Annals of Operations Research
https://doi.org/10.1007/s10479-024-06033-1

ORIGINAL RESEARCH

Toward interpretable machine learning: evaluating models of heterogeneous predictions

Ruixun Zhang[1,2,3,4]

Received: 17 March 2023 / Accepted: 24 April 2024

Abstract
AI and machine learning have made significant progress in the past decade, powering many applications in FinTech and beyond. But few machine learning models, especially deep learning models, are interpretable by humans, creating challenges for risk management and model improvements. Here, we propose a simple yet powerful framework to evaluate and interpret any black-box model with binary outcomes and explanatory variables, and heterogeneous relationships between the two. Our new metric, the signal success share (SSS) cross-entropy loss, measures how well the model captures the relationship along any feature or dimension, thereby providing actionable guidance on model improvements. Simulations demonstrate that our metric works for heterogeneous and nonlinear predictions, and distinguishes itself from traditional loss functions in evaluating model interpretability. We apply the methodology to an example of predicting loan defaults with real data. Our framework is more broadly applicable to a wide range of problems in financial and information technology.

Keywords Machine learning · Interpretability · Heterogeneous prediction · Bayesian statistics · Loan default

1 Introduction

AI and machine learning have made significant progress in technology and finance in the last decade (LeCun et al., 2015; Goodfellow et al., 2016; Russell and Norvig, 2016; Giglio et al., 2022). Successful applications span a wide range of areas that influence our society,[1]

[1] Examples include computer vision (Szeliski, 2010; Szegedy et al., 2016) natural language processing (Hirschberg and Manning, 2015; Young et al., 2018), health care (Hanson III and Marshall, 2001; Shen et al., 2017; Esteva et al., 2019), robotics (Lin, 1993; Mnih et al., 2015), algorithmic trading (Heaton et al., 2017; Henrique et al., 2019), derivative pricing (Hutchinson et al., 1994; Culkin and Das, 2017; De Spiegeleer et al., 2018), and risk management (Khandani et al., 2010; Bisias et al., 2012). With the recent breakthrough

✉ Ruixun Zhang
zhangruixun@pku.edu.cn

[1] School of Mathematical Sciences, Peking University, Beijing, China

[2] Center for Statistical Science, Peking University, Beijing, China

[3] National Engineering Laboratory for Big Data Analysis and Applications, Peking University, Beijing, China

[4] Laboratory for Mathematical Economics and Quantitative Finance, Peking University, Beijing, China

Published online: 10 May 2024

Meanwhile, AI and especially deep learning suffer from a lack of interpretability (Murdoch et al., 2019). Deep neural networks are commonly perceived as black-box models and difficult to understand. This limitation has hindered the adoption of powerful machine learning models in many areas including finance. For example, if a trading algorithm has lost 20% in a week, investors need to understand why; if a loan application is denied using an algorithm, the General Data Protection Regulation (GDPR) requires that the consumer is given a right to explanation; if a student is not admitted to a school, one cannot simply blame a black-box model that made the decision. Such examples exist in many high-stake applications of modern AI, and interpretability is a key aspect of machine learning that heavily influences its future.

In economics and finance, researchers have used linear models for decades, which are perhaps the most primitive and explainable form of machine learning. Here, models serve as tools to understand the underlying dynamics of the economic or financial problem. In response to the rapid development of AI technology and the sea of new data sets available every day, there is an increasing interest in applying machine learning techniques to economics and finance; see, for example, Mullainathan and Spiess (2017) and Giglio et al. (2022). In fact, the investment management industry has successfully adopted advanced AI and machine learning techniques for many years because they care more about the prediction performance (Patterson, 2010; De Prado, 2018; Zuckerman, 2019). Improving the interpretability of these methods can potentially unlock their power to understand complex dynamics in economics and finance.

Here we propose a new framework and a new metric, which we call the signal success share (SSS) cross-entropy loss, to evaluate any black-box model on its interpretability, including linear models, tree-based models such as boosting trees and random forests, and deep neural networks. By interpretability, we mean how well the model captures relationships between the target variable and explanatory variables, also referred to as "descriptive accuracy" in the recent statistics literature (Murdoch et al., 2019), as opposed to how well the black-box model predicts the target variable, which is commonly referred to as "predictive accuracy".

Interpretable machine learning has received an increasing amount of attention recently (Rudin, 2014; Arras et al., 2017; Ahmad et al., 2018; Nemati et al., 2018; Chen et al., 2019; Molnar, 2019; Rudin, 2019; Davis et al., 2023). [2] Our methodology is based on a simple idea of conditioning in Bayesian statistics. It contributes to the literature as a model-agnostic method that can evaluate any black-box model with binary outcomes, binary explanatory variables, and heterogeneous relationships between the two across different samples. In particular, our SSS cross-entropy loss measures how well the model captures the relationship along any feature or dimension as well as any interactions.

We demonstrate that our metric works across models with homogeneous predictions, heterogeneous predictions, and predictions with nonlinear interactions. We also highlight its difference from traditional loss functions such as the mean squared error (MSE) because it precisely captures the interpretability but not the predictive ability of the model, thereby providing actionable guidance on model improvements as a supplement to traditional loss functions. We highlight the influence of sample size on our methodology, and demonstrate its practical applicability with an example of loan default prediction using real data.

Footnote 1 continued
of AlphaGo beating human experts in the game of Go (Silver et al., 2016, 2017), people wonder where the limit of AI is.

[2] For interpretation methods specific to deep neural networks, see Yosinski et al. (2015); Montavon et al. (2018) for example. Another class of machine learning models that is interpretable is decision trees (see, for example, recent discussions by Bertsimas et al. (2019) and Bertsimas and Kallus (2020)). Here we discuss general interpretation frameworks that are model-agnostic.

In this paper, we illustrate and validate our framework in the context of understanding loan default.[3] More broadly, many problems in economics, finance, healthcare, and technology have a common mathematical structure of binary predictions, and can therefore benefit from our framework to interpret complicated machine learning models for these problems.

In Sect. 2 we introduce our framework and the signal success share (SSS) cross-entropy loss. Section 3 shows how to use the new metric to evaluate models with heterogeneous predictions on synthetic data. Section 4 applies the methodology to a problem with real data. Section 5 concludes.

2 Problem statement and the framework

Many problems are concerned with the relationship between a target variable, Y, and a set of explanatory variables (or factors), $F_1, \ldots, F_n$. For example, the default or refinancing of a loan is affected by loan characteristics such as principal and maturity, and individual characteristics such as gender, age, income, and risk appetite; the merger and acquisition decision of two companies is affected by their managers' psychology, respective market share, industry outlook, and the regulatory environment; the return of a stock is affected by company fundamentals, market factors, and the sentiment of investors. In all of these examples, uncovering the relationship between Y and $F_1, \ldots, F_n$ is at the core of understanding the underlying problem. In this paper, we consider an example to understand the dynamics of loan default, but our technique applies to general problems of such a form.

2.1 Predicting the default of loans

We consider M loans and $Y_i = 1$ denotes default for the i-th loan and 0 otherwise, for $i = 1, \ldots, M$. Each loan's default is determined by a few underlying factors:

$$Y_i = \mathcal{F}\left(F_{1,i}, \ldots, F_{N,i}\right) + \epsilon_i \tag{1}$$

where $F_{1,i}, \ldots, F_{N,i}$ are the values of N factors for loan i that are potentially correlated, ϵ_i is the idiosyncratic error for loan i, and $\mathcal{F}(\cdot)$ is the function that determines the relationship between the target variable and explanatory variables. The target variable and some of the explanatory variables are observable, but many other explanatory variables and the specific functional form $\mathcal{F}(\cdot)$ are not. A researcher estimates $\mathcal{F}(\cdot)$ based on a set of observed data.

Suppose a prediction model for Y_i is given as:

$$\hat{Y}_i = \hat{\mathcal{F}}\left(F_{1,i}, \ldots, F_{N,i}\right) \tag{2}$$

where we use $\hat{\mathcal{F}}(\cdot)$ to denote the estimate for the unknown relationship $\mathcal{F}(\cdot)$. Here $\hat{\mathcal{F}}(\cdot)$ can be simple linear models and decision trees, or any black box models such as random forests and deep neural networks.

The goal of this paper is not to find the best $\hat{\mathcal{F}}(\cdot)$ but rather, given a model $\hat{\mathcal{F}}(\cdot)$, to develop a metric that interprets the underlying dynamics of $\hat{\mathcal{F}}(\cdot)$, and evaluates whether $\hat{\mathcal{F}}(\cdot)$ captures the true relationship between Y_i and F_i as specified by $\mathcal{F}(\cdot)$.

[3] In recent years, researchers have focused on why loans and mortgages default (Campbell and Dietrich, 1983; Campbell and Cocco, 2015; Serrano-Cinca et al., 2015) or get prepaid (Keys et al., 2016), and how that affects the dynamics of household finance (Campbell, 2006) and the financial system as a whole (Khandani et al., 2013).

How do we tackle this problem today? There are at least three different angles. The first is to assume that the underlying dynamics is linear, and to use linear functions to approximate $\hat{\mathcal{F}}(\cdot)$. This approach is attractive because linear functions are easy to understand and straightforward to interpret. However, it only captures first-order relationships and ignores nonlinearities between the target variable and explanatory variables, and cannot harness the power of more complicated machine learning models. In fact, highly nonlinear deep neural networks have been shown to provide impressive performance for applications such as computer vision and natural language processing (LeCun et al., 2015; Goodfellow et al., 2016). The question is whether finance can benefit from the same development in these techniques.

The second approach is to focus only on the prediction performance of $\hat{\mathcal{F}}(\cdot)$, as captured by some loss function $Loss(Y_i, \hat{Y}_i)$ such as the cross-entropy loss and MSE. This approach is the foundation for developing models that are highly predictive of the target variable, but it does not guide us towards interpretable models, nor does it tell us when models are overfitting. In addition, having models that are highly predictive does not guarantee an improved understanding of the underlying dynamics of the economic problem we study. Therefore, developing an approach to interpret black-box models is essential to their adoption in finance and economics.

The third approach is the so-called "first-order" methods, which use derivative-based techniques to study what happens to the predicted $\hat{Y}$ if one feature in the model is removed or locally perturbed. This has led to techniques such as feature importance (Friedman et al., 2001) and the partial dependence plot (Friedman, 2001). Although useful for interpreting simple models, these "first-order" methods only capture the average marginal relationship of one feature and do not work for heterogeneous predictions. Imagine a model that correctly captures a relationship with a positive sign for half of the data, and a relationship with a negative sign for the other half. Such methods would yield zero net marginal prediction while the model has actually captured useful dynamics.

The metric we propose in the next section aims to tackle the problems in these approaches.

2.2 The signal success share (SSS) cross-entropy loss

Lift and lift bias. Imagine that we have two different models to predict the default of a loan. How do we tell which one is better at capturing the relationship between the target variable, Y, and a particular factor, say, F_1? Intuitively, the model should capture directional relationships between Y and F_1. In particular, if we consider the binary-valued factor F_1, we can calculate the observed lift of this factor

$$Lift = \frac{\text{Default Rate}|F_1 = 1}{\text{Default Rate}|F_1 = 0}$$

and compare this to the predicted lift from the model

$$pLift = \frac{\mathbb{P}(Default|F_1 = 1)}{\mathbb{P}(Default|F_1 = 0)}.$$

This transforms the problem from the space of default probability to the space of the lift of a particular factor. If a model is interpretable and captures the relationship between the default probability and F_1 well, $pLift$ should be close to $Lift$. In this space, the overall lift bias of a model (as a percentage) is defined to be:

$$\text{Overall Lift Bias} = 100\% \cdot \left(\frac{pLift}{Lift} - 1\right).$$

However, just looking at the overall bias of a lift model is obviously inadequate because the positive and negative errors from different samples can cancel out. This problem is particularly concerning for underlying dynamics that are heterogeneous across samples. For example, if the model over-predicts the lift for all loans from male applicants and under-predicts the lift for all loans from female applicants, the Overall Lift Bias may still be small. Therefore, we need a better metric to evaluate how well the model captures $Lift$, preferably at the level of each loan i.

SSS: a sample-level measure of lift. In order to evaluate $pLift$ of the model at the sample level, we need to define an observable binary outcome whose probability depends on the lift but not on the raw prediction of the model. Our methodology is based on a simple idea of conditioning in Bayesian statistics.

Consider all loans that have defaulted. Each default either came from a loan where a particular factor, F_1, is active (1) or inactive (0). This is a binary outcome that we actually observe. Furthermore, the probability that a given default came from a loan where the factor is active depends only on the lift and the marginal probability that the factor is active. This can be shown using Bayes' rule:

$$\begin{aligned} \mathbb{P}(F_1 = 1|Default) &= \frac{\mathbb{P}(F_1 = 1)\mathbb{P}(Default|F_1 = 1)}{\mathbb{P}(F_1 = 1)\mathbb{P}(Default|F_1 = 1) + \mathbb{P}(F_1 = 0)\mathbb{P}(Default|F_1 = 0)} \\ &= \frac{\pi \cdot \mathbb{P}(Default|F_1 = 1)/\mathbb{P}(Default|F_1 = 0)}{\pi \cdot \mathbb{P}(Default|F_1 = 1)/\mathbb{P}(Default|F_1 = 0) + (1-\pi)} \\ &= \frac{\pi \cdot Lift}{\pi \cdot Lift + (1-\pi)} \end{aligned} \tag{3}$$

where π is the marginal probability that a factor is active, which can be easily observed from data, or set to 0.5 when bucketing continuous features to create the binary factor F_1.

We call the (conditional) probability in (3) the signal success share, or SSS in short, because an active factor ($F_1 = 1$) can be treated as a signal and a default ($Y_i = 1$) can be treated as a "success" event.

Now, for each observed loan default, we define

$$p_i(k) = \frac{\pi \cdot pLift_i(k)}{\pi \cdot pLift_i(k) + (1-\pi)}, \tag{4}$$

to be the predicted SSS, or the predicted probability of a loan having an active k-th factor given a default, where

$$pLift_i(k) = \mathbb{P}(Default|F_k = 1)/\mathbb{P}(Default|F_k = 0)$$

is the predicted lift for the k-th factor. This has transformed the problem of evaluating lift to a standard binary prediction problem where the prediction is given by the predicted SSS in (4), and the true labels are whether the k-th factor of a defaulted loan is active, given by $F_{k,i} = 1$.

SSS cross-entropy loss. To motivate our loss function of the sample-level metric SSS, we first review how to measure the goodness of fit for a general binary classifier, such as the model for predicting loan default in our example. If Y_i is an observed binary outcome (0 or 1), one common approach to evaluate a model for predicting $p_i = \mathbb{P}(Y_i = 1)$ is to calculate the cross-entropy loss, or the log loss, for that prediction:

$$LogLoss_i = Y_i \log(p_i) + (1 - Y_i)\log(1 - p_i).$$

The cross-entropy loss has strong roots in information theory and is widely adopted when training models of binary labels, such as the logistic regression. In particular, it penalizes the predictions for each *individual* binary outcome and is, therefore, sensitive to heterogeneous underlying dynamics and predictions.

The cross-entropy loss can be used to evaluate models for lift. Because lift is one component of the overall loan default model, one way to evaluate a lift model is simply to measure how good the overall model is at predicting defaults. The problem with this approach is that the lift is confounded with other features in the model. We want to evaluate how good the lift model is, thereby specifically measuring interpretability with regard to one feature, independent of how good the rest of the overall model is.

Therefore, we evaluate a lift model and, therefore, how well the model captures the relationship with all factors, by calculating the SSS cross-entropy loss:

$$\text{SSS cross-entropy loss} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{n}\sum_{i=1}^{n}\big(-F_{k,i}\log p_i(k) - (1-F_{k,i})\log(1-p_i(k))\big) \quad (5)$$

where n is the total number of defaulted loans. We emphasize that although the loss in (5) is defined by combining the cross-entropy loss for all N factors in the overall SSS cross-entropy loss, one can also break down the loss for each factor depending on the specific application. We report the SSS cross-entropy loss for individual factors in our analysis in both Sects. 3 and 4.

3 Evaluating models with heterogeneous predictions

In this section, we show that the SSS cross-entropy loss can indeed provide interpretations for how well a model captures heterogeneous predictions. We also provide more intuition on how it is different from traditional loss functions for the overall prediction model. We use a simulated dataset because, to validate these intuitions, we need to know the ground truth of the underlying dynamics (1).

3.1 Simulation setup

Underlying dynamics. Suppose the default probability of a loan is determined by N factors and their interactions through a logistic function:

$$\mathbb{P}(Default) = logistic\left(\sum_{k=1}^{N}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i} + \epsilon_i\right),$$

and the actual default Y_i (zero or one) is a Bernoulli draw with this probability:

$$Y_i \sim \text{Bernoulli}\left(logistic\left(\sum_{k=1}^{N}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i} + \epsilon_i\right)\right). \quad (6)$$

Here factors $F_1, \ldots, F_N$ are (potentially correlated) Bernoulli random variables with equal probability to be 0 or 1.[4] The beta coefficients are uniform random variables between -1 and 1, $\epsilon_i \sim N(0, 1)$, and $logistic(x) = (1 + \exp(-x))^{-1}$. All values are IID (independently and identically distributed) draws across loans $i = 1, \ldots, M$. We simulate $M = 50,000$ loans

[4] The assumption that factors are binary is non-essential for our framework. Continuous factors can be discretized into buckets as demonstrated in the empirical example in Sect. 4.

in our analysis in Sects. 3.2–3.5 and $M = 10,000$ and $1,000$ loans in Sect. 3.6 to validate that the methodology works for a small number of samples.

We emphasize that the factors $F_1, \ldots, F_N$ can potentially be correlated. We use Gaussian copula (Emrich and Piedmonte, 1991; Li, 2000) to simulate correlated binary factors, which is a technique widely used in the literature (Fernandez et al., 2012; Das et al., 2018; Siah et al., 2021). Given the marginal probability π that a factor is active (with value one), and a $N \times N$ positive definite correlation matrix, $\Sigma \in S^N_{++}$, with pairwise correlations ρ, we first draw a random column vector $\nu = [\nu_1, \ldots, \nu_N]'$ where ν_k's are independent standard normal random variables following $N(0, 1)$. Next we compute $z = \Sigma^{1/2}\nu$, where $\Sigma^{1/2}$ denotes the Cholesky decomposition of Σ. The correlated random variables z then follow $z \sim N(0, \Sigma)$. Finally, we generate the binary factors as $F_k = \mathrm{I}\{\Phi(z_k) > 1 - \pi\}$ where Φ is the cumulative distribution function of the standard normal distribution, so that the marginal probability π is preserved.[5]

Models. Suppose that we have black-box models that capture some, but not all, relationships of the underlying dynamics (6). For example:

$$\hat{\mathcal{F}}_1(\cdot) = \beta_{1,i} F_{1,i} \tag{7}$$

$$\hat{\mathcal{F}}_2(\cdot) = \beta_{1,i} F_{1,i} + bias \tag{8}$$

$$\hat{\mathcal{F}}_3(\cdot) = \beta_{1,i} F_{1,i} \cdot bias \tag{9}$$

$$\hat{\mathcal{F}}_4(\cdot) = \beta_{1,i} F_{1,i} + \beta_{2,i} F_{2,i} \tag{10}$$

$$\hat{\mathcal{F}}_5(\cdot) = \beta_{1,i} F_{1,i} + \beta_{3,i} F_{3,i} + \beta_{5,i} F_{5,i} \tag{11}$$

$$\hat{\mathcal{F}}_6(\cdot) = \epsilon_i \tag{12}$$

$$\hat{\mathcal{F}}_7(\cdot) = \text{Trees or Neural Networks} \tag{13}$$

$$\cdots \tag{14}$$

$$\hat{\mathcal{F}}_8(\cdot) = \beta_{1,i} F_{1,i} + \cdots + \beta_{N,i} F_{N,i}. \tag{15}$$

Here (7) captures the first factor but not the others. (8) and (9) capture the first factor but with an additive and multiplicative bias respectively. (10) and (11) capture a subset of all factors. (12) only captures the noise and is unlikely to generalize. (13) can be a tree-based model or neural network. (15) successfully captures the full model.

Our goal in this paper is not to develop methods to come up with such models, but rather to develop a framework and a metric that uncover the relationships they have captured, given these (potentially black-box) models.

Outline of validation. In the following sections, we progressively consider three different sub-cases of true underlying dynamics (6), and apply our framework to study a range of models that capture some or all relationships of the true dynamics. The first is homogeneous (non-heterogeneous) predictions which serve as a baseline:

$$Y_i \sim \text{Bernoulli}\left(logistic\left(\beta_1 F_{1,i} + \cdots + \beta_N F_{N,i} + \epsilon_i\right)\right), \tag{16}$$

in which $\beta_1, \ldots, \beta_N$ are homogeneous across all loans, indicating that the mechanism through which the factors affect the default is the same across all loans. In this scenario, traditional methods like regression work well. Does our method work?

[5] Strictly speaking, the correlation matrix Σ is the correlation matrix of latent variable z and not the correlation matrix of F. However, a higher pairwise correlation for z corresponds to a higher pairwise correlation for F, so it suffices for us to model different levels of dependence between factors. This specification of correlation is consistent with the previous literature on the simulation of correlated binary outcomes; see, for example, Table 2 of Siah et al. (2021).

The second is heterogeneous predictions:

$$Y_i \sim \text{Bernoulli}\left(logistic\left(\beta_{1,i}F_{1,i} + \cdots + \beta_{N,i}F_{N,i} + \epsilon_i\right)\right). \tag{17}$$

Here each loan has a different set of β's, indicating that the mechanism through which the factors affect the default is different for different loans. This happens for many problems in practice because such mechanisms can be confounded with other unobserved factors, such as the instrumented principal component analysis model of Kelly et al. (2019) in the context of asset pricing. Traditional methods break down when dealing with highly heterogeneous predictions. If we employ more complicated machine learning models for such relationships, does our framework provide guidance on what relationships the model has or has not captured?

The third is predictions with nonlinear terms:

$$Y_i \sim \text{Bernoulli}\left(logistic\left(\sum_{k=1}^{N}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i} + \epsilon_i\right)\right). \tag{18}$$

Here the model captures nonlinear interactions between factors. Does our framework uncover these relationships?

We answer these three questions in the following sections. In particular, the goal of the simulation analysis is to establish a baseline that, when models progressively know more about the underlying relationships, the SSS cross-entropy loss can reflect that information. Therefore, we will test different oracle models that know different levels of ground truth and see if our proposed loss function can capture how much each model knows.

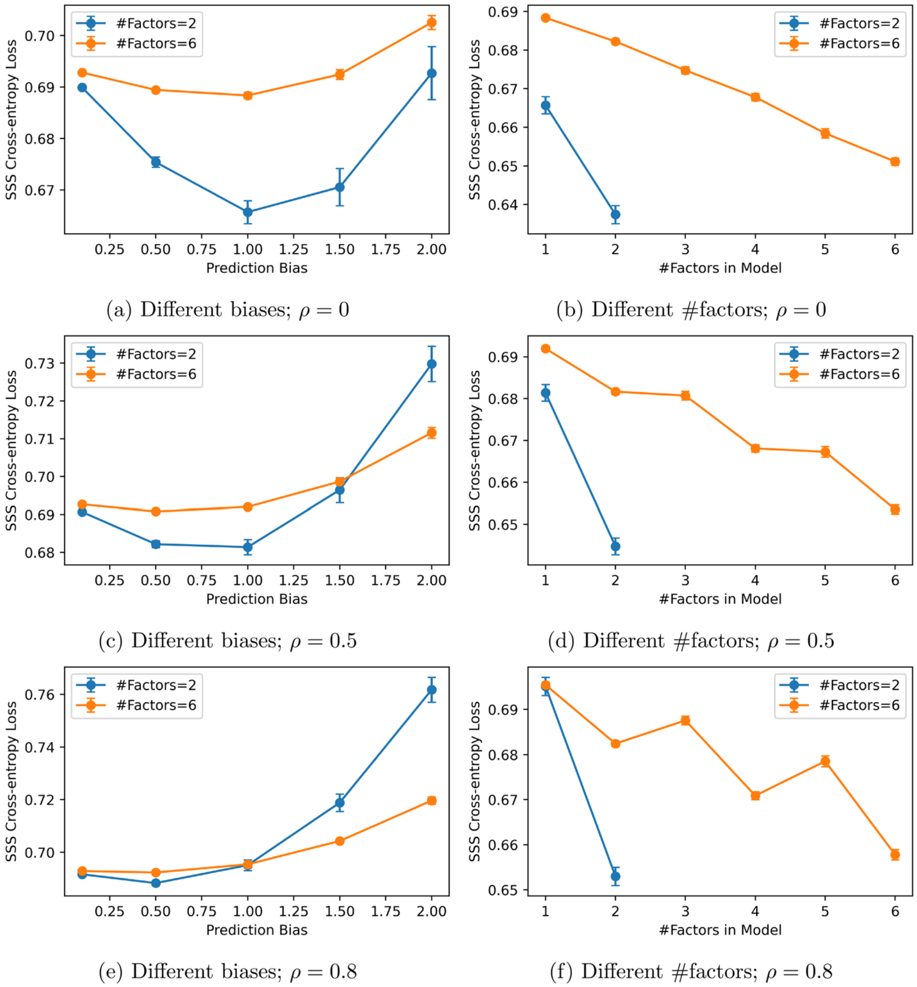
3.2 Homogeneous predictions

Suppose the underlying true dynamics for a loan default are homogeneous following (16). In other words, the beta coefficients are the same across all loans, and traditional methods like regression can recover the true relationships consistently. Therefore, it is a basic requirement for any metric to recover the true relationships in this case, as a sanity check.

In Fig. 1, c, e, for three different levels of correlation $\rho = 0$, 0.5, and 0.8, we show the SSS cross-entropy loss for a class of models:

$$\hat{\mathcal{F}}(\cdot) = bias \cdot \beta_1 F_{1,i} \tag{19}$$

where $bias$ ranges between 0 and 2. Here different models $\hat{\mathcal{F}}$ give different predictions based on the first factor. The blue line shows a case with $N = 2$ true factors and the orange line shows a case with $N = 6$ true factors. In both cases the true model $\hat{\mathcal{F}}$ only observes a partial set of one factor. When the correlation ρ between factors is not too high, the SSS cross-entropy loss is indeed minimized when the true relationship is recovered ($bias = 1$). When the correlation ρ is very high, it becomes increasingly difficult to recover the true beta based on the SSS cross-entropy loss, which is a common challenge for all models with highly collinear explanatory variables.



(a) Different biases; $\rho = 0$

(b) Different #factors; $\rho = 0$

(c) Different biases; $\rho = 0.5$

(d) Different #factors; $\rho = 0.5$

(e) Different biases; $\rho = 0.8$

(f) Different #factors; $\rho = 0.8$

Fig. 1 The three figures on the left show the SSS cross-entropy loss (y-axis) for a class of models in (19), where *bias* ranges from 0 to 2 (x-axis). The three figures on the right show the SSS cross-entropy loss (y-axis) for a class of models in (20), where more relationships are captured progressively (x-axis). The correlation between factors $\rho = 0$, 0.5, and 0.8

In Fig. 1b, d, f, for three different levels of correlation $\rho = 0$, 0.5, and 0.8, we compare several models that progressively capture more relationships:

$$\begin{cases} \hat{\mathcal{F}}_1(\cdot) = \beta_1 F_{1,i} \\ \hat{\mathcal{F}}_2(\cdot) = \beta_1 F_{1,i} + \beta_2 F_{2,i} \\ \cdots \\ \hat{\mathcal{F}}_N(\cdot) = \beta_1 F_{1,i} + \cdots + \beta_N F_{N,i}. \end{cases} \tag{20}$$

The horizontal axis corresponds to the number of factors captured in the model. In general, the SSS cross-entropy loss decreases when more relationships are recovered, as expected. When the correlation ρ between factors becomes higher, the decrease in loss is not as smooth as the case when the correlation is low, but optimizing the model using this loss function should still guide the model towards recovering more true factors.

These two examples demonstrate that the SSS cross-entropy loss provides the same recovery of the true underlying factors and their betas as linear models when the prediction is homogeneous. The recovery is robust when some factors are unobserved by the model, and is the strongest when the correlation between factors is not too high.

3.3 Heterogeneous predictions

Next, we consider heterogeneous underlying dynamics following (17). In other words, the beta coefficients are different for different loans, which is a more realistic description of many real-world applications because of unobserved confounding factors to the features in the model. Traditional linear models break down with highly heterogeneous relationships, and more complicated machine learning models are necessary to capture such relationships. The real strength of our framework becomes evident in providing interpretability in this case.

Similar to the previous section, Fig. 2a, c, e compare a class of models that progressively captures more heterogeneous relationships for three different levels of correlation:

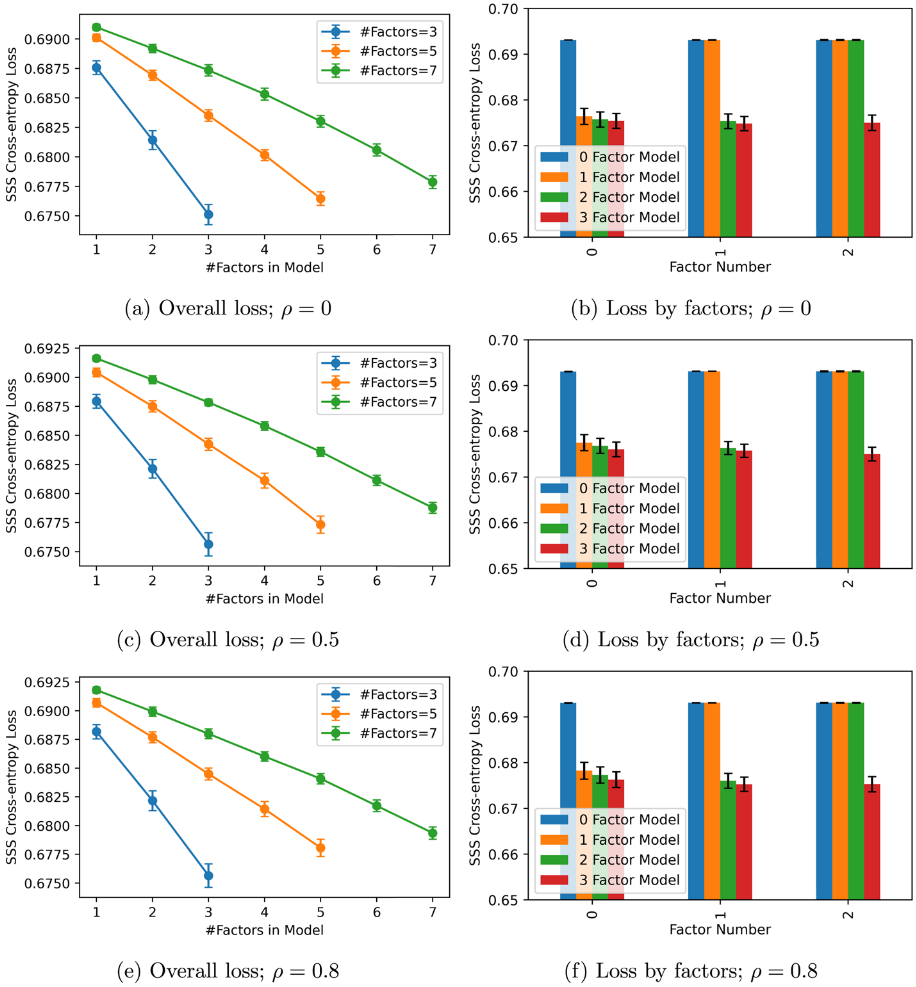
$$\begin{cases} \hat{\mathcal{F}}_1(\cdot) = \beta_{1,i} F_{1,i} \\ \hat{\mathcal{F}}_2(\cdot) = \beta_{1,i} F_{1,i} + \beta_{2,i} F_{2,i} \\ \quad \cdots \\ \hat{\mathcal{F}}_N(\cdot) = \beta_{1,i} F_{1,i} + \cdots + \beta_{N,i} F_{N,i} \end{cases} \tag{21}$$

The blue (orange, green) line corresponds to the case with $N = 3$ (5, 7) true underlying factors, and the horizontal axis corresponds to progressively more factors captured by the model. As desired, SSS cross-entropy loss decreases when more heterogeneous effects are recovered. In other words, optimizing for the SSS cross-entropy loss will eventually lead to models that capture the full underlying dynamics, and the models are interpretable in the sense that the SSS cross-entropy loss guides them to capture the relationship between the target variable and each factor as much as possible.

This simple example demonstrates that the SSS cross-entropy loss can capture heterogeneous relationships. In fact, the β's in our simulated dataset follows the uniform distribution from -1 to 1, and therefore the average relationship for any factor is close to 0. Traditional methods like the partial dependence plot (Friedman, 2001) would yield zero marginal relationships and fail to measure the heterogeneity that the model has captured.

Although the aggregate SSS cross-entropy loss shown in Fig. 2a, c, e tells us which model captures more relationships, it does not explicitly guide the model developer towards which *particular factor* the model has captured well (or not well). Thanks to the construction of SSS cross-entropy loss in (5), we can in fact break down this metric by each factor.

As an example, we consider a true default dynamic with three heterogeneous relationships that corresponds to $N = 3$ in (17), and break down the SSS cross-entropy loss by each factor for models in (21). The results are summarized in Figs. 2b, d, f for three different levels of correlation ρ. Different colors represent models that capture different numbers of factors, ranging from 0 to 3. For example, the blue bars suggest that the zero-factor model has a high loss for all factors; the orange bars suggest that the one-factor model has a low loss for the 0-th factor and a high loss for the other two factors; the red bars suggest that the three-factor model has a low loss for all factors.



(a) Overall loss; $\rho = 0$

(b) Loss by factors; $\rho = 0$

(c) Overall loss; $\rho = 0.5$

(d) Loss by factors; $\rho = 0.5$

(e) Overall loss; $\rho = 0.8$

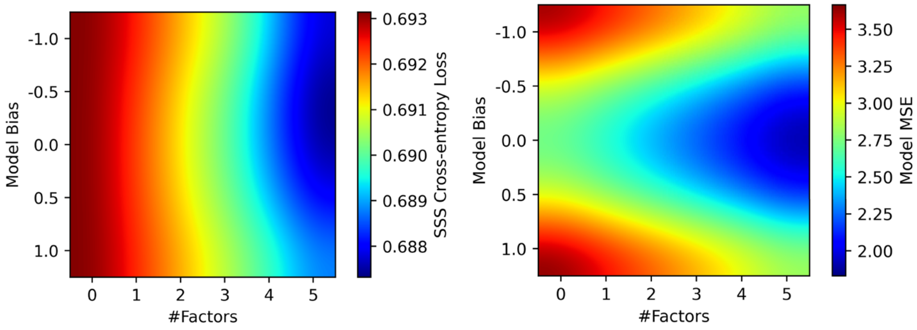
(f) Loss by factors; $\rho = 0.8$

Fig. 2 The three figures on the left show the SSS cross-entropy loss (y-axis) for a class of models in (21), where more relationships are captured progressively (x-axis). The three figures on the right show the SSS cross-entropy loss (y-axis) broken down by factors (x-axis) for a class of models that capture more relationships progressively. The correlation between factors $\rho = 0, 0.5$, and 0.8

This example demonstrates that the SSS cross-entropy loss can successfully point out that a particular relationship is well captured by the model, while another relationship is not. The ability to break down the loss by each factor provides actionable guidance for model improvements, and traditional loss functions like MSE for the original model $\hat{\mathcal{F}}(\cdot)$ cannot achieve this.

3.4 Difference from traditional loss

To provide more intuition and highlight the differences between the SSS cross-entropy loss and traditional loss functions like MSE, we consider the same heterogeneous underlying



(a) SSS cross-entropy loss

(b) Mean Squared Error

Fig. 3 SSS cross-entropy loss (**a**) and MSE (**b**) for a class of models in (22), where more relationships are captured progressively (x-axis), and model bias varies from -1 to 1 (y-axis)

dynamics as (17) with ten factors:

$$Y_i \sim \text{Bernoulli}\left(logistic\left(\beta_{1,i}F_{1,i} + \cdots + \beta_{10,i}F_{10,i} + \epsilon_i\right)\right).$$

We evaluate the SSS cross-entropy loss of the following models:

$$\begin{cases} \hat{\mathcal{F}}_1(\cdot) = bias + \beta_{1,i}F_{1,i} \\ \hat{\mathcal{F}}_2(\cdot) = bias + \beta_{1,i}F_{1,i} + \beta_{2,i}F_{2,i} \\ \quad \cdots \\ \hat{\mathcal{F}}_N(\cdot) = bias + \beta_{1,i}F_{1,i} + \cdots + \beta_{N,i}F_{N,i} \end{cases} \tag{22}$$

which progressively capture more relationships, and have a potentially non-zero bias term.

Figure 3a shows the SSS cross-entropy loss as the number of factors from the model increases from 0 to 5, and as the model bias varies between -1 and 1. We can see that the horizontal direction is the main axis of variation, which suggests that the SSS cross-entropy loss predominantly measures the strength of relationships the model has captured, but not the bias in the model itself. This reflects precisely the design of the SSS metric in 3 which is intended to capture the *lift* of model predictions with respect to a factor rather than the *level* of model predictions.

In contrast, Fig. 3b shows MSE for the same set of models, which varies along both directions. This sharp difference demonstrates that traditional loss functions like MSE would prefer, for example, a model with no bias but a small number of learned factor relationships, to a model with many learned factor relationships but some bias. On the other hand, the SSS cross-entropy loss only focuses on one dimension, which is how well the model has captured the relationship between the target variable and explanatory variables.

In other words, if we have an over-fitting model that only captures noise:

$$\hat{\mathcal{F}}(\cdot) = \epsilon_i,$$

MSE would still indicate that the model is good because it decreases loss on the training data, but the SSS cross-entropy loss would not because the model did not capture any meaningful relationships between the target variable and explanatory variables. This is a fundamental difference between the SSS cross-entropy loss and traditional loss functions which makes the former suitable for measuring model interpretability.

We would like to emphasize that we do not intend to suggest that traditional loss functions like MSE are inferior to our proposed SSS cross-entropy loss here. In fact, they are widely adopted and useful metrics to measure the *overall performance* of model predictions, which is essential to train models that fit the data well. Instead, we are highlighting the difference between the two classes of loss functions. The SSS cross-entropy loss provides a useful supplement to MSE in that it provides a different dimension of information from MSE, and that information can help the model to be more interpretable by measuring how well the relationship between the target variable and each factor is captured. In practice, we suggest using traditional loss functions like MSE as the primary metric for model training, while using the SSS cross-entropy loss either as auxiliary constraints or diagnostic metrics to guide model improvements. We demonstrate an example using real data in Sect. 4.

3.5 Predictions with nonlinear terms

Finally, we consider the case where the underlying dynamic is nonlinear with respect to the factors. As a special case of nonlinear relationships, suppose the true underlying default dynamics include interactions of the factors following (18), and we evaluate the following models with and without the nonlinear terms:

$$\begin{cases} \hat{\mathcal{F}}_1(\cdot) = \beta_{1,i}F_{1,i} \\ \quad \cdots \\ \hat{\mathcal{F}}_N(\cdot) = \beta_{1,i}F_{1,i} + \cdots + \beta_{N,i}F_{N,i} \\ \hat{\mathcal{F}}_{2,Interact}(\cdot) = \beta_{1,i}F_{1,i} + \beta_{2,i}F_{2,i} + \beta_{1,2,i}F_{1,i}F_{2,i} \\ \quad \cdots \\ \hat{\mathcal{F}}_{N,Interact}(\cdot) = \sum_{k=1}^{N}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i}. \end{cases} \tag{23}$$

Fig. 4a, c, e compare the SSS cross-entropy loss for models with and without nonlinear relationships. The blue (orange) line corresponds to the case with $N = 3$ (5) true underlying factors, and the solid (dashed) version corresponds to models with (without) interaction terms. Similar to Sect. 3.3, as the number of factors in the model (horizontal axis) progressively increases, the SSS cross-entropy loss decreases. More importantly, our loss also decreases when more nonlinear interactions are recovered, a feature that would be desired for machine learning models.

In particular, we consider the default dynamic with three true underlying factors:

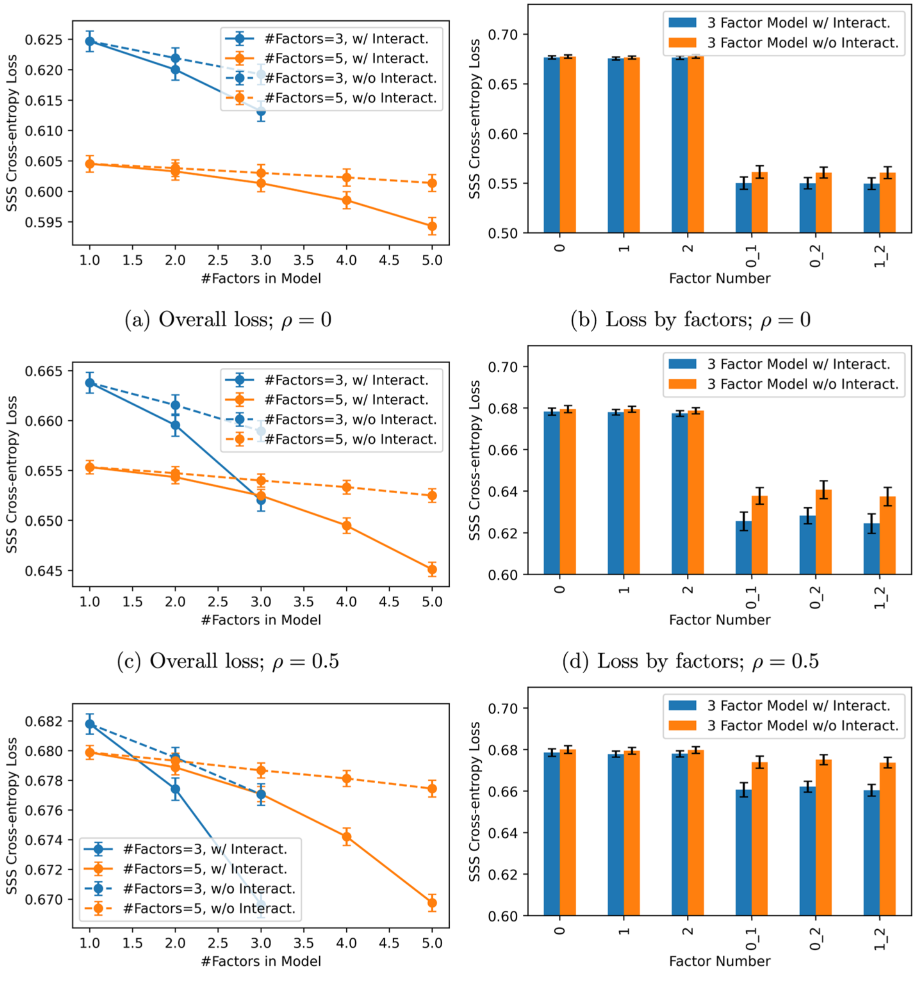
$$Y_i = \sum_{k=1}^{3}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i} + \epsilon_i$$

and two models with and without the nonlinear terms:

$$\hat{\mathcal{F}}_1(\cdot) = \sum_{k=1}^{3}\beta_{k,i}F_{k,i}$$

$$\hat{\mathcal{F}}_2(\cdot) = \sum_{k=1}^{3}\beta_{k,i}F_{k,i} + \sum_{k\neq l}\beta_{k,l,i}F_{k,i}F_{l,i}.$$

We again break down the SSS cross-entropy loss by each factor and each interaction term, shown in Figs. 4b, d, f. Blue bars represent the model that captures the interactions, and orange bars represent the model that does not capture the interactions. The SSS cross-entropy loss



(a) Overall loss; $\rho = 0$

(b) Loss by factors; $\rho = 0$

(c) Overall loss; $\rho = 0.5$

(d) Loss by factors; $\rho = 0.5$

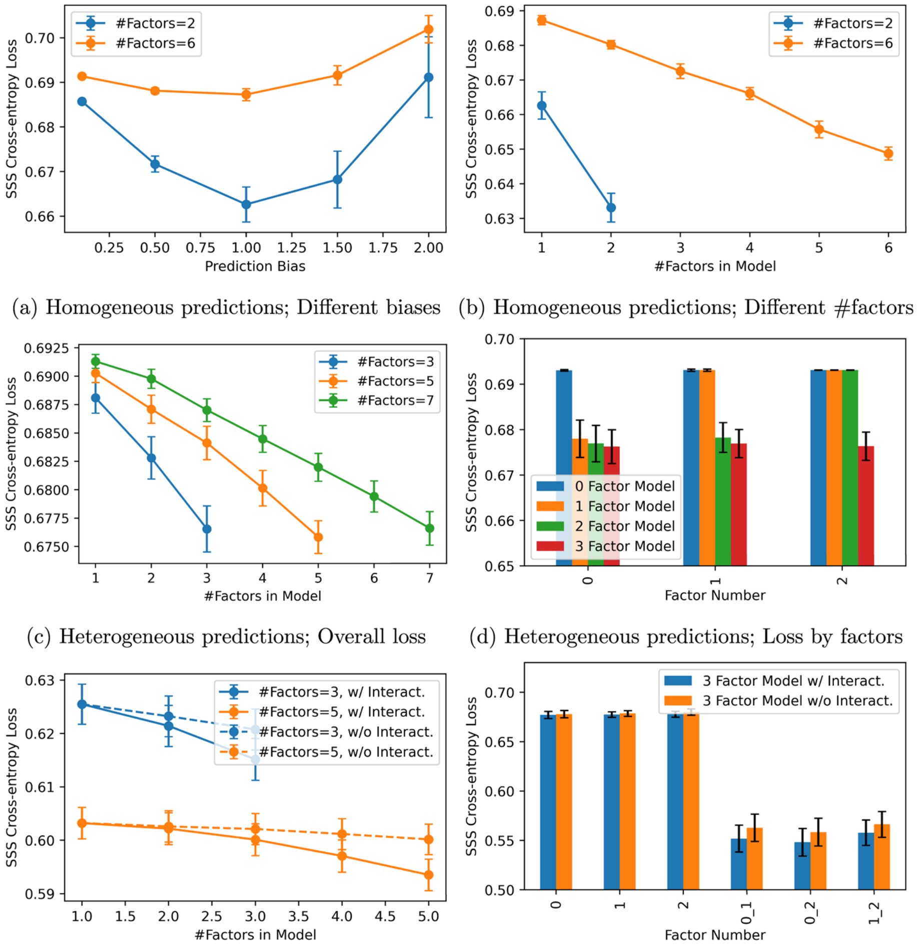
(e) Overall loss; $\rho = 0.8$

(f) Loss by factors; $\rho = 0.8$

Fig. 4 The three figures on the left show the SSS cross-entropy loss (y-axis) for a class of models in (23), where more relationships are captured progressively (x-axis), with and without interactions (solid and dashed lines). The three figures on the right show the SSS cross-entropy loss (y-axis) broken down by factors and interactions (x-axis) for the same class of models in (23). The correlation between factors $\rho = 0, 0.5$, and 0.8
a

for interactions is indeed higher for the model without interactions, and the loss for the marginal relationship of each factor is the same. This implies that the SSS cross-entropy loss can distinguish which interaction the model has not captured, again providing actionable guidance for model improvements in the nonlinear case.

To summarize, we have demonstrated that our methodology provides a measure to evaluate how well models capture relationships between the target variable and explanatory variables, for both heterogeneous and nonlinear relationships. The measure works with correlated factors because it depends on the lift of model predictions from each factor. Although correlations between factors will affect the specific level of the SSS cross-entropy loss, they



(a) Homogeneous predictions; Different biases (b) Homogeneous predictions; Different #factors

(c) Heterogeneous predictions; Overall loss (d) Heterogeneous predictions; Loss by factors

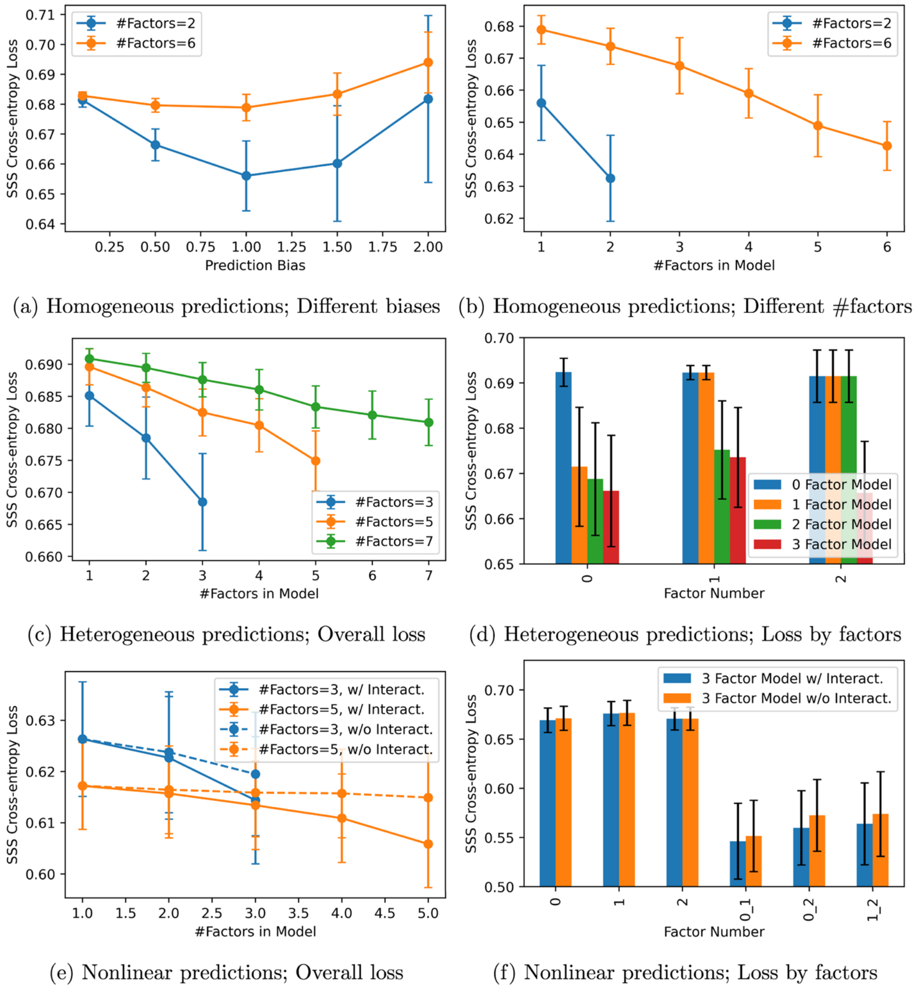
(e) Nonlinear predictions; Overall loss (f) Nonlinear predictions; Loss by factors

Fig. 5 The SSS cross-entropy loss for the analysis in Sects. 3.2–3.5 on a synthetic dataset with 10, 000 samples and $\rho = 0$

do not change the fact that the loss decreases when the model improves its ability to capture the lift from a specific factor. As a result, this provides a new tool to interpret black-box models.

3.6 Influence of number of samples

In this section, we repeat the previous analysis for synthetic datasets with a smaller number of samples. Figure 5 shows the results on a dataset with 10,000 samples, including the scenarios of homogeneous predictions (Fig. 5a, b), heterogeneous predictions (Fig. 5c, d), and nonlinear interactions (Fig. 5e, f). The results are very similar to those in Sects. 3.2–3.5, except that the bootstrap confidence intervals are wider. However, the SSS cross-entropy losses are still



(a) Homogeneous predictions; Different biases (b) Homogeneous predictions; Different #factors

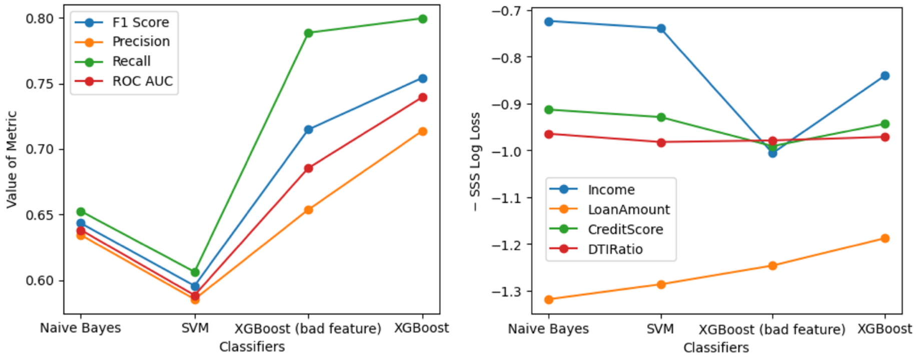
(c) Heterogeneous predictions; Overall loss (d) Heterogeneous predictions; Loss by factors

(e) Nonlinear predictions; Overall loss (f) Nonlinear predictions; Loss by factors

Fig. 6 The SSS cross-entropy loss for the analysis in Sects. 3.2–3.5 on a synthetic dataset with 1000 samples and $\rho = 0$

different enough in most cases to distinguish whether a particular relationship is captured by the model.

Furthermore, Fig. 6 stress-tests our results on an even smaller dataset with only 1,000 samples. The results are still very similar to those in Sects. 3.2–3.5 in terms of point estimates, but the bootstrap confidence intervals are much wider, as expected. While the SSS cross-entropy losses are still useful to evaluate how well a marginal relationship is captured by the model in most cases, it becomes challenging to distinguish interactions after accounting for uncertainties in the estimate of the loss function. This highlights that users should be cautious with implications from such analysis on very small datasets, and additional techniques such as data augmentation may be used to mitigate this issue.



(a) Model performance metrics

(b) SSS cross-entropy loss

Fig. 7 Model performance metrics and the SSS cross-entropy loss for the loan default example with real data

4 Application to real data

We use the publicly available Kaggle Loan Default Prediction Dataset [6] for our analysis. The data contains individual loans with information on whether the loan defaulted (1) or not (0), and over a dozen loan characteristics. The loan data is heavily imbalanced with a small fraction of defaults compared with non-defaults. We apply the popular synthetic minority over-sampling technique (SMOTE) technique (Chawla et al., 2002) to derive a balanced dataset of 21,272 loans with approximately half defaults and half non-defaults. We randomly split the data into five folds and apply cross-validation to train all models. We always report metrics on the test dataset.

Following previous studies using the same dataset (Moula et al., 2017; Sheikh et al., 2020; Madaan et al., 2021), we include four explanatory variables in our models as a proof of concept because they are found to be the most useful for predicting the default of loans. The four chosen features are the income, credit score, and debt-to-income (DTI) ratio of the borrower, and the dollar amount of the loan. All features are standardized to have zero mean and unit variance before feeding to the model.

We train three different types of models using this dataset, including the naive Bayes classifier, support vector machines (SVM) (Boser et al., 1992), and boosting trees as implemented by XGBoost (Chen and Guestrin, 2016). In order to demonstrate that the SSS cross-entropy loss measures how well a model captures relationships with each feature and, more importantly, how it provides researchers with guidance on model improvements, we first train the naive Bayes classifier and SVM with all four features. However, we train XGBoost using only three real features, and the other feature, income of the borrower, is replaced by a random feature following the uniform distribution from −2 to 2. We hope that the SSS cross-entropy loss can highlight that XGBoost is not learning the relationship with respect to income well and, therefore, guide the researcher to find out that efforts need to be invested to improve this particular feature.

Figure 7a shows several performance metrics for the models above including the F1 score, precision, recall, and the area under the receiver operating characteristic curve (ROC AUC). The horizontal axis represents different models. The XGBoost (bad feature) represents the XGBoost model mentioned above, which has the best performance among the three models

[6] https://www.kaggle.com/datasets/nikhil1e9/loan-default/data. Accessed 15 Nov, 2023.

although it is missing one feature, thanks to its strong learning capabilities. From these performance metrics alone, it is difficult for researchers to realize any problem with the XGBoost model.

We then compute the SSS cross-entropy loss of the three models with respect to each feature. Although the models are trained using features with continuous values, we discretize each variable when computing the SSS cross-entropy loss by setting all feature values above its median to be one, and feature values below its median to be zero. As a result, SSS captures the lift of each feature in terms of its values in the top half compared with values in the bottom half.[7]

Figure 7b shows the SSS cross-entropy loss with respect to each feature, which provides useful supplementary information in addition to the performance metrics in Fig. 7a. Although the XGBoost (bad feature) model achieves very good performances, the blue line shows that it has a low SSS cross-entropy loss for the income feature compared to other models, which implies that it is very poor in capturing the relationship with that feature as compared to other models. This is not surprising because the XGBoost (bad feature) model is not able to use the true income feature in the first place. With this information, researchers should immediately investigate potential issues for XGBoost and focus on the income feature.

Finally, we add the true income feature back for XGBoost and train a new model. The results are shown in the last column in both subfigures in Fig. 7. Not surprisingly, this improves the performance of XGBoost even further in terms of the ROC AUC, and the SSS cross-entropy loss reflects this improvement.

5 Conclusion

AI and machine learning have made significant progress and impact on society. In particular, they have shown great success in learning complex patterns. In addition to using these models for prediction, the ability to interpret what a model has learned is receiving increasing attention. Interpretable models have the benefit of being both concise and convincing. In practice, an interpretable model can help domain experts troubleshoot the inner workings of a complex model to improve its accuracy and domain applicability.

Based on a simple idea from conditioning in Bayesian statistics, here we provide a framework to improve the interpretability of any black-box model, which is crucial to unlock the power of AI in finance and economics. We demonstrate the effectiveness of our framework and the new metric, signal success share (SSS) cross-entropy loss, through extensive simulation and an application with real data. The framework is developed and validated using binary explanatory variables and therefore works best under this scenario. We demonstrate a simple example of how to transform continuous variables into binary ones, but we emphasize that further work should be done to fully understand the generalizability to continuous variables.

Although we develop our methodology in the context of predicting loan defaults, our methodology is more generally applicable to any model involving a binary prediction, including insurance claims, school applications, click-through-rate prediction in online advertising, and many more problems in financial and information technology.

[7] More generally, researchers can define their own discretization scheme if they are more interested in the lift in other regions of the feature. It is worth noting that researchers only need to perform this discretization for each factor independently, rather than all factors jointly. In other words, this is fairly realistic in practice because they do not need data to cover all combinations of the binary discretization of all factors.

Acknowledgements The authors thank Omkar Muralidharan, Dennis Sun, Kathy Zhong, and seminar participants at New York University for helpful comments and discussion. Ruixun Zhang's research is supported by the National Key R&D Program of China (2022YFA1007900), the National Natural Science Foundation of China (12271013,72342004), and the Fundamental Research Funds for the Central Universities (Peking University).

References

Ahmad, M. A., Eckert, C., & Teredesai, A. (2018). Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics* (pp. 559–560).

Arras, L., Horn, F., Montavon, G., Müller, K.-R., & Samek, W. (2017). What is relevant in a text document?": An interpretable machine learning approach. *PloS one*, *12*.

Bertsimas, D., Dunn, J., & Mundru, N. (2019). Optimal prescriptive trees. *INFORMS Journal on Optimization, 1*, 164–183.

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science, 66*, 1025–1044.

Bisias, D., Flood, M., Lo, A. W., & Valavanis, S. (2012). A survey of systemic risk analytics. *Annual Review of Financial Economics, 4*, 255–296.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual workshop on computational learning theory* (pp. 144–152).

Campbell, J. Y. (2006). Household finance. *The Journal of Finance, 61*, 1553–1604.

Campbell, J. Y., & Cocco, J. F. (2015). A model of mortgage default. *The Journal of Finance, 70*, 1495–1554.

Campbell, T. S., & Dietrich, J. K. (1983). The determinants of default on insured conventional residential mortgage loans. *The Journal of Finance, 38*, 1569–1581.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems* (pp. 8928–8939).

Chen, T., & Guestrin, C. (2016) Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Culkin, R., & Das, S. R. (2017). Machine learning in finance: The case of deep learning for option pricing. *Journal of Investment Management, 15*, 92–100.

Das, S., Rousseau, R., Adamson, P. C., & Lo, A. W. (2018). New business models to accelerate innovation in pediatric oncology therapeutics: A review. *JAMA Oncology, 4*, 1274–1280.

Davis, R., Lo, A. W., Mishra, S., Nourian, A., Singh, M., Wu, N., & Zhang, R. (2023). Explainable machine learning models of consumer credit risk. *The Journal of Financial Data Science, 5*, 9–39.

De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Hoboken: Wiley.

De Spiegeleer, J., Madan, D. B., Reyners, S., & Schoutens, W. (2018). Machine learning for quantitative finance: Fast derivative pricing, hedging and fitting. *Quantitative Finance, 18*, 1635–1643.

Emrich, L. J., & Piedmonte, M. R. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician, 45*, 302–304.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine, 25*, 24–29.

Fernandez, J.-M., Stein, R. M., & Lo, A. W. (2012). Commercializing biomedical research through securitization techniques. *Nature Biotechnology, 30*, 964–975.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning. Springer series in statistics New York*. New York: Springer.

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

Giglio, S., Kelly, B., & Xiu, D. (2022). Factor models, machine learning, and asset pricing. *Annual Review of Financial Economics, 14*, 337–368.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT Press.

Hanson, C. W., III., & Marshall, B. E. (2001). Artificial intelligence applications in the intensive care unit. *Critical Care Medicine, 29*, 427–435.

Heaton, J., Polson, N., & Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry, 33*, 3–12.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications, 124*, 226–251.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science, 349*, 261–266.

Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance, 49*, 851–889.

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics, 134*, 501–524.

Keys, B. J., Pope, D. G., & Pope, J. C. (2016). Failure to refinance. *Journal of Financial Economics, 122*, 482–499.

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance, 34*, 2767–2787.

Khandani, A. E., Lo, A. W., & Merton, R. C. (2013). Systemic risk and the refinancing ratchet effect. *Journal of Financial Economics, 108*, 29–45.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.

Li, D. X. (2000). On default correlation: A copula function approach. *Journal of Fixed Income, 9*, 43–54.

Lin, L.-J. (1993). Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science.

Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. In *IOP conference series: Materials science and engineering* (Vol. 1022, p. 012042). IOP Publishing.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529–533.

Molnar, C. (2019). Interpretable machine learning (Lulu. com).

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing, 73*, 1–15.

Moula, F. E., Guotai, C., & Abedin, M. Z. (2017). Credit default prediction modeling: an application of support vector machine. *Risk Management, 19*, 158–187.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives, 31*, 87–106.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences, 116*, 22071–22080.

Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., & Buchman, T. G. (2018). An interpretable machine learning model for accurate prediction of sepsis in the ICU. *Critical Care Medicine, 46*, 547–553.

Patterson, S. (2010). Letting the machines decide. *The Wall Street Journal*, *13*.

Rudin, C. (2014). Algorithms for interpretable machine learning. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1519–1519).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*, 206–215.

Russell, S. J., and P. Norvig, 2016, Artificial Intelligence: A Modern Approach (Malaysia; Pearson Education Limited,).

Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in p2p lending. *PloS One*, *10*.

Sheikh, M. A., Goel, A. K. & Kumar, T. (2020). An approach for prediction of loan approval using machine learning algorithm. In *2020 International conference on electronics and sustainable communication systems (ICESC)*, (pp. 490–494). IEEE.

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering, 19*, 221–248.

Siah, K. W., Xu, Q., Tanner, K., Futer, O., Frishkopf, J. J., & Lo, A. W. (2021). Accelerating glioblastoma therapeutics via venture philanthropy. *Drug Discovery Today, 26*, 1744–1749.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature, 529*, 484.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature, 550*, 354–359.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Berlin: Springer.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579 .

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine, 13*, 55–75.

Zuckerman, G. (2019). The man who solved the market: How Jim Simons launched the quant revolution (Portfolio).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.