Article

Responsible Machine Learning Techniques

Interpretable Models, Post-hoc Explanation, and Discrimination Testing

Navdeep Gill [1,†], Patrick Hall [1,3,†,*], Kim Montgomery [1,†], and Nicholas Schmidt [2,†]

[1] H2O.ai
[2] BLDS, LLC
[3] George Washington University
* Correspondence: phall@h2o.ai; nschmidt@bldsllc.com
† All authors contributed equally to this work.

Version December 18, 2019 submitted to Information

Abstract: This manuscript outlines a viable approach for training and evaluating machine learning (ML) systems for high-stakes, human-centered, or regulated applications using common Python programming tools. The accuracy and intrinsic interpretability of two types of constrained models, monotonic gradient boosting machines (MGBM) and explainable neural networks (XNN), a deep learning architecture well-suited for structured data, are assessed on simulated data with known feature importance and discrimination characteristics and on publicly available mortgage data. For maximum transparency and the potential generation of personalized adverse action notices, the constrained models are analyzed using post-hoc explanation techniques including plots of partial dependence (PD) and individual conditional expectation (ICE) and global and local Shapley feature importance. The constrained model predictions are also tested for disparate impact (DI) and other types of discrimination using adverse impact ratio (AIR), marginal effect (ME), standardized mean difference (SMD), and additional straightforward group fairness measures. By combining interpretable models, post-hoc explanation, and discrimination testing with accessible software tools, this text aims to describe the art of the possible for important ML applications that require high accuracy and interpretability and minimal discrimination.

Keywords: Machine Learning; Neural Network; Gradient Boosting Machine; Interpretable; Explanation; Fairness; Disparate Impact; Python

---

0. Introduction

Responsible artificial intelligence (AI) has been variously conceptualized as AI-based products or projects that use transparent technical mechanisms, that create appealable decisions or outcomes, that perform reliably and in a trustworthy manner over time, that exhibit minimal social discrimination, and that are designed by humans with diverse experiences, both in terms of demographics and professional backgrounds, i.e. ethics, social sciences, and technology.[1] Although responsible AI is today a somewhat broad and amorphous notion, at least one aspect is crystal clear: ML models, a common application of AI, have problems that responsible practitioners should likely attempt to remediate. ML models can be inaccurate and unappealable black-boxes, even with the application of newer post-hoc explanation techniques [1].[2] ML models can perpetuate and exacerbate discrimination

---

[1] See: Responsible Artificial Intelligence, *Responsible AI: A Framework for Building Trust in Your AI Solutions*, PwC's Responsible AI, Responsible AI Practices.
[2] See: *When a Computer Program Keeps You in Jail*.

[2], [3], [4]. ML models can be hacked, resulting in manipulated model outcomes or the exposure of proprietary intellectual property or sensitive training data [5], [6], [7], [8]. While this manuscript makes no claim that the interdependent issues of opaqueness, discrimination, or security vulnerabilities in ML have been solved (even as singular entities, much less as complex intersectional phenomena), Sections 1, 2, and 3 do propose some specific technical countermeasures, in the form of interpretable models, post-hoc explanation, and DI and discrimination testing implemented in widely available, free, and open source Python tools, to address a subset of these vexing problems for high-stakes, human-centered, or regulated ML applications.[3,4]

Section 1 describes methods and materials, including simulated and collected training datasets, interpretable and constrained model architectures, post-hoc explanations used to create an appealable decision-making framework, tests for DI and other social discrimination, and public and open source software resources. In Section 2, interpretable and constrained modeling results are compared to less interpretable and unconstrained models and post-hoc explanation and discrimination testing results are also presented for interpretable models. Section 3 then discusses some nuances of the outlined modeling, explanation, and discrimination testing methods and results. Section 4 closes this manuscript with a brief summary of the outlined methods, materials, results, and discussion.

1. Materials and Methods

Detailed descriptions of notation, training data, ML models, post-hoc explanation techniques, discrimination testing methods, and software resources are organized in Section 1 as follows:

- **Notation**: spaces, datasets, & models – §1.1
- **Training data**: simulated data & collected mortgage data – §1.2 and §1.3
- **ML models**: constrained, interpretable MGBM & XNN models – §1.4 and §1.5
- **Post-hoc explanation techniques**: PD, ICE, & Shapley values – §1.6 and §1.7
- **Discrimination testing methods**: AIR, ME, and SMD – §1.8
- **Software resources**: GitHub repository associated with Sections 1 and 2 – §1.9

To provide a sense of accuracy differences, performance of more interpretable constrained ML models and less interpretable unconstrained ML models is compared on simulated data and collected mortgage data. The simulated data, based on the well-known Friedman datasets and with known feature importance and discrimination characteristics, is used to gauge the validity of interpretable modeling, post-hoc explanation, and discrimination testing techniques [10], [11]. The mortgage data is sourced from the Home Mortgage Disclosure Act (HMDA) database.[5] Because unconstrained ML models, like gradient boosting machines (GBMs) (e.g. [12], [13]) and artificial neural networks (ANNs) (e.g. [14], [15], [16], [17]), can be difficult to understand, trust, and appeal, even after the application of post-hoc explanation techniques, explanation analysis and discrimination testing are applied only to the constrained interpretable ML models [1], [18], [19]. Here, MGBMs[6] and XNNs ([20] [21]) will serve as those more interpretable models for subsequent explanatory and discrimination analysis.

Post-hoc explanation and discrimination testing techniques are applied to constrained, interpretable models trained on the mortgage data to provide a template workflow for future users of similar methods and tools. Presented explanation techniques include PD, ICE, and Shapley values

---

[3] This text and associated software are not, and should not be construed as, legal advice or requirements for regulatory compliance.

[4] In the United States (US), interpretable models, explanations, DI testing, and the model documentation they enable may be required under the Civil Rights Acts of 1964 and 1991, the Americans with Disabilities Act, the Genetic Information Nondiscrimination Act, the Health Insurance Portability and Accountability Act, the Equal Credit Opportunity Act (ECOA), the Fair Credit Reporting Act (FCRA), the Fair Housing Act, Federal Reserve SR 11-7, and the European Union (EU) Greater Data Privacy Regulation (GDPR) Article 22 [9].

[5] See: Mortgage data (HMDA).

[6] As implemented in `XGBoost` or `h2o`.

[13], [22], [23], [24]. PD, ICE, and Shapley values provide direct, global, and local summaries and descriptions of constrained models without resorting to the use of intermediary and approximate surrogate models. Discussed discrimination testing methods include AIR, ME, and SMD [2], [25], [26].[7] Accuracy and other confusion matrix metrics are also reported by demographic segment [27]. All outlined materials and methods are implemented in open source Python code, and are made available in the software resources associated delineated in Subsection 1.9.

1.1. Notation

To facilitate descriptions of data and modeling, explanatory, and discrimination testing techniques, notation for input and output spaces, datasets, and models is defined.

1.1.1. Spaces

- Input features come from the set $\mathcal{X}$ contained in a P-dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$. An arbitrary, potentially unobserved, or future instance of $\mathcal{X}$ is denoted $\mathbf{x}$, $\mathbf{x} \in \mathcal{X}$.
- Labels corresponding to instances of $\mathcal{X}$ come from the set $\mathcal{Y}$.
- Learned output responses of models are contained in the set $\hat{\mathcal{Y}}$.

1.1.2. Datasets

- The input dataset $\mathbf{X}$ is composed of observed instances of the set $\mathcal{X}$ with a corresponding dataset of labels $\mathbf{Y}$, observed instances of the set $\mathcal{Y}$.
- Each i-th observation of $\mathbf{X}$ is denoted as $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \ldots, x_{P-1}^{(i)}]$, with corresponding i-th labels in $\mathbf{Y}, \mathbf{y}^{(i)}$, and corresponding predictions in $\hat{\mathbf{Y}}, \hat{\mathbf{y}}^{(i)}$.
- $\mathbf{X}$ and $\mathbf{Y}$ consist of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- Each j-th input column vector of $\mathbf{X}$ is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \ldots, x_j^{(N-1)}]^T$.

1.1.3. Models

- A type of ML model g, selected from a hypothesis set $\mathcal{H}$, is trained to represent an unknown signal-generating function f observed as $\mathbf{X}$ with labels $\mathbf{Y}$ using a training algorithm $\mathcal{A}$: $\mathbf{X}, \mathbf{Y} \xrightarrow{\mathcal{A}} g$, such that $g \approx f$.
- g generates learned output responses on the input dataset $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and on the general input space $g(\mathcal{X}) = \hat{\mathcal{Y}}$.
- The model to be explained and tested for discrimination is denoted as g.

1.2. Simulated Data

Simulated data is created based on a function first proposed in Friedman [10] and extended in Friedman *et al.* [11]:

$$f(\mathbf{X}) = 10 \sin(\pi \, \mathbf{X}_{\text{Friedman},1} \, \mathbf{X}_{\text{Friedman},2}) + 20(\mathbf{X}_{\text{Friedman},3} - 0.5)^2 + 10 \, \mathbf{X}_{\text{Friedman},4} + 5 \, \mathbf{X}_{\text{Friedman},5} \quad (1)$$

where $\mathbf{X}_{\text{Friedman},j}$ are random uniform features in $[0, 1]$. In Friedman's texts, a Gaussian noise term was added to create a continuous output variable for testing spline regression methodologies. In this manuscript, the signal generating function and input features are modified in several ways. Two binary features, a categorical feature with five discrete levels, and a bias term are introduced into f to add a degree of complexity that may more closely mimic real-world settings. For binary classification analysis, the Gaussian noise term is replaced with noise drawn from a logistic distribution and coefficients are re-scaled to be one fifth of the size of those used by Friedman, and any $f(\mathbf{X})$ value

[7] Part 1607 - Uniform Guidelines on Employee Selection Procedures (1978) §1607.4.

above 0 is classified as a positive outcome, while $f(\mathbf{X})$ values less than or equal to zero are designated as negative outcomes. Finally, f is augmented with two hypothetical protected class-control features with known dependencies on the binary outcome to allow for discrimination testing. The simulated data is generated to have eight input features, twelve after numeric encoding of categorical features, and a binary outcome, two class-control features, and 100,000 rows. The simulated data is then split into a training and test set, with 80,000 and 20,000 observations, respectively. Within the training set, a 5 fold cross validation indicator is used for training and assessing all models. For an exact specification of the simulated data, see the software resources referenced in Subsection 1.9.

1.3. Mortgage Data

The US HMDA, originally enacted in 1975, requires many financial institutions that originate mortgage products to provide certain data about many of the mortgage-related products that they either deny or originate on an annual basis. This information is first provided to the Consumer Financial Protection Bureau (CFPB), which subsequently releases some of the data to the public. Regulators often use HMDA data to, "...show whether lenders are serving the housing needs of their communities; they give public officials information that helps them make decisions and policies; and they shed light on lending patterns that could be discriminatory."[5] In addition to regulatory use, public advocacy groups use these data for similar purposes, and the lenders themselves use the data to benchmark their community outreach relative to their peers. The publicly available data that the CFPB releases includes information such as the lender, the type of loan, loan amount, loan to value (LTV) ratio, debt to income (DTI) ratio, and other important financial descriptors. The data also include information on each borrower and co-borrower's race, ethnicity, gender, and age. Because the data includes information on these protected class characteristics, certain metrics that can be indicative of discrimination in lending can be calculated directly using the HDMA data.

The mortgage dataset analyzed here is a random sample of consumer-anonymized loans from the HDMA database. These loans are a subset of all originated mortgage loans in the 2018 HMDA data that were chosen to represent a relatively comparable group of consumer mortgages. A selection of features is used to predict whether a loan is *high-priced*, i.e. the annual percentage rate (APR) charged was 150 basis points (1.5%) or more above a survey-based estimate of other similar loans offered around the time of the given loan. After data cleaning and preprocessing to encode categorical features and create missing markers, the mortgage data contains ten input features and the binary outcome, *high-priced*. The data is split into a training set with 160,338 loans and a marker for 5 fold cross validation and a test set containing 39,662 loans. While lenders would almost certainly use more information than the selected features to determine whether to offer and originate a high-priced loan, the selected input features (LTV ratio, DTI ratio, property value, loan amount, introductory interest rate, customer income, etc.) are likely to be some of the most influential factors that a lender would consider. Ultimately, the HMDA data represent the most comprehensive source of data on highly-regulated mortgage lending that is publicly available, which makes it an ideal dataset to use for the types of analyses set forth in Sections 1 and 2.

1.4. Monotonic Gradient Boosting Machine

MGBMs constrain typical GBM training to consider only tree splits that obey user-defined positive and negative monotonicity constraints, with respect to each X_j and $\mathbf{y}$ independently. The MGBM remains an additive combination of B trees trained by gradient boosting, T_b, and each tree learns a set of splitting rules that respect monotonicity constraints, Θ_b^{mono}.

$$g^{\text{MGBM}}(\mathbf{x}) = \sum_{b=1}^{B} T_b\left(\mathbf{x}; \Theta_b^{\text{mono}}\right) \tag{2}$$

As in unconstrained GBM, Θ_b^{mono} is selected in a greedy, additive fashion by minimizing a regularized loss function that considers known target labels, $\mathbf{y}$, the predictions of all subsequently trained trees in the MGBM, $g_{b-1}^{\text{MGBM}}(\mathbf{X})$, and a regularization term that penalizes complexity in the current tree, $\Omega(T_b)$. For the b-th iteration, the loss function, $\mathcal{L}_b$, can generally be defined as:

$$\mathcal{L}_b = \sum_{i=0}^{N-1} l(y^{(i)}, g_{b-1}^{\text{MGBM}}(\mathbf{x}^{(i)}), T_b(\mathbf{x}^{(i)}; \Theta_b^{\text{mono}})) + \Omega(T_b) \tag{3}$$

In addition to $\mathcal{L}_b$, g^{MGBM} training is characterized by additional splitting rules and constraints on tree node weights. Each binary splitting rule, $\theta_{b,j,k} \in \Theta_b$, is associated with a feature, X_j, is the k-th split associated with X_j in T_b, and results in left and right child nodes with a numeric weights, $\{w_{b,j,k,L}, w_{b,j,k,R}\}$. For terminal nodes, $\{w_{b,j,k,L}, w_{b,j,k,R}\}$ can be direct numeric components of some g^{MGBM} prediction. For two values of some feature X_j, $x_j^\alpha \leq x_j^\beta$, g^{MGBM} is positive monotonic with respect to some X_j if $g^{\text{MGBM}}(x_j^\alpha) \leq g^{\text{MGBM}}(x_j^\beta)\ \forall\ x_j^\alpha \leq x_j^\beta \in X_j$. The following rules and constraints ensure positive monotonicity in Θ_b, where the prediction for each value results in $T_b(x_j^\alpha; \Theta_b) = w_\alpha$ and $T_b(x_j^\beta; \Theta_b) = w_\beta$.

1. For the first and highest split in T_b involving X_j, any $\theta_{b,j,0}$ resulting in the left child weight being greater than the right child weight, $T(x_j; \theta_{b,j,0}) = \{w_{b,j,0,L}, w_{b,j,0,R}\}$ where $w_{b,j,0,L} > w_{b,j,0,R}$, is not considered.
2. For any subsequent left child node involving X_j, any $\theta_{b,j,k\geq 1}$ resulting in $T(x_j; \theta_{b,j,k\geq 1}) = \{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$ where $w_{b,j,k\geq 1,L} > w_{b,j,k\geq 1,R}$, is not considered.
3. Moreover, for any subsequent left child node involving X_j, $T(x_j; \theta_{b,j,k\geq 1}) = \{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$, $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$ are bound by the associated $\theta_{b,j,k-1}$ set of node weights, $\{w_{b,j,k-1,L}, w_{b,j,k-1,R}\}$, such that $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} < \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.
4. (1) and (2) are also applied to all right child nodes, except that for right child nodes $w_{b,j,k,L} \leq w_{b,j,k,R}$ and $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} \geq \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.

Note that for any one X_j and $T_b \in g^{\text{MGBM}}$ left subtrees will alway produce lower predictions than right subtrees, and that any $g^{\text{MGBM}}(\mathbf{x})$ is an addition of each T_b output, with the application of a monotonic logit or softmax link function for classification problems. Moreover, each tree's root node corresponds to some constant node weight that by definition obeys monotonicity constraints, $T(x_j^\alpha; \theta_{b,0}) = T(x_j^\beta; \theta_{b,0}) = w_{b,0}$. Together these additional splitting rules and node weight constraints ensure that $g^{\text{MGBM}}(x_j^\alpha) \leq g^{\text{MGBM}}(x_j^\beta)\ \forall\ x_j^\alpha \leq x_j^\beta \in X_j$. For a negative monotonic constraint, i.e. $g^{\text{MGBM}}(x_j^\alpha) \geq g^{\text{MGBM}}(x_j^\beta)\ \forall\ x_j^\alpha \leq x_j^\beta \in X_j$, left and right splitting rules and node weight constraints are switched. Also consider that MGBM models with independent monotonicity constraints between some X_j and $\mathbf{y}$ likely restrict non-monotonic interactions between multiple X_j. Moreover, if monotonicity constraints are not applied to all $X_j \in \mathbf{X}$, any strong non-monotonic signal in training data associated with some important X_j maybe forced onto some other arbitrary unconstrained X_j under some g^{MGBM} models, compromising the end goal of interpretability.

Herein, two g^{MGBM} models are trained. One on the simulated data and one on the mortgage data. In both cases, positive and negative monotonic constraints for each X_j are selected using domain knowledge, random grid search is used to determine other hyperparameters, and five-fold cross validation and test partitions are used for model assessment. For exact parameterization of the two g^{MGBM} models, see the software resources referenced in Subsection 1.9.

1.5. Explainable Neural Network

XNNs are an alternative formulation of additive index models in which the ridge functions are neural networks [20]. XNNs also bare a strong resemblance to generalized additive models (GAMs) and so-called explainable boosting machines (EBMs or GA^{2}M), i.e. GAMs which consider main effects and a small number of 2-way interactions and may also incorporate boosting into their training [13], [28]. Hence, XNNs enable users to tailor interpretable neural network architectures to a given prediction problem and to visualize model behavior by plotting ridge functions. XNNs are composed of a global bias term, μ_0, K individually specified neural networks, n_k with scale parameters γ_k, and the inputs to each n_k are themselves a linear combination of modeling inputs, $\sum_j \beta_{k,j} x_j$.

$$g^{\text{XNN}}(\mathbf{x}) = \mu_0 + \sum_{k=0}^{K-1} \gamma_k n_k \Big(\sum_{j=0}^{J=\mathcal{P}-1} \beta_{k,j} x_j \Big) \tag{4}$$

g^{XNN} is comprised of 3 meta-layers:

1. The first and deepest meta-layer, composed of K linear $\sum_j \beta_{k,j} x_j$ hidden units, is known as the *projection layer* and is fully connected to each input feature, X_j. Each hidden unit in the projection layer may optionally include a bias term.
2. The second meta-layer contains K hidden and separate n_k ridge functions, or *subnetworks*. Each n_k is a neural network, which can be parameterized to suit a given modeling task. To facilitate direct interpretation and visualization, the input to each subnetwork is the 1-dimensional output of its associated projection layer hidden unit, $\sum_j \beta_{k,j} x_j$. Each n_k can contain several bias terms.
3. The output meta-layer, called the *combination layer*, is another linear unit comprised of a global bias term, μ_0, and the K weighted 1-dimensional outputs of each subnetwork, $\gamma_k n_k(\sum_j \beta_{k,j} x_j)$. Again, subnetwork output is restricted to 1-dimension for interpretation and visualization purposes.

Here, each g^{XNN} is trained by mini-batch stochastic gradient descent (SGD) on the simulated data and mortgage data. Each g^{XNN} is assessed in five training folds and in a test data partition. L_1 regularization is applied to both the projection and combination layers to induce a sparse and interpretable model, where each n_k subnetwork and corresponding combination layer γ_k are ideally associated with an important X_j or combination thereof. The g^{XNN} models appear highly sensitive to weight initialization and batch size. Be aware that g^{XNN} model architectures may require manual and judicious feature selection due to long training times. For more details regarding g^{XNN} training, see the software resources in Subsection 1.9.

1.6. Partial Dependence and Individual Conditional Expectation

PD plots are a widely-used method for describing and plotting the average predictions of a complex model g across some partition of data $\mathbf{X}$ for some interesting input feature X_j [13]. ICE plots are a newer method that describes the local behavior of g for a single instance $\mathbf{x} \in \mathcal{X}$ [22]. PD and ICE can be overlaid in the same plot to compensate for known weaknesses of PD (e.g. inaccuracy in the presence of strong interactions and correlations [22], [29]), to identify interactions modeled by g, and to create a holistic global and local portrait of the predictions for some g and X_j [22].

Following Friedman *et al.* [13] a single feature $X_j \in \mathbf{X}$ and its complement set $\mathbf{X}_{\mathcal{P} \setminus \{j\}} \in \mathbf{X}$ (where $X_j \cup \mathbf{X}_{\mathcal{P} \setminus \{j\}} = \mathbf{X}$) is considered. $\text{PD}(X_j, g)$ for a given feature X_j is estimated as the average output of the learned function $g(\mathbf{X})$ when all the observations of X_j are set to a constant $x \in \mathcal{X}$ and $\mathbf{X}_{\mathcal{P} \setminus \{j\}}$ is left unchanged. $\text{ICE}(x_j, \mathbf{x}, g)$ for a given instance $\mathbf{x}$ and feature x_j is estimated as the output of $g(\mathbf{x})$ when x_j is set to a constant $x \in \mathcal{X}$ and all other features $\mathbf{x} \in \mathbf{X}_{\mathcal{P} \setminus \{j\}}$ are left untouched. PD and ICE curves are usually plotted over some set of constants $x \in \mathcal{X}$, as displayed in Section 2. Due to known problems for PD in the presence of strong correlation and interactions, PD should not be used alone. PD should be paired with ICE or be replaced with accumulated local effect (ALE) plots [22], [29].

1.7. Shapley Values

Shapley explanations are a class of additive, locally accurate feature contribution measures with long-standing theoretical support [23], [30]. Shapley explanations are the only possible locally accurate and globally consistent feature contribution values, meaning that Shapley explanation values for input features always sum to $g(\mathbf{x})$ for some $\mathbf{x} \in \mathcal{X}$ and that Shapley explanation values should never decrease in magnitude for some x_j when g is changed such that x_j truly makes a stronger contribution to $g(\mathbf{x})$ [23], [24]. For some instance $\mathbf{x} \in \mathcal{X}$, Shapley explanations take the form:

$$g(\mathbf{x}) = \phi_0 + \sum_{j=0}^{j=\mathcal{P}-1} \phi_j \mathbf{z}_j \tag{5}$$

In Equation 5, $\mathbf{z} \in \{0,1\}^{\mathcal{P}}$ is a binary representation of $\mathbf{x}$ where 0 indicates missingness. Each ϕ_j is the local feature contribution value associated with x_j and ϕ_0 is the average of $g(\mathbf{X})$. Each ϕ_j is a weighted combination of model scores, $g_x(\mathbf{x})$, with x_j, $g_x(S \cup \{j\})$, and the model scores without x_j, $g_x(S)$, for every subset of features S not including j, $S \subseteq \mathcal{P} \setminus \{j\}$, where g_x incorporates the mapping between $\mathbf{x}$ and the binary vector $\mathbf{z}$.

$$\phi_j = \sum_{S \subseteq \mathcal{P} \setminus \{j\}} \frac{|S|!(\mathcal{P} - |S| - 1)!}{\mathcal{P}!} [g_x(S \cup \{j\}) - g_x(S)] \tag{6}$$

Local, per-instance explanations using Shapley values tend to involve ranking x_j by ϕ_j values or delineating a set of the X_j names associated with the k-largest ϕ_j values for some $\mathbf{x}$, where k is some small positive integer, say 5. Global explanations are typically the absolute mean of the ϕ_j associated with a given X_j across all of the observations in some set $\mathbf{X}$.

Shapley values can be estimated in different ways, many of which are intractable for datasets with large $\mathcal{P}$. Tree SHAP is a specific implementation of Shapley explanations that relies on traversing internal decision tree structures to efficiently estimate the contribution of each x_j for some $g(\mathbf{x})$ [24]. Tree SHAP (SHapley Additive exPlanations) has been implemented efficiently in popular gradient boosting libraries such as `h2o`, `LightGBM`, and `XGBoost`, and Tree SHAP is used to calculate accurate and consistent global and local feature importance for MGBM models in Sections 1 and 2. Deep SHAP is an approximate Shapley value technique that creates SHAP values for ANNs [23]. Deep SHAP is implemented in the `shap` package and is used to generate SHAP values for the two g^{XNN} models discussed in Sections 1 and 2.

1.8. Discrimination Testing Metrics

Metrics for testing discrimination in this text are:

$$\text{AIR} \equiv \frac{\Pr(\hat{\mathbf{y}} = 1 | \mathbf{X}_p = 1)}{\Pr(\hat{\mathbf{y}} = 1 | \mathbf{X}_c = 1)} \tag{7}$$

$$\text{ME} \equiv 100 \cdot (\Pr(\hat{\mathbf{y}} = 1 | \mathbf{X}_c = 1) - \Pr(\hat{\mathbf{y}} = 1 | \mathbf{X}_p = 1)) \tag{8}$$

$$\text{SMD} \equiv \frac{\bar{\hat{\mathbf{y}}}_p - \bar{\hat{\mathbf{y}}}_c}{\sigma_{\hat{\mathbf{y}}}} \tag{9}$$

where $\mathbf{X}_p$ and $\mathbf{X}_c$ represent binary markers created from some demographic attribute $\mathbf{X}_j$, c denotes the control group (often whites or males), p indicates a protected group, and conditional probabilities, $\Pr(\hat{\mathbf{y}} | \mathbf{X}_j = \alpha)$, are evaluated over all $\mathbf{x}^{(i)} \in \{\mathbf{X}_j | x_j^{(i)} = \alpha\}$. Accuracy in each demographic is also considered as a measure of validity for a model and a demographic group.

1.9. Software Resources

Python code to reproduce discussed results is available at: `https://github.com/h2oai/article-information-2019`. The primary Python packages employed are: `numpy` and `pandas` for data manipulation, `h2o`, `keras`, `shap`, and `tensorflow` for modeling, explanation, and discrimination testing, and `matplotlib` for plotting.

2. Results

Results are laid out for the simulated and mortgage datasets. Accuracy is compared for unconstrained, less interpretable g^{GBM} and g^{ANN} models and constrained, more interpretable g^{MGBM} and g^{XNN} models. Then, for the g^{MGBM} and g^{XNN} models, intrinsic interpretability, post-hoc explanation, and discrimination testing results are presented.

2.1. Simulated Data Results

Results for constrained models on the simulated data are displayed in Subsections 2.1.1 – 2.1.3. Model fit is roughly uniform for g^{GBM}, g^{MGBM}, g^{ANN}, and g^{XNN} on the simulated test data. Given that little or no trade-off is required in terms of model to fit to use the constrained models, intrinsic interpretability, post-hoc explainability, and discrimination are explored further for the g^{MGBM} and g^{XNN} models. ...

2.1.1. Constrained vs. Unconstrained Model Fit Assessment

Table 1 presents a variety of fit metrics for the g^{GBM}, g^{MGBM}, g^{ANN}, and g^{XNN} on the simulated test data. g^{GBM} exhibits the best performance, but all models give relatively similar fit results. Interpretability and explainability benefits of the constrained models appear to come at little cost to overall model performance, or in the case of g^{ANN} and g^{XNN}, no cost at all. g^{XNN} actually shows slightly better fit than g^{ANN} across accuracy, area under the curve (AUC), logloss, and root mean squared error (RMSE). Accuracy is measured at the best F1 threshold for each model.

Table 1. Fit metrics for g^{GBM}, g^{MGBM}, g^{ANN}, and g^{XNN} on the simulated test data.

Model	Accuracy	AUC	Logloss	RMSE
g^{GBM}	0.775	0.857	0.474	0.394
g^{MGBM}	0.763	0.846	0.498	0.405
g^{ANN}	0.757	0.850	0.480	0.398
g^{XNN}	0.758	0.851	0.479	0.397

2.1.2. Interpretability and Post-hoc Explanation Results

Tree SHAP values are reported in the margin space, prior to the application of the logit link function, and the numeric value of the reported values can interpreted as the absolute mean impact of each X_j on g^{MGBM} in the margin space.

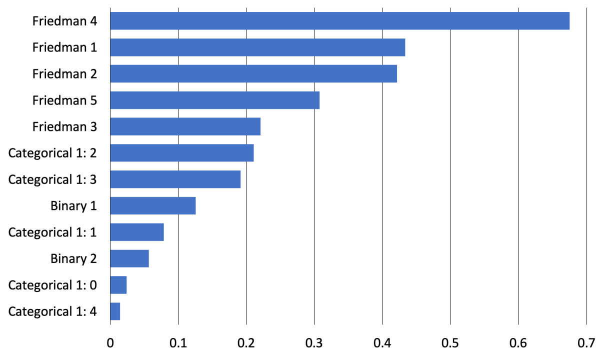

Figure 1. Global mean Tree SHAP feature importance for g^{MGBM} on the simulated test data. Tree SHAP values are reported in the margin space, prior to the application of the logit link function.

PD and ICE are always displayed with a histogram herein to highlight any sparse regions in an input feature's domain. Because most ML models will always issue a prediction on any datum with a correct schema, it's crucial to consider whether a given model learned enough about an observation to make an accurate prediction. Viewing PD and ICE along with a histogram is a convenient method to visually assess whether a prediction is reasonable and based on sufficient training data.

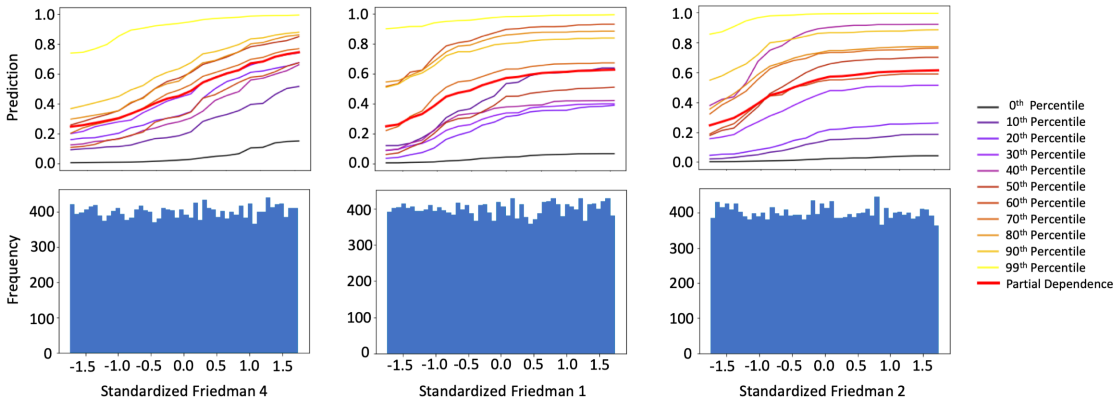

Figure 2. PD, ICE for 10 observations across selected percentiles of $g^{\text{MGBM}}(\mathbf{X})$, and histograms for the three most important input features of g^{MGBM} on the simulated test data.

Deep SHAP values are reported in the probability space, after the application of the logit link function. They are also calculated from the projection layer of g^{XNN}.

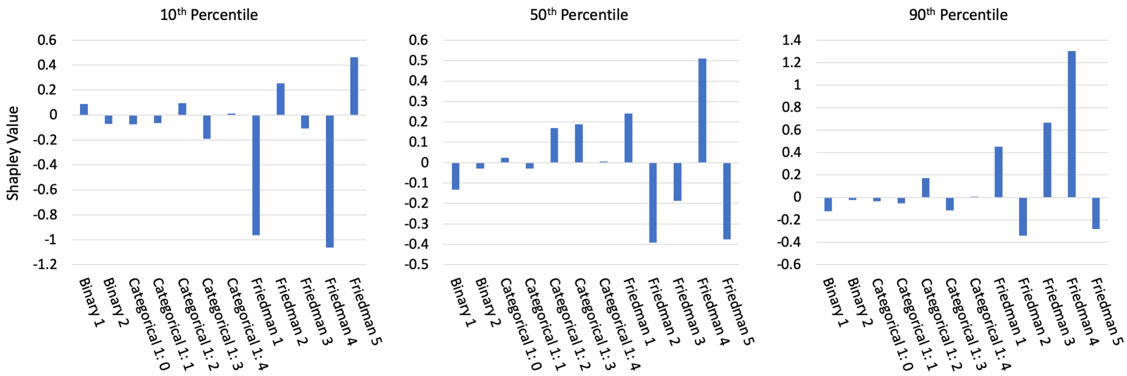

Figure 3. Tree SHAP values for three observations across selected percentiles of $g^{\text{MGBM}}(\mathbf{X})$ for the simulated test data.

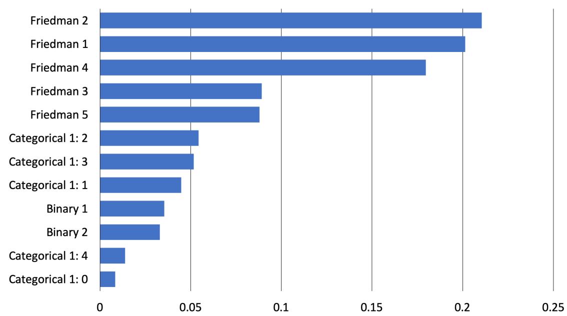

Figure 4. Global mean Deep SHAP feature importance for g^{XNN} on the simulated test data.

Ridge functions are reminiscent of basis functions ... distinctive simplistic functions

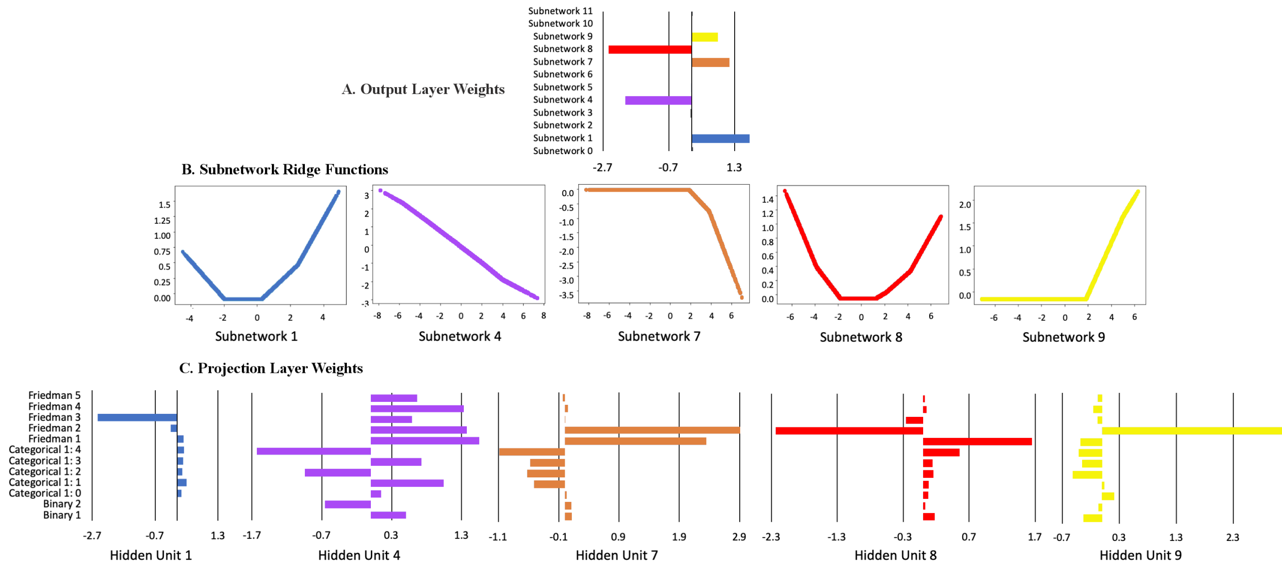

Figure 5. A. Output layer γ_k weights, **B.** corresponding n_k ridge functions, and **C.** associated projection layer β_k weights for g^{XNN} on the simulated test data.

Deep SHAP values are reported in the probability space, after the application of the logit link function. They are also calculated from the projection layer of g^{XNN}.

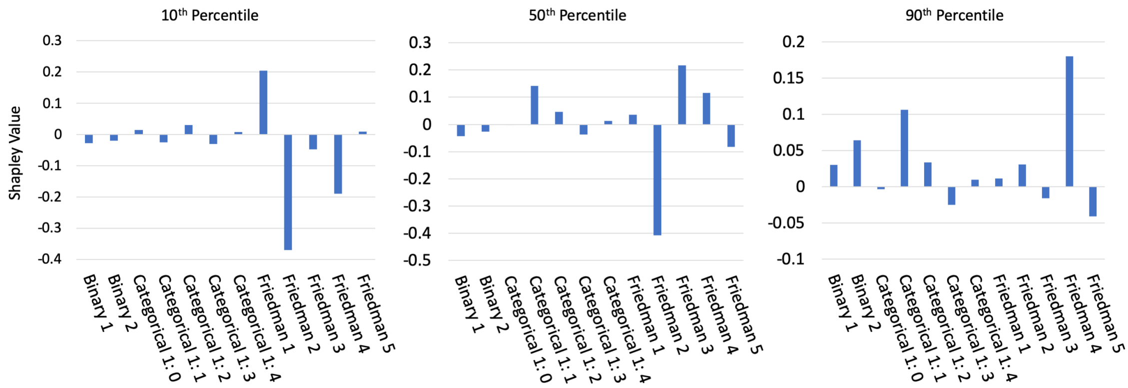

Figure 6. Deep SHAP values for three observations across selected percentiles of g^{XNN} on the simulated test data.

Deep SHAP values are reported in the probability space, after the application of the logit link function.

2.1.3. Discrimination Testing Results

Table 2. Group size and accuracy for g^{MGBM} and g^{XNN} on the simulated test data.

Class	N	Model	Accuracy
Protected 1	3057.0	g^{MGBM} g^{XNN}	0.770 0.764
Control 1	16943	g^{MGBM} g^{XNN}	0.739 0.751
Protected 2	9916	g^{MGBM} g^{XNN}	0.758 0.761
Control 2	10084	g^{MGBM} g^{XNN}	0.729 0.745

Table 3. AIR, ME and SMD for g^{MGBM} and g^{XNN} on the simulated test data.

Model	Protected Class	Control Class	AIR	ME	SMD
g^{MGBM}	Protected 1	Control 1	0	0%	0
	Protected 2	Control 2	0	0%	0
g^{XNN}	Protected 1	Control 1	0	0%	0
	Protected 2	Control 2	0	0%	0

2.2. Mortgage Data Results

Results for the mortgage data are presented in Subsections 2.2.1 – 2.2.3. g^{ANN} and g^{XNN} outperform g^{GBM} and g^{MGBM} on the mortgage data, but as in Subsection 2.1.1 the constrained variants of both model architectures do not show large differences in model performance with respect to unconstrained variants. Assuming that small fit differences on static test data do not outweigh the need for intrinsic model interpretability and post-hoc explainability in high-stakes, human-centered, or regulated applications, only g^{MGBM} and g^{XNN} interpretability, post-hoc explainability, and discrimination testing results are presented. For g^{MGBM}, intrinsic interpretability is evaluated with PD and ICE plots of mostly monotonic prediction behavior for several important X_j, and post-hoc Shapley explanation analysis is used to create global and local feature importance. For g^{XNN}, inherent interpretability manifests as plots of sparse γ_k output layer weights, n_k subnetwork ridge functions, and sparse β_j weights in the projection layer. Post-hoc Shapley explanation techniques are also used to generate global and local feature importance for g^{XNN}. Both g^{MGBM} and g^{XNN} are evaluated for discrimination using AIR, ME, SMD, and other measures.

2.2.1. Constrained vs. Unconstrained Model Fit Assessment

Table 4 shows that g^{ANN} and g^{XNN} noticeably outperform g^{GBM} and g^{MGBM} on the mortgage data. This is at least partially due to the preprocessing required to present directly comparable post-hoc explainability results and to use neural networks and tensorflow, e.g. numerical encoding of categorical features and missing values. This preprocessing appears to hamstring some of the tree-based models' inherent capabilities. g^{GBM} models trained on non-encoded data with missing values repeatedly produced AUC values of ~ 0.81.

Table 4. Fit metrics for g^{GBM}, g^{MGBM}, g^{ANN}, and g^{XNN} on the mortgage test data.

Model	Accuracy	AUC	Logloss	RMSE
g^{GBM}	0.795	0.828	0.252	0.276
g^{MGBM}	0.765	0.814	0.259	0.278
g^{ANN}	0.865	0.871	0.231	0.262
g^{XNN}	0.869	0.868	0.233	0.263

Regardless of the fit differences between the two families of hypothesis models, the difference between the fit of constrained and unconstrained variants within the two types of models is small for the GBMs and negligible for ANNs, 3% and < 1% worse fit respectively, averaged across the metrics reported in Table 4.

2.2.2. Interpretability and Post-hoc Explanation Results

Global Shapley feature importance for g^{MGBM} on the mortgage test data is reported in Figure 7. g^{MGBM} places high importance on LTV ratio, perhaps too high, and also weighs DTI ratio, property value, loan amount, and introductory rate period heavily in many of its predictions.

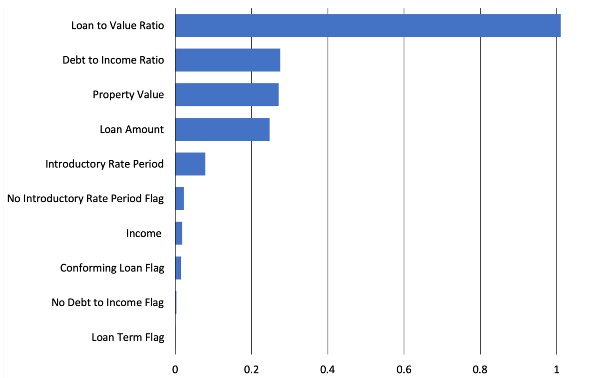

Figure 7. Global mean absolute Tree SHAP feature importance for g^{MGBM} on the mortgage test data.

The potential over-emphasis of LTV ratio, and the de-emphasis of income, likely an important feature from a business perspective, and the encoded no introductory rate period flag feature may contribute to the decreased performance of g^{MGBM} as compared to g^{XNN}.

Domain knowledge was used to positively constrain DTI ratio and LTV ratio and to negatively constrain income and the loan term flag under g^{MGBM}. The monotonicity constraints for DTI ratio and LTV ratio are confirmed for $g^{\text{MGBM}}(\mathbf{X})$ on the mortgage test data in Figure 8. Both DTI ratio and LTV ratio display positive monotonic behavior at all selected percentiles for ICE and on average with PD. Because PD curves generally follow the patterns of the ICE curves for both features, it's also likely that no strong interactions are at play for DTI ratio and LTV ratio under g^{MGBM}. Of course, the monotonicity constraints themselves can dampen the effects of non-monotonic interactions under g^{MGBM}, even if they do exist in the data, and this rigidity could also play a role in the performance differences between g^{MGBM} and g^{XNN}, which does allow for the modeling of non-monotonic interactions. DTI ratio and LTV ratio also appear to have sparse regions in their univariate distributions. The monotonicity constraints likely play to the advantage of g^{MGBM} in this regard, as g^{MGBM} appears to carry reasonable predictions learned from populous domains into the sparse domains of both features.

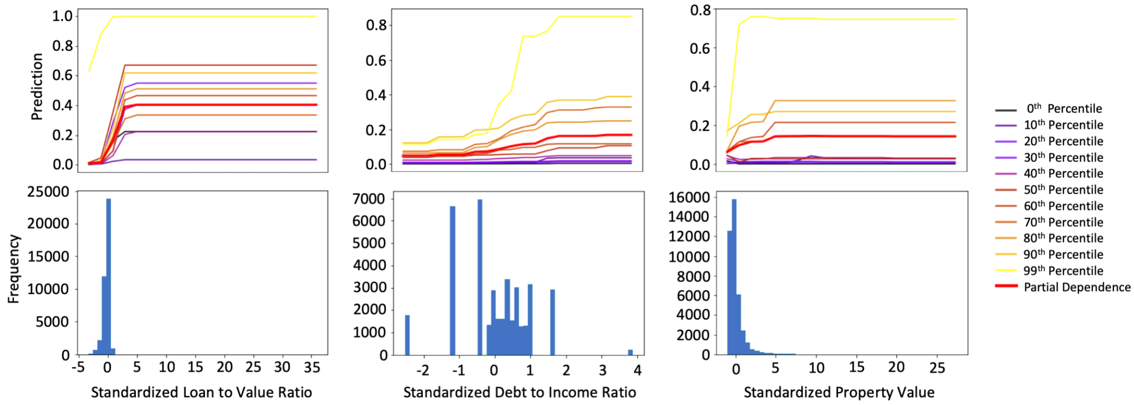

Figure 8. PD, ICE for 10 individuals across selected percentiles of $g^{\text{MGBM}}(\mathbf{X})$, and histograms for the three most important input features of g^{MGBM} on the mortgage test data.

Figure 8 also displays PD and ICE for the unconstrained feature property value. Unlike DTI ratio and LTV ratio, PD for property value does not always follow the patterns established by ICE curves. While PD shows monotonically increasing prediction behavior on average, apparently influenced by large predictions at extreme $g^{\text{MGBM}}(\mathbf{X})$ percentiles, ICE curves for individuals at the 40$^{\text{th}}$ percentile of $g^{\text{MGBM}}(\mathbf{X})$, and lower, exhibit different prediction behavior with respect to property value. Some individuals at these lower percentiles display monotonically decreasing prediction behavior while others appear to show fluctuating prediction behavior. Property value is strongly right-skewed, with little data regarding high-value property from which g^{MGBM} can learn. For the most part, reasonable predictions do appear to be carried from more densely populated regions to more sparsely populated regions. However, prediction fluctuations at lower $g^{\text{MGBM}}(\mathbf{X})$ percentiles are visible, and in a sparse region of property value. Such fluctuations can be caused by local data characteristics, modeled local interactions, or unfortunately, by overfitting or leakage of strong non-monotonic signal from important constrained features into the modeled behavior of non-constrained features. This divergence of PD and ICE can be indicative of an interaction affecting property value under $g^{\text{MGBM}}(\mathbf{X})$ [22], and drivers of such interactions can often be elucidated using surrogate decision trees [54].

In Figure 9, local Tree SHAP values are displayed for selected individuals at the 10$^{\text{th}}$, 50$^{\text{th}}$, and 90$^{\text{th}}$ percentiles of $g^{\text{MGBM}}(\mathbf{X})$ in the mortgage test data. The selected individuals show an expected progression of mostly negative Shapley values at the 10$^{\text{th}}$ percentile, a mixture of positive and negative Shapley values at the 50$^{\text{th}}$ percentile, mostly positive Shapley values the 90$^{\text{th}}$ percentile, and with globally important features driving most local model decisions.

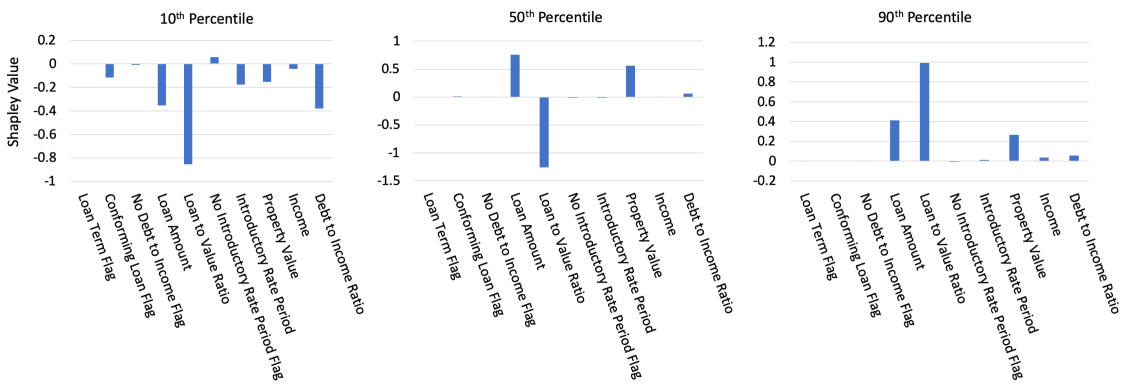

Figure 9. Tree SHAP values for three individuals across selected percentiles of $g^{\text{MGBM}}(\mathbf{X})$ for the mortgage test data.

Deeper significance for Figure 9 lies in the ability to accurately summarize any single $g^{\text{MGBM}}(\mathbf{x})$ prediction in this manner, which is generally important for enabling logical appeal or override of

ML-based decisions, and specifically important in the context of lending, where applicable regulations often require lenders to provide consumer-specific reasons for denying credit to an individual. In the US, applicable regulations are typically ECOA and FCRA, and the consumer-specific reasons are commonly known as adverse actions codes.

Figure 10 displays global feature importance for g^{XNN} on the mortgage test data. g^{XNN} distributes importance more evenly across business drivers and puts stronger emphasis on the no introductory rate period flag feature than does g^{MGBM}. Like g^{MGBM}, g^{XNN} puts little emphasis on the other flag features.

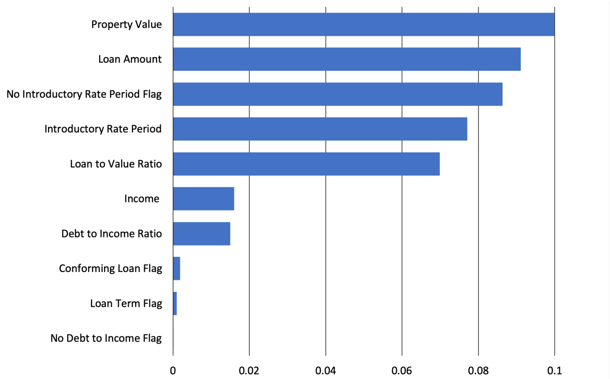

Figure 10. Global mean absolute Deep SHAP feature importance for g^{XNN} on the mortgage test data.

As compared to g^{MGBM}, g^{XNN} assigns higher importance to property value and loan amount, and lower importance on LTV ratio, and the income related features, DTI ratio and income.

The capability of g^{XNN} to model nonlinear phenomenon and high-degree interactions, and to do so in an interpretable manner, is on display in Figure 11. 11 **A** presents the sparse γ_k weights of the g^{XNN} output layer in which the n_k subnetworks with $k \in \{0, 1, 2, 3, 5, 8, 9\}$ have large magnitude weights and n_k subnetworks, $k \in \{4, 6, 7\}$, have small or near-zero magnitude weights. Distinctive ridge functions that feed into those large magnitude γ_k weights are highlighted in 11 **B** and color-coded to pair with their corresponding γ_k weight. As in the subsection 2.1.2, n_k ridge function plots vary with the output of the corresponding projection layer $\sum_j \beta_{k,j} x_j$ hidden unit, with weights displayed in matching colors in 11 **C**. In both the simulated and mortgage data, g^{XNN} n_k ridge functions appear to be elementary functional forms that the output layer learns to combine to generate accurate predictions, perhaps reminiscent of the visual primitives often learned by low layers of pattern-detecting convolutional neural networks (CNNs) [31].

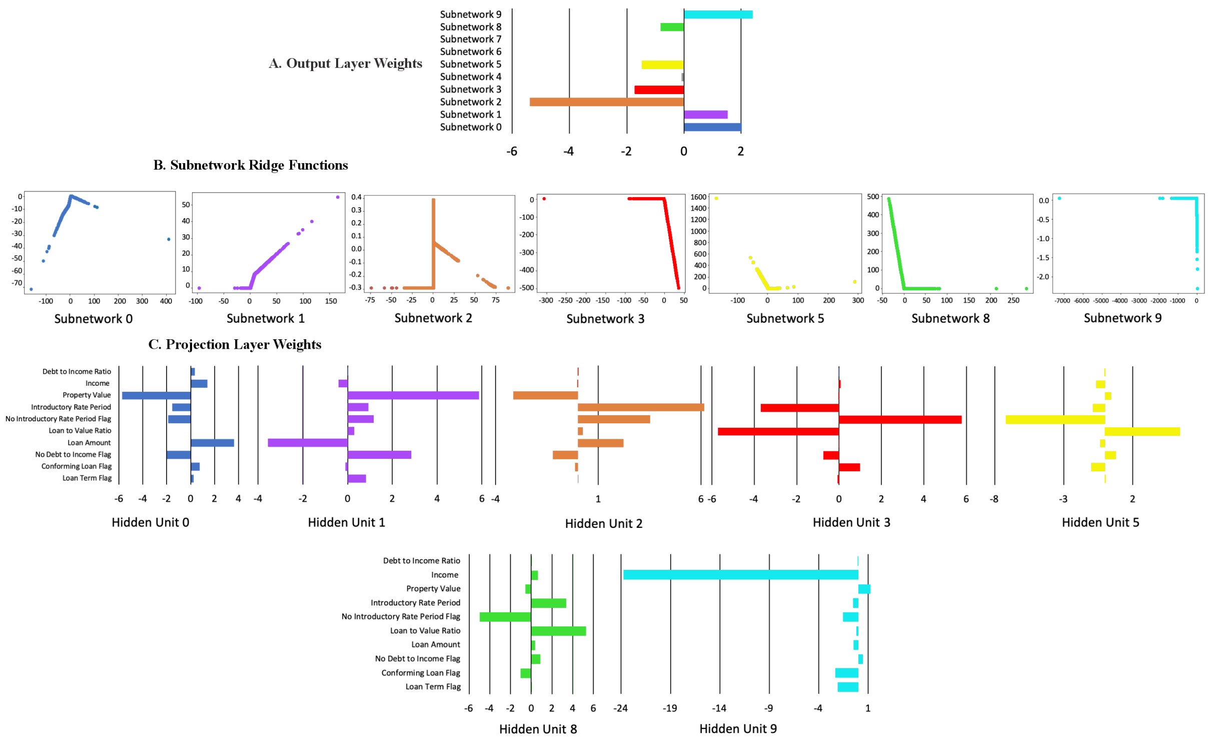

Figure 11. A. Output layer γ_k weights, **B.** corresponding n_k ridge functions, and **C.** associated projection layer β_k weights for g^{XNN} on the mortgage test data.

11 **C** displays the sparse β_k weights of the projection layer $\sum_j \beta_{k,j} x_j$ hidden units that are associated with each n_k subnetwork ridge function. For instance, subnetwork n_3 is influenced by large weights for LTV ratio, no introductory rate period flag, and introductory rate period, whereas subnetwork n_9 is nearly completely dominated by the weight for income. In combination, Figure 11 **A**, **B**, and **C** help practitioners understand which original input X_j features are weighed heavily in each n_k subnetwork, and which n_k subnetworks have a strong influence on $g^{\text{XNN}}(\mathbf{X})$.

To compliment the global interpretability of g^{XNN}, Figure 12 displays local Shapley values for selected individuals, estimated from the projection layer using Deep SHAP in the g^{XNN} probability space.

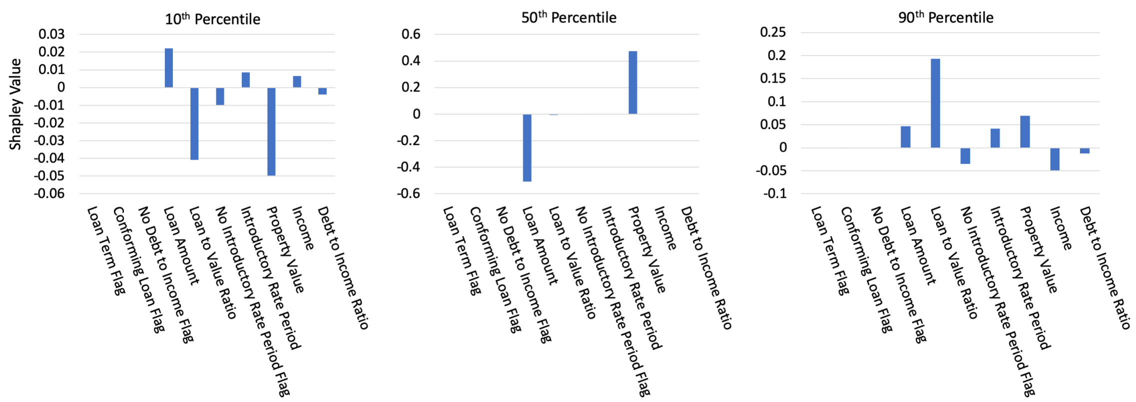

Figure 12. Deep SHAP values for three individuals across selected percentiles of g^{XNN} on the mortgage test data.

While the Shapley values appear to follow the roughly increasing pattern established in Figures 3, 6, and 9 their true value is their ability to be calculated for any $g^{\text{XNN}}(\mathbf{x})$ prediction, as a means to summarize model reasoning and allow for appeal and override of specific ML-based decisions, even for neural network architectures.

2.2.3. Discrimination Testing Results

Table 5. Group size and accuracy for g^{MGBM} and g^{XNN} on the mortgage test data.

Class	**N**	**Model**	Accuracy
Black	2608	g^{MGBM} g^{XNN}	0.654 0.702
White	28361	g^{MGBM} g^{XNN}	0.817 0.857
Female	8301	g^{MGBM} g^{XNN}	0.768 0.822
Male	13166	g^{MGBM} g^{XNN}	0.785 0.847

Table 6. AIR, ME and SMD for g^{MGBM} and g^{XNN} on the mortgage test data.

Model	**Protected Class**	**Control Class**	**AIR**	**ME**	**SMD**
g^{MGBM}	Black Female	White Male	0.776 0.948	18.3% 4.1%	0.628 0.084
g^{XNN}	Black Female	White Male	0.743 0.955	21.4% 3.6%	0.621 0.105

3. Discussion

3.1. The Burgeoning Python Ecosystem for Responsible Machine Learning

MGBM and XNN interpretable model architectures were selected for this text because they are straightforward variants of popular unconstrained ML models. If practitioners are working with GBM and ANN models, it should be relatively uncomplicated for them to evaluate the constrained versions of these models. The same can be said of the presented explanation methods and discrimination tests. Due to their post-hoc nature, they can often be shoe-horned into existing ML work flows and pipelines. While these approaches are promising responses to the black-box and discrimination problems in ML, they are just a small part of a burgeoning ecosystem of research and Python tools for responsible ML. Figure 13 is a work flow blueprint that illustrates some of the additional steps that may be required to build a fully understandable and trustworthy ML system.[8] While all the methods mentioned in Figure 13 play an important role in increasing human trust and understanding of ML, a few pertinent references and Python resources are highlighted below as further reading.

[8] See: Toward Responsible Machine Learning for details regarding Figure 13.

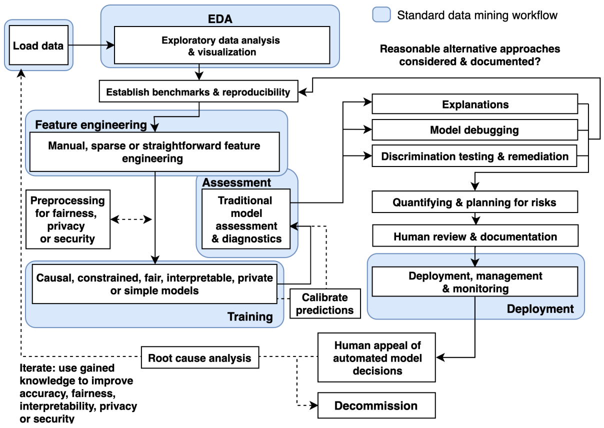

Figure 13. A diagram of a proposed holistic ML work flow in which interpretable models, post-hoc explanations, discrimination testing and remediation techniques, and other review and appeal mechanisms can create an understandable and trustworthy ML system.

Any discussion of interpretable ML models would be incomplete without references to the seminal work of the Rudin group at Duke University and EBM or GA^{2}M models, pioneered by researchers at Microsoft and Cornell. In keeping with a major theme of this manuscript, models from these leading researchers and several other kinds of interpretable ML models are now available as open source Python packages. Among others, practitioners can now evaluate EBM in the `interpret` package, optimal sparse decision trees, GAMs in the `pyGAM` package, a variant of Friedman's RuleFit in the `skope-rules` package, monotonic calibrated interpolated lookup tables in `tensorflow/lattice`, and *this looks like that* interpretable deep learning [32], [33], [34], [35].[9,10] Additional, relevant references and Python functionality include:

- **Exploratory data analysis (EDA)**: `H2OAggregatorEstimator` in `h2o` [36].
- **Sparse feature extraction**: `H2OGeneralizedLowRankEstimator` in `h2o` [37].
- **Privacy preprocessing and private models**: differential privacy and private models in `diffprivlib` and `tensorflow/privacy` [38], [39], [40], [41].
- **Post-hoc explanation**: structured data explanations with `alibi` and `PDPbox`, image classification explanations with `DeepExplain`, and natural language explanations with `allennlp` [42], [43], [44].
- **Discrimination testing**: with `aequitas` and `Themis`.
- **Discrimination remediation**: Reweighing, adversarial de-biasing, learning fair representations, and reject option classification with `AIF360` [45], [46], [47], [48].
- **Model debugging**: with `foolbox`, `SALib`, `tensorflow/cleverhans`, and `tensorflow/model-analysis` [49], [50], [51], [52].
- **Model documentation**: models cards [53], e.g. GPT-2 model card, Object Detection model card.

See: Awesome Machine Learning Interpretability for a longer, community-curated metalist of related software packages and resources.

3.2. *Interpretability, Explainability, Appeal, and Compliance*

Interpretable model architectures and post-hoc explanations play an important role in increasing transparency into model mechanisms and predictions. As seen in Sections 1 and 2, interpretable models often enable users to enforce domain knowledge-based constraints on model behavior, to ensure that models obey reasonable expectations, and to gain data-derived insights into the modeled problem domain. Post-hoc explanations generally help describe and summarize mechanisms and decisions,

[9] See: Optimal sparse decision trees.
[10] See: *This looks like that* interpretable deep learning.

potentially yielding an even clearer understanding of ML models. Together they can allow for human learning from ML, certain types of regulatory compliance, and crucially, human appeal or override of automated model decisions [54]. Interpretable models and post-hoc explanations are likely good candidates for ML uses cases under the FCRA, ECOA, GDPR and other regulations that may require explanations of model decisions, and they are already used in the financial services industry today for model validation and other purposes.[11,12] Writ large, transparency in ML also facilitates additional responsible AI processes such as model debugging, model documentation, and logical appeal and override processes, some which may also be required by applicable regulations.[13] Among these, appeal may deserve the most attention. ML models are often wrong.[14] For high-stakes, human-centered, or regulated applications that are trusted with mission- or life-critical decisions, the ability to appeal or override inevitable wrong decisions is not only a possible prerequisite for regulatory compliance, but also an important failsafe procedure for those affected by ML decisions.

3.3. Impact of Discrimination Testing on Model Use and Adoption

3.4. Viable Discrimination Remediation Approaches

3.5. Intersectionality of Interpretability, Explainability, Discrimination, and Security in ML

The black-box nature of ML, the perpetuation or exacerbation of discrimination by ML, or the security vulnerabilities inherent in ML are each serious and difficult problems on their own. However, evidence is mounting that these harms can also manifest as complex intersectional challenges, e.g. the *fairwashing* or *scaffolding* of biased models with ML explanations, the privacy harms of ML explanations, or the adversarial poisoning of ML models to become discriminatory [8], [18], [19].[15,16,17] Again, this text makes no claims that the opacity, discrimination, or security problems in ML have been solved, even treated as independent problems. Instead, this text aims to highlight these issues as both singular entities and non-static intersectional phenomena. Practitioners should of course consider the discussed interpretable modeling, post-hoc explanation, and discrimination testing approaches as at least partial remedies to the black-box and discrimination issues in ML. However, they should also consider that explanations can ease model stealing, data extraction, and membership inference attacks and that explanations can mask ML discrimination. Additionally, high-stakes, human-centered, or regulated ML systems should generally be built and tested with robustness to adversarial attacks as a primary design consideration, and specifically to prevent ML models from being poisoned or otherwise altered to become discriminatory. Accuracy, discrimination, and security characteristics of a system can change over time as well. Simply testing for these problems at training time, as presented in Sections 1 and 2, is not adequate for high-stakes, human-centered, or regulated ML systems. Accuracy, discrimination, and security should be monitored in real-time and over time, as long as a model is deployed.

---

11 See: *Deep Insights into Explainability and Interpretability of Machine Learning Algorithms and Applications to Risk Management*.

12 Unfortunately, many non-consistent explanation methods can result in drastically different global and local feature importance values across different models trained on the same data or even for refreshing a model with augmented training data [55]. Consistency and accuracy guarantees are perhaps a factor in the growing momentum behind Shapley values as a candidate technique for generating consumer-specific adverse action notices for explaining and appealing automated ML-based decisions in highly-regulated settings such as credit lending [56].

13 E.g.: US Federal Reserve Bank Supervision and Regulation (SR) Letter 11-7: Guidance on Model Risk Management.

14 "All models are wrong, but some are useful." – George Box, Statistician (1919 - 2013)

15 See: *Tay, Microsoft's AI chatbot, gets a crash course in racism from Twitter*.

16 While the focus of this paper is not ML security, proposed best-practices from that field do point to transparency of ML systems as a mitigating factor for some ML attacks and hacks [52]. High system complexity is sometimes considered a mitigating influence as well [57]. This is sometimes known as the *transparency paradox* in data privacy and security, and it likely applies to ML security as well, especially in the context of interpretable ML models and post-hoc explanation.[17]

17 See: *The AI Transparency Paradox*.

4. Conclusion

This text puts forward results on simulated data to provide a rough validation of constrained ML models, post-hoc explanation techniques, and discrimination testing methods. These same modeling, explanation, and discrimination testing approaches are then applied to more realistic mortgage data to provide an example of a responsible ML work flow for high-stakes, human-centered, or regulated ML applications. The discussed methodologies are solid steps toward interpretability, explanation, and minimal discrimination for ML decisions, which should ultimately enable increased fairness and logical appeal processes for ML decision subjects. Of course there is more to the responsible practice of ML than interpretable models, post-hoc explanation, and discrimination testing, even from a technology perspective, and Section 3 also points out numerous additional references and open source Python software assets that are available to researchers and practitioners today to increase human trust and understanding in ML systems. While the messy, complex, and human problems of racism, sexism, privacy violations, and cyber crime can probably not be solved by technology alone, this work (and many, many others) illustrate numerous ways for ML practitioners to become part of the solution to these problems, instead of perpetuating and exacerbating them.

Author Contributions: NG, data cleaning, GBM and MGBM assessment and results; PH, primary author; KM, ANN and XNN implementation, assessment, and results; NS, secondary author, data simulation and collection, and discrimination testing.

Funding: This work received no external funding.

Acknowledgments: Wen Phan for work in formalizing notation. Sue Shay for editing. Andrew Burt for ideas around the transparency paradox.

Conflicts of Interest: XNN was first made public by the corporate model validation team at Wells Fargo bank. Wells Fargo is a customer of, and investor in, H2O.ai and a customer of BLDS, LLC.

Abbreviations

The following abbreviations are used in this text: AI – artificial intelligence, AIR - adverse impact ratio, ALE - accumulated local effect, ANN – artificial neural network, APR – annual percentage rate, AUC – area under the curve, CNN – convolutional neural network, CFPB – Consumer Financial Protection Bureau, DI – disparate impact, DTI – debt to income, EBM or GA^2M – explainable boosting machine, i.e. variants GAMs that consider two-way interactions and may incorporate boosting into training, ECOA - Equal Credit Opportunity Act, EDA – exploratory data analysis, EU – European Union, FCRA – Fair Credit Reporting Act, GAM – generalized additive model, GBM – gradient boosting machine, GDPR - General Data Protection Regulation, HMDA – Home Mortgage Disclosure Act, ICE – individual conditional expectation, LTV – loan to value, ME – marginal effect, MGBM – monotonic gradient boosting machine, ML – machine learning, PD – partial dependence, RMSE – root mean square error, SGD – stochastic gradient descent, SHAP – Shapley additive explanation, SMD - standardized mean difference, SR – supervision and regulation, US – United States, XNN – explainable neural network.

References

1. Rudin, C. Please Stop Explaining Black Box Models for High Stakes Decisions and Use Interpretable Models Instead. *arXiv preprint arXiv:1811.10154* **2018**. URL: https://arxiv.org/pdf/1811.10154.pdf.
2. Feldman, M.; Friedler, S.A.; Moeller, J.; Scheidegger, C.; Venkatasubramanian, S. Certifying and Removing Disparate Impact. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 259–268. URL: https://arxiv.org/pdf/1412.3756.pdf.
3. Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness Through Awareness. Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ACM, 2012, pp. 214–226. URL: https://arxiv.org/pdf/1104.3913.pdf.
4. Buolamwini, J.; Gebru, T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. Conference on Fairness, Accountability and Transparency, 2018, pp. 77–91. URL: http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.

5. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J. The Security of Machine Learning. *Machine Learning* **2010**, *81*, 121–148. URL: http://people.ischool.berkeley.edu/~tygar/papers/SML/sec_mach_learn_journal.pdf.
6. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing Machine Learning Models via Prediction APIs. 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 601–618. URL: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_tramer.pdf.
7. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 3–18. URL: https://arxiv.org/pdf/1610.05820.pdf.
8. Shokri, R.; Strobel, M.; Zick, Y. Privacy Risks of Explaining Machine Learning Models. *arXiv preprint arXiv:1907.00164* **2019**. URL:https://arxiv.org/pdf/1907.00164.pdf.
9. Williams, M.; others. *Interpretability*; Fast Forward Labs, 2017. URL: https://www.cloudera.com/products/fast-forward-labs-research.html.
10. Friedman, J.H. A Tree-structured Approach to Nonparametric Multiple Regression. In *Smoothing techniques for curve estimation*; Springer, 1979; pp. 5–22. URL: http://inspirehep.net/record/140963/files/slac-pub-2336.pdf.
11. Friedman, J.H.; others. Multivariate Adaptive Regression Splines. *The annals of statistics* **1991**, *19*, 1–67. URL: https://projecteuclid.org/download/pdf_1/euclid.aos/1176347963.
12. Friedman, J.H. Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics* **2001**, pp. 1189–1232. URL: https://projecteuclid.org/download/pdf_1/euclid.aos/1013203451.
13. Friedman, J.H.; Hastie, T.; Tibshirani, R. ***The Elements of Statistical Learning***; Springer: New York, 2001. URL: https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf.
14. Recht, B.; Re, C.; Wright, S.; Niu, F. HOGWILD: A Lock-free Approach to Parallelizing Stochastic Gradient Descent. Advances in Neural Information Processing Systems (NIPS), 2011, pp. 693–701. URL: https://papers.nips.cc/paper/4390-hogwild-a-lock-free-approach-to-parallelizing-stochastic-gradient-descent.pdf.
15. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580* **2012**. URL: https://arxiv.org/pdf/1207.0580.pdf.
16. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the Importance of Initialization and Momentum in Deep Learning. International Conference on Machine Learning, 2013, pp. 1139–1147. URL: http://proceedings.mlr.press/v28/sutskever13.pdf.
17. Zeiler, M.D. ADADELTA: an Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701* **2012**. URL: https://arxiv.org/pdf/1212.5701.pdf.
18. Aïvodji, U.; Arai, H.; Fortineau, O.; Gambs, S.; Hara, S.; Tapp, A. Fairwashing: the Risk of Rationalization. *arXiv preprint arXiv:1901.09749* **2019**. URL: https://arxiv.org/pdf/1901.09749.pdf.
19. Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; Lakkaraju, H. How Can We Fool LIME and SHAP? Adversarial Attacks on Post-hoc Explanation Methods. *arXiv preprint arXiv:1911.02508* **2019**. URL: https://arxiv.org/pdf/1911.02508.pdf.
20. Vaughan, J.; Sudjianto, A.; Brahimi, E.; Chen, J.; Nair, V.N. Explainable Neural Networks Based on Additive Index Models. *arXiv preprint arXiv:1806.01933* **2018**. URL: https://arxiv.org/pdf/1806.01933.pdf.
21. Yang, Z.; Zhang, A.; Sudjianto, A. Enhancing Explainability of Neural Networks Through Architecture Constraints. *arXiv preprint arXiv:1901.03838* **2019**. URL: https://arxiv.org/pdf/1901.03838.pdf.
22. Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics* **2015**, *24*. URL: https://arxiv.org/pdf/1309.6392.pdf.
23. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NIPS)*; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds.; Curran Associates, Inc., 2017; pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.
24. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent Individualized Feature Attribution for Tree Ensembles. In *Proceedings of the 2017 ICML Workshop on Human Interpretability in Machine Learning (WHI 2017)*; Kim, B.; Malioutov, D.M.; Varshney, K.R.; Weller, A., Eds.; ICML WHI 2017, 2017; pp. 15–21. URL: https://openreview.net/pdf?id=ByTKSo-m-.

25. Cohen, J. ***Statistical Power Analysis for the Behavioral Sciences***; Routledge, 2013.
26. Cohen, J. A Power Primer. *Psychological Bulletin* **1992**, *112*, 155. URL: https://www.ime.usp.br/~abe/lista/pdfn45sGokvRe.pdf.
27. Zafar, M.B.; Valera, I.; Gomez Rodriguez, M.; Gummadi, K.P. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification Without Disparate Mistreatment. Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017, pp. 1171–1180. URL: https://arxiv.org/pdf/1610.08452.pdf.
28. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate Intelligible Models with Pairwise Interactions. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013, pp. 623–631. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.352.7682&rep=rep1&type=pdf.
29. Apley, D.W. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *arXiv preprint arXiv:1612.08468* **2016**. URL: https://arxiv.org/pdf/1612.08468.pdf.
30. Shapley, L.S.; Roth, A.E.; others. ***The Shapley value: Essays in Honor of Lloyd S. Shapley***; Cambridge University Press, 1988. URL: http://www.library.fa.ru/files/Roth2.pdf.
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. URL: https://s3.us-east-2.amazonaws.com/hkg-website-assets/static/pages/files/DeepLearning.pdf.
32. Hu, X.; Rudin, C.; Seltzer, M. Optimal Sparse Decision Trees. *arXiv preprint arXiv:1904.12847* **2019**. URL: https://arxiv.org/pdf/1904.12847.pdf.
33. Friedman, J.H.; Popescu, B.E.; others. Predictive Learning Via Rule Ensembles. *The Annals of Applied Statistics* **2008**, *2*, 916–954. URL: https://projecteuclid.org/download/pdfview_1/euclid.aoas/1223908046.
34. Gupta, M.; Cotter, A.; Pfeifer, J.; Voevodski, K.; Canini, K.; Mangylov, A.; Moczydlowski, W.; Van Esbroeck, A. Monotonic Calibrated Interpolated Lookup Tables. *The Journal of Machine Learning Research* **2016**, *17*, 3790–3836. URL: http://www.jmlr.org/papers/volume17/15-243/15-243.pdf.
35. Chen, C.; Li, O.; Barnett, A.; Su, J.; Rudin, C. This Looks Like That: Deep Learning for Interpretable Image Recognition. Proceedings of Neural Information Processing Systems (NeurIPS), 2019. URL: https://arxiv.org/pdf/1806.10574.pdf.
36. Wilkinson, L. Visualizing Big Data Outliers through Distributed Aggregation. *IEEE Transactions on Visualization & Computer Graphics* **2018**. URL: https://www.cs.uic.edu/~wilkinson/Publications/outliers.pdf.
37. Udell, M.; Horn, C.; Zadeh, R.; Boyd, S.; others. Generalized Low Rank Models. *Foundations and Trends® in Machine Learning* **2016**, *9*, 1–118. URL: https://www.nowpublishers.com/article/Details/MAL-055.
38. Holohan, N.; Braghin, S.; Mac Aonghusa, P.; Levacher, K. Diffprivlib: The IBM Differential Privacy Library. *arXiv preprint arXiv:1907.02444* **2019**. URL: https://arxiv.org/pdf/1907.02444.pdf.
39. Ji, Z.; Lipton, Z.C.; Elkan, C. Differential Privacy and Machine Learning: A Survey and Review. *arXiv preprint arXiv:1412.7584* **2014**. URL: https://arxiv.org/pdf/1412.7584.pdf.
40. Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; Erlingsson, Ú. Scalable Private Learning with PATE. *arXiv preprint arXiv:1802.08908* **2018**. URL: https://arxiv.org/pdf/1802.08908.pdf.
41. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 308–318. URL: https://arxiv.org/pdf/1607.00133.pdf.
42. Wachter, S.; Mittelstadt, B.; Russell, C. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. *Harv. JL & Tech.* **2017**, *31*, 841. URL: https://arxiv.org/pdf/1711.00399.pdf.
43. Ancona, M.; Ceolini, E.; Oztireli, C.; Gross, M. Towards Better Understanding of Gradient-based Attribution Methods for Deep Neural Networks. 6th International Conference on Learning Representations (ICLR 2018), 2018. URL: https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/249929/Flow_ICLR_2018.pdf.
44. Wallace, E.; Tuyls, J.; Wang, J.; Subramanian, S.; Gardner, M.; Singh, S. AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models. *arXiv preprint arXiv:1909.09251* **2019**. URL: https://arxiv.org/pdf/1909.09251.pdf.
45. Kamiran, F.; Calders, T. Data Preprocessing Techniques for Classification Without Discrimination. *Knowledge and Information Systems* **2012**, *33*, 1–33. URL: https://bit.ly/2lH95lQ.

46. Zhang, B.H.; Lemoine, B.; Mitchell, M. Mitigating Unwanted Biases with Adversarial Learning. Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. ACM, 2018, pp. 335–340. URL: https://arxiv.org/pdf/1801.07593.pdf.
47. Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; Dwork, C. Learning Fair Representations. International Conference on Machine Learning, 2013, pp. 325–333. URL: http://proceedings.mlr.press/v28/zemel13.pdf.
48. Kamiran, F.; Karim, A.; Zhang, X. Decision Theory for Discrimination-aware Classification. 2012 IEEE 12th International Conference on Data Mining. IEEE, 2012, pp. 924–929. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.722.3030&rep=rep1&type=pdf.
49. Rauber, J.; Brendel, W.; Bethge, M. Foolbox: A Python Toolbox to Benchmark the Robustness of Machine Learning Models. *arXiv preprint arXiv:1707.04131* **2017**. URL: https://arxiv.org/pdf/1707.04131.pdf.
50. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; Matyasko, A.; Behzadan, V.; Hambardzumyan, K.; Zhang, Z.; Juang, Y.L.; Li, Z.; Sheatsley, R.; Garg, A.; Uesato, J.; Gierke, W.; Dong, Y.; Berthelot, D.; Hendricks, P.; Rauber, J.; Long, R. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768* **2018**. URL: https://arxiv.org/pdf/1610.00768.pdf.
51. Amershi, S.; Chickering, M.; Drucker, S.M.; Lee, B.; Simard, P.; Suh, J. Modeltracker: Redesigning Performance Analysis Tools for Machine Learning. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 2015, pp. 337–346. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/amershi.CHI2015.ModelTracker.pdf.
52. Papernot, N. A Marauder's Map of Security and Privacy in Machine Learning: An overview of current and future research directions for making machine learning secure and private. Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security. ACM, 2018. URL: https://arxiv.org/pdf/1811.01134.pdf.
53. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model Cards for Model Reporting. Proceedings of the Conference on Fairness, Accountability, and Transparency. ACM, 2019, pp. 220–229. URL: https://arxiv.org/pdf/1810.03993.pdf.
54. Hall, P. On the Art and Science of Machine Learning Explanations. KDD '19 XAI Workshop Proceedings, 2019. URL: https://arxiv.org/pdf/1810.02909.pdf.
55. Molnar, C. ***Interpretable Machine Learning***; christophm.github.io, 2018. URL: https://christophm.github.io/interpretable-ml-book/.
56. Bracke, P.; Datta, A.; Jung, C.; Sen, S. Machine Learning Explainability in Finance: an Application to Default Risk Analysis **2019**. URL: https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2019/machine-learning-explainability-in-finance-an-application-to-default-risk-analysis.pdf.
57. Hoare, C.A.R. The 1980 ACM Turing Award Lecture. *Communications* **1981**. URL: http://www.cs.fsu.edu/~engelen/courses/COP4610/hoare.pdf.