Explainable Performance*

Sullivan Hué† Christophe Hurlin‡ Christophe Pérignon§ Sébastien Saurin¶

December 13, 2022

Abstract

We introduce the XPER (eXplainable PERformance) methodology to measure the specific contribution of the input features to the predictive or economic performance of a model. Our methodology offers several advantages. First, it is both model-agnostic and performance metric-agnostic. Second, XPER is theoretically founded as it is based on Shapley values. Third, the interpretation of the benchmark, which is inherent in any Shapley value decomposition, is meaningful in our context. Fourth, XPER is not plagued by model specification error, as it does not require re-estimating the model. Fifth, it can be implemented either at the model level or at the individual level. In an application based on auto loans, we find that performance can be explained by a surprisingly small number of features. XPER decompositions are rather stable across metrics, yet some feature contributions switch sign across metrics. Our analysis also shows that explaining model *forecasts* and model *performance* are two distinct tasks.

Keywords: Machine learning; Explainability; Performance metric; Shapley value

*We thank for their comments Jean-Edouard Colliard, Serge Darolles, Jean-François Dupuy, Emmanuel Flachaire, Thierry Foucault, Julien Grand-Clément, Emmanuel Kemel, Sébastien Laurent, Jean-Michel Poggi, Gilbert Saporta, Olivier Scaillet, Nicolas Vieille, participants at the CISEM 2022, the 2022 Quantitative Finance and Financial Econometrics (Aix-Marseille School of Economics), the 2022 Financial Econometrics Day (University Paris Nanterre), and seminar participants at the University of Orléans. We thank the Institut Universitaire de France (IUF), the ACPR Chair in Regulation and Systemic Risk, and the French National Research Agency (Ecodec ANR-11-LABX-00474, MLEforRisk ANR-21-CE26-0007, and CaliBank ANR-19-CE26-0002-02) for supporting our research.

†Aix-Marseille University (Aix-Marseille School of Economics), CNRS, and EHESS. Email: sullivan.hue@univ-amu.fr

‡University of Orléans (LEO) and Institut Universitaire de France (IUF), Rue de Blois, 45067 Orléans, France. Email: christophe.hurlin@univ-orleans.fr

§HEC Paris, 1 Rue de la Libération, 78350 Jouy-en-Josas, France. Email: perignon@hec.fr

¶University of Orléans (LEO), Rue de Blois, 45067 Orléans, France. Email: sebastien.saurin@univ-orleans.fr

1 Introduction

Why is the AUC of a given machine learning classifier equal to 0.7? Which features explain this performance? Why did this AUC drop from 0.9 in the training sample to 0.7 in the test sample? What are the contributions of the different features to the MSE of a black-box regression model? What are the main features explaining the economic performance of a customer churn model? These are some of the questions we aim to answer in this paper. To do so, we develop a general methodology, called eXplainable PERformance (XPER), which measures the marginal contribution of a particular feature to the performance of a regression or classification model.

Being able to identify the driving forces of the economic or statistical performance of a predictive model lies at the very core of modelling and is of great importance for both data scientists and business experts basing their decisions on such model. For instance, it can allow data-scientists to understand the source of, and mitigate, overfitting, to improve feature selection, or to address heterogeneity issues by identifying groups of individuals on which the model performs poorly. Moreover, it can benefit business experts by enabling them to make more informed decisions, e.g., investing in the factor with the strongest impact in terms of P&L or assessing the economic value of a given feature.

However, designing a general framework to explain model performance is a challenging task. First, it needs to be adapted to the wide variety of performance measures. In practice, dozens of statistical performance metrics are commonly used for classification and regression models: R^2, RMSE, AUC, PCC, specificity, sensitivity, Brier score, Gini index, etc. Similarly, in many applications the performance of the model is assessed in monetary terms, hence requiring an economic performance metric to be broken down. Second, it needs to accommodate any model: parametric or not, linear or not, econometric model or machine learning algorithm, etc.

To overcome these challenges, we propose a general framework based on Shapley values (Shapley (1953)). While the Shapley values decomposes a *payoff* among *players* in a *game*, the XPER values decomposes a *performance metric* among *features* in a *model*. More precisely, XPER decomposition

allows to break down the difference between a performance metric obtained on a test sample and a benchmark value, among the features of the model. Formally, an XPER value is defined as the weighted average contribution of a given feature to a performance metric, obtained in a set of coalitions of other features. For instance, evaluating the XPER value of x_1 in a three-feature model implies to assess the incremental performance due to x_1 by successively considering four subsets or coalitions of features: one coalition including no features, only x_2, only x_3, and both x_2 and x_3. Although the Shapley methodology is well known, its application in the context of model performance explanation is not trivial. Indeed, many Shapley values can be defined given the assumptions made on the model, on the data, and on the features excluded from the coalitions. Unfortunately, as of today, there is no consensus about the need to re-estimate the model for each coalition, to marginalise the metric over the features excluded from the coalitions, to choose a specific benchmark for the performance metric, to replace the features excluded from the coalitions by ad-hoc values, to take into account the dependence between features, etc.[1]

In this paper, we choose to define XPER values by considering the expectation of the performance metric with respect to the joint distribution of the features excluded from the coalition. Compared to alternative approaches, we show that a key advantage of this definition is that the benchmark value has a meaningful interpretation: it corresponds to the performance metric that we would obtain on a hypothetical test sample in which the target variable is independent from all the features included in the model, i.e., when the model is completely misspecified. To grasp the intuition, let us consider a hypothetical model with three features and an AUC of 0.90 in the test sample. XPER allows to allocate the difference between the AUC and a benchmark value of 0.50 among all features. The benchmark value corresponds to the AUC that we would obtain if the model had been evaluated on a hypothetical test sample where the three features were independent from the target variable. In this case, the benchmark value equal to 0.50, corresponds to the AUC of a random predictor. The difference of 0.40 between the AUC and the benchmark captures the *over-performance* of the

[1]For more discussions about the computation of Shapley values in other contexts, see Strumbelj and Kononenko (2010), Lundberg and Lee (2017), Owen and Prieur (2017), Chantreuil et al. (2019), Redell (2019), Kumar et al. (2020), Sundararajan and Najmi (2020), Aas et al. (2021), or Singal et al. (2022).

model due to the dependence between its features and the target variable. Having XPER values of 0.20, 0.12, and 0.08 means that features x_1, x_2, and x_3 contribute respectively to 50%, 30%, and 20% of the over-performance of the model.

XPER values offer many other advantages. First, they are theoretically grounded and satisfy the desirable properties of a Shapley decomposition, such as symmetry, monotonicity, and dummy variable. Second, they are *model-agnostic* as they permit to interpret the predictive performance of any type of econometric or machine learning model. Third, they are *metric-agnostic* as they can be used with any type of performance metric: predictive accuracy (AUC, Gini, PCC), goodness of fit (R^2), information criteria (AIC, BIC), statistical loss function (MSE, MAE, Q-like), or economic performance metric (profit and loss function). Fourth, we can evaluate XPER either at the global/model level or at the local/individual level. This is one of the main differences with respect to the SHAP values proposed by Lundberg and Lee (2017). At the global level, the XPER value of a given feature measures its contribution to the performance of the model. At the local level, the XPER value of a given feature measures the *contribution of the feature* to the *contribution of an individual* to the model performance. Fifth, XPER requires neither re-estimating the model nor picking an ad-hoc value for features excluded from the coalition. The former means that XPER is not plagued by model specification error or omitted variable bias (see Grömping (2007) for a discussion in the specific case of the R^2 decomposition). The latter avoids to get an attribution scheme for the performance which is dependent on a specific value for the excluded features (Israeli, 2007).

We illustrate our decomposition method for different models and performance metrics using both theoretical examples and Monte Carlo simulations. We highlight the interpretation of estimated XPER values and show that they can be used to identify features improving or impairing the accuracy of a model. We propose an estimation method of XPER values both for models with few features, for which it is possible to go through all the feature coalitions, and for models with a large number of features for which this approach is unfeasible. Finally, we implement our methodology

on a credit scoring model trained on a database of auto loans provided by an international bank. We provide a set of graphical representations allowing an efficient and accurate diagnostic of feature contributions. The main empirical insights are the following. A small number of features can explain a surprisingly large part of the model performance. Moreover, the XPER decomposition is rather stable across performance metrics, yet some feature contributions can vary drastically, and can even turn negative, from one performance metric to the next. The empirical analysis confirms that explaining model forecasts (through feature importance or SHAP) and model performance are two distinct and complementary tasks.

Our paper contributes to the burgeoning literature on explainable artificial intelligence (XAI). One well-known limitation of AI and machine learning methods comes from their opacity and lack of explainability. Most of these algorithms are considered as black boxes in the sense that the corresponding outcomes cannot be easily explained to final users nor related to the initial features. Recently, various explainability methods have been designed to explain black-box models by measuring which features most affect its output.[2] Standard XAI methods include the partial dependence plots (Friedman (2001)), the individual conditional expectation (Goldstein et al. (2015)), the accumulated local effects (Apley and Zhu (2020)), the global or local surrogate models (e.g., the LIME methodology of Ribeiro et al. (2016)), etc. One of the most cited methods is the Shapley additive explanation (SHAP) of Lundberg and Lee (2017). SHAP distributes the prediction of a model among its features.[3] While model predictive performance obviously depends on predictions, the contribution of a feature to the performance metric also depends on the true value of the target variable. Therefore, SHAP and XPER are not likely to be equivalent.[4] For instance, at the global level, XPER values can be either positive or negative while SHAP values are by design only positive.

[2]See Molnar (2020) for a survey and Wang et al. (2022) for a study of the importance of algorithmic transparency for AI-using firms.

[3]Many developments of SHAP have been recently proposed (Sundararajan et al., 2017; Lundberg et al., 2018; Agarwal et al., 2019; Aas et al., 2021; Senoner et al., 2021, etc.)

[4]Bowen and Ungar (2020) introduce the concept of Generalized SHAP, which measures feature importance with respect to some function of the model output. They also show the non-equivalence between SHAP and Generalized SHAP. A similar result is obtained in Borup et al. (2022) who show that for time-series models, the importance of individual predictors in determining the predicted target values does not necessarily align with the predictors' roles in determining forecasting accuracy.

When both are positive, we show that the relative contribution of a feature can be high to explain the outcome and low to explain performance. Furthermore, at the individual level, a feature can be positively related to the prediction score but negatively to the performance metric.

XPER also shares similarities with the Shapley Feature Importance (SFIMP) metric of Casalicchio et al. (2019).[5] Indeed, both approaches allow to measure and to visualize the contribution of a feature to the performance of a model on out-of-sample observations. However, XPER allows to conduct both global and local analyses thanks to the individual decomposition of the performance metric. Although similar in spirit, XPER also contrasts with the Performance-Based Shapley Value (PBSV) metric of Borup et al. (2022). Their approach also allows to measure the contribution of a feature to the performance of a model on out-of-sample observations, similarly to XPER. However, the PBSV metric is specifically designed to handle predictions obtained from sequence of fitted models on time-series data whereas XPER can be applied to predictions of a model that has been estimated or trained once on time-series or cross-section data.[6]

Our paper also contributes to the rich literature on the decomposition of performance metrics. Numerous methods have been proposed for the MSE (Theil, 1971; Ahlburg, 1984), for various inequality measures (Bourguignon, 1979; Shorrocks, 1980, 1982, 1984), and for the R^2 (see Grömping (2015) for a survey).[7] Unlike the previously mentioned authors, some break down performance metrics using Shapley values (Stufken, 1992; Lipovetsky and Conklin, 2001; Israeli, 2007; Grömping, 2007; Huettner and Sunder, 2012; Redell, 2019). These decompositions are both specific to a performance metric and a model. Furthermore, some of them re-estimate the model on different subsets of features, which may lead to omitted variable bias. Others use ad-hoc values for the features excluded from the coalitions. As shown below, XPER overcomes these limitations.

[5] Casalicchio et al. (2019) also propose a Permutation-based Feature Importance (PFI) method where the predictive performance is measured once with and once without permuted values of the feature of interest. Permuting the values of a feature breaks the association between the feature and the target variable and results in a large drop in performance if the considered feature is important.

[6] There are also differences in the computation of PBSV and XPER values. For example, XPER marginalises the performance metric over the features which are not included in the coalition by considering their distribution in the test set, whereas the PBSV approach relies on their distribution in the training set.

[7] Examples of R^2 decompositions include Green et al. (1978), Lindeman et al. (1980), Kruskal (1987), Chevan and Sutherland (1991), Genizi (1993), and Johnson (2000).

The rest of this paper is structured as follows. We start with an intuitive presentation of XPER in Section 2. In Section 3, we introduce the framework of analysis and the concept of performance metric. Section 4 introduces the XPER value decomposition of performance metrics. Section 5 describes the estimation procedure with two illustrations of the use of XPER values. Section 6 presents the empirical application and Section 7 concludes the paper.

2 An intuitive primer on XPER

In this section, we introduce XPER using a simple classification problem with $y \in \{0, 1\}$ the target variable and three features $\{x_1, x_2, x_3\} \in \mathbb{R}^3$. We estimate a model $\hat{f}(x_1, x_2, x_3)$ on a training sample and evaluate its predictive performance by considering its AUC on a test sample.[8] If none of the three features provides any useful information to predict the target variable, the AUC is at 0.5 (random classification). We refer to this value as the benchmark AUC, denoted ϕ_0. On the contrary, when the AUC is greater than 0.5, at least one feature contains some relevant information to predict the target variable.

Consider now the following question: What is the relative contribution of each feature to this predictive performance? We answer this question using XPER in order to break down the difference between the AUC obtained on the test sample and the benchmark value, among the features of the model. Let us denote by ϕ_j the XPER contribution of feature x_j to the predictive performance of the model. For instance, the AUC of the model can be 0.78 and the XPER decomposition can be as follows:

	AUC		ϕ_0		ϕ_1		ϕ_2		ϕ_3
Test sample	0.78	=	0.50	+	0.14	+	0.10	+	0.04

This decomposition indicates that the feature x_1 is the main driver of the predictive performance

[8] The receiver operating characteristic (ROC) curve is a graphical diagnostic tool for binary classifiers that allows considering a variation on its discrimination threshold. It corresponds to the plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The AUC (area under the curve) or AUROC (area under the receiver operating characteristics) measure can be interpreted as the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one. The AUC ranges from 0.5 (random classification) to 1 (prefect classification).

of the model as it explains half of the difference between the AUC of the model and its benchmark ($= 0.14/(0.78 - 0.50)$). The second most informative feature is x_2 as it explains another 35.7% ($= 0.10/(0.78 - 0.50)$) of the difference. Finally, x_3 explains the remaining 14.3%.

Such performance decomposition can prove handy in several contexts. First, XPER can help to rationalize a potential *heterogeneity* in the predictive performance of a model. Indeed, one can compute the XPER decomposition of a given performance measure on a subset of the test sample. For instance, let us consider two subsets of the test sample, denoted A and B. The performance of the model can be evaluated and decomposed with XPER independently in each subset.[9] The illustrative results displayed in the table below indicate that the model tends to under-perform in A and to over-perform in B. In this case, XPER values can be used to explain the sources of such over(under)-performance of the model. For instance, if the XPER decomposition of the AUC over the subsets A and B is as follows, the under-performance (respectively over-performance) observed in A (in B) is only due to feature x_1.

	AUC		ϕ_0		ϕ_1		ϕ_2		ϕ_3
Subset A	0.65	=	0.50	+	0.01	+	0.10	+	0.04
Subset B	0.85	=	0.50	+	0.21	+	0.10	+	0.04

Second, XPER can help to understand the origin of *overfitting*. Overfitting occurs when a model performs significantly better on the training sample than it does on the test sample. XPER can indeed identify some features which contribute more to the performance of the model in the training sample than in the test sample, and thus explains overfitting. In the table below, the drop in AUC between the training sample (0.90) and the test sample (0.78) illustrates a typical overfitting issue. In this case, the overfitting problem is entirely due to x_1.

	AUC		ϕ_0		ϕ_1		ϕ_2		ϕ_3
Training	0.90	=	0.5	+	0.20	+	0.15	+	0.05
Test	0.78	=	0.5	+	0.08	+	0.15	+	0.05
Training - Test	0.12	=	0	+	0.12	+	0	+	0

[9] At the limit, if the subset only contains one individual, XPER can be used to distribute among the features the accuracy of the prediction made by the model for this individual. We call this metric the individual performance.

Third, XPER can be applied to any statistical performance metrics (e.g., R^2, Brier Score, Gini index, etc.), but also to any *economic performance metrics* (e.g., return-on-investment, profit-and-loss or P&L, excess return of a financial portfolio, customer lifetime value, etc.). As an example, we can decompose the P&L of a credit scoring model defined as:

$$P\&L = \sum_{i=1}^{n} (1 - \hat{y}_i)(1 - y_i) \times profit + (1 - \hat{y}_i) y_i \times loss$$

where $profit$ is the money made on any reimbursed loan ($y_i = 0$) and $loss$ is the money lost on any defaulted loan ($y_i = 1$). The P&L can be broken down as follows:

P&L		ϕ_0		ϕ_1		ϕ_2		ϕ_3
\$10,000	=	\$2,000	+	\$1,000	+	\$5,000	+	\$2,000

The remaining question is how to compute XPER values. The XPER value ϕ_j corresponds to the Shapley value (Shapley (1953)) of the j-th feature associated to the decomposition of the performance metric. A Shapley value measures the impact of a player on a payoff by assessing the changes in the payoff, when this player is included or not in the game. These changes depend on the combination (coalition) of players already considered in the game. By analogy, we decompose a performance metric (payoff) among the features (players) of the model (game).

Contrary to the Shapley value associated to the predicted outcome usually considered in explainable artificial intelligence (e.g., SHAP methodology of Lundberg and Lee (2017)), here the payoff depends both on the features and the target variable. Thus, XPER computation requires to take into account the dependence between the target variable and the features included or excluded from the coalitions. As a consequence, explaining the predictive performance of a model is not equivalent to explaining its outcome, even if both explanations are complementary. Furthermore, as some features are excluded from the coalitions, the computation of XPER values requires either to re-estimate the model, to plug an ad hoc value for the excluded features, or to marginalise over their distribution. We will discuss all these points in detail in the following sections.

3 Framework and performance metrics

We now introduce a general framework allowing us to formally present the XPER methodology. To do so, we consider a classification or regression problem involving a target variable denoted y taking values in $\mathcal{Y}$. The latter is defined as $\mathcal{Y} = \{0, 1\}$ in case of a classification problem, or as $\mathcal{Y} \subset \mathbb{R}$ in case of a regression problem. The q-vector $\mathbf{x} \in \mathcal{X}$ refers to input (explanatory) features with $\mathcal{X} \subset \mathbb{R}^q$. We denote by $f : \mathbf{x} \to \hat{y}$ an econometric model or a machine learning algorithm, where $\hat{y} \in \mathcal{Y}$ is either a classification output, or regression output, such as $\hat{y} = f(\mathbf{x})$. In a classification problem, we assume that the classifier also produces conditional probabilities $\hat{P}(\mathbf{x}) = \hat{\mathbb{P}}(y = 1|\mathbf{x})$. As we impose no constraint on the model $f(.)$, it may be parametric or not, linear or not, a weak learner or an ensemble method, etc. For simplicity, for parametric models we exclude the parameters from the notation, i.e., $f(\mathbf{x}) \equiv f(\mathbf{x}; \theta)$. The model is estimated (parametric model) or trained (machine learning algorithm) once for all on a training sample $S_T = \{\mathbf{x}_i, y_i\}_{i=1}^T$. The sample size T is considered as fixed and we impose no constraint on it. The corresponding trained model can be written either as $\hat{f}_T(.)$ or $\hat{f}(.)$. The predictive performance of the model is evaluated on a test sample $S_n = \{\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i)\}_{i=1}^n$ for n individuals.

We consider a performance metric (PM), defined as an assessment measure of the predictive model performance. This definition encompasses a large variety of metrics. In case of regression models, MSE, MAE, or R^2 are standard PMs, whereas the AUC, Brier score, and Gini index are standard examples of PMs for classification models. More generally, any information criteria (AIC, BIC), loss function (Qlike, Log-Loss), or economic performance indicator (profit function) can be considered as PM.

Definition 1. *Formally, a sample performance metric* $PM_n \in \Theta \subseteq \mathbb{R}$ *associated to a model* $\hat{f}(.)$ *and a test sample* S_n *is defined as:*

$$PM_n = \tilde{G}_n(y_1, ..., y_n; \hat{f}(\mathbf{x_1}), ..., \hat{f}(\mathbf{x_n})) = G_n(\mathbf{y}; \mathbf{X}), \tag{1}$$

where $\mathbf{y} = (y_1, ..., y_n)'$ *and* $\mathbf{X} = (\mathbf{x}_1, .., \mathbf{x}_n)'$.

For instance, $\Theta = [0, 1]$ for AUC, R^2, Brier Score, and $\Theta = \mathbb{R}^+$ for MSE, MAE, etc. We introduce the following assumptions on the PM.

Assumption 1. *The sample performance metric increases over Θ with predictive performance.*

Assumption 1 simplifies the interpretation of the performance metrics. For instance, both the R^2 and the AUC satisfy this assumption, as they increase with the predictive performance of the model. Differently, when dealing with performance metrics that are negatively correlated with performance, e.g. MSE, we consider the opposite of the metric.

Assumption 2. *The sample performance metric satisfies an additive property such that:*

$$G_n(\mathbf{y}; \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} G(y_i; \mathbf{x}_i; \hat{\delta}_n), \tag{2}$$

where $G(y_i; \mathbf{x}_i; \hat{\delta}_n)$ denotes an individual contribution to the performance metric and $\hat{\delta}_n$ is a nuisance parameter which depends on the test sample S_n.

For simplicity, we only consider models for which the outcome y_i for instance i only depends on features $\mathbf{x}_i = (x_{i,1}, ..., x_{i,n})'$. For regression or classification models with cross-sectional interactions (e.g., spatial econometrics model) or time-series dependence, notations have to be adjusted such that $\hat{y}_i = \widehat{f}(\mathbf{w}_i)$ where $\mathbf{x}_i \subseteq \mathbf{w}_i$, $\exists j \neq i : \mathbf{x}_j \subseteq \mathbf{w}_i$ and/or $y_j \subseteq \mathbf{w}_i$. Then, the additive property becomes $G_n(\mathbf{y}; \mathbf{X}) = n^{-1} \sum_{i=1}^{n} G\left(y_i; \mathbf{w}_i; \hat{\delta}_n\right)$.

Assumption 3. *The sample performance metric $G_n(\mathbf{y}; \mathbf{X})$ converges to the population performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0))$, where $\mathbb{E}_{y,\mathbf{x}}(.)$ refers to the expected value with respect to the joint distribution of y and $\mathbf{x}$, and $\delta_0 = plim\ \hat{\delta}_n$. In addition, $\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0))$ exists and is finite.*

These assumptions are consistent with a wide range of performance measures. In Appendix A, we provide the expression of the function $G(y; \mathbf{x}; \delta_0)$ for standard performance metrics associated to regression or classification models. For instance, consider a linear regression model $\hat{y} = \hat{f}(\mathbf{x})$ and

a R^2 as the performance metric, we have:

$$R^2 = G_n(\mathbf{y}; \mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n} G(y_i; \mathbf{x}_i; \hat{\delta}_n) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2},$$

with $G(y_i; \mathbf{x}_i; \hat{\delta}_n) = 1 - \hat{\delta}_n^{-1}(y_i - \hat{f}(\mathbf{x}_i))^2$ and $\hat{\delta}_n = n^{-1}\sum_{j=1}^{n}(y_j - \bar{y})^2$. The corresponding population R^2 is defined as:

$$\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0)) = 1 - \frac{1}{\sigma_y^2}\mathbb{E}_{y,\mathbf{x}}\left((y - \hat{f}(\mathbf{x}))^2\right),$$

with $G(y; \mathbf{x}; \delta_0) = 1 - \delta_0^{-1}(y - \hat{f}(\mathbf{x}))^2$ and $\delta_0 = \sigma_y^2$ the variance of the target variable.

4 XPER values

4.1 Definition

Our objective is to identify the contribution of the model's features to its predictive performance, as measured by a PM on a test set. We measure this contribution through Shapley values (Shapley, 1953). The Shapley values is a solution concept used in game theory that involves fairly distributing a payoff $Val(x_1, ..., x_q)$ among several players $x_1, ..., x_q$ working in a coalition. The Shapley value ϕ_j measures the impact of a player x_j on the payoff by assessing the changes in $Val(x_1, ..., x_q)$ when this player is included or not in the coalition. This amounts to assess the marginal effects of the player on the payoff. Each marginal effect depends on the combination of players already considered in the game. We refer to a combination of players as a coalition, S. We denote $\mathbf{x}^S$ the vector of players included in coalition S and $\mathbf{x}^{\overline{S}}$ the vector of players excluded from coalition S, such as $\{\mathbf{x}\} = \{\mathbf{x}^S\} \cup \{\mathbf{x}^{\overline{S}}\} \cup \{x_j\}$ and $\mathbf{x} = (x_1, ..., x_q)$. The Shapley value is defined as a weighted average of the marginal contributions associated to each coalition S.

Definition 2 (Shapley (1953))**.** *The Shapley value contribution of player x_j to the payoff is:*

$$\phi_j = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})} w_S \left[Val(\mathbf{x}^S \cup \{x_j\}) - Val(\mathbf{x}^S)\right], \tag{3}$$

$$w_S = \frac{|S|!\,(q - |S| - 1)!}{q!}, \tag{4}$$

with $Val(.)$ the payoff, S a coalition or a subset of players, excluding the player of interest x_j, $|S|$ the number of players in the coalition, and $\mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})$ the partition of the set $\{\mathbf{x}\} \setminus \{x_j\}$.

By analogy, we decompose the performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$ (the "payoff") among the features $x_1, ..., x_q$ (the "players") of the model $\hat{f}(\mathbf{x})$. The main difference with the previous notations is that the payoff $Val(\mathbf{x}; y) = \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$ depends not only on the features $x_1, ..., x_q$, but also on the target variable y. Thus, we need to consider the dependence between the target variable and the features $\mathbf{x}^S$ included in coalition S. Beyond this first difference, it is necessary to define what is meant by including or excluding a feature from the coalition within a model. Two solutions can be considered: *re-estimating* the model or *marginalising* over the features excluded from the coalitions, and we are going to consider both in turn.[10]

An intuitive way to measure the impact of a feature on a performance metric would be to assess the changes in the performance by re-estimating the model with or without this particular feature while considering all the possible coalitions of other features. For instance, for a model with three features x_1, x_2, x_3, the computation of the Shapley value $\tilde{\phi}_1$ associated to x_1 requires to estimate 4 sub-models, namely without any feature, with x_2 only, with x_3 only, and with x_2, x_3 and then to re-estimate the same sub-models while including x_1 as an additional feature. Then, one considers the differences in performance metrics, say R^2, associated to these 4 sub-models (coalitions) estimated with or without x_1, namely $R^2(y, \hat{f}_1(x_1)) - R^2(y, \hat{f}_2(\emptyset))$, $R^2(y, \hat{f}_3(x_1, x_2)) - R^2(y, \hat{f}_4(x_2))$, $R^2(y, \hat{f}_5(x_1, x_3)) - R^2(y, \hat{f}_6(x_3))$, and $R^2(y, \hat{f}_7(x_1, x_2, x_3)) - R^2(y, \hat{f}_8(x_2, x_3))$ where all the sub-models $\hat{f}_j$ are estimated on the train test. Finally, the Shapley value $\tilde{\phi}_1$ is defined as a weighted average of the *expected value* of these differences, e.g., $\mathbb{E}_{y,x_1,x_2}(R^2(y, \hat{f}_3(x_1, x_2))) - \mathbb{E}_{y,x_2}(R^2(y, \hat{f}_4(x_2)))$ where $\mathbb{E}_{y,x_j}$ denotes the expectation with respect to the joint distribution of (y, x_j). More generally, for a coalition S with features $\mathbf{x}^S$ the corresponding sub-model $\hat{f}_S(\mathbf{x}^S)$

[10] A third approach consists in replacing the features excluded from the coalitions by ad-hoc values, typically 0 or the mean of the feature. However, the predictive performance decomposition is then dependent of these values.

is associated to a performance metric $G_S(y; \mathbf{x}^S; \delta_0)$. Similarly, when the feature of interest x_j is included in the sub-model, the metric becomes $G_S(y; x_j, \mathbf{x}^S; \delta_0)$. Thus, the re-estimation based Shapley value $\tilde{\phi}_j$ is obtained by summing the weighted marginal contributions associated to all the sub-models $\hat{f}_S(\mathbf{x}^S)$ and averaging each contribution with respect to the joint distribution of $\{y, \mathbf{x}^S\}$ or $\{y, \mathbf{x}^S, x_j\}$:

$$\tilde{\phi}_j = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})} w_S \left[\mathbb{E}_{y, \mathbf{x}^S, x_j}(G_S(y; x_j; \mathbf{x}^S; \delta_0)) - \mathbb{E}_{y, \mathbf{x}^S}(G_S(y; \mathbf{x}^S; \delta_0)) \right]. \tag{5}$$

This approach was adopted by Lipovetsky and Conklin (2001), Israeli (2007), and Huettner and Sunder (2012) to decompose the R^2 of a linear model. However, re-estimating a sub-model $\hat{f}_S(\mathbf{x}^S)$ with only a subset of the initial features inherently leads to model specification error. For instance, in a linear regression model, excluding a relevant variable may induce an omitted variable bias. More generally, as noted by Casalicchio et al. (2019), it can lead to different results of the learning algorithm since different relationships can be learned due to the absence of features. Thus, the model specification error necessarily distorts the Shapley value $\tilde{\phi}_j$ and thus may lead to an unreliable decomposition of the performance metric.

In this paper, we propose an alternative solution that consists in marginalising the performance metric over the features excluded from the coalitions. [11] The first advantage of this approach, is that there is no need to re-estimate any sub-model with a subset of features. This amounts to consider the expected value of the performance metric with respect to the features $\mathbf{x}^{\overline{S}}$ excluded from the coalitions, while leaving the model $\hat{f}(\mathbf{x})$ unchanged. For instance, when considering a coalition with x_1 only in a three-feature model $\hat{f}(x_1, x_2, x_3)$, we compute the expectation of the performance metric, say R^2, with respect to the excluded variables x_2 and x_3, i.e., $\mathbb{E}_{x_2,x_3}(R^2(y, \hat{f}(x_1, x_2, x_3)))$ where $\hat{f}(.)$ is the already fitted model. Then, we consider the expectation of the performance metric with respect to the joint distribution of the variables included in the coalition and the target, i.e., y and x_1 in our example. Thus, we get an *expected value* $\mathbb{E}_{y,x_1}\mathbb{E}_{x_2,x_3}(R^2(y, \hat{f}(x_1, x_2, x_3)))$, where the

[11] In general, the marginalisation refers to a method which consists in summing over the possible values of one variable to determine the marginal contribution of another.

first expectation refers to the averaging effect, whereas the second one refers to the marginalisation effect. The marginalisation-based Shapley value is then computed by averaging all the differences in the expected performance metrics obtained with or without the feature of interest, for all the coalitions of other features.

Formally, we define XPER as the marginalisation-based Shapley value associated to a given model $\hat{f}(\mathbf{x})$ and a corresponding performance metric $G(y; \mathbf{x}; \delta_0)$.

Definition 3 (XPER)**.** *The XPER value associated to the feature x_j is defined as:*

$$\phi_j = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})} w_S \left[\underbrace{\mathbb{E}_{y, x_j, \mathbf{x}^S}}_{averaging} \underbrace{\mathbb{E}_{\mathbf{x}^{\overline{S}}}}_{marginalisation} \left(G\left(y; \mathbf{x}, x_j; \delta_0\right)\right) - \underbrace{\mathbb{E}_{y, \mathbf{x}^S}}_{averaging} \underbrace{\mathbb{E}_{x_j \mathbf{x}^{\overline{S}}}}_{marginalisation} \left(G\left(y; \mathbf{x}; \delta_0\right)\right) \right],$$

with S a coalition or a subset of features, excluding the feature of interest x_j, and $\mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})$ the partition of the set $\{\mathbf{x}\} \setminus \{x_j\}$.

The XPER value ϕ_j measures the weighted average marginal contribution of the feature x_j to the performance metric over all feature coalitions. The marginal contribution is defined as the difference between the expected values of the performance metric obtained while including or not the feature of interest x_j within a coalition. In definition (3), the term $\mathbb{E}_{\mathbf{x}^{\overline{S}}}$ refers to the marginalisation with respect to the features which are excluded from the coalition S. The second expectation $\mathbb{E}_{y, x_j, \mathbf{x}^S}\left(G\left(y; \mathbf{x}; \delta_0\right)\right)$ refers to an averaging effect, i.e., an expectation with respect to the joint distribution of the features $\mathbf{x}^S$ included in the coalition with x_j, and the target variable y.[12] XPER values ϕ_j are defined as theoretical and unobserved quantities. The corresponding estimates will be presented in Section 5.

[12]Note that one could argue about that the use of an unconditional expectation to marginalise features can lead to including unrealistic instances when features are correlated (Molnar, 2020). This concern is not specific to XPER and it could be solved by using conditional expectations instead of unconditional ones (Aas et al., 2021). However, considering conditional expectation leads to measures that are no longer Shapley values as they do not satisfy at least the symmetry property (Sundararajan and Najmi, 2020; Janzing et al., 2020)

4.2 Properties and interpretation of XPER values

The XPER values satisfy different properties which are particularly relevant for statistical performance analysis. First, the XPER values exhibit an efficiency property:

Proposition 4. *(Efficiency) The sum of the XPER values ϕ_j, $\forall j = 1, ..., q$, is equal to the difference between the population performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0))$ and a benchmark such as:*

$$\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0)) = \phi_0 + \sum_{j=1}^{q} \phi_j, \tag{6}$$

where the benchmark $\phi_0 = \mathbb{E}_{\mathbf{x}}\mathbb{E}_y\left(G(y; \mathbf{x}; \delta_0)\right)$ corresponds to the performance metric associated to a population where the target variable is independent from all features considered in the model.

One of the main advantages of the XPER decomposition is that the benchmark value ϕ_0 has a nice interpretation: it corresponds to the performance metric that we would obtain on a hypothetical sample in which the target variable y is independent from all model features $\mathbf{x}$, i.e., in a case where the model $\hat{f}(\mathbf{x})$ is fully misspecified. For instance, if we consider the AUC as performance metric, then the benchmark ϕ_0 corresponds to the AUC associated to a random predictor and is equal to 0.5. For the sensitivity (true positive rate), the benchmark corresponds to the probability $\Pr(\hat{y} = 1)$, for the specificity (true negative rate) the benchmark is $\Pr(\hat{y} = 0)$, etc. Formally, the benchmark value ϕ_0 is the expected value of the population metric $\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0)$ obtained under the assumption that model features are independent of the target, and is then equal to $\mathbb{E}_y\mathbb{E}_{\mathbf{x}}(G(y; \mathbf{x}; \delta_0)$.

Thus, thanks to the marginalisation-based definition of XPER, we can decompose any population performance metric into two parts: (i) a base value ϕ_0 obtained in a hypothetical case where y and $\mathbf{x}$ would be independent, and (ii) a component determined by the feature contributions, which depend on their dependence with the target, i.e., their relevance. By definition, this second component is equal to the sum of the XPER values ϕ_j associated to the model features.

$$\underbrace{\mathbb{E}_{y,\mathbf{x}}(G(y; \mathbf{x}; \delta_0)}_{\text{population performance metric}} = \underbrace{\mathbb{E}_y\mathbb{E}_{\mathbf{x}}(G(y; \mathbf{x}; \delta_0)}_{\text{expected value under independence}} + \underbrace{\sum_{j=1}^{q} \phi_j}_{\text{feature contributions}},$$

The XPER values also satisfy the other main properties (dummy, symmetry, monotonicity) of the Shapley values:

Proposition 5. *(Dummy) If the model feature x_j does not have any impact on the performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$, then its XPER value ϕ_j is null, i.e., $\phi_j = 0$.*

Proposition 6. *(Monotonicity) If a feature x_j contributes more to the performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$ than a feature x_s then $\phi_j > \phi_s$.*

Proposition 7. *(Symmetry) If two features x_j and x_s contribute equally to the performance metric $\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$ across all coalitions then $\phi_j = \phi_s$.*

To illustrate these properties, we consider a model with three features and pick x_1 as the feature of interest. In Table 1, we report all the coalitions among the set $\{x_2, x_3\}$ (column 1), the associated weights computed according to equation (4) (column 2), and the marginal contributions (column 3) used to compute the XPER value ϕ_1.

Table 1: Components of ϕ_1 in a three-feature model

S	w_S	$\mathbb{E}_{\mathbf{x}^{\overline{S}}}\mathbb{E}_{y,x_1,\mathbf{x}^S}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,\mathbf{x}^{\overline{S}}}\mathbb{E}_{y,\mathbf{x}^S}\left(G\left(y;\mathbf{x};\delta_0\right)\right)$
$\{\emptyset\}$	$1/3$	$\mathbb{E}_{x_2,x_3}\mathbb{E}_{y,x_1}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_2,x_3}\mathbb{E}_{y}\left(G\left(y;\mathbf{x};\delta_0\right)\right)$
$\{x_2\}$	$1/6$	$\mathbb{E}_{x_3}\mathbb{E}_{y,x_1,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_3}\mathbb{E}_{y,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right)$
$\{x_3\}$	$1/6$	$\mathbb{E}_{x_2}\mathbb{E}_{y,x_1,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_2}\mathbb{E}_{y,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)$
$\{x_2,x_3\}$	$1/3$	$\mathbb{E}_{y,x_1,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1}\mathbb{E}_{y,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)$

Note: This table provides details about the XPER value computation, i.e., the coalitions (column 1), the associated weights according to equation (4) (column 2), and the marginal contributions (column 3).

The XPER value ϕ_1 associated to x_1 is computed by multiplying the weights (column 2) to the marginal contributions (column 3) and summing over all coalitions, such as:

$$\begin{aligned}
\phi_1 &= \frac{1}{3}\left(\mathbb{E}_{x_2,x_3}\mathbb{E}_{y,x_1}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \phi_0\right) \\
&+ \frac{1}{6}\left(\mathbb{E}_{x_3}\mathbb{E}_{y,x_1,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_3}\mathbb{E}_{y,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{6}\left(\mathbb{E}_{x_2}\mathbb{E}_{y,x_1,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_2}\mathbb{E}_{y,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{3}\left(\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) - \mathbb{E}_{x_1}\mathbb{E}_{y,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right).
\end{aligned}$$

Using the same approach, we can compute the contributions ϕ_2 and ϕ_3. By summing over all features, all terms cancel each other out except $\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))$ and ϕ_0 (see Appendix B). Thus, we verify the efficiency property, i.e., $\sum_{j=1}^{3}\phi_j = \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) - \phi_0$.

As a rule, there is no analytical expression of the XPER values ϕ_j. Indeed, the function $G(y;\mathbf{x};\delta_0)$ is in general non-linear in y and $\mathbf{x}$, as the model $\hat{f}(.)$ may be highly non-linear in $\mathbf{x}$ (e.g., machine learning model) and/or the performance metric may be non-linear in y and $\mathbf{x}$. Then, the expectation cannot be computed analytically and the XPER values are obtained numerically. The only exception corresponds to the case of quadratic loss functions, e.g., MSE, R^2, etc., associated to the linear regression model (see Appendix C for other examples). For instance, consider a linear regression model $\hat{f}(\mathbf{x}_i) = \sum_{j=1}^{q}\hat{\beta}_j x_{i,j}$ estimated on a training set, and the R^2 as sample performance metric. We assume that the data generating process (DGP) associated to the test sample $S_n = \{\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i)\}_{i=1}^{n}$ satisfies $\mathbb{E}(\mathbf{x}) = 0_q$ and $\mathbb{V}(\mathbf{x}) = diag(\sigma^2_{x_j}), \forall j = 1, ..., q$. We denote by σ_y^2 the variance of the target variable and by σ_{y,x_j} its covariance with feature x_j. Then, the XPER contribution ϕ_j of feature x_j to the R^2 is:

$$\phi_j = \frac{2\hat{\beta}_j \sigma_{y,x_j}}{\sigma_y^2}, \qquad \forall j = 1, ..., q. \tag{7}$$

Formally, ϕ_j depends on the estimated parameter $\hat{\beta}_j$ (training set) and the covariance (test set) between x_j and the target variable y, i.e., σ_{y,x_j}. If the DGP of the training and test samples are identical, XPER values ϕ_j are positive or null.[13] The dummy property $\phi_j = 0$ is either satisfied if the feature has no impact on the model ($\hat{\beta}_j = 0$) or if the feature is uncorrelated with the target variable on the test sample ($\sigma_{y,x_j} = 0$). Similarly, a variable x_j has a larger contribution to the R^2 than a feature x_s as soon as $\hat{\beta}_j \sigma_{y,x_j} > \hat{\beta}_s \sigma_{y,x_s}$, meaning that x_j is more related to the target variable than x_s both in-sample (through $\hat{\beta}_j$) and out-of-sample (through σ_{y,x_s}).

[13] If the DGP of the training and test samples are identical, model parameters $\hat{\beta}_j$ and covariance σ_{y,x_j} have the same sign, which means $\phi_j > 0$. Otherwise, XPER values may be negative.

4.3 Individual XPER values

The XPER framework allows to conduct a global analysis of the model predictive performance through feature contributions ϕ_j, as well as a local analysis at individual level. Under assumption 2, we define individual XPER values as follows:

Definition 8. *(Individual XPER) The individual XPER value $\phi_{i,j}$ associated to individual i and model feature j satisfies:*

$$\phi_j = \mathbb{E}_{y,\mathbf{x}}(\phi_{i,j}(y_i; \mathbf{x}_i)), \tag{8}$$

where the random variable $\phi_{i,j}(y_i; \mathbf{x}_i)$ corresponds to individual i and feature j contribution to the performance metric defined as:

$$\phi_{i,j}(y_i; \mathbf{x}_i) = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})} w_S \left[\mathbb{E}_{\mathbf{x}^{\overline{S}}} \left(G\left(y_i; x_{i,j}, \mathbf{x}_i^S; \delta_0\right)\right) - \mathbb{E}_{x_j, \mathbf{x}^{\overline{S}}} \left(G\left(y_i; \mathbf{x}_i^S; \delta_0\right)\right) \right].$$

The individual XPER value can be interpreted as the contribution of individual i and model feature j to the performance metric. For a given realisation $(y_i, \mathbf{x}_i)$, the corresponding individual contribution to the performance metric can be broken down into:

$$G(y_i; \mathbf{x}_i; \delta_0) = \phi_{i,0} + \sum_{j=1}^{q} \phi_{i,j}, \tag{9}$$

where $\phi_{i,j}$ is the realisation of $\phi_{i,j}(y_i; \mathbf{x}_i)$ and $\phi_{i,0}$ is the realisation of $\phi_{i,0}(y_i) = \mathbb{E}_{\mathbf{x}}(G(y_i; \mathbf{x}; \delta_0))$. The individual benchmark $\phi_{i,0}$ corresponds to the individual contribution to the performance metric obtained for a population where the target variable y_i is independent from the features $\mathbf{x}_i$. Therefore, the difference between the individual contribution to the performance metric $G(y_i; \mathbf{x}_i; \delta_0)$ and the individual benchmark $\phi_{i,0}$ is explained by individual XPER values $\phi_{i,j}$.

To illustrate this point, consider the same linear regression framework as before and a R^2 performance metric. Then, the individual XPER value $\phi_{i,j}$ can be expressed as:

$$\phi_{i,j} = \sigma_y^{-2} \left[\hat{\beta}_j \left(x_{i,j} - \mathbb{E}(x_j)\right) \left(2y_i - 2\hat{\beta}_0 - \sum_{k \neq j}^{q} \hat{\beta}_k (x_{i,k} + \mathbb{E}(x_k))\right) - \hat{\beta}_j^2 (x_{i,j}^2 - \mathbb{E}(x_j^2)) \right]. \tag{10}$$

The value $\phi_{i,j}$ measures the contribution of feature x_j to $G(y_i; \mathbf{x}_i; \delta_0) = 1 - \sigma_y^{-2}(y_i - \hat{f}(\mathbf{x}_i))^2$. A positive individual XPER value $\phi_{i,j}$ means that incorporating the information included in feature x_j improves model prediction for individual i compared to the benchmark, hence increases R^2. Several comments can be made here. First, the dummy property 5 holds: if $\hat{\beta}_j = 0$, i.e., if the feature has no impact on the model outcome, the feature x_j does not have any impact on the R^2 for all individuals. Second, the closer the realization of feature x_j is to its expected value $\mathbb{E}(x_j)$, the lower is the contribution of this feature to the performance metric, for all individuals. A similar result occurs when x_j^2 is close to its expected value. Indeed, when the characteristics of an individual are close to the mean values over the population, her contribution to the predictive performance of the model is also close to the average contribution of other individuals.

5 Estimation

5.1 Definitions

In this section, we discuss the estimation procedure for the XPER values ϕ_j and $\phi_{i,j}$, based on a test sample $S_n = \{\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i)\}_{i=1}^n$. Below, we assume that the model has a small number of features, typically $q \leq 10$.[14] When the number of features q exceeds few units, an approximation of the XPER values is required and we propose a modified version of the Kernel SHAP method of Lundberg and Lee (2017) (see Appendix D for more details).

Under the additive property (assumption 2), the XPER value ϕ_j can be estimated by a weighted average of individual contributions differences.

Proposition 9. *A consistent estimator of the XPER value associated to x_j is:*

$$\hat{\phi}_j = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})} w_S \left[\frac{1}{n^2} \sum_{u=1}^{n} \sum_{v=1}^{n} G\left(y_v; x_{v,j}, \mathbf{x}_v^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right) - \frac{1}{n^2} \sum_{u=1}^{n} \sum_{v=1}^{n} G\left(y_v; x_{u,j}, \mathbf{x}_v^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right) \right], \quad (11)$$

with S a coalition, i.e., a subset of features, excluding the feature of interest x_j, and $\mathcal{P}(\{\mathbf{x}\} \setminus \{x_j\})$ the partition of the set $\{\mathbf{x}\} \setminus \{x_j\}$.

[14] The standard estimation framework is only feasible for a small number of features q. Indeed, the computation of the XPER values ϕ_j according to definition 3 becomes cumbersome as the number of features increases. Indeed, it requires to consider $q \times 2^{q-1}$ coalitions of features, e.g., 5,120 for $q = 10$ and 10,485,760 for $q = 20$, etc.

Table 2 illustrates the computation of the estimated value $\hat{\phi}_1$ associated to feature x_1 in a model with three features (x_1, x_2, x_3). For each coalition of features (column 1), we report the corresponding weight (column 2) along with the estimated marginal contribution of feature x_1 to the performance metric (column 3). The intuition is as follows: the sum over index u refers to the marginalisation effect, whereas the sum over index v refers to the averaging effect. For a given instance v, we compute its average performance metric by replacing the variables $x^{\overline{S}}$ which are *not included* in the coalition by the corresponding values observed for all the instances of the test sample (marginalisation). Then, we compute the average performance metric for all the instances v (averaging).

Table 2: Computation of the XPER value $\hat{\phi}_1$ in a three-feature model

S	w_S	$\frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{v,j}, \mathbf{x}_v^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right) - \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{u,j}, \mathbf{x}_v^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right)$
$\{\emptyset\}$	$1/3$	$\frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{v,1}, x_{u,2}, x_{u,3}; \hat{\delta}_n\right) - \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{u,1}, x_{u,2}, x_{u,3}; \hat{\delta}_n\right)$
$\{x_2\}$	$1/6$	$\frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{v,1}, x_{v,2}, x_{u,3}; \hat{\delta}_n\right) - \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{u,1}, x_{v,2}, x_{u,3}; \hat{\delta}_n\right)$
$\{x_3\}$	$1/6$	$\frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{v,1}, x_{u,2}, x_{v,3}; \hat{\delta}_n\right) - \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{u,1}, x_{u,2}, x_{v,3}; \hat{\delta}_n\right)$
$\{x_2, x_3\}$	$1/3$	$\frac{1}{n}\sum_{v=1}^{n} G\left(y_v; x_{v,1}, x_{v,2}, x_{v,3}; \hat{\delta}_n\right) - \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} G\left(y_v; x_{u,1}, x_{v,2}, x_{v,3}; \hat{\delta}_n\right)$

Note: This table displays details of empirical XPER value computation, i.e., the coalitions (column 1), the associated weights (column 2) and the estimated marginal contributions (column 3)

Similarly to the estimation of features ϕ_j, we can estimate the *individual* XPER values $\phi_{i,j}$. For any individual i of the test sample of S_n and any feature x_j, a consistent estimator of the individual XPER values $\phi_{i,j}$ is defined as:

$$\hat{\phi}_{i,j} = \sum_{S \subseteq \mathcal{P}(\{\mathbf{x}\}\setminus\{x_j\})} w_S \left[\frac{1}{n}\sum_{u=1}^{n} G\left(y_i; x_{i,j}, \mathbf{x}_i^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right) - \frac{1}{n}\sum_{u=1}^{n} G\left(y_i; x_{u,j}, \mathbf{x}_i^S, \mathbf{x}_u^{\overline{S}}; \hat{\delta}_n\right)\right]. \tag{12}$$

By definition, these individual XPER values satisfy:

$$\hat{\phi}_j = \frac{1}{n}\sum_{i=1}^{n} \hat{\phi}_{i,j}. \tag{13}$$

5.2 Illustrations

In this section, we provide two illustrations of the use of XPER values in the case of Monte Carlo simulations for which we control for (i) the DGP for the train and test sets, and (ii) the model for which we want to explain the predictive performance.

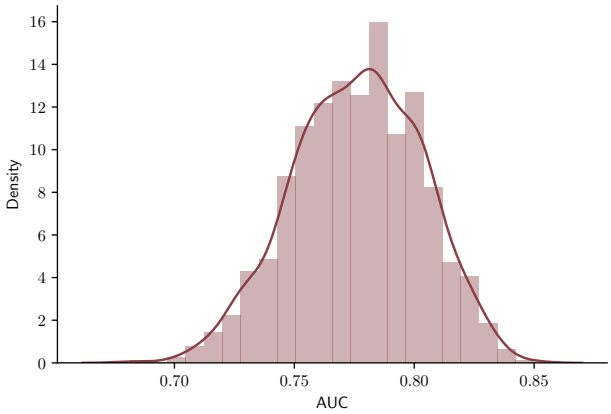
Illustration 1: XPER decomposition of the AUC of a classification model. As a first example, XPER values are used to explain the predictive performance of a probit regression model, measured by the AUC computed on a test set S_n. In order to understand the mechanisms of the XPER decomposition, we consider here a white-box model for which one can clearly explain how it produces predictions and what the influencing variables are. The DGP is given by a latent variable model such that:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

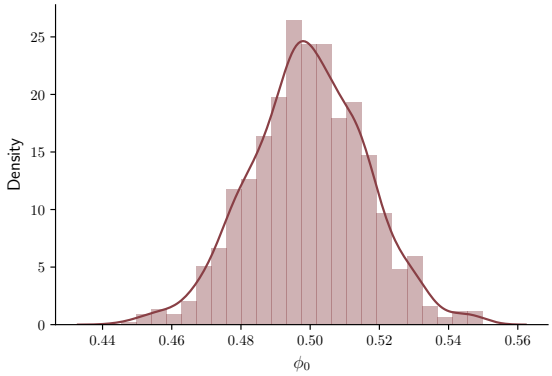
with $y_i^* = \omega_i\beta + \varepsilon_i$, $\omega_i = (1 : \mathbf{x}_i^{'})$ and ε_i an i.i.d. error term with $\varepsilon_i \sim \mathcal{N}(0,1)$. We consider three i.i.d. features such that $\mathbf{x}_i = (x_{i,1}, x_{i,2}, x_{i,3})^{'} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $diag(\Sigma) = (1.2, 1, 1)$. The true vector of parameters is $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^{'} = (0.05, 0.5, 0.5, 0)^{'}$ with β_0 the intercept. To illustrate the dummy property, we assume that the third feature x_3 does not affect the target variable.

We simulate $K = 5,000$ pseudo-samples $\{y_i^s, \mathbf{x}_i^s\}_{i=1}^{T+n}$ of size 1,000, for $s = 1, \ldots, K$. For each pseudo-sample, we use the first $T = 700$ observations to estimate a probit model and the remaining ones as test set S_n to compute the AUC and the corresponding XPER values according to equation (11). For instance, the estimated parameters obtained for the simulation $s = 1$ are equal to $\{\hat{\beta}_0^s, \hat{\beta}_1^s, \hat{\beta}_2^s, \hat{\beta}_3^s\} = \{0.0109, 0.4943, 0.5234, 0.0688\}$ and the AUC is equal to 0.7775. The associated feature contributions are the following:

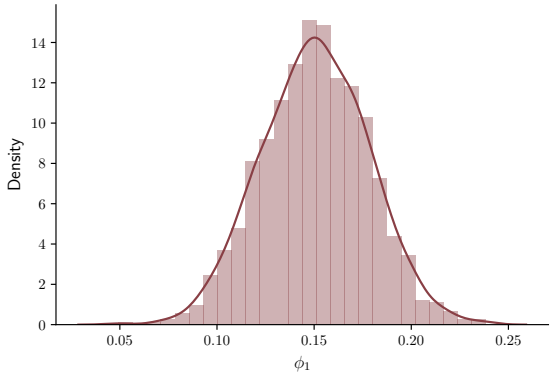
$$\underbrace{0.7775}_{AUC} = \underbrace{0.4984}_{\hat{\phi}_0} + \underbrace{0.1716}_{\hat{\phi}_1} + \underbrace{0.1098}_{\hat{\phi}_2} + \underbrace{(-0.0023)}_{\hat{\phi}_3}.$$



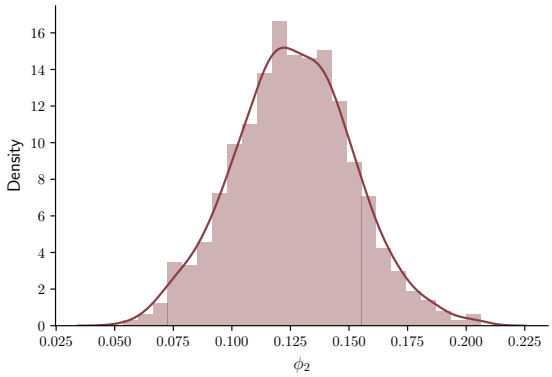
(a) AUC



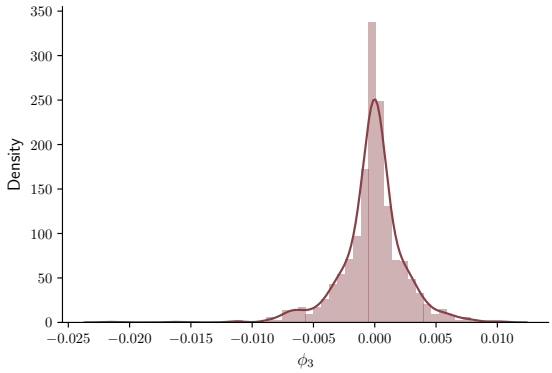
(b) ϕ_0



(c) ϕ_1



(d) ϕ_2



(e) ϕ_3

Figure 1: Empirical distributions of AUC and XPER values

Note: This figure displays the empirical distributions of the AUC and XPER values on the test sample according to the DGP detailed in Illustration 1. XPER values are divided by the difference between the AUC of the model and the benchmark value to be comparable between the training and the test sample. The solid red lines refer to kernel density estimations.

As expected, the estimated benchmark $\hat{\phi}_0$ is close to 0.5. As a reminder, an AUC equal to 0.5 is associated to a random predictor. Hence, the benchmark $\hat{\phi}_0$ corresponds to the AUC of the model $\hat{f}(\mathbf{x}_i) = \mathbf{x}_i'\hat{\beta}$ that we would obtain on a virtual test sample where the target variable is independent from all features. The difference between the estimated AUC and this hypothetical benchmark is explained by the feature contributions. We verify that these contributions are positive or null for all features. The feature with the largest variance (x_1) has also the largest contribution to the predictive ability of the model ($0.1716/(0.7775 - 0.4984) \simeq 62\%$). On the contrary, as the third feature is excluded from the model, its contribution to the AUC is close to zero (dummy property). Figure 1 displays the empirical distributions of AUC, benchmark values $\hat{\phi}_0$, and XPER values associated to features x_1, x_2, and x_3, computed from the K simulations. It confirms the robustness of our analysis, but also illustrates the possibility to make inference on XPER values by Bootstrap or other numerical method when the computing time is reasonable.

A local analysis of feature contributions to the AUC for simulation $s = 1$ is detailed in Table 3. In the second column, we report the estimated individual contributions to the AUC for a subset of instances of the test sample S_n. The AUC of the probit model (0.7775), corresponds to the average of individual contributions (cf. equation (13)). Consider the instance $i = 4$ with a target value equal to 1, its individual contribution (0.5334) is smaller than the AUC because its estimated probability 0.3524 is lower than $(\sum_{i=1}^n y_i)^{-1} \sum_{i=1}^n \hat{P}(\mathbf{x}_i) y_i = 0.6081$, i.e., the average probability for individuals with $y_i = 1$. It reflects the greater uncertainty of the model about the true type of this individual. Conversely, the instance $i = 1$, associated to a larger probability $\hat{P}(\mathbf{x}_3) = 0.6614$, has a larger individual contribution (0.9) to the AUC. This instance tends to increase the predictive performance of the model. More generally, we observe in Figure 2 that for individuals with $y_i = 1$ (respectively $y_i = 0$), the contributions $G(y_i; \mathbf{x}_i; \hat{\delta}_n)$ increase (respectively decrease) with the probabilities estimated by the model.

The third column of Table 3 displays the individual benchmark. For a binary classification model, the benchmark $\hat{\phi}_{i,0} \equiv \hat{\phi}_{i,0}(y_i)$ only takes two values. In our simulations, for individuals

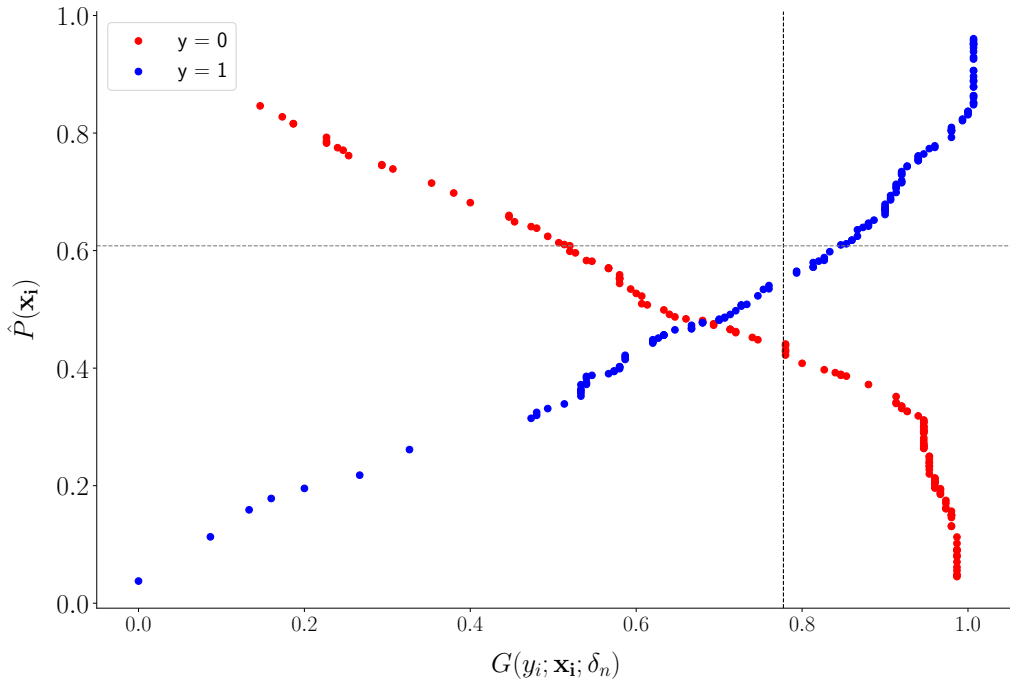


Figure 2: Estimated probabilities as a function of individual contributions to the AUC

Note: This figure displays estimated probabilities (y-axis) as a function of individual contributions to the AUC. Blue (red) dots refer to individuals with target value equal to 1 (0). The horizontal dotted line refers to $(\sum_{i=1}^{n} y_i)^{-1} \sum_{i=1}^{n} \hat{P}(\mathbf{x}_i) y_i = 0.6082$ and the vertical dotted one corresponds to the $AUC = 0.7775$.

$y_i = 1$ (respectively $y_i = 0$), this value is equal to 0.5001 (respectively 0.4967). Remind that the individual benchmark corresponds to the contribution to the AUC obtained from a random predictor for an individual with a target value $y_i = 1$ or $y_i = 0$. Consider the first instance $i = 1$ with $y_i = 1$, $G(y_1; \mathbf{x}_1; \hat{\delta}_n) = 0.9$ and $\hat{\phi}_{1,0} = 0.5001$. As $G(y_1; \mathbf{x}_1; \hat{\delta}_n) > \hat{\phi}_{1,0}$, it means that the features allow the probit model to better predict the event y_1 for this instance than a random predictor. On the contrary, for individual $i = 297$ for which $G(y_{297}; \mathbf{x}_{297}; \hat{\delta}_n) = 0.2533 < \hat{\phi}_{297,0} = 0.4967$, the probit model is less efficient than a random predictor. Indeed, for this instance the estimated probability given by the model $\hat{P}(\mathbf{x}_{297}) = 0.7616$ is high, whereas the event does not occur ($y_{297} = 0$).

Finally, columns 4, 5 and 6 report the XPER values associated to features x_1, x_2, x_3. First, we verify that feature x_3 has close to no impact on the AUC for all instances (dummy property). Second, the local analysis reveals the heterogeneity of contributions to the AUC depending on individual

Table 3: Illustration of AUC XPER values in a three-fold logit model

	$G(y_i;\mathbf{x}_i;\hat{\delta}_n)$	$\hat{\phi}_{i,0}$	$\hat{\phi}_{i,1}$	$\hat{\phi}_{i,2}$	$\hat{\phi}_{i,3}$	y_i	$\hat{\mathbb{P}}(y_i=1\|\mathbf{x}_i)$
i=1	0.9000	0.5001	0.2450	0.1771	-0.0221	1	0.6614
i=2	1.0000	0.5001	0.3975	0.1168	-0.0144	1	0.8369
i=3	1.0067	0.5001	0.2142	0.3203	-0.0279	1	0.8785
i=4	0.5334	0.5001	-0.1245	0.1660	-0.0083	1	0.3524
i=5	0.6134	0.4967	0.1311	0.0211	-0.0356	0	0.5076
...	...	...	...	...	...	...	...
i=296	0.5934	0.4967	-0.0145	0.1344	-0.0233	0	0.5346
i=297	0.2533	0.4967	-0.1232	-0.1237	0.0035	0	0.7616
i=298	0.2267	0.4967	-0.2364	-0.0236	-0.0100	0	0.7828
i=299	0.9867	0.4967	0.3583	0.1196	0.0121	0	0.0894
i=300	0.9600	0.4967	0.1316	0.3069	0.0249	0	0.1982
	0.7775	0.4984	0.1716	0.1098	-0.0023	0.4967	0.4941

Note: This table displays individual contributions to the AUC, individual benchmarks, and XPER values associated to each feature x_j, $j = 1, 2, 3$, in a three-fold logit model. The last row of the table reports average values of the columns.

characteristics. Remind that at the global level, the contribution of feature x_1 is higher than feature x_2. However, at the local level, we observe for instance $i = 3$ that the contribution of feature x_1 is smaller than feature x_2, i.e., $0.21428 < 0.3203$. Third, contrary to global feature contributions, we observe that some individual contributions are negative. For instance, contribution of feature x_2 for individual $i = 297$ is equal to $\hat{\phi}_{297,2} = -0.1232$. It means that, for this instance, this feature tends to disturb the model to predict the true target value $y_i = 0$.

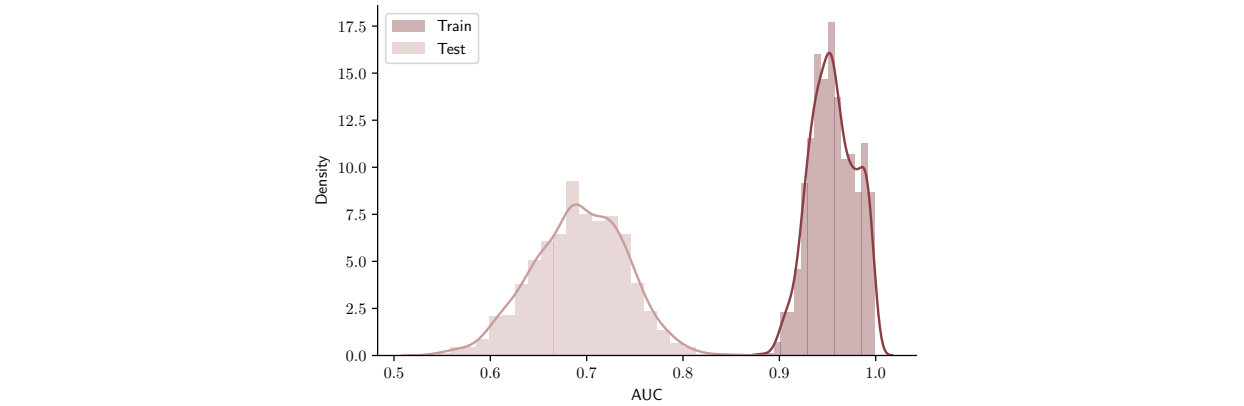
Illustration 2: Explaining Overfitting. As a second example, XPER values are used to detect the origin of overfitting. The overfitting of a model can arise for at least two reasons: (1) an improper control of the bias-variance trade-off through model hyperparameters, or (2) a shift of the feature distributions between the training and test sample. Let us illustrate these two cases with the following Monte Carlo simulations.

Case 1: Consider a DGP given by:

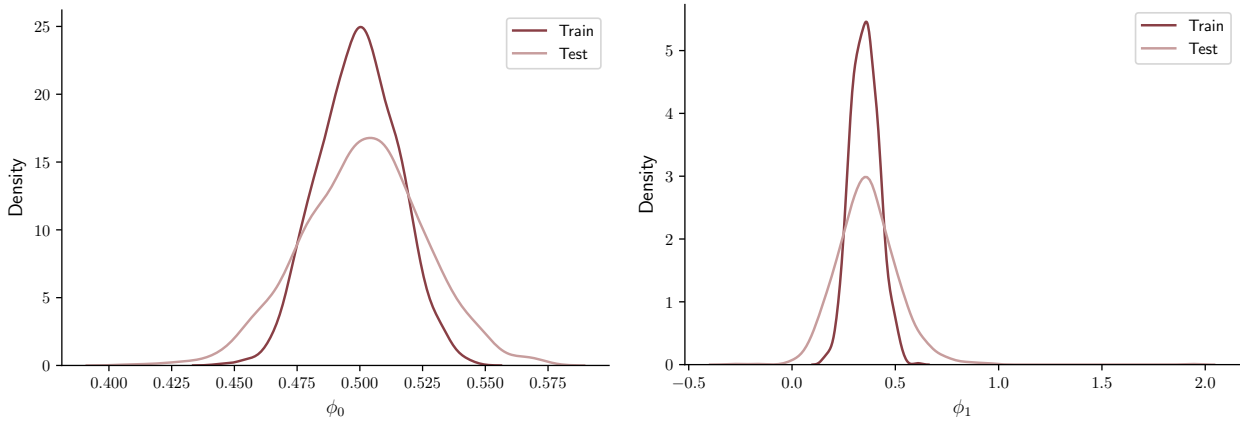
$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

with $y_i^* = \omega_i\beta + \varepsilon_i$ a latent variable, $\omega_i = (1 : \mathbf{x}_i^{'})$, and ε_i an i.i.d. error term with $\varepsilon_i \sim \mathcal{N}(0, 1)$.

We consider three independent features such that $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $diag(\Sigma) = (1.3, 1.2, 1.1)$. The true vector of parameters is $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)' = (0.05, 0.5, 0.5, 0.5)'$ with β_0 the intercept. We generate $K = 5,000$ pseudo-samples $\{y_i^s, \mathbf{x}_i^s\}_{i=1}^{T+n}$ of size $1,000$ using this DGP. Then, we estimate a decision tree using 5-fold cross validation on the first $T = 700$ observations of each pseudo-sample and we use the remaining $n = 300$ observations as a test sample. In order to generate overfitting, we impose a minimum tree-depth of 6 nodes for only three features in the model. For each trained model, we implement XPER to decompose the effect of the features on the AUC of the training and the test samples. We display in Figure 3a the empirical distributions of the AUC. As expected, the trained tree models are overfitting the data, illustrated by the relatively low AUC values obtained on the test samples compared to the training samples. The empirical distributions of the XPER values reported in other panels of Figure 3 show that this drop in the performance does not come from a particular feature. Indeed, the XPER contributions to the AUC are relatively close between the training and the test sample for all features. Thus, when overfitting is due to an improper control of the bias-variance trade-off, we observe a large decrease of the performance metric along with a stability of XPER values between the training and the test sample. Therefore, XPER can be used as a reverse engineering tool to detect wrong settings of hyperparameters.

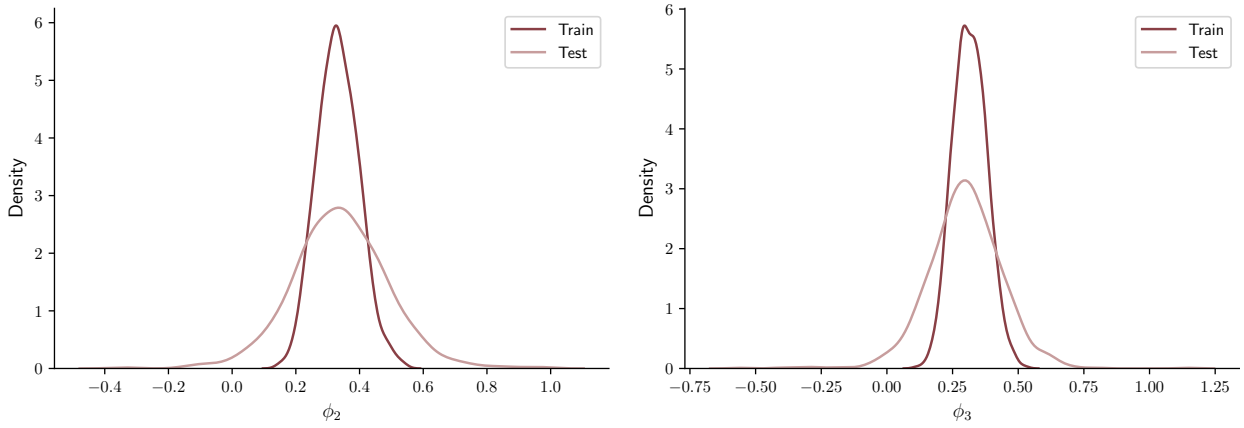


(a) AUC



(b) ϕ_0

(c) ϕ_1

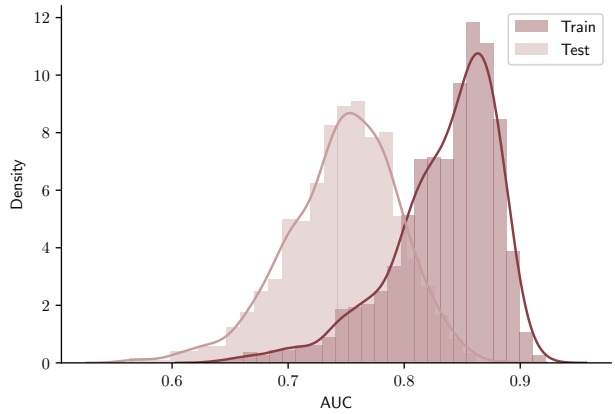


(d) ϕ_2

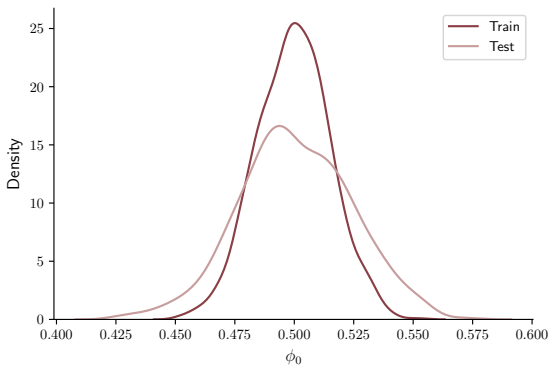
(e) ϕ_3

Figure 3: Empirical distributions of AUC and XPER values in case of overfitting due to improper control of the bias-variance trade-off

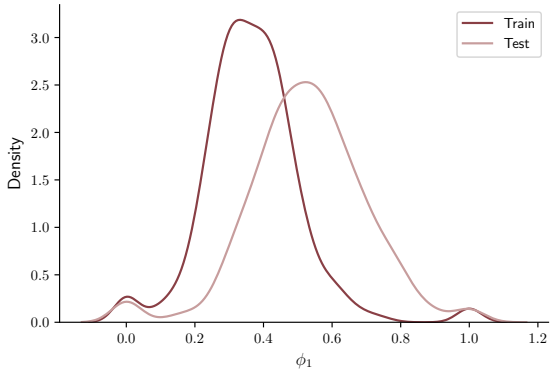
Note: This figure displays the empirical distributions of the AUC and XPER values on the training (dark color) and test (light color) sample according to the framework detailed in Illustration 2, case 1. XPER values are divided by the difference between the AUC of the model and the benchmark value to be comparable between the training and the test sample. The solid lines refer to kernel density estimations.



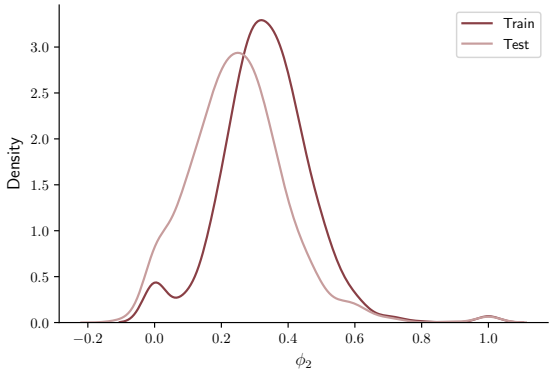
(a) AUC



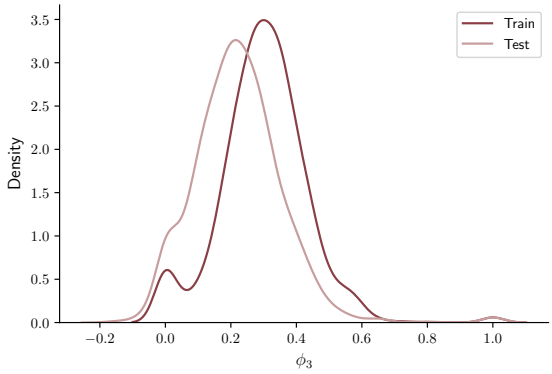
(b) ϕ_0



(c) ϕ_1



(d) ϕ_2



(e) ϕ_3

Figure 4: Empirical distributions of AUC and XPER values in case of overfitting due to a shift of the distribution of the features between the training and the test sample

Note: This figure displays the empirical distributions of the AUC and XPER values on the training (dark color) and test (light color) sample according to the framework detailed in Illustration 2, case 2. XPER values are divided by the difference between the AUC of the model and the benchmark value to be comparable between the training and the test sample. The solid lines refer to kernel density estimations.

Case 2: As previously mentioned, overfitting can also arise from a shift of the feature distributions between the training and the test sample. In this second experiment, we consider two distinct DGPs for the training and the test sample. For the former, we keep the same DGP as in case 1. For the test sample, we assume an increase in the variance of the first feature while keeping other parameters unchanged, such that $diag(\tilde{\mathbf{\Sigma}}) = (3, 1.2, 1.1)$. As in case 1, we generate $K = 5,000$ pseudo-samples $\{y_i^s, \mathbf{x}_i^s\}_{i=1}^{T+n}$ of size 1,000 ($T = 700$ and $n = 300$). For each pseudo-sample, we estimate a decision tree with a depth between 1 to 5 using 5-fold cross validation. Setting a relatively low tree depth avoids overfitting due to an improper control of the bias-variance trade-off. In Figure 4a, we observe a decrease in AUC between the training and test samples. Contrary to the previous case, this decrease is due to the shift of the distribution of x_1 which has also an impact on XPER values. More precisely, in Figure 4 we observe that the contribution of feature x_1 to the AUC increases from the training to the test sample whereas the contribution of the other features decreases. Thus, observing a decrease of the performance metric along with a change in the XPER values from the training sample to the test sample, may indicate a change in the data structure which is not captured by the model and not related to hyperparameter settings.

6 Empirical application

6.1 Data and model

We implement our methodology on a proprietary database of auto loans provided by an international bank. Such loans are granted to individuals to purchase either new or second-hand cars. For each borrower, we know whether he or she has defaulted ($y = 1$) or not ($y = 0$) on the loan. Given the sensitive nature of the data, we had to randomly under-sample individuals to set the default rate to an arbitrary 20% level. Besides benefits in terms of confidentiality, setting a high arbitrary default rate also protects us against concerns arising from using an unbalanced database. After under-sampling, our database includes 7,440 borrowers. Besides the default target variable, we have access to ten features on the loan (funding amount, funded duration, vehicle price, down-

payment) and on the borrower (job tenure, age, marital status, monthly payment in percentage of income, home ownership status, credit event). We divide the database into a stratified training (70%) and test (30%) samples to have the same default rate in both.

We provide in Table 4 some summary statistics about the features and the target variable. In our dataset, a typical loan amounts to around 11,500 euros, finances a 13,000 euro car, and lasts for 56 months. A typical borrower is 45 years old, married, not owning his or her home, has spent nine years in the same job, experienced no credit event over the past six months, provides less than a 50% down payment, and allocates 10% of his or her monthly income to reimburse the car loan. To get a first sense of the role of each feature on default, we display in Figure 5 their distributions separately for defaulting and non-defaulting borrowers. This preliminary test indicates that the list of discriminating feature includes age, credit event, down payment, marital and ownership status.

Table 4: Summary Statistics

	Count	Mean	Std.	Minimum	25%	50%	75%	Maximum
Job tenure	7,440	9.3298	9.9787	0	2	5	15	58
Age	7,440	45.1691	14.7965	18	33	46	55	89
Car price	7,440	12,935	6,204	546	8,149	11,950	16,500	47,051
Funding amount	7,440	11,461	6,019	546	6,846	10,382	15,000	30,000
Loan duration	7,440	56.2176	19.3833	6	48	60	72	96
Monthly payment	7,440	0.1051	0.0611	0.0051	0.0690	0.0947	0.1304	2.6300
Downpayment	7,440	0.0897		0				1
Credit event	7,440	0.0220		0				1
Married	7,440	0.5347		0				1
Homeowner	7,440	0.3848		0				1
Default	7,440	0.2000		0				1

Note: This table displays summary statistics for each feature used in the XGBoost model as well as the target variable. For each categorical feature the standard deviation (Std.) and the quartiles (25%, 50% and 75%) are not displayed.

Using the training sample, we estimate an XGBoost model to predict default.[15] We selected this type of model because it is recognised as one of the most powerful scoring engines (Gunnarsson

[15]See Baesens et al. (2003), Lessmann et al. (2015), and Gunnarsson et al. (2021) for applications of XGBoost and other machine-learning algorithms in credit scoring.

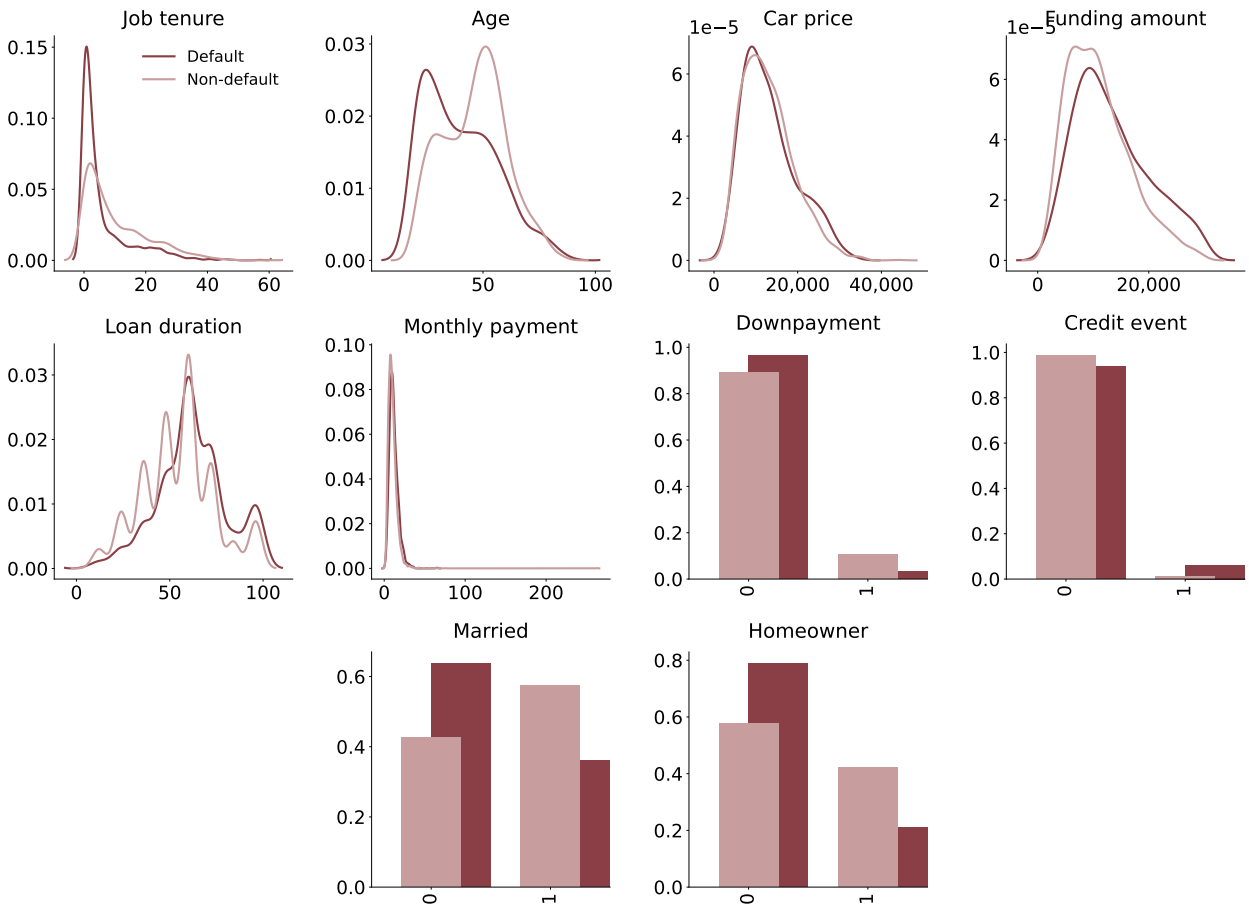


Figure 5: Features distribution by default class

Note: This figure displays the distribution of the features by default class on the training sample, using kernel density estimation for continuous features. Dark red refers to defaulting borrowers and light red to non-defaulting borrowers.

et al., 2021). Another reason for using an XGBoost is its black-box nature. Indeed, while the XPER methodology is model-agnostic, we believe it is interesting to assess its usefulness when used with a particularly complex and opaque algorithm. We select the hyperparameters of the XGBoost using stratified five-fold cross-validation based on a balanced accuracy criteria and a random search algorithm (Bergstra and Bengio, 2012).

Table 5 displays the values of six performance metrics for the XGBoost model obtained on the training and test samples. We choose these six different performance metrics in order to consider all the main categories of performance measures. Specifically, (1) the AUC measures the *discriminatory*

ability of the model, (2) the Brier Score evaluates the *accuracy of probabilities*, and (3) the Balanced Accuracy, PCC, Sensitivity, and Specificity assess the *correctness of categorical predictions*. As shown in Table 5, the XGBoost has an AUC of 0.7521, a Brier Score of 0.1433, and a PCC of 79.53 on the test sample. We observe some over-fitting in the model as its performances drops slightly from the training sample to the test sample. Despite this, the model displays standard performances in credit scoring (Gunnarsson et al., 2021; Lessmann et al., 2015).

Table 5: XGBoost Performances

Sample	AUC	Brier Score	BA	PCC	Sensitivity	Specificity
Train	0.8969	0.0958	0.7243	86.98	0.4818	0.9669
Test	0.7521	0.1433	0.5869	79.53	0.2399	0.9339

Note: This table displays the performances of the XGBoost model on the training and the test sample.

6.2 XPER decomposition

We apply the XPER methodology to decompose several performance measures of the XGBoost model. To do so, we pick one performance metric from each category, namely AUC, Brier Score, and Balanced Accuracy. We display in Figure 6 the decomposition of the AUC among the ten features. For ease of presentation, we express the feature contributions in percentage of the spread between the AUC and its benchmark (see Equation 6). As shown in Table 5, the AUC in the test sample is equal to 0.7521, which is significantly better than the benchmark value of 0.5 obtained for a random predictor. We see that around 40% of this over-performance is coming from the *funding amount* feature. The second most contributing feature is *job tenure*, which accounts for another 18%. It is interesting to note that with only two features, we can explain more than half of the performance of the model. Next, we have five features which contribute each for another 8-10% to the performance. At the other side of the spectrum, *down payment* is the feature contributing the least to the AUC. This feature does not help the model to better predict default than a random predictor as its XPER value is close to 0 and even slightly negative. We believe this negative value for *down payment* is particularly intriguing: it is possible for a feature to be selected by the

algorithm in the training phase and to have a detrimental effect on the performance in the test sample. It illustrates the ability of the XPER decomposition to trace the origin of overfitting in machine learning.

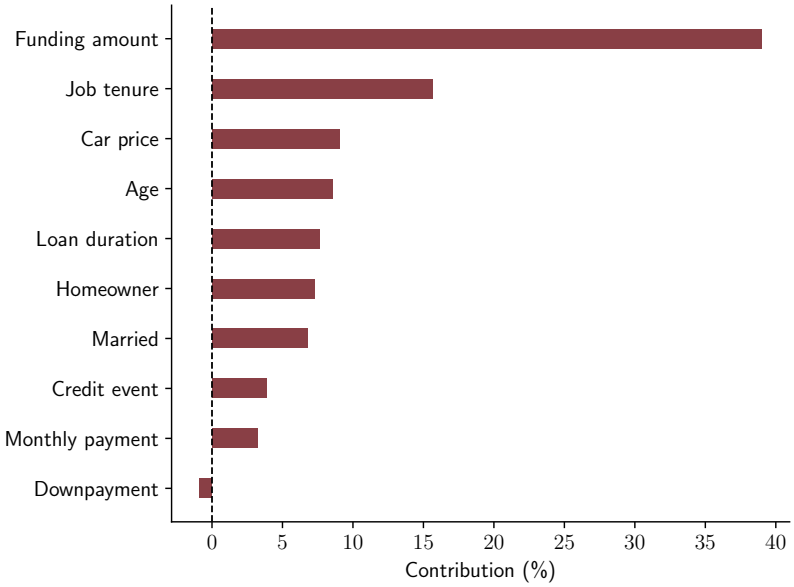


Figure 6: XPER decomposition of the AUC

Note: This figure displays the XPER values of the AUC of the XGBoost model estimated on the test sample. The contributions are expressed in percentage. The vertical dotted line refers to a contribution equal to 0.

Next, we study in Figure 7 the robustness of the XPER decomposition across performance metrics: AUC, Brier Score, and Balanced Accuracy. Maybe surprisingly, the XPER decomposition remains quite similar across metrics, although the considered metrics belong to three different categories of performance measures. For instance, the funding amount and the job tenure account for approximately 60% of each performance metric. However, for some features, the XPER values vary quite drastically across metrics. For instance, the *car price* effect increases the AUC but decreases the Balanced accuracy.

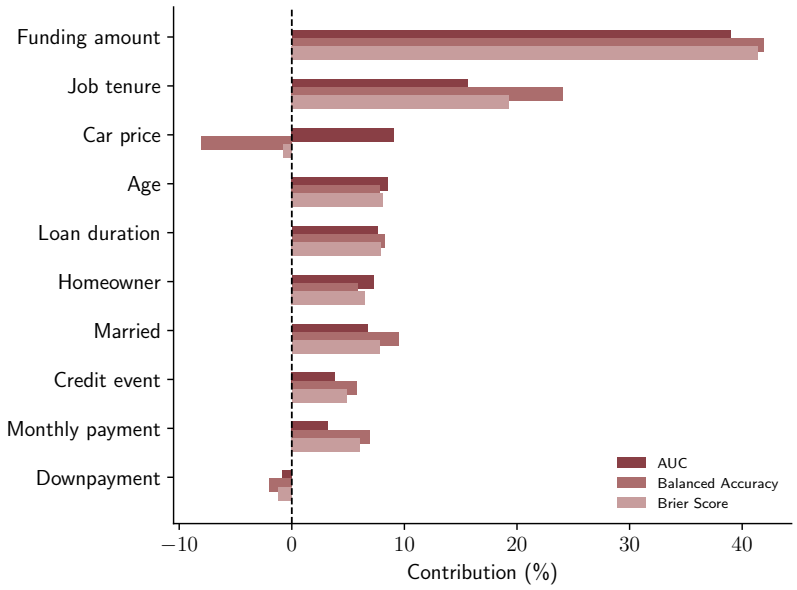


Figure 7: XPER decomposition of the AUC, the Balanced Accuracy, and the Brier Score

Note: This figure displays the XPER values of the AUC, Balanced Accuracy, and Brier Score of the XGBoost model estimated on the test sample. The contributions are expressed in percentage. The contributions are ranked by decreasing order according to the AUC, the most important one starting at the top of the graphic. The vertical dash line refers to a contribution equal to 0.

6.3 XPER vs. standard feature contributions

In this subsection, we compare the XPER performance decomposition to standard feature contribution methods commonly used in machine learning to explain the output of a black-box model, namely feature importance and SHAP values.

First, we contrast in Figure 8 the XPER values of the AUC and the XGBoost-based feature importance. The latter computes for a feature x_j, the average increase in accuracy obtained by splitting nodes using this particular feature. For ease of comparison, we divide each feature contribution by the sum of the ten feature contributions. As shown in Figure 8, the result is rather striking. Indeed, the two methodologies lead to very different contributions as some dominating feature in a given methodology play a minor role in the other. For instance, *credit event* exhibits the highest feature importance but it is only the $8th$ most contributing feature according to XPER. Differently, funding amount plays a very important role to explain performance but does not contribute much

in terms of feature importance. Overall, this figure clearly shows that explaining model forecasts and model performance are two distinguished and complementary tasks. An important implication is that relying on feature importance to guess which features drive performance can be misleading.

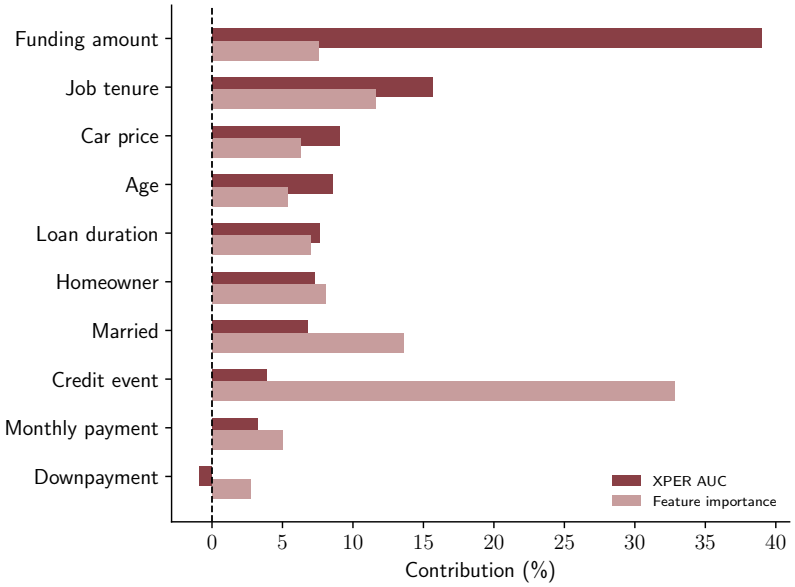


Figure 8: XPER vs. feature importance

Note: This figure displays (1) the XPER values of the AUC and (2) the XGBoost-based feature importances. The contributions are expressed in percentage. The contributions are ranked in decreasing order according to the AUC. The vertical dotted line refers to a contribution equal to 0.

Second, we compare in Figure 9 the XPER decomposition of the AUC with the SHAP values introduced by Lundberg and Lee (2017). In our context, the SHAP values assess the impact of the different features on the probabilities of default of each borrower. As conventionally done, we take the average absolute SHAP values for each feature to assess the feature contribution at the aggregate or model level. As before, we scale each feature contribution by the sum of all features contributions to make them comparable with the XPER values. In Figure 9, we see that SHAP and XPER provide for some features very different information. For instance, the *car price* contribution is more than twice as important for SHAP than for XPER. Similarly, the *funding amount* contribution is around 40% for XPER whereas only 28% for SHAP. Moreover, a more fundamental difference between SHAP and XPER values is that the former is by construction positive whereas the latter can turn

negative. Indeed, as shown in Figure 7, *car price* has a negative impact on the Balanced Accuracy of the model. We conclude that XPER delivers different, incremental information over SHAP at the model level. In the next sub-section, we will show that this is also true at the individual level.

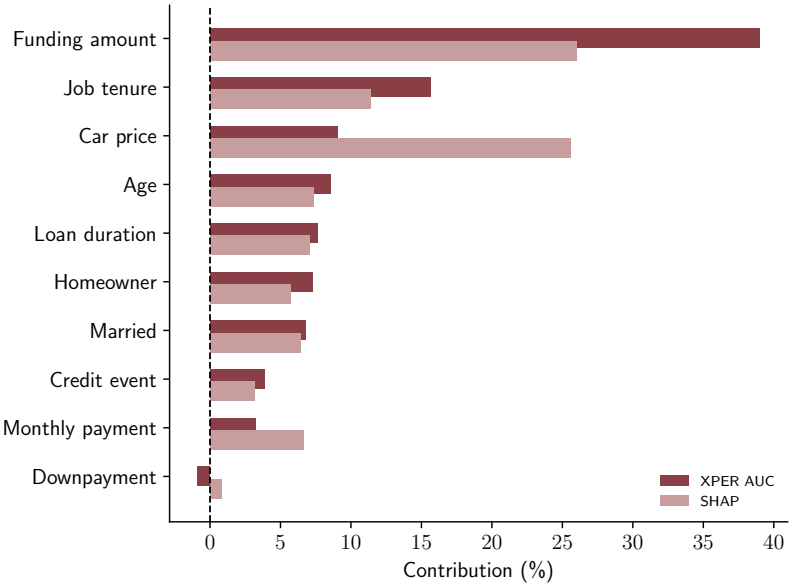


Figure 9: XPER vs. SHAP

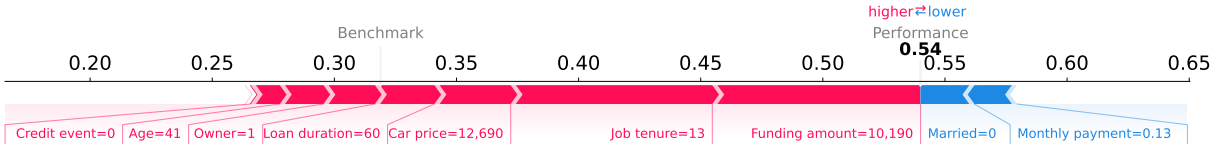
Note: This figure displays (1) the XPER values of the AUC and (2) the average absolute SHAP values. The contributions are expressed in percentage. The contributions are ranked in decreasing order according to the AUC. The vertical dotted line refers to a contribution equal to 0.

6.4 Individual XPER decomposition

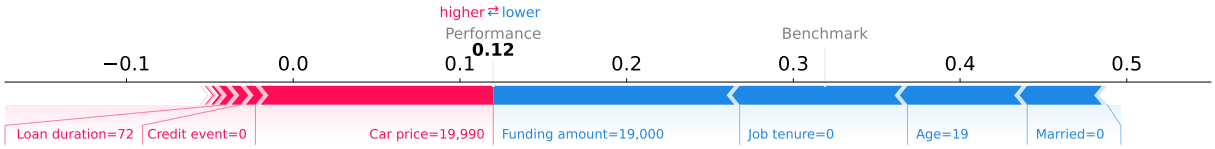
We now analyse the impact of the various features on the performance metric but we now do it for each borrower individually.

We start by analysing in Figure 10 the XPER decomposition for two sample borrowers. These force plots enable us to decompose the individual performance of each borrower, as defined in Equation 9. By doing so, they allow us to understand why some individuals contribute more to the AUC of the model than others. In each panel of the Figure 10, *Performance* refers to the contribution of the borrower to the AUC of the model and *Benchmark* to their benchmark value, i.e., $\phi_{i,0}$ in Equation 9. For each borrower, the features increasing (respectively decreasing) the performance appear in red (blue). Borrower #3 has a relatively high individual AUC compared to

borrower #28 (both have the same benchmark). The over-performance of borrower #3 is mainly due to the large positive XPER values for *funding amount*, *job tenure*, and *car price*. It also comes from the small negative XPER values for the marital status (*married*) and the share of the monthly payment in the borrower's income (*monthly payment*).



(a) Borrower #3



(b) Borrower #28

Figure 10: Force plots of individual XPER values

Note: This figure displays the XPER decomposition (for the AUC) of two loan borrowers (see Equation 8). Borrower #3 did not default on his loan and has a probability of default of 8% according to the XGBoost ((Panel (a)). Borrower #28 did not default on his loan and has a probability of default of 57% according to the XGBoost (Panel (b)). *Performance* refers to the individual level of the AUC whereas *Benchmark* represents the individual contribution to the AUC associated to a population where the target variable y_i is independent from the features $\mathbf{x}_i$. The red color refers to positive XPER values, i.e., features increasing performance. The blue color refers to negative XPER values, i.e., features decreasing performance.

To better understand the relative influence of each feature for the two borrowers, we analyse their risk-profiles and probabilities of default predicted by the model. Let us start with borrower #3. He is 41 years old, homeowner, has a stable job, and applied for a loan to buy a moderately-priced car. He provided a down payment and experienced no past credit event. Intuitively, we would naturally classify this borrower as low-risk and this is confirmed by the 8% default probability estimated by the XGBoost model. Thus, as borrower #3 eventually did not default on his loan, his contribution to the AUC is high. The situation of borrower #28 is quite different as he exhibits a higher risk

profile (young, jobless, not married, relatively large credit amount, no down payment). Yet, the model remains quite undecided about his capacity to pay back the loan with a 57% estimated default probability. As the AUC measures the discriminatory ability of the model, this uncertainty leads to a low individual contribution, and even lower than the benchmark value.

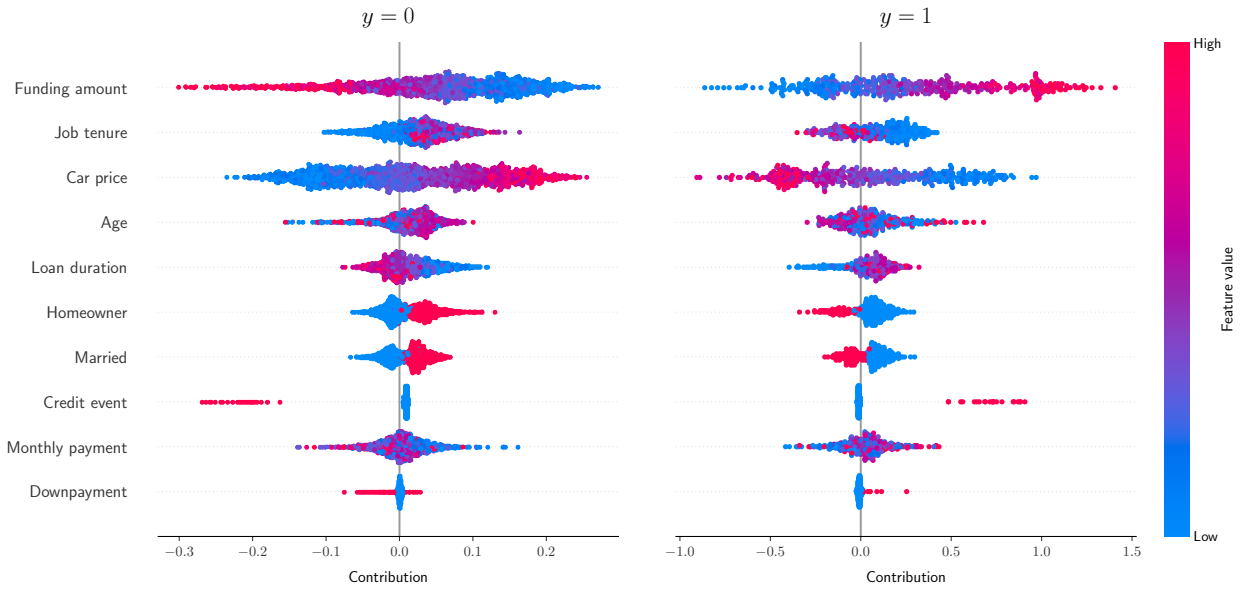


Figure 11: Summary plots of individual XPER values

Note: This figure displays the individual XPER values for each feature used in the XGBoost model. Each dot represents the value for a given borrower (see Equation 8). We display the results for the borrowers who paid back their loans (left graphic) and those who do not (right graphic).

We then consider the entire sample of borrowers. In Figure 11, we display the XPER values for each feature as a function of the feature value. We analyse these results according to two types of borrowers: non-defaulting borrowers (y=0) and defaulting borrowers (y=1). We clearly see that depending on the value of the feature and the type of borrower, we know if this feature contributes to increase or decrease the performance of the model. For instance, for a non-defaulting borrower (left panel), a relatively high job tenure is associated to a positive XPER value. This result is due to the fact that a relatively long job tenure tends to lower the probability of default in the model. Hence, this increases the ability of the model to distinguish him from the defaulting borrowers and boosts the XPER value. On the opposite, for a defaulting borrower (right panel), a relatively high

job tenure leads to a negative XPER value and thus decreases his contribution to the AUC of the model.

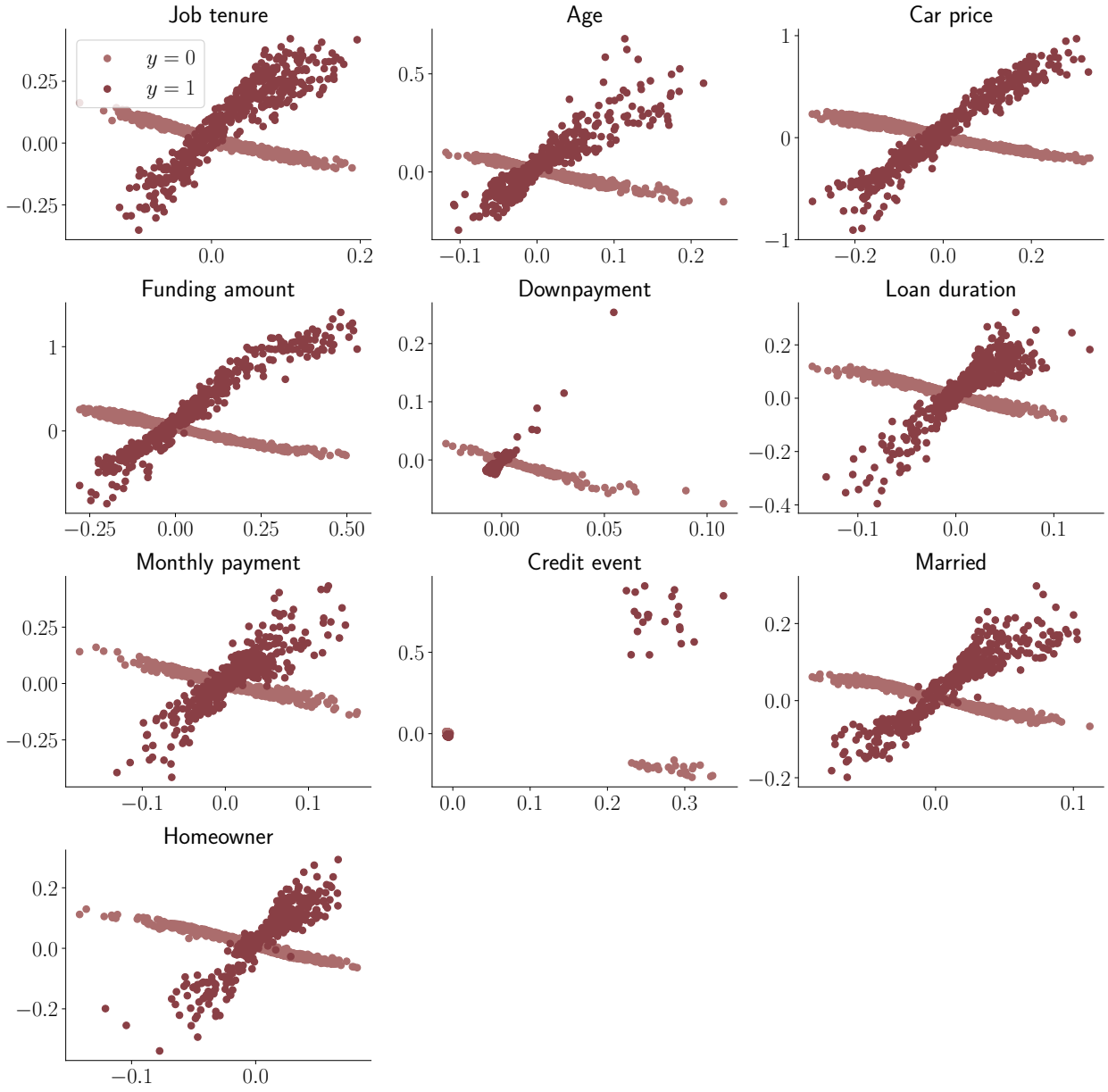


Figure 12: Individual XPER vs. SHAP

Note: This figure displays the relationship between the SHAP values (x-axis) and the XPER values (y-axis) for each feature used in the XGBoost model. Defaulting borrowers are represented in dark red whereas non-defaulting borrowers display in light red.

In a final step, we compare the individual XPER values to the SHAP values. We see in Figure 12 that knowing the SHAP value does not allow someone to infer the XPER value. Indeed, we clearly see that there is no bijective relationship between the SHAP and XPER values. For instance, for the *age* feature of defaulting borrowers, a SHAP value of 0.1 can be associated to XPER values ranging from 0.25 to 0.65. Notice that these discrepancies are more severe for individuals who defaulted on their loans. This finding brings another piece of evidence supporting the idea that explaining model forecasts (feature importance or SHAP) differs markedly from explaining model performance. This remains true both at the aggregate and individual levels.

7 Conclusion

In this paper, we have introduced XPER, a methodology designed to measure the feature contributions to the performance of an econometric or machine learning model. We have built on existing interpretability tools developed in machine learning (feature importance, Shapley values, SHAP) but with the distinct objective of focusing on model performance and not on model predictions $\hat{y}$. Given the fact that performance depends not only on $\hat{y}$ but also on y, one can expect that the driving factors of the model predictions differ from those driving performance. In this paper, we have shown empirically that the discrepancies can be sizable. In such a case, it may become misleading to use $\hat{y}$-based feature contributions instead of performance-based feature contributions.

The XPER methodology offers several advantages. First, it is both model-agnostic and metric-agnostic. Second, XPER is theoretically founded as it is based on Shapley values. Third, the interpretation of the benchmark is meaningful in our context. Fourth, XPER is not plagued by any model specification error as it does not require re-estimating the model. Fifth, our methodology can be implemented either at the global (model) level or at the individual (borrower) level. The latter enables us to understand why some instances contribute more to the performance of the model than others.

While the numerical application conducted in this paper was mainly illustrative, several promis-

ing applications could be envisioned in the future. Examples of such applications include studying the main driving factors of the economic performance generated by an AI-enhanced business. It would obviously be interesting to investigate the usefulness of our methodology to explain the performance of a financial portfolio, and especially so when it is generated by a robo-advisor. Alternatively, one could give a dollar value to different sources of information used in a given machine-learning application, depending on their respective contributions to the overall performance. Finally, XPER could go beyond performance. Indeed, it could for instance be used with any function of both $\hat{y}$ and y. This situation arises for instance when one assess the algorithmic fairness of a machine-learning model. One could identify the features at the origin of the lack of fairness of an given model, and acts accordingly to enhance fairness (see Lundberg, 2020).

A Examples of performance metrics

Table A1: Performance metrics

Metrics	$G_n(\mathbf{y}, \mathbf{x})$	$G(y_i; \mathbf{x}_i; \hat{\delta}_n)$	$\hat{\delta}_n$
MAE	$\frac{1}{n}\sum_{i=1}^n \left\lvert y_i - \hat{f}(\mathbf{x}_i)\right\rvert$	$\left\lvert y_i - \hat{f}(\mathbf{x}_i)\right\rvert$	$\emptyset$
MSE	$\frac{1}{n}\sum_{i=1}^n \left(y_i - \hat{f}(\mathbf{x}_i)\right)^2$	$\left(y_i - \hat{f}(\mathbf{x}_i)\right)^2$	$\emptyset$
R2	$1 - \frac{\sum_{i=1}^n \left(y_i - \hat{f}(\mathbf{x}_i)\right)^2}{\sum_{j=1}^n (y_j - \bar{y})^2}$	$1 - \hat{\delta}_n^{-1}\left(y_i - \hat{f}(\mathbf{x}_i)\right)^2$	$n^{-1}\sum_{j=1}^n (y_j - \bar{y})^2$
Accuracy	$\frac{1}{n}\sum_{i=1}^n \left(y_i\hat{f}(\mathbf{x}_i) + (1-y_i)(1-\hat{f}(\mathbf{x}_i))\right)$	$y_i\hat{f}(\mathbf{x}_i) + (1-y_i)(1-\hat{f}(\mathbf{x}_i))$	$\emptyset$
BA	$\frac{1}{n}\sum_{i=1}^n \frac{1}{2}\left[\frac{y_i\hat{f}(\mathbf{x}_i)}{\frac{1}{n}\sum_{j=1}^n y_j} + \frac{(1-y_i)(1-\hat{f}(\mathbf{x}_i))}{\frac{1}{n}\sum_{j=1}^n (1-y_j)}\right]$	$\frac{1}{2}\left[\hat{\delta}_{n_1}^{-1}\left(y_i\hat{f}(\mathbf{x}_i)\right) + \hat{\delta}_{n_2}^{-1}\left((1-y_i)(1-\hat{f}(\mathbf{x}_i))\right)\right]$	$\hat{\delta}_{n_1} = \frac{1}{n}\sum_{j=1}^n y_j$ $\hat{\delta}_{n_2} = \frac{1}{n}\sum_{j=1}^n (1-y_j)$
Brier score	$\frac{1}{n}\sum_{i=1}^n \left(y_i - \hat{P}(\mathbf{x}_i)\right)^2$	$\left(y_i - \hat{P}(\mathbf{x}_i)\right)^2$	$\emptyset$
Precision	$\frac{1}{n}\sum_{i=1}^n \left(\frac{y_i\hat{f}(\mathbf{x}_i)}{\frac{1}{n}\sum_{j=1}^n \hat{f}(\mathbf{x}_j)}\right)$	$\hat{\delta}_n^{-1} y_i\hat{f}(\mathbf{x}_i)$	$\frac{1}{n}\sum_{j=1}^n \hat{f}(\mathbf{x}_j)$
Sensitivity	$\frac{1}{n}\sum_{i=1}^n \left(\frac{y_i\hat{f}(\mathbf{x}_i)}{\frac{1}{n}\sum_{j=1}^n y_j}\right)$	$\hat{\delta}_n^{-1} y_i\hat{f}(\mathbf{x}_i)$	$\frac{1}{n}\sum_{j=1}^n y_j$
Specificity	$\frac{1}{n}\sum_{i=1}^n \left(\frac{(1-y_i)(1-\hat{f}(\mathbf{x}_i))}{\frac{1}{n}\sum_{j=1}^n (1-y_j)}\right)$	$\hat{\delta}_n^{-1}(1-y_i)(1-\hat{f}(\mathbf{x}_i))$	$\frac{1}{n}\sum_{j=1}^n (1-y_j)$
	$\frac{\sum_{i=1}^n \sum_{j=1}^n (1-y_i)y_j I(\hat{P}(\mathbf{x}_i) < \hat{P}(\mathbf{x}_j))}{\sum_{j=1}^n y_j \sum_{j=1}^n (1-y_j)}$	$\frac{1}{n}\sum_{i=1}^n \left((1-y_i) \times \hat{\delta}_{n_1}(\mathbf{x}_i)\right)\hat{\delta}_{n_2}^{-1}$	$\hat{\delta}_{n_1}(\mathbf{x}_i) = \frac{1}{n}\sum_{j=1}^n y_j I(\hat{P}(\mathbf{x}_i) < \hat{P}(\mathbf{x}_j))$
AUC	$I(\hat{P}(\mathbf{x}_i) < \hat{P}(\mathbf{x}_j)) = \begin{cases} 0 & \text{if } \hat{P}(\mathbf{x}_i) > \hat{P}(\mathbf{x}_j) \\ 0.5 & \text{if } \hat{P}(\mathbf{x}_i) = \hat{P}(\mathbf{x}_j) \\ 1 & \text{if } \hat{P}(\mathbf{x}_i) < \hat{P}(\mathbf{x}_j) \end{cases}$		$\hat{\delta}_{n_2} = \frac{1}{n^2}\sum_{j=1}^n y_j \sum_{j=1}^n (1-y_j)$

Note: This table displays the expression of sample performance metrics $G_n(\mathbf{y}, \mathbf{x})$, individual contribution to the sample performance metric $G(y_i; \mathbf{x}_i; \hat{\delta}_n)$, and the corresponding nuisance parameter $\hat{\delta}_n$. The solid black line allows us to distinguish between regression and classification performance metrics.

B Additive property in a three-fold model

In a three-fold model, the XPER value ϕ_j can be written as follows:

$$\phi_j = \frac{1}{3}\left(\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) - \phi_0\right) + M_j(G\left(y;\mathbf{x};\delta_0\right)) \tag{16}$$

where $M_j(G\left(y;\mathbf{x};\delta_0\right))$, for $j = 1, 2, 3$, is defined as:

$$\begin{aligned}
M_1(G\left(y;\mathbf{x};\delta_0\right)) = &\frac{1}{6}\left(\mathbb{E}_{x_2}\mathbb{E}_{y,x_1,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_2}\mathbb{E}_{y,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{6}\left(\mathbb{E}_{x_3}\mathbb{E}_{y,x_1,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_3}\mathbb{E}_{y,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{3}\left(\mathbb{E}_{x_2,x_3}\mathbb{E}_{y,x_1}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1}\mathbb{E}_{y,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right)
\end{aligned}$$

$$\begin{aligned}
M_2(G\left(y;\mathbf{x};\delta_0\right)) = &\frac{1}{6}\left(\mathbb{E}_{x_1}\mathbb{E}_{y,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_2}\mathbb{E}_{y,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{6}\left(\mathbb{E}_{x_3}\mathbb{E}_{y,x_1,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_2,x_3}\mathbb{E}_{y,x_1}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{3}\left(\mathbb{E}_{x_1,x_3}\mathbb{E}_{y,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_2}\mathbb{E}_{y,x_1,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right)
\end{aligned}$$

$$\begin{aligned}
M_3(G\left(y;\mathbf{x};\delta_0\right)) = &\frac{1}{6}\left(\mathbb{E}_{x_2}\mathbb{E}_{y,x_1,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_2,x_3}\mathbb{E}_{y,x_1}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{6}\left(\mathbb{E}_{x_1}\mathbb{E}_{y,x_2,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_1,x_3}\mathbb{E}_{y,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right) \\
&+ \frac{1}{3}\left(\mathbb{E}_{x_1,x_2}\mathbb{E}_{y,x_3}\left(G\left(y;\mathbf{x};\delta_0\right)\right) - \mathbb{E}_{x_3}\mathbb{E}_{y,x_1,x_2}\left(G\left(y;\mathbf{x};\delta_0\right)\right)\right).
\end{aligned}$$

Thus, the XPER values satisfy the additive property:

$$\sum_{j=1}^{3}\phi_j = \underbrace{\sum_{j=1}^{3}\frac{1}{3}\left[\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) - \phi_0\right]}_{\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))-\phi_0} + \underbrace{\sum_{j=1}^{3}M_j(G\left(y;\mathbf{x};\delta_0\right))}_{=\ 0},$$

or equivalently,

$$\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) = \phi_0 + \sum_{j=1}^{3}\phi_j.$$

C Examples of XPER value decomposition

Below we provide several illustrations of the XPER value decomposition for several standard performance metrics.

Illustration 1: Regression model and MSE. Consider a linear regression model $\hat{f}(\mathbf{x}_i) = \sum_{j=1}^{q} \hat{\beta}_j x_{i,j}$, and the (opposite of the) MSE as sample performance metric. We assume that the DGP generating the test sample $S_n = \{\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i)\}_{i=1}^{n}$ satisfies $\mathbb{E}(\mathbf{x}) = 0_q$ and $\mathbb{V}(\mathbf{x}) = diag(\sigma^2_{x_j})$ $\forall j = 1, ..., q$. We denote by σ^2_y the variance of the target variable and by σ_{y,x_j} the covariance between the feature x_j and the target variable. Then, the contributions ϕ_j of features x_j to the (opposite of the) MSE satisfy the efficiency property such that:

$$\underbrace{2\sum_{j=1}^{q} \hat{\beta}_j \sigma_{y,x_j} - \sum_{j=1}^{q} \hat{\beta}_j^2 \sigma^2_{x_j} - \sigma^2_y}_{\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))} = \underbrace{- \sum_{j=1}^{q} \hat{\beta}_j^2 \sigma^2_{x_j} - \sigma^2_y}_{\phi_0} + \sum_{j=1}^{q} \underbrace{2\hat{\beta}_j \sigma_{y,x_j}}_{\phi_j}. \tag{17}$$

Formally, the XPER value ϕ_j depends on the estimated parameter $\hat{\beta}_j$ (estimation sample) and the covariance between x_j and the target variable y (test sample), i.e., σ_{y,x_j}. The XPER values ϕ_j, $\forall j = 1, ..., q$, are positive or null.[16] The dummy property $\phi_j = 0$ is either satisfied if the feature has no impact on the model ($\hat{\beta}_j = 0$) or if the feature is uncorrelated with the target variable on the test sample ($\sigma_{y,x_j} = 0$). Similarly, a variable x_j has a larger MSE contribution than a feature x_s as soon as $\hat{\beta}_j \sigma_{y,x_j} > \hat{\beta}_s \sigma_{y,x_s}$, meaning that x_j is more related to the target variable than x_s both in-sample (through $\hat{\beta}_j$) and out-sample (through σ_{y,x_s}). The benchmark ϕ_0 corresponds to the MSE that we would obtain by applying the model to data generated by a DGP where the target variable is independent from the features.

[16] If the DGPs of the training and test samples are similar, we expect the model parameters $\hat{\beta}_j$ and covariances σ_{y,x_j} to have the same sign, which means $\phi_j > 0$.

Illustration 2: Regression model and R^2. Consider the same linear regression model as in the previous illustration with the R^2 as performance metric. The XPER values ϕ_j of the R^2 satisfy:

$$\underbrace{\frac{\sigma_{y,\hat{y}}}{\sigma_y^2}}_{\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))} = \underbrace{-\frac{\sigma_{y,\hat{y}}}{\sigma_y^2}}_{\phi_0} + \sum_{j=1}^{q} \underbrace{\frac{2\hat{\beta}_j \sigma_{y,x_j}}{\sigma_y^2}}_{\phi_j}. \tag{18}$$

Feature contributions ϕ_j are equal to those obtained for the MSE, normalized by the variance of the target variable. The main difference comes from the benchmark. This benchmark corresponds to the R^2 associated to the model $\hat{f}(\mathbf{x})$ when it is applied on a test sample issued from a population where the target variable is independent from all model features. We might expect the benchmark to be equal to 0. However, this benchmark is negative. To illustrate this result, we have to distinguish the estimation DGP from the test DGP. Usually, both DGPs are the same but here this distinction allows us to consider two cases. In the first case, the target variable does not depend on the features, meaning that the estimated model is reduced to a constant, i.e., $\hat{y} = \hat{c}$. Whatever the dependence between the features and the target variable in the test DGP, the population R^2 is equal to 0. In the second case, the features are correlated to the target variable in the estimation DGP. As usually, if the test and estimation DGPs are the same then the population R^2 is equal to a given positive value $\gamma \in]0, 1]$. On the contrary, when $\mathbf{x} \perp y$ in the test DGP, then the estimated model is misspecified as the parameters $\hat{\beta}$ are different from 0 whereas they should be. The R^2 in this situation corresponds to our benchmark ϕ_0. The information contained in model features is now misleading the model compared to a model excluding them. Therefore, the estimated model R^2 is lower than the R^2 associated to model without features. As the R^2 of a model without features is equal to 0, then the estimated model has necessarily a negative population R^2, i.e., $\phi_0 < 0$.

Illustration 3: Classification model and accuracy. Consider a logistic regression model and the accuracy as sample performance metric with

$$\hat{f}(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \hat{P}(\mathbf{x}_i) = 1/\left[1 + \exp\left(-\mathbf{x}_i\hat{\beta}\right)\right] > \pi \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where $\pi \in [0, 1]$ refers to a cutoff value. We denote by $\sigma_{y,\hat{f}(\mathbf{x})}$ the covariance between the target

variable and the classification output. Then, the efficiency property becomes:

$$\underbrace{2\sigma_{y,\hat{f}(\mathbf{x})} + 2P(\mathbf{x})\hat{P}(\mathbf{x}) + 1 - P(\mathbf{x}) - \hat{P}(\mathbf{x})}_{\mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0))} = \underbrace{2P(\mathbf{x})\hat{P}(\mathbf{x}) + 1 - P(\mathbf{x}) - \hat{P}(\mathbf{x})}_{\phi_0} + \underbrace{2\sigma_{y,\hat{f}(\mathbf{x})}}_{\sum_{j=1}^{q}\phi_j} \tag{20}$$

with $P(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x})$. Unlike the example of the MSE, the Shapley values ϕ_j, $\forall j = 1, ..., n$ do not have any analytical expression. As expected, the XPER values depend on the covariance between the target variable y and the classification output $\hat{f}(\mathbf{x})$. The benchmark ϕ_0 corresponds to the accuracy that we would obtain by applying the model to data generated by a DGP where the target variable is independent from the features. This approach can be extended to any classification metric, such as:

$$\text{Precision: } \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) = \mathbb{P}(y = 1) + \sum_{j=1}^{q}\phi_j \tag{21}$$

$$\text{Sensitivity: } \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) = \mathbb{P}(\hat{y} = 1) + \sum_{j=1}^{q}\phi_j \tag{22}$$

$$\text{Specificity: } \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) = \mathbb{P}(\hat{y} = 0) + \sum_{j=1}^{q}\phi_j \tag{23}$$

$$\text{AUC: } \mathbb{E}_{y,\mathbf{x}}(G(y;\mathbf{x};\delta_0)) = 0.5 + \sum_{j=1}^{q}\phi_j. \tag{24}$$

Given the non-linearity of the logit model, the XPER values ϕ_j and the population metrics do not have analytical expressions, contrary to the benchmark. For the precision, the sensitivity, the specificity, and the AUC the benchmark ϕ_0 has a nice interpretation. For instance, for the AUC, the benchmark corresponds to the AUC associated to a random predictor and is equal to 0.5.

D Feasible estimation with many features

As shown in section 3, the XPER value ϕ_j relies on every possible coalition for each feature. The total number of coalitions required to compute $(\phi_1, \ldots, \phi_q)$ is equal to $q \times 2^{(q-1)}$ coalitions, with $2^{(q-1)}$ the number of coalitions for feature j. As the total number of coalitions quickly increases with the number of features ($5,120$ for $q = 10$ and $10,485,760$ for $q = 20$), XPER values turn out to be cumbersome to compute or estimate in practice.[17]

To overcome this issue, a standard approach in the literature is to approximate XPER values. Intuitively, the idea is to rely only on a subset of coalitions to compute XPER values rather than on its total number in order to reduce the number of computations required. To do so, we propose an approximation method of XPER values which is based on the recent model-agnostic approach of Lundberg and Lee (2017) called Kernel SHAP. The main advantage of this approach is that it can be used for any application as it does not depend on the model $f(.)$ nor on the performance metric $G_n(\mathbf{y}; \mathbf{X})$. Denote as S_k a coalition randomly drawn for $k = 1, \ldots, K$, $K < q \times 2^{(q-1)}$ the total number of coalitions drawn, and $\tilde{S}_k = \{\mathbf{x}^{\bar{S}_k}\} \cup \{\mathbf{x}_j\}$ the vector of features not included in coalition S_k. Formally, the approximation of XPER values $\hat{\phi}_j$ is based on a subset K of coalitions S_k and on the following model

$$G_{n,k}(\mathbf{y}; \mathbf{X}) = \phi_0 + \sum_{j=1}^{q} \phi_j z_{k,j}, \tag{25}$$

where

$$G_{n,k}(\mathbf{y}; \mathbf{X}) = \frac{1}{n^2} \sum_{u=1}^{n} \sum_{v=1}^{n} G\left(y_v; \mathbf{x}_v^{S_k}, \mathbf{x}_u^{\overline{S}_k}; \hat{\delta}_n\right),$$

is a sample performance metric associated to the coalition S_k, and $z_{k,j}$ is equal to 1 if the feature j belongs to the coalition S_k, and 0 otherwise. In matrix notation, we have

$$\mathbf{G_n}(\mathbf{y}; \mathbf{X}) = \mathbf{Z}\boldsymbol{\phi},$$

[17] This issue is related to the general concept of Shapley value and is not specific to our approach.

where

$$\underset{(K\times 1)}{\mathbf{G_n}}(\mathbf{y};\mathbf{X}) = \begin{pmatrix} G_{n,1}(\mathbf{y};\mathbf{X}) \\ \vdots \\ G_{n,K}(\mathbf{y};\mathbf{X}) \end{pmatrix}, \underset{(K\times q+1)}{\mathbf{Z}} = \begin{pmatrix} 1 & z_{1,1} & .. & z_{1,q} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_{K,1} & .. & z_{K,q} \end{pmatrix}, \underset{(q+1\times 1)}{\boldsymbol{\phi}} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_q \end{pmatrix}.$$

This approach relies on the comparison of K values of the performance metric obtained for K randomly drawn coalitions involving different subset of variables.[18] The features being present in some coalitions and missing in others, the estimation of this regression allows to approximate the impact of each feature on the performance metric that would be obtained for the subset k of features, and thus to approximate estimated XPER values $\hat{\phi}_j$. Moreover, contrary to Equation (11) this approach estimates the impact of every feature all at once, similarly to parameters' estimation in a traditional ordinary least squares regression, and is much faster as it only involves the calculation of K coalitions rather than $q \times 2^{(q-1)}$.

Recall that in Equation (11) the sum of marginal contribution to the performance metric is weighted as some coalitions are more informative than others. To this end, Lundberg and Lee (2017) propose the Kernel Shap which is a function that attributes a higher weight to the most informative coalitions, and which also takes into account the fact that some coalitions are drawn while others are not. The Kernel Shap function attributes a weight w_{S_k} to coalition S_k such as

$$w_{S_k} = \frac{(q-1)}{\frac{q!}{|S_k|!(q-|S_k|)!}|S_k|\,(q-|S_k|)}, \tag{26}$$

where $|S_k|$ is the number of features included in the coalition S^k. To take into account these weights, Lundberg and Lee (2017) propose to estimate Equation (25) by Weighted Least Squares (WLS).[19] In practice, a relatively large value of K allows to obtain accurate approximations of $\hat{\phi}_j$.

Finally, this approach can also be used to approximate individual contributions to the estimated

[18] Theses coalitions are randomly drawn from the total number of coalitions involving every feature, i.e., from 2^q coalitions.

[19] Note that $S_k = \{\mathbf{x}\}$ and $S_k = \{\emptyset\}$ are not considered in the set of randomly individual coalitions as they lead to infinite weights. The K coalitions are thus drawn from $2^q - 2$ coalitions. See Lundberg and Lee (2017) for more details on the approximation approach of Shapley values.

XPER values based on the following model

$$G_k\left(y_i;\mathbf{x}_i\right)=\phi_{i,0}+\sum_{j=1}^{q}\phi_{i,j}z_{k,j}, \tag{27}$$

where

$$G_k\left(y_i;\mathbf{x}_i\right)=\frac{1}{n}\sum_{u=1}^{n}G\left(y_i;\mathbf{x}_i^{S_k},\mathbf{x}_u^{\tilde{S}_k};\hat{\delta}_n\right).$$

Equation (27) is then estimated for each observation i of the sample, and the weight w_{S_k} remains the same as defined previously.

References

Aas, K., Jullum, M., and Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502.

Agarwal, A., Dhamdhere, K., and Sundararajan, M. (2019). A new interaction index inspired by the taylor series.

Ahlburg, D. A. (1984). Forecast evaluation and improvement using Theil's decomposition. *Journal of Forecasting*, 3(3):345–351.

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

Baesens, B., Setiono, R., Mues, C., and Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3):312–329.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 12:281–305.

Borup, D., Coulombe, G., Rapach, D. E., Schütte, E. C. M., and Schwenk-Nebbe, S. (2022). The Anatomy of Out-of-Sample Forecasting Accuracy. *Federal Reserve Bank of Atlanta, Working paper series*.

Bourguignon, F. (1979). Decomposable Income Inequality Measures. *Econometrica*, 47(4):901–920.

Bowen, D. and Ungar, L. (2020). Generalized shap: Generating multiple types of explanations in machine learning. *arXiv preprint arXiv:2006.07155*.

Casalicchio, G., Molnar, C., and Bischl, B. (2019). Visualizing the feature importance for black box models. In *Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer International Publishing.

Chantreuil, F., Courtin, S., Fourrey, K., and Lebon, I. (2019). A note on the decomposability of inequality measures. *Social Choice and Welfare*, 53(2):283–298.

Chevan, A. and Sutherland, M. (1991). Hierarchical partitioning. *The American Statistician*, 45(2):90–96.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Genizi, A. (1993). Decomposition of R^2 in multiple regression with correlated regressors. *Statistica Sinica*, pages 407–420.

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

Green, P. E., Carroll, J. D., and DeSarbo, W. S. (1978). A new measure of predictor variable importance in multiple regression. *Journal of Marketing Research*, 15(3):356–360.

Grömping, U. (2015). Variable importance in regression models. *Wiley interdisciplinary reviews: Computational statistics*, 7(2):137–152.

Grömping, U. (2007). Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician*, 61(2):139–147.

Gunnarsson, B. R., vanden Broucke, S., Baesens, B., Óskarsdóttir, M., and Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, 295(1):292–305.

Huettner, F. and Sunder, M. (2012). Axiomatic arguments for decomposing goodness of fit according to Shapley and Owen values. *Electronic Journal of Statistics*, 6:1239–1250.

Israeli, O. (2007). A Shapley-based decomposition of the R-square of a linear regression. *Journal of Economic Inequality*, 5(2):199–212.

Janzing, D., Minorics, L., and Blöbaum, P. (2020). Feature relevance quantification in explainable ai: A causal problem. In *International Conference on artificial intelligence and statistics*, pages 2907–2916. PMLR.

Johnson, J. W. (2000). A heuristic method for estimating the relative weight of predictor variables in multiple regression. *Multivariate Behavioral Research*, 35(1):1–19.

Kruskal, W. (1987). Relative importance by averaging over orderings. *The American Statistician*, 41(1):6–10.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136.

Lindeman, R. H., Merenda, P. F., and Gold, R. Z. (1980). *Introduction to Bivariate and Multivariate Analysis*. Glenview.

Lipovetsky, S. and Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330.

Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Molnar, C. (2020). *Interpretable machine learning*. Lulu.com.

Owen, A. B. and Prieur, C. (2017). On Shapley Value for Measuring Importance of Dependent Inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002.

Redell, N. (2019). Shapley decomposition of R-squared in machine learning models. *arXiv preprint arXiv:1908.09718*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "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*, KDD '16, pages 1135–1144. Association for Computing Machinery.

Senoner, J., Netland, T., and Feuerriegel, S. (2021). Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*.

Shapley, L. (1953). The value of a player in n-person games. *Contributions to the Theory of Games*, 2:307–317.

Shorrocks, A. F. (1980). The Class of Additively Decomposable Inequality Measures. *Econometrica*, 48(3):613–625.

Shorrocks, A. F. (1982). Inequality decomposition by factor components. *Econometrica*, 50(1):193–211.

Shorrocks, A. F. (1984). Inequality Decomposition by Population Subgroups. *Econometrica*, 52(6):1369–1385.

Singal, R., Besbes, O., Desir, A., Goyal, V., and Iyengar, G. (2022). Shapley meets uniform: An axiomatic framework for attribution in online advertising. *Management Science*.

Strumbelj, E. and Kononenko, I. (2010). An Efficient Explanation of Individual Classifications using Game Theory. *The Journal of Machine Learning Research*, 11:1–18.

Stufken, J. (1992). On hierarchical partitioning. *The American Statistician*, 46:70–71.

Sundararajan, M. and Najmi, A. (2020). The many Shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Theil, H. (1971). *Applied Economic Forecasting*. North-Holland, Amsterdam.

Wang, Q., Huang, Y., Jasin, S., and Singh, P. V. (2022). Algorithmic transparency with strategic users. *Management Science*.