. For $\tau \in [-1, 0]$, let $\epsilon_{i_0,\tau} = \tau(m_{i_0} - q_{i_0,\alpha})$. For $\tau \in [0,1]$, let $\epsilon_{i_0,\tau} = \tau(q_{i_0,1-\alpha} - m_{i_0})$. Finally define*

$$\lambda_i^{(i_0,\tau)} = \lambda_i^{((X_1)^{i_0},\ldots,(X_N)^{i_0}),\epsilon_{i_0,\tau}}$$

Theorem 2.4. *Let*

$$Q_{N,i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N} \lambda_i^{(i_0,\tau)} \delta_{X_i,\hat{Y}_i,Y_i}.$$

$Q_{N,i_0,\tau}$ is solution of the minimization program $\min_\nu$: $\nu \mapsto KL(\nu, Q_N)$ under the constraint that

$$E_\nu(X^{i_0}) = m_{i_0} + \epsilon_{i_0,\tau}.$$

This theorem enables to reweight the observations of each variable so that its mean increases or decreases. In Definition 2.3, τ indicates a change of mean proportional to the range of empirical values of the variable i_0. More precisely, $\tau = 0$ yields no change of mean, $\tau = -1$ changes the mean from m_{i_0} to the (small) quantile $q_{i_0,\alpha}$ and $\tau = 1$ changes the mean from m_{i_0} to the (large) quantile $q_{i_0,1-\alpha}$.

Proof. The proof follows directly from Theorem 2.1 with $\Phi(X, \hat{Y}, Y) = X^{i_0}$ and $\varphi(\epsilon) = m_{i_0} + \epsilon$, applied with $\epsilon = \epsilon_{i_0,\tau}$. ■

Hence we have defined a strategy to resample the data in order to stress individually each variable while keeping as much as possible the original distribution. Note that the generation of the new warped sample does not involve the creation of new $(X_i, \hat{Y}_i, Y_i)$ but only fast computations of weights λ_i's. This makes it possible to deal with very large databases without computing new values for new observations. We can thus highlight the effect of every variable in the decision rule as follows.

3 Explainable models using perturbed distributional entries

A machine learning algorithm aims to find the link between two random variables X and Y with distribution Q_{obs}. $X \in E$ stands for the covariates and $Y \in \mathcal{Y}$ has to be predicted. Let ℓ be a loss function quantifying how a predictor is close enough to the variable to be predicted. Then the goal of machine learning is to build a rule f such that $\ell(f(X), Y)$ is small for all observations that are similar to the observations at hand. In the framework of statistical learning, we assume that we have at hand a learning sample $\mathcal{D}_L = \{(X_{i,L}, Y_{i,L}) i = 1, \ldots, n\}$ drawn from an unknown distribution Q_{obs} which is approximated using its empirical version $Q^{\mathbb{L}}_{\text{obs},n} = \frac{1}{n}\sum_{i=1}^{n} \delta_{(X_i,Y_i)}$. Let f_n be a decision rule calibrated to minimize criteria that depend on the empirical loss function $\mathbb{E}_{Q^{\mathbb{L}}_{\text{obs},n}} \ell(f_n(X), Y) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_n(X_{i,L}), Y_{i,L})$ and possibly other terms designed to prevent overfitting of the data. The algorithm is designed to perform on every similar data and its accuracy is assessed on test samples $\mathcal{D}_T = \{(X_i, Y_i) i = 1, \ldots, N\}$ drawn by using the same distribution Q_{obs} and the empirical distribution $Q_{\text{obs},N}$. To quantify the change of each variable we will change the test set according to Theorem 2.4 by changing the mean of each variable $i_0 \in \{1, \ldots, p\}$ and generating training sets $\mathcal{D}_{i_0,\tau}$ with empirical distribution $Q_{\text{obs},N,i_0,\tau}$ for different values of τ. We point out that this procedure amounts to change weights associated to the variables, the predictions and the true outputs $(X_i, f_n(X_i), Y_i)$. Hence we do not monitor changes with respect to particular individuals' decisions, since the data do not change, but the changes in the global behavior of the algorithm.

In particular, we consider how the perturbations in the distribution of $f_n(\mathcal{D}_T)$ impact the indicators that are meaningful with respect to the behavior of the algorithm such as the error rates including global error, false positive but also the changes with respect to the proportion of quantities predicted and their variability. Such changes are studied for each variable and for different amounts of perturbation, which enables to stress the importance of each variable and the particular bias it induces.
Hence explainability in this paper has to be understood in a general framework. We try to determine the global effect of each variable in the learning rule and how a particular variation of each variable affects the accuracy of the prediction, enabling to understand how the algorithm evolves when a characteristic of the observations is modified. Our framework is closely related to what is done in computer code experiment when dealing with sensitivity analysis. More precisely, in [14] a reweighing method has been proposed and studied. The method allows to measure the effect on a probability of failure of a perturbation performed on an input distribution

We point out that this point of view is different from previous works where the importance of each variable was considered. Sparse models (see for instance in [3] for general introduction on the importance of sparsity) enable to identify few important variables. Importance indicators have also been developed in machine learning to detect which variables play a key role in the algorithm. For instance importance of variables is often computed using feature importance or Gini indices (see in [18] or [22]). Yet such indexes are computed without investigating the particular effects of each variable and without explaining its particular role in the decision process. We now detail three particular cases encountered in machine learning: two-class classification, multi-class classification and the regression case.

3.1 The case of binary classification

Consider the case of a classification in two cases, i.e where $Y_i, f_n(X_i)$ belong to $\{0, 1\}$ for all $i = 1 \ldots, N$. This case corresponds to the two-case classification problem for which the usual loss function is $\ell(Y, f(X)) = \mathbf{1}\{Y \neq f(X)\}$. We suggest to consider the following indicators for the perturbed distributions $\frac{1}{N}\sum_{i=1}^{N} \lambda_i^{(i_0,\tau)} \delta_{(Y_i,\hat{Y}_i,Y_i)}$.

Understanding the classification rule in this case corresponds first to monitor the evolution of the error rate. So the first index is the error rate

$$\text{ER}_{i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N} \lambda_i^{(i_0,\tau)} \mathbf{1}\{f_n(X_i) \neq Y_i\}.$$

We suggest to plot $\text{ER}_{i_0,\tau}$ as a function of τ for $\tau \in [-1, 1]$ for each $i_0 \in \{1, \ldots, p\}$. The case $\tau = 0$ provides the baseline of the algorithm performance without perturbation of the learning sample. Here, we highlight the variables which produce most confusion in the error so for which the variability among the two predicted class is the most important, hampering the prediction error rate.
Then, we can consider more precisely the error term since it can be decomposed into the true and the false

positive rate. So consider the false positive rate

$$\mathrm{FPR}_{i_0,\tau} = \frac{\frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\mathbf{1}\left\{Y_i \neq 1\right\}}{\frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\mathbf{1}\left\{f_n(X_i) = 1\right\}}.$$

and the true positive rate

$$\mathrm{TPR}_{i_0,\tau} = \frac{\frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\mathbf{1}\left\{f_n(X_i) = 1\right\}}{\frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\mathbf{1}\left\{Y_i = 1\right\}}.$$

A ROC curve corresponding to perturbations of the variable i_0 can then be obtained by plotting pairs $(\mathrm{FPR}_{i_0,\tau}, \mathrm{TPR}_{i_0,\tau})$ for a large number of values of $\tau \in [-1, 1]$. We then obtain the evolution of both errors when τ evolves, for a sharper analysis of the evolution of the error.

Finally, the influence of each variable on the prediction may be quantified by computing the proportion of predicted observations with label equal to one

$$\mathrm{P1}_{i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)} f_n(X_i)$$

that we suggest to plot similarly as $\mathrm{ER}_{i_0,\tau}$. The obtained figure provides a way to understand the particular influence of the variables and their variations to obtain a given decision $Y = 1$, whatever the veracity of the prediction but pointing out which variable should be modified and in which sense in order to change a given decision.

Remark. *The case where the labels are taken as $\{-1, 1\}$ and for which the loss function is the hinge loss can be tackled in a similar way with the corresponding changes in previous definitions.*

3.2 The case of multi-class classification

Consider now the case of a classification into k different categories. In this case $Y_i, f_n(X_i)$ belong to $\{1, \ldots, k\}$ for all $i = 1 \ldots, N$ where $k \in \mathbb{N}$ is fixed. In this case, the error rate $\mathrm{ER}_{i_0,\tau}$ can be defined and plotted as in the binary classification case.

The evolution, with respect to τ, of the number of individuals predicted to belong to a given class must be here expressed for all classes. Hence, for all $j \in \{1, \ldots, k\}$, we suggest to consider the proportion of j criterion

$$\mathrm{Pj}_{i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\mathbf{1}\left\{f_n(X_i) = j\right\},$$

which denotes the proportion of individuals assigned to the j-class. For all $j = 1, \ldots, k$, these quantities $\mathrm{Pj}_{i_0,\tau}$ can be plotted similarly as $\mathrm{P1}_{i_0,\tau}$.

These criteria are the natural generalization of the indicators defined for the two-class classification.

3.3 The case of continuous regression

Consider now the case of a real valued regression where $Y_i, f_n(X_i) \in \mathbb{R}$ for $i = 1 \ldots, N$. In order to understand the effects of each variable, first we consider, the mean criterion

$$\mathrm{M}_{i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)} f_n(X_i),$$

which will indicate how a change in the variable will modify the output of the learned regression. Second the variance criterion

$$\mathrm{V}_{i_0,\tau} = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\left(f_n(X_i) - \mathrm{M}_{i_0,\tau}\right)^2$$

is meant to study the stability of the regression with respect to the perturbation of the variables. Finally the root mean square error (RMSE) criterion

$$\mathrm{RMSE}_{i_0,\tau} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\lambda_i^{(i_0,\tau)}\left(f_n(X_i) - Y_i\right)^2}$$

is analogous to the classification error criterion since it enables to detect possibly misleading variable or confusing variables when learning the regression.

For each $i_0 \in \{1, \ldots, p\}$, these three criteria can be plotted as a function of τ for $\tau \in [-1, 1]$.

4 Use Case of interpretability through resampling

4.1 Two class classification

To illustrate the performance of the procedure we propose, we consider the *Adult Income* dataset. It contains 29.825 instances consisting in the values of 14 attributes, 6 numeric and 8 categorical, and a categorization of each person as having an income of more or less than 50, 000$ per year. This attribute will be the target variable Y that has to be predicted. Hence this variable is either $Y = 1$ corresponding to a high income or $Y = 0$ corresponding to a low income.

We train three different classifiers (Logit Regression, XGboost and Random Forest[1]) in order to predict this label.

Then we perform the sensivity analysis by computing the analysis described in Section 3 and providing different indicators. This analysis is performed for 50 different random choices of the learning and test samples, which enables to provide a confidence interval for each indicator, and also to assess the stability of the classifiers.

[1] `R` command `glm` and packages `xgboost` and `ranger`.

For all classifiers, we present in Fig. 1 *(Left)* the role played by each variable in the decision. This plot highlights the role played by the variable education number. The more educated, the higher the income will be and inversely. The two variables *LcapitalGain* and *LcapitalLoss* are also testimonial of high incomes since people with large income have more money on their bank account or may easily contract debts but the contrary is not true. It is worth pointing out the role played by the age variable which appears clearly in the figure: young people have smaller income but increasing the age is not enough to increase the income.

We then present in Fig. 1 *(Center)* the evolution of the classification error when all variables are shifted according to τ, increasing or decreasing. The three models enable to select the same couple of variables that are important for the accuracy of the prediction when they increase: education number and numbers of hours worked pro week. The latter makes the prediction task the most difficult when it increases. Indeed, people working a large number of hours per week may not always increase their income since it relies on different factors but people with high income also work a large number of weekly hours. Hence these two variables play an important role in the prediction and their changes impact the prediction error.

Finally, the evolution of the False Positive Rate and True Positive Rate is presented in Fig. 1 *(Right).*

4.2 Multiclass classification

We now consider the *Iris* dataset which serves as a toy example for many methods. This dataset is composed of 150 observations with 4 variables used to predict a label into three categories: *setosa*, *versicolor*, *virginica*. To predict the labels, we used an Extreme Gradient Boosting model and a Random Forest classifier (in Fig. 2). We first present for both models the Classification error. Then the two other subfigures show the effects of increasing or decreasing the 4 parameters, i.e the width or the length of the sepal or petal is shown for all classes. We recover the well known result that the width of the sepal is the main parameter which enables to differentiate the class *Setosa* while the differentiation between the two other remaining classes is less obvious.

4.3 Regression case

We use now our strategy on the Boston Housing dataset. These data deal with houses prices in Boston. It contains 506 observations with 13 variables that should be used to predict the price of the house to be sold. When considering an optimized Random Forest algorithm, the importance calculated as described in [2], enables to select the 5 most important variables as follows: *lstat* (15227), *rm* (14852), *dis* (2413), *crim* (2144) and *nox* (2042).

Our analysis goes further than this score. Indeed, if these variables are shown to play an important role, their impact on the predictions is made understandable with our methodology. In particular we point out the non linear influence of the variables depending whether they increase or decrease, as shown in Fig. 3. For instance the size of a house (with variable code rm for the average number of rooms) is an important factor that makes the price increase while smaller houses are not always the cheapest since sellers find other arguments than size in such cases. Other variables show more linear influence such as age for instance.

Other criteria could be studied looking at the gradient of the functions plotted in Fig. 3 to identify, in the presence of a large number of variables, few variables that have the largest impact with respect to a chosen criterion.

5 Conlusion and Future Work

Explainability is a difficult task and has many interpretations. In this work we focused on the analysis of the variables importance and their impact on a decision rule. When building a surface response in computer code experiments, the prediction algorithm is applied to new entries to explore its possible outcomes. In the machine learning framework the issue is quite different since the test input variables must follow the distribution of the learning sample. Therefore, evaluating the decision rule at all possible points does not make any sense. Hence we have proposed an information theory procedure to obtain perturbations of the original variables without loosing the information conveyed by the initial distribution. The proposed solution amounts to resample the observation points of the testing sample, leading to very fast computations and to the construction of new indices that enable to understand the weight and the direction played by each variable. This method must be restricted to large datasets since we only explore the observations we have already that should be in a sufficiently large number.

This resampling method is presented here in view of the explainability of a machine learning algorithm. Yet it will be used in future work for a robust analysis of an algorithm since the resampled observations can be seen as stressed input variables and the produced indices as stability indices that highlight the resiliency of the decision to contamination of the observations.

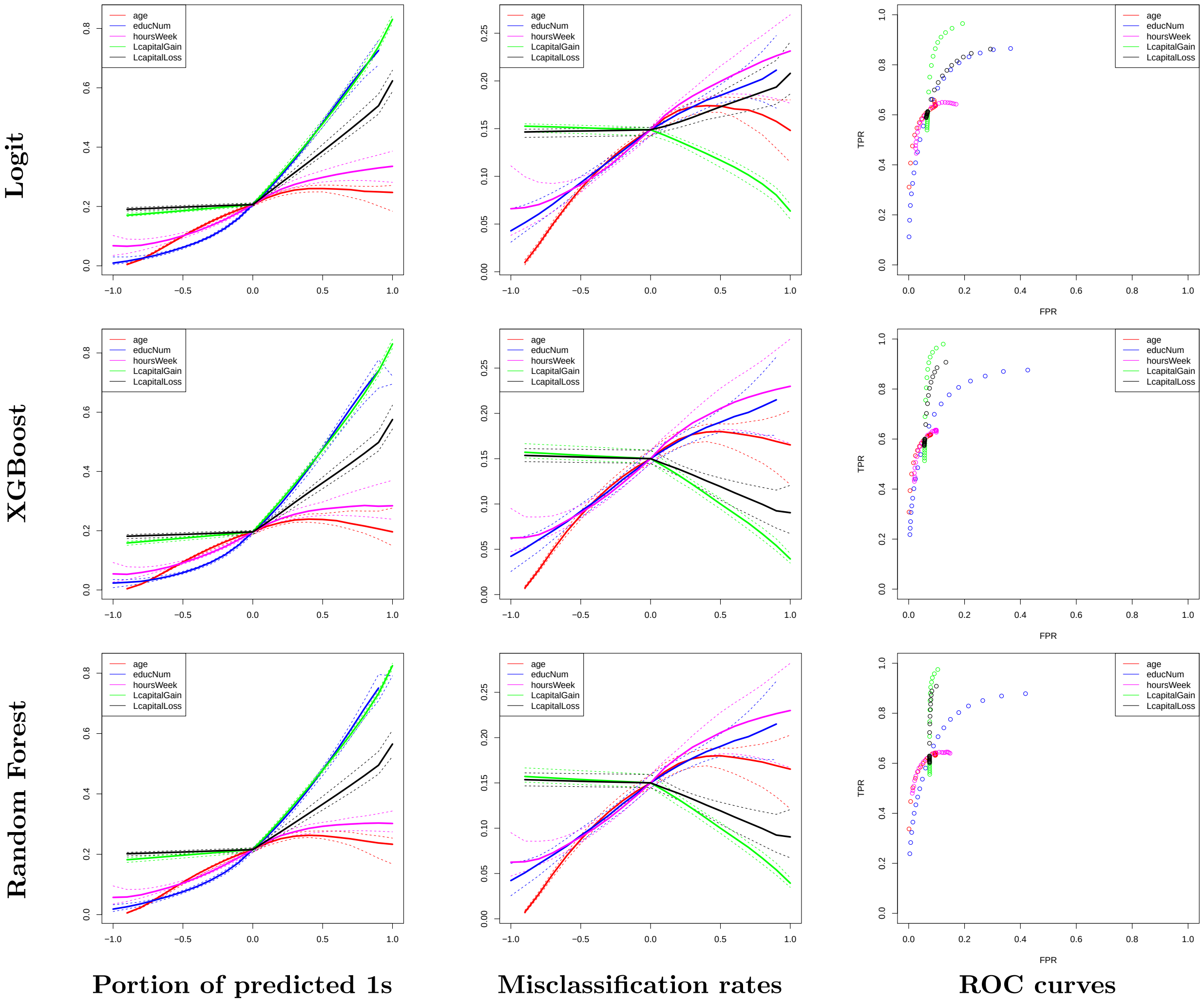

Figure 1: Results of Section 4.1 on the *Adult income* dataset. **(Left)** Portion of predicted ones (*High Income*) with respect to the explanatory variable perturbation τ. **(Center)** Classification Error in the *Adult income* dataset with respect to τ. **(Left-Center)** There is no perturbation if $\tau = 0$, and larger or lower values of τ indicate that larger or lower values of the explanatory variable receive more weight, respectively. The dashed lines represent the 10% and 90% quantiles of the indicators, over the randomly sampled test and learning bases. The plain lines represent the medians. Here, the analysis of the predictions highlights the importance of the *age* and *LcapitalGain* variables. Large values of the explanatory variable *hoursWeek* also yield a difficult classification. **(Right)** Evolution of Roc Curves in the *Adult income* dataset. As for the classification errors, we observe that large values of the variable *hoursWeek* make the classification difficult.

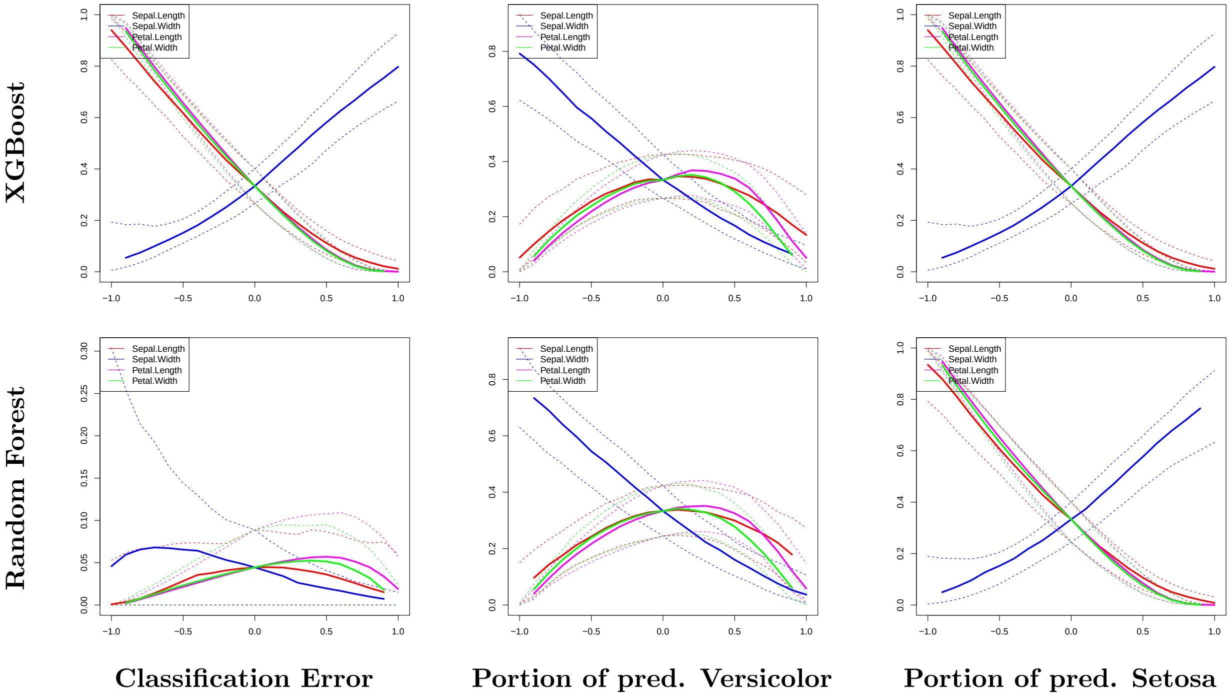

Figure 2: Evaluation of the classification error and the prediction with respect to the explanatory variable perturbation τ, on the *Iris* dataset (Section 4.2). The quantity τ and the plain/dashed lines have the same signification as in Fig. 1. **(Top)** XGBoost Model. The sepal width enables to differenciate the *Setosa* class. **(Bottom)** Random Forest Model. The sepal width again enables to differenciate the *Setosa* class.

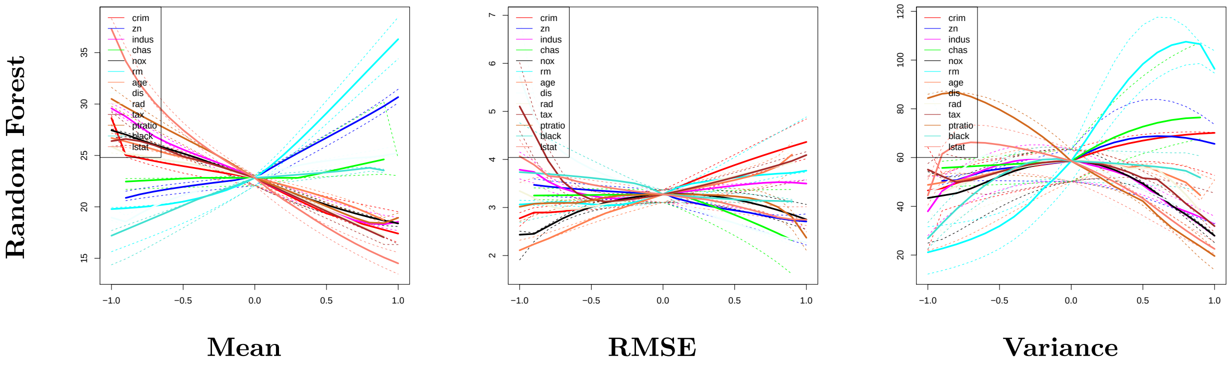

Figure 3: Results obtained on the *Boston Housing* dataset (Section 4.3) with respect to the explanatory variable perturbation τ . The quantity τ and the plain/dashed lines have the same signification as in Fig. 1.

References

[1] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] Peter Bühlmann and Sara Van De Geer. Introduction. In *Statistics for High-Dimensional Data*, pages 1–6. Springer, 2011.

[4] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, New York, NY, USA, 2015. ACM.

[5] Mark Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 24–30, 1995.

[6] Imre Csiszár. *I*-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pages 146–158, 1975.

[7] Imre Csiszár. Sanov property, generalized *I*-projection and a conditional limit theorem. *The Annals of Probability*, pages 768–793, 1984.

[8] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[9] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.

[10] Baptiste Gregorutti. *Random forests and variable selection : analysis of the flight data recorders for aviation safety*. Theses, Université Pierre et Marie Curie - Paris VI, March 2015.

[11] Patrick Hall, Navdeep Gill, and Mark Chan. Practical techniques for interpreting machine learning models: Introductory open source examples using python, h2o, and xgboost, 2018.

[12] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[13] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[14] Paul Lemaître, Ekatarina Sergienko, Aurélie Arnaud, Nicolas Bousquet, Fabrice Gamboa, and Bertrand Iooss. Density modification-based reliability sensitivity analysis. *Journal of Statistical Computation and Simulation*, 85(6):1200–1223, 2015.

[15] Zachary C. Lipton. The mythos of model interpretability. In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, pages 96–100, 2016.

[16] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 150–158, New York, NY, USA, 2012. ACM.

[17] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[18] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, May 2004.

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[20] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[21] Ilya M Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical modelling and computational experiments*, 1(4):407–414, 1993.

[22] Hastie Trevor, Tibshirani Robert, and Friedman JH. The elements of statistical learning: data mining, inference, and prediction, 2009.

[23] Leslie Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books (AZ), 2013.