

Article

Responsible Machine Learning Techniques

Interpretable Models, Post-hoc Explanation, and Discrimination Testing

Navdeep Gill ^{1,†}, Patrick Hall ^{1,3,t,*}, Kim Montgomery ^{1,†}, and Nicholas Schmidt ^{2,†}

¹ H2O.ai

² BLDS, LLC

³ George Washington University

* Correspondence: phall@h2o.ai; nschmidt@bldslc.com

† All authors contributed equally to this work.

Version December 12, 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), 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 provide a template workflow 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, SMD, and other group fairness metrics – §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]. 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. [11], [12]) and artificial neural networks (ANNs) (e.g. [13], [14], [15], [16]), 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], [17], [18]. Here, MGBMs⁶ and XNNs ([19] [20]) 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 more realistic template workflow for future users of similar methods and tools. Presented explanation techniques include PD, ICE, and

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](#).

⁶⁷ Shapley values [12], [21], [22], [23]. PD, ICE, and Shapley values provide direct, global, and local
⁶⁸ summaries and descriptions of constrained models without resorting to the use of intermediary and
⁶⁹ approximate surrogate models. Discussed discrimination testing methods include measures of DI
⁷⁰ with marginal effects, AIR and SMD [2], [24], [25].⁷ Accuracy and other confusion matrix metrics are
⁷¹ also reported by demographic segment [26]. All outlined materials and methods are implemented in
⁷² open source Python code, and are made available in the software resources associated delineated in
⁷³ Subsection 1.9.

⁷⁴ 1.1. Notation

⁷⁵ To facilitate descriptions of data and modeling, explanatory, and discrimination testing techniques,
⁷⁶ notation for input and output spaces, datasets, and models is defined.

⁷⁷ 1.1.1. Spaces

- ⁷⁸ • Input features come from the set \mathcal{X} contained in a P -dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$. An
⁷⁹ arbitrary, potentially unobserved, or future instance of \mathcal{X} is denoted $\mathbf{x}, \mathbf{x} \in \mathcal{X}$.
- ⁸⁰ • Labels corresponding to instances of \mathcal{X} come from the set \mathcal{Y} .
- ⁸¹ • Learned output responses of models are contained in the set $\hat{\mathcal{Y}}$.

⁸² 1.1.2. Datasets

- ⁸³ • The input dataset \mathbf{X} is composed of observed instances of the set \mathcal{X} with a corresponding dataset
⁸⁴ of labels \mathbf{Y} , observed instances of the set \mathcal{Y} .
- ⁸⁵ • Each i -th observation of \mathbf{X} is denoted as $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \dots, x_{P-1}^{(i)}]$, with corresponding i -th labels
⁸⁶ in $\mathbf{Y}, \mathbf{y}^{(i)}$, and corresponding predictions in $\hat{\mathbf{Y}}, \hat{\mathbf{y}}^{(i)}$.
- ⁸⁷ • \mathbf{X} and \mathbf{Y} consist of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- ⁸⁸ • Each j -th input column vector of \mathbf{X} is denoted as $\mathbf{X}_j = [x_j^{(0)}, x_j^{(1)}, \dots, x_j^{(N-1)}]^T$.

⁸⁹ 1.1.3. Models

- ⁹⁰ • A type of ML model g , selected from a hypothesis set \mathcal{H} , is trained to represent an unknown
⁹¹ signal-generating function f observed as \mathbf{X} with labels \mathbf{Y} using a training algorithm $\mathcal{A}: \mathbf{X}, \mathbf{Y} \xrightarrow{\mathcal{A}} g$,
⁹² such that $g \approx f$.
- ⁹³ • g generates learned output responses on the input dataset $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and on the general input
⁹⁴ space $g(\mathcal{X}) = \hat{\mathcal{Y}}$.
- ⁹⁵ • The model to be explained and tested for discrimination is denoted as g .

⁹⁶ 1.2. Simulated Data

⁹⁷ 1.3. Mortgage Data

⁹⁸ The training data contains 33 total features and 144,000 rows, each representing a unique loan, and
⁹⁹ a fold identifier to ensure consistent 5-fold cross-validation accuracy and error measurements across
¹⁰⁰ different types of models. Consumer finance and loan descriptors are used for training. Demographic
¹⁰¹ features are not used in model training. The mortgage test data contains 36,000 loans.

¹⁰² 1.4. Monotonic Gradient Boosting Machine

¹⁰³ MGBMs constrain typical GBM training to consider only tree splits that obey user-defined positive
¹⁰⁴ and negative monotonicity constraints, with respect to each X_j and y independently. The MGBM

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

¹⁰⁵ remains an additive combination of B trees trained by gradient boosting, T_b , and each tree learns a set
¹⁰⁶ of splitting rules that respect monotonicity constraints, Θ_b^{mono} .

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

¹⁰⁷ As in unconstrained GBM, Θ_b^{mono} is selected in a greedy, additive fashion by minimizing a regularized
¹⁰⁸ loss function that considers known target labels, \mathbf{y} , the predictions of all subsequently trained trees in
¹⁰⁹ the MGBM, $g_{b-1}^{\text{MGBM}}(\mathbf{X})$, and a regularization term that penalizes complexity in the current tree, $\Omega(T_b)$.
¹¹⁰ For the b -th iteration, the loss function, \mathcal{L}_b , can generally be defined as:

$$\mathcal{L}_b = \sum_{i=0}^{N-1} l(y^{(i)}, g_{b-1}^{\text{MGBM}}(\mathbf{x}^{(i)}), T_b(\mathbf{x}^{(i)}; \Theta_b^{\text{mono}})) + \Omega(T_b) \quad (2)$$

¹¹¹ In addition to \mathcal{L}_b , g^{MGBM} training is characterized by additional splitting rules and constraints on
¹¹² tree node weights. Each binary splitting rule, $\theta_{b,j,k} \in \Theta_b$, is associated with a feature, X_j , is the
¹¹³ k -th split associated with X_j in T_b , and results in left and right child nodes with a numeric weights,
¹¹⁴ $\{w_{b,j,k,L}, w_{b,j,k,R}\}$. For terminal nodes, $\{w_{b,j,k,L}, w_{b,j,k,R}\}$ can be direct numeric components of some
¹¹⁵ g^{MGBM} prediction. For two values of some feature X_j , $x_j^\alpha \leq x_j^\beta$, g^{MGBM} is positive monotonic with
¹¹⁶ 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
¹¹⁷ ensure positive monotonicity in Θ_b , where the prediction for each value results in $T_b(x_j^\alpha; \Theta_b) = w_\alpha$
¹¹⁸ and $T_b(x_j^\beta; \Theta_b) = w_\beta$.

- ¹¹⁹ 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
¹²⁰ 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
¹²¹ considered.
- ¹²² 2. For any subsequent left child node involving X_j , any $\theta_{b,j,k \geq 1}$ resulting in $T(x_j; \theta_{b,j,k \geq 1}) =$
¹²³ $\{w_{b,j,k \geq 1,L}, w_{b,j,k \geq 1,R}\}$ where $w_{b,j,k \geq 1,L} > w_{b,j,k \geq 1,R}$, is not considered.
- ¹²⁴ 3. Moreover, for any subsequent left child node involving X_j , $T(x_j; \theta_{b,j,k \geq 1}) = \{w_{b,j,k \geq 1,L}, w_{b,j,k \geq 1,R}\}$,
¹²⁵ $\{w_{b,j,k \geq 1,L}, w_{b,j,k \geq 1,R}\}$ are bound by the associated $\theta_{b,j,k-1}$ set of node weights,
¹²⁶ $\{w_{b,j,k-1,L}, w_{b,j,k-1,R}\}$, such that $\{w_{b,j,k \geq 1,L}, w_{b,j,k \geq 1,R}\} < \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.
- ¹²⁷ 4. (1) and (2) are also applied to all right child nodes, except that for right child nodes $w_{b,j,k,L} \leq$
¹²⁸ $w_{b,j,k,R}$ and $\{w_{b,j,k \geq 1,L}, w_{b,j,k \geq 1,R}\} \geq \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.

¹²⁹ Note that for any one X_j and $T_b \in g^{\text{MGBM}}$ left subtrees will always produce lower predictions than
¹³⁰ right subtrees, and that any $g^{\text{MGBM}}(\mathbf{x})$ is an addition of each T_b output, with the application of
¹³¹ a monotonic logit or softmax link function for classification problems. Moreover, each tree's root
¹³² node corresponds to some constant node weight that by definition obeys monotonicity constraints,
¹³³ $T(x_j^\alpha; \theta_{b,0}) = T(x_j^\beta; \theta_{b,0}) = w_{b,0}$. Together these additional splitting rules and node weight constraints
¹³⁴ 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.
¹³⁵ $g^{\text{MGBM}}(x_j^\alpha) \geq g^{\text{MGBM}}(x_j^\beta) \forall x_j^\alpha \leq x_j^\beta \in X_j$, left and right splitting rules and node weight constraints are
¹³⁶ switched. Also consider that MGBM models with independent monotonicity constraints between some
¹³⁷ X_j and \mathbf{y} likely restrict non-monotonic interactions between multiple X_j . Moreover, if monotonicity
¹³⁸ constraints are not applied to all $X_j \in \mathbf{X}$, any strong non-monotonic signal in training data associated
¹³⁹ with some important X_j maybe forced onto some other arbitrary unconstrained X_j under some g^{MGBM}
¹⁴⁰ models, compromising the end goal of interpretability.

¹⁴¹ Herein, two g^{MGBM} models are trained. One on the simulated data and one on the mortgage
¹⁴² data. In both cases, positive and negative monotonic constraints for each X_j are selected using domain
¹⁴³ knowledge, random grid search is used to determine other hyperparameters, and five-fold cross
¹⁴⁴ validation and test partitions are used for model assessment. For exact parameterization of the two
¹⁴⁵ g^{MGBM} models, see the software resources referenced in Subsection 1.9.

¹⁴⁶ 1.5. Explainable Neural Network

¹⁴⁷ XNNs are an alternative formulation of additive index models in which the ridge functions are
¹⁴⁸ neural networks [19]. XNNs also bear a strong resemblance to generalized additive models (GAMs)
¹⁴⁹ and so-called explainable boosting machines (EBMs or GA²M), i.e. GAMs which consider main
¹⁵⁰ effects and a small number of 2-way interactions and may also incorporate boosting into their training
¹⁵¹ [12], [27]. Hence, XNNs enable users to tailor interpretable neural network architectures to a given
¹⁵² prediction problem and to visualize model behavior by plotting ridge functions. XNNs are composed
¹⁵³ of a global bias term, μ_0 , K individually specified neural networks, n_k with scale parameters γ_k , and
¹⁵⁴ the inputs to each n_k are themselves a linear combination of modeling inputs, $\sum_j \beta_{k,j} x_j$.

$$g^{\text{XNN}}(\mathbf{x}) = \mu_0 + \sum_{k=0}^{K-1} \gamma_k n_k \left(\sum_{j=0}^{J=\mathcal{P}-1} \beta_{k,j} x_j \right) \quad (3)$$

¹⁵⁵ g^{XNN} is comprised of 3 meta-layers:

- ¹⁵⁶ 1. The first and deepest meta-layer, composed of K linear $\sum_j \beta_{k,j} x_j$ hidden units, is known as the
¹⁵⁷ *projection layer* and is fully connected to each input feature, X_j . Each hidden unit in the projection
¹⁵⁸ layer may optionally include a bias term.
- ¹⁵⁹ 2. The second meta-layer contains K hidden and separate n_k ridge functions, or *subnetworks*. Each
¹⁶⁰ n_k is a neural network, which can be parameterized to suit a given modeling task. To facilitate
¹⁶¹ direct interpretation and visualization, the input to each subnetwork is the 1-dimensional output
¹⁶² of its associated projection layer hidden unit, $\sum_j \beta_{k,j} x_j$. Each n_k can contain several bias terms.
- ¹⁶³ 3. The output meta-layer, called the *combination layer*, is another linear unit comprised of a global bias
¹⁶⁴ 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,
¹⁶⁵ subnetwork output is restricted to 1-dimension for interpretation and visualization purposes.

¹⁶⁶ Here, each g^{XNN} is trained by mini-batch stochastic gradient descent (SGD) on the simulated
¹⁶⁷ data and mortgage data. Each g^{XNN} is assessed in five training folds and in a test data partition.
¹⁶⁸ L_1 regularization is applied to both the projection and combination layers to induce a sparse and
¹⁶⁹ interpretable model, where each n_k subnetwork and corresponding combination layer γ_k are ideally
¹⁷⁰ associated with an important X_j or combination thereof. The g^{XNN} models appear highly sensitive to
¹⁷¹ weight initialization and batch size. Be aware that g^{XNN} model architectures may require manual and
¹⁷² judicious feature selection due to long training times. For more details regarding g^{XNN} training, see
¹⁷³ the software resources in Subsection 1.9.

¹⁷⁴ 1.6. Partial Dependence and Individual Conditional Expectation

¹⁷⁵ PD plots are a widely-used method for describing and plotting the average predictions of a
¹⁷⁶ complex model g across some partition of data \mathbf{X} for some interesting input feature X_j [12]. ICE plots
¹⁷⁷ are a newer method that describes the local behavior of g for a single instance $\mathbf{x} \in \mathcal{X}$ [21]. PD and ICE
¹⁷⁸ can be overlaid in the same plot to compensate for known weaknesses of PD (e.g. inaccuracy in the
¹⁷⁹ presence of strong interactions and correlations [21], [28]), to identify interactions modeled by g , and
¹⁸⁰ to create a holistic global and local portrait of the predictions for some g and X_j [21].

¹⁸¹ Following Friedman *et al.* [12] a single feature $X_j \in \mathbf{X}$ and its complement set $\mathbf{X}_{\mathcal{P} \setminus j} \in \mathbf{X}$ (where
¹⁸² $X_j \cup \mathbf{X}_{\mathcal{P} \setminus j} = \mathbf{X}$) is considered. $\text{PD}(X_j, g)$ for a given feature X_j is estimated as the average output of
¹⁸³ 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
¹⁸⁴ 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
¹⁸⁵ 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
¹⁸⁶ usually plotted over some set of constants $x \in \mathcal{X}$, as displayed in Section 2. Due to known problems
¹⁸⁷ for PD in the presence of strong correlation and interactions, PD should not be used alone. PD should
¹⁸⁸ always be paired with ICE or be replaced with accumulated local effect (ALE) plots [21], [28].

189 1.7. *Shapley Values*

190 Shapley explanations are a class of additive, locally accurate feature contribution measures with
191 long-standing theoretical support [22], [29]. Shapley explanations are the only possible locally accurate
192 and globally consistent feature contribution values, meaning that Shapley explanation values for
193 input features always sum to $g(\mathbf{x})$ for some $\mathbf{x} \in \mathcal{X}$ and that Shapley explanation values should never
194 decrease in magnitude for some x_j when g is changed such that x_j truly makes a stronger contribution
195 to $g(\mathbf{x})$ [22], [23]. For some instance $\mathbf{x} \in \mathcal{X}$, Shapley explanations take the form:

$$g(\mathbf{x}) = \phi_0 + \sum_{j=0}^{j=\mathcal{P}-1} \phi_j \mathbf{z}_j \quad (4)$$

196 In Equation 4, $\mathbf{z} \in \{0, 1\}^{\mathcal{P}}$ is a binary representation of \mathbf{x} where 0 indicates missingness. Each ϕ_j is the
197 local feature contribution value associated with x_j and ϕ_0 is the average of $g(\mathbf{X})$. Each ϕ_j is a weighted
198 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
199 every subset of features S not including j , $S \subseteq \mathcal{P} \setminus \{j\}$, where g_x incorporates the mapping between \mathbf{x}
200 and the binary vector \mathbf{z} .

$$\phi_j = \sum_{S \subseteq \mathcal{P} \setminus \{j\}} \frac{|S|!(\mathcal{P} - |S| - 1)!}{\mathcal{P}!} [g_x(S \cup \{j\}) - g_x(S)] \quad (5)$$

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

205 Shapley values can be estimated in different ways, many of which are intractable for datasets
206 with large \mathcal{P} . Tree SHAP is a specific implementation of Shapley explanations that relies on traversing
207 internal decision tree structures to efficiently estimate the contribution of each x_j for some $g(\mathbf{x})$ [23].
208 Tree SHAP (SHapley Additive exPlanations) has been implemented efficiently in popular gradient
209 boosting libraries such as `h2o`, `LightGBM`, and `XGBoost`, and Tree SHAP is used to calculate accurate
210 and consistent global and local feature importance for MGBM models in Sections 1 and 2. Deep SHAP
211 is an approximate Shapley value technique that creates SHAP values for ANNs [22]. Deep SHAP
212 is implemented in the `shap` package and is used to generate SHAP values for the two g^{XNN} models
213 discussed in Sections 1 and 2.

214 1.8. *Discrimination Metrics and Test Description*

215 1.9. *Software Resources*

216 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
217 manipulation, `h2o`, `keras`, `shap`, and `tensorflow` for modeling, explanation, and discrimination
218 testing, and `matplotlib` and `seaborn` for plotting.

220 2. **Results**

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

²²⁵ 2.1. Simulated Data Results

²²⁶ 2.1.1. Constrained vs. Unconstrained Model Fit Assessment

Table 1. Fit metrics for g^{GBM} , g^{MGBM} , g^{ANN} , and g^{XNN} on the simulated test data. Accuracy is measured at the best F1 threshold for each model.

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}				
g^{XNN}	0.758	0.851	0.479	0.397

²²⁷ 2.1.2. Interpretability and Post-hoc Explanation Results

Figure 1. Global mean Tree SHAP feature importance for g^{MGBM} on the simulated test data.

Figure 2. PD, ICE across deciles, and histograms for the three most important input features for g^{MGBM} on the simulated test data.

Figure 3. Mean Tree SHAP values across quintiles for the three most important input features for g^{MGBM} on the simulated test data.

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

Figure 5. Ridge functions for the three most important input features for g^{XNN} on the simulated test data.

Figure 6. Mean Deep SHAP values across quintiles for the three most important input features for g^{XNN} on the simulated test data.

²²⁸ 2.1.3. Discrimination Testing Results

²²⁹ 2.2. Mortgage Data Results

²³⁰ 2.2.1. Constrained vs. Unconstrained Model Fit Assessment

Table 2. Fit metrics for g^{GBM} , g^{MGBM} , g^{ANN} , and g^{XNN} on the mortgage test data. Accuracy is measured at the best F1 threshold for each model.

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}				
g^{XNN}	0.869	0.868	0.233	0.263

²³¹ 2.2.2. Interpretability and Post-hoc Explanation Results

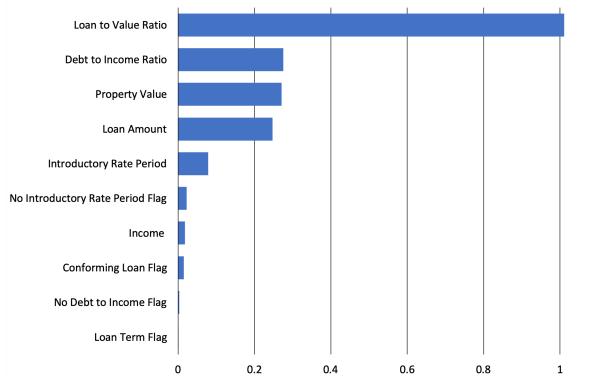


Figure 7. Global mean absolute Tree SHAP feature importance for g^{MGBM} on the mortgage test data. Tree SHAP values are reported in the margin space, prior to the application of the logit link function.

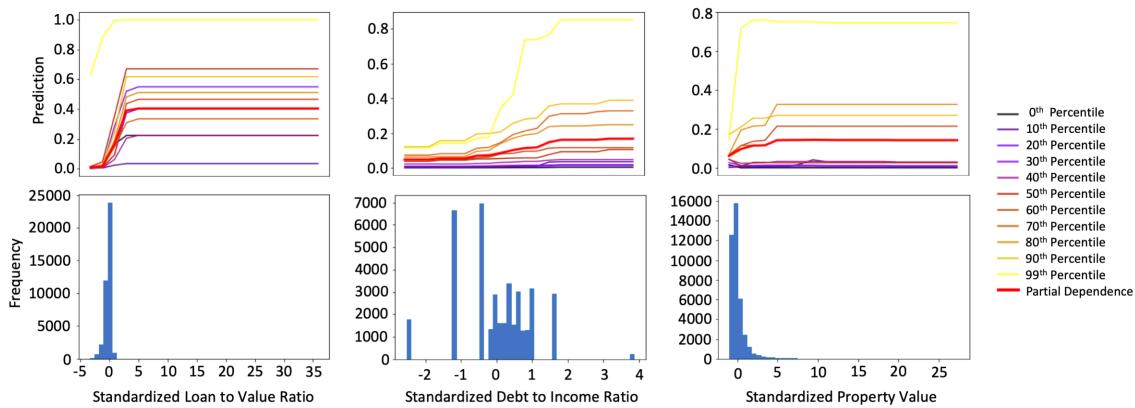


Figure 8. PD, ICE across selected percentiles, and histograms for the three most important input features for g^{MGBM} on the mortgage test data.

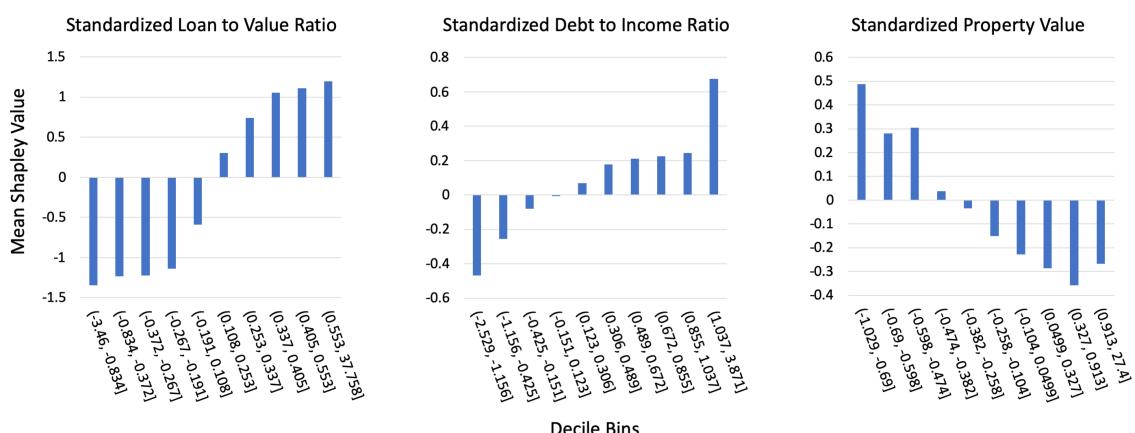


Figure 9. Mean Tree SHAP values across quintiles for the three most important input features for g^{MGBM} on the mortgage test data.

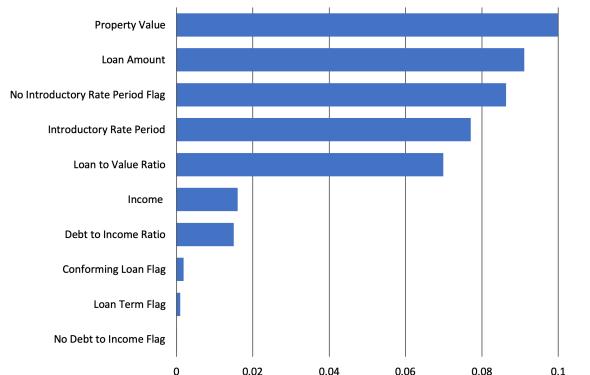


Figure 10. Global mean absolute Deep SHAP feature importance for g^{XNN} on the mortgage test data. Deep SHAP values are reported in the probability space, after the application of the logit link function.

Figure 11. Ridge functions for the three most important input features for g^{XNN} on the mortgage test data.

Figure 12. Mean Deep SHAP values across quintiles for the three most important input features for g^{XNN} on the mortgage test data.

232 2.2.3. Discrimination Testing Results

233 3. Discussion

234 3.1. The Burgeoning Python Ecosystem for Responsible Machine Learning

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

⁸ See: https://github.com/jphall1663/hc_ml for details regarding the work flow in Figure 13.

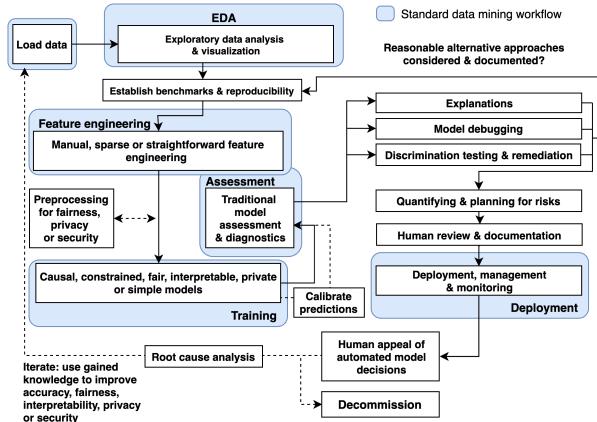


Figure 13. A diagram of a proposed holistic ML work flow in which interpretable models, post-hoc explanations, discrimination testing and remediation techniques, and other review and appeal mechanisms can create an understandable and trustworthy ML system.

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 GA²M 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 [30], [31], [32], [33].^{9,10} Additional, relevant references and Python functionality include:

- **Exploratory data analysis (EDA):** H2OAggregatorEstimator in [h2o](#) [34].
- **Sparse feature extraction:** H2OGeneralizedLowRankEstimator in [h2o](#) [35].
- **Privacy preprocessing and private models:** differential privacy and private models in [diffprivlib](#) and [tensorflow/privacy](#) [36], [37], [38], [39].
- **Post-hoc explanation:** structured data explanations with [alibi](#) and [PDPbox](#), image classification explanations with [DeepExplain](#), and natural language explanations with [allenNLP](#) [40], [41], [42].
- **Discrimination testing:** with [aequitas](#) and [Themis](#).
- **Discrimination remediation:** Reweighting, adversarial de-biasing, learning fair representations, and reject option classification with [AIF360](#) [43], [44], [45], [46].
- **Model debugging:** with [foolbox](#), [SALib](#), [tensorflow/cleverhans](#), and [tensorflow/model-analysis](#) [47], [48], [49], [50].
- **Model documentation:** models cards [51], e.g. https://github.com/openai/gpt-2/blob/master/model_card.md, <https://modelcards.withgoogle.com/object-detection>.

See: <https://github.com/jphall1663/awesome-machine-learning-interpretability> for a longer, curated list 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

⁹ Optimal sparse decision trees: <https://github.com/xiyanghu/OSDT>.

¹⁰ *This looks like that* interpretable deep learning: <https://github.com/cfchen-duke/ProtoPNet>.

domain. Post-hoc explanations generally help describe and summarize mechanisms and decisions, 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 [52]. 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.

288 3.3. Impact of Discrimination Testing on Model Use and Adoption

289 3.4. Viable Discrimination Remediation Approaches

290 3.5. Intersectionality of Interpretability, Explainability, Discrimination, and Security in ML

291 The black-box nature of ML, the perpetuation or exacerbation of discrimination by ML, or the
292 security vulnerabilities inherent in ML are each serious and difficult problems on their own. However,
293 evidence is mounting that these harms can also manifest as complex intersectional challenges, e.g. the
294 *fairwashing* or *scaffolding* of biased models with ML explanations, the privacy harms of ML explanations,
295 or the adversarial poisoning of ML models to become discriminatory [8], [17], [18].^{15,16,17} Again, this
296 text makes no claims that the opacity, discrimination, or security problems in ML have been solved,
297 even treated as independent problems. Instead, this text aims to highlight these issues as both singular
298 entities and non-static intersectional phenomena. Practitioners should of course consider the discussed
299 interpretable modeling, post-hoc explanation, and discrimination testing approaches as at least partial
300 remedies to the black-box and discrimination issues in ML. However, they should also consider that
301 explanations can ease model stealing, data extraction, and membership inference attacks and that
302 explanations can mask ML discrimination. Additionally, high-stakes, human-centered, or regulated
303 ML systems should generally be built and tested with robustness to adversarial attacks as a primary
304 design consideration, and specifically to prevent ML models from being poisoned or otherwise altered
305 to become discriminatory. Accuracy, discrimination, and security characteristics of a system can change
306 over time as well. Simply testing for these problems at training time, as presented in Sections 1 and 2,

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 [53]. 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 [54].

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 [50]. High system complexity is sometimes considered a mitigating influence as well [55]. 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: [Andrew Burt HBR article - upcoming](#).

307 is not adequate for high-stakes, human-centered, or regulated ML systems. Accuracy, discrimination,
308 and security should be monitored in real-time and over time, as long as a model is deployed.

309 4. Conclusion

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

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

327 **Funding:** This work received no external funding.

328 **Acknowledgments:** Wen Phan for work in formalizing notation. **BLDS editor** for editing. Andrew Burt for ideas
329 around ML transparency, model debugging, and the transparency paradox.

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

332 Abbreviations

333 The following abbreviations are used in this text: AI – artificial intelligence, AIR - adverse impact ratio, ALE -
334 accumulated local effect, ANN – artificial neural network, DI – disparate impact, EBM or GA²M – explainable
335 boosting machine, i.e. variants GAMs that consider two-way interactions and may incorporate boosting into
336 training, ECOA - Equal Credit Opportunity Act, EDA – exploratory data analysis, EU – European Union, FCRA
337 – Fair Credit Reporting Act, GAM – generalized additive model, GBM – gradient boosting machine, GDPR -
338 General Data Protection Regulation, HMDA – Home Mortgage Disclosure Act ICE – individual conditional
339 expectation, MGBM – monotonic gradient boosting machine, ML – machine learning, PD – partial dependence,
340 SGD – stochastic gradient descent, SHAP – Shapley additive explanation, SMD - standardized mean difference,
341 SR – supervision and regulation, US – United States, XNN – explainable neural network.

342 References

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

- 354 5. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J. The Security of Machine Learning. *Machine Learning* **2010**,
355 *81*, 121–148. URL: http://people.ischool.berkeley.edu/~tygar/papers/SML/sec_mach_learn_journal.pdf.
- 356 6. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing Machine Learning Models via
357 Prediction APIs. 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 601–618. URL:
358 https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_tramer.pdf.
- 359 7. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning
360 Models. 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 3–18. URL: <https://arxiv.org/pdf/1610.05820.pdf>.
- 361 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>.
- 362 9. Williams, M.; others. *Interpretability*; Fast Forward Labs, 2017. URL: <https://www.cloudera.com/products/fast-forward-labs-research.html>.
- 363 10. Friedman, J.H.; others. Multivariate Adaptive Regression Splines. *The annals of statistics* **1991**, *19*, 1–67.
364 URL: https://projecteuclid.org/download/pdf_1/euclid-aos/1176347963.
- 365 11. Friedman, J.H. Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics* **2001**,
366 pp. 1189–1232. URL: https://projecteuclid.org/download/pdf_1/euclid-aos/1013203451.
- 367 12. Friedman, J.H.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer: New York, 2001.
368 URL: https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf.
- 369 13. Recht, B.; Re, C.; Wright, S.; Niu, F. HOGWILD: A Lock-free Approach to Parallelizing
370 Stochastic Gradient Descent. Advances in Neural Information Processing Systems (NIPS), 2011,
371 pp. 693–701. URL: <https://papers.nips.cc/paper/4390-hogwild-a-lock-free-approach-to-parallelizing-stochastic-gradient-descent.pdf>.
- 372 14. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving Neural Networks
373 by Preventing Co-adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580* **2012**. URL: <https://arxiv.org/pdf/1207.0580.pdf>.
- 374 15. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the Importance of Initialization and Momentum
375 in Deep Learning. International Conference on Machine Learning, 2013, pp. 1139–1147. URL: <http://proceedings.mlr.press/v28/sutskever13.pdf>.
- 376 16. Zeiler, M.D. ADADELTA: an Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701* **2012**. URL:
377 <https://arxiv.org/pdf/1212.5701.pdf>.
- 378 17. Aïvodji, U.; Arai, H.; Fortineau, O.; Gambs, S.; Hara, S.; Tapp, A. Fairwashing: the Risk of Rationalization.
379 *arXiv preprint arXiv:1901.09749* **2019**. URL: <https://arxiv.org/pdf/1901.09749.pdf>.
- 380 18. Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; Lakkaraju, H. How Can We Fool LIME and SHAP? Adversarial
381 Attacks on Post-hoc Explanation Methods. *arXiv preprint arXiv:1911.02508* **2019**. URL: <https://arxiv.org/pdf/1911.02508.pdf>.
- 382 19. Vaughan, J.; Sudjianto, A.; Brahimi, E.; Chen, J.; Nair, V.N. Explainable Neural Networks Based on Additive
383 Index Models. *arXiv preprint arXiv:1806.01933* **2018**. URL: <https://arxiv.org/pdf/1806.01933.pdf>.
- 384 20. Yang, Z.; Zhang, A.; Sudjianto, A. Enhancing Explainability of Neural Networks Through Architecture
385 Constraints. *arXiv preprint arXiv:1901.03838* **2019**. URL: <https://arxiv.org/pdf/1901.03838.pdf>.
- 386 21. Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking Inside the Black Box: Visualizing Statistical
387 Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*
388 **2015**, *24*. URL: <https://arxiv.org/pdf/1309.6392.pdf>.
- 389 22. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural
390 Information Processing Systems (NIPS)*; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.;
391 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>.
- 392 23. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent Individualized Feature Attribution for Tree Ensembles.
393 In *Proceedings of the 2017 ICML Workshop on Human Interpretability in Machine Learning (WHI 2017)*; Kim,
394 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->.
- 395 24. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Routledge, 2013.
- 396 25. Cohen, J. A Power Primer. *Psychological Bulletin* **1992**, *112*, 155. URL: <https://www.ime.usp.br/~abe/lista/pdfn45sGokvRe.pdf>.

- 407 26. Zafar, M.B.; Valera, I.; Gomez Rodriguez, M.; Gummadi, K.P. Fairness Beyond Disparate Treatment &
408 Disparate Impact: Learning Classification Without Disparate Mistreatment. Proceedings of the 26th
409 International Conference on World Wide Web. International World Wide Web Conferences Steering
410 Committee, 2017, pp. 1171–1180. URL: <https://arxiv.org/pdf/1610.08452.pdf>.
- 411 27. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate Intelligible Models with Pairwise Interactions.
412 Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data
413 Mining. ACM, 2013, pp. 623–631. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.352.7682&rep=rep1&type=pdf>.
- 414 28. 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>.
- 415 29. Shapley, L.S.; Roth, A.E.; others. *The Shapley value: Essays in Honor of Lloyd S. Shapley*; Cambridge
416 University Press, 1988. URL: <http://www.library.fa.ru/files/Roth2.pdf>.
- 417 30. Hu, X.; Rudin, C.; Seltzer, M. Optimal Sparse Decision Trees. *arXiv preprint arXiv:1904.12847* **2019**. URL:
418 <https://arxiv.org/pdf/1904.12847.pdf>.
- 419 31. Friedman, J.H.; Popescu, B.E.; others. Predictive Learning Via Rule Ensembles. *The Annals of Applied
420 Statistics* **2008**, *2*, 916–954. URL: https://projecteuclid.org/download/pdfview_1/euclid.aoas/1223908046.
- 421 32. Gupta, M.; Cotter, A.; Pfeifer, J.; Voevodski, K.; Canini, K.; Mangylov, A.; Moczydłowski, W.; Van Esbroeck,
422 A. Monotonic Calibrated Interpolated Lookup Tables. *The Journal of Machine Learning Research* **2016**,
423 *17*, 3790–3836. URL: <http://www.jmlr.org/papers/volume17/15-243/15-243.pdf>.
- 424 33. Chen, C.; Li, O.; Barnett, A.; Su, J.; Rudin, C. This Looks Like That: Deep Learning for Interpretable
425 Image Recognition. Proceedings of Neural Information Processing Systems (NeurIPS), 2019. URL:
426 <https://arxiv.org/pdf/1806.10574.pdf>.
- 427 34. Wilkinson, L. Visualizing Big Data Outliers through Distributed Aggregation. *IEEE Transactions on
428 Visualization & Computer Graphics* **2018**. URL: <https://www.cs.uic.edu/~wilkinson/Publications/outliers.pdf>.
- 429 35. Udell, M.; Horn, C.; Zadeh, R.; Boyd, S.; others. Generalized Low Rank Models. *Foundations and Trends®
430 in Machine Learning* **2016**, *9*, 1–118. URL: <https://www.nowpublishers.com/article/Details/MAL-055>.
- 431 36. Holohan, N.; Braghin, S.; Mac Aonghusa, P.; Levacher, K. Diffprivlib: The IBM Differential Privacy Library.
432 *arXiv preprint arXiv:1907.02444* **2019**. URL: <https://arxiv.org/pdf/1907.02444.pdf>.
- 433 37. 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>.
- 434 38. Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; Erlingsson, Ú. Scalable Private Learning
435 with PATE. *arXiv preprint arXiv:1802.08908* **2018**. URL: <https://arxiv.org/pdf/1802.08908.pdf>.
- 436 39. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with
437 Differential Privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications
438 Security. ACM, 2016, pp. 308–318. URL: <https://arxiv.org/pdf/1607.00133.pdf>.
- 439 40. Wachter, S.; Mittelstadt, B.; Russell, C. Counterfactual Explanations without Opening the Black Box:
440 Automated Decisions and the GDPR. *Harv. JL & Tech.* **2017**, *31*, 841. URL: <https://arxiv.org/pdf/1711.00399.pdf>.
- 441 41. Ancona, M.; Ceolini, E.; Oztireli, C.; Gross, M. Towards Better Understanding of Gradient-based Attribution
442 Methods for Deep Neural Networks. 6th International Conference on Learning Representations (ICLR
443 2018), 2018. URL: https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/249929/Flow_ICLR_2018.pdf.
- 444 42. Wallace, E.; Tuyls, J.; Wang, J.; Subramanian, S.; Gardner, M.; Singh, S. AllenNLP Interpret: A Framework
445 for Explaining Predictions of NLP Models. *arXiv preprint arXiv:1909.09251* **2019**. URL: <https://arxiv.org/pdf/1909.09251.pdf>.
- 446 43. Kamiran, F.; Calders, T. Data Preprocessing Techniques for Classification Without Discrimination.
447 *Knowledge and Information Systems* **2012**, *33*, 1–33. URL: <https://bit.ly/2lH95IQ>.
- 448 44. Zhang, B.H.; Lemoine, B.; Mitchell, M. Mitigating Unwanted Biases with Adversarial Learning.
449 Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. ACM, 2018, pp. 335–340. URL:
450 <https://arxiv.org/pdf/1801.07593.pdf>.

- 458 45. Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; Dwork, C. Learning Fair Representations. International
459 Conference on Machine Learning, 2013, pp. 325–333. URL: <http://proceedings.mlr.press/v28/zemel13.pdf>.
- 460 46. Kamiran, F.; Karim, A.; Zhang, X. Decision Theory for Discrimination-aware Classification. 2012 IEEE 12th
461 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>.
- 462 47. Rauber, J.; Brendel, W.; Bethge, M. Foolbox: A Python Toolbox to Benchmark the Robustness of Machine
463 Learning Models. *arXiv preprint arXiv:1707.04131* 2017. URL: <https://arxiv.org/pdf/1707.04131.pdf>.
- 464 48. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown,
465 T.; Roy, A.; Matyasko, A.; Behzadan, V.; Hamardzumyan, K.; Zhang, Z.; Juang, Y.L.; Li, Z.; Sheatsley,
466 R.; Garg, A.; Uesato, J.; Gierke, W.; Dong, Y.; Berthelot, D.; Hendricks, P.; Rauber, J.; Long, R. Technical
467 Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768* 2018. URL:
468 <https://arxiv.org/pdf/1610.00768.pdf>.
- 469 49. Amershi, S.; Chickering, M.; Drucker, S.M.; Lee, B.; Simard, P.; Suh, J. Modeltracker: Redesigning
470 Performance Analysis Tools for Machine Learning. Proceedings of the 33rd Annual ACM Conference on
471 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>.
- 472 50. Papernot, N. A Marauder’s Map of Security and Privacy in Machine Learning: An overview of current and
473 future research directions for making machine learning secure and private. Proceedings of the 11th ACM
474 Workshop on Artificial Intelligence and Security. ACM, 2018. URL: <https://arxiv.org/pdf/1811.01134.pdf>.
- 475 51. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru,
476 T. Model Cards for Model Reporting. Proceedings of the Conference on Fairness, Accountability, and
477 Transparency. ACM, 2019, pp. 220–229. URL: <https://arxiv.org/pdf/1810.03993.pdf>.
- 478 52. Hall, P. On the Art and Science of Machine Learning Explanations. KDD ’19 XAI Workshop Proceedings,
479 2019. URL: <https://arxiv.org/pdf/1810.02909.pdf>.
- 480 53. Molnar, C. *Interpretable Machine Learning*; christophm.github.io, 2018. URL: <https://christophm.github.io/interpretable-ml-book/>.
- 481 54. Bracke, P.; Datta, A.; Jung, C.; Sen, S. Machine Learning Explainability in Finance: an Application to Default
482 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>.
- 483 55. Hoare, C.A.R. The 1980 ACM Turing Award Lecture. *Communications* 1981. URL: <http://www.cs.fsu.edu/~engelen/courses/COP4610/hoare.pdf>.

484 490 © 2019 by the authors. Submitted to *Information* for possible open access publication
491 under the terms and conditions of the Creative Commons Attribution (CC BY) license
492 (<http://creativecommons.org/licenses/by/4.0/>).