Article

Responsible Machine Learning Techniques

Interpretable Models, Post-hoc Explanation, and Disparate Impact 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 7, 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 realistic, 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 straightforward group fairness measures. By combining innovations in 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] Even if responsible AI feels like a somewhat broad and amorphous notion, at least one aspect should be crystal clear: ML models, a common application of AI, have problems that responsible practitioners should likely attempt to remediate. ML models can be inaccurate, noncompliant, and unappealable black-boxes, even with the application of newer post-hoc explanation techniques [1].[2] ML models can perpetuate and exacerbate discrimination [2], [3], [4], [5]. ML models can be hacked, resulting in manipulated model outcomes

---

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".

or the exposure of proprietary intellectual property or sensitive training data [6], [7], [8]. Though 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), this text does 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 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 outline of proposed additional steps to increase human trust and understanding in ML.

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**: – §1.8
- **Software resources**: GitHub repository for this text; utilized & useful Python packages – §1.9

To provide a sense of accuracy differences, this manuscript compares the performance of more interpretable constrained ML models and less interpretable unconstrained ML models on simulated data and on 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] 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.

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, MGBMs[6] and XNNs ([19] [20]) will serve as those more interpretable models for subsequent explanatory and discrimination analysis. 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 U.S., interpretable models, explanations, disparate impact 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 (E.U.) Greater Data Privacy Regulation (GDPR) Article 22 [9].

[5] See: Mortgage data (HMDA).

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

Shapley values [12], [21], [22], [23]. 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 measures of DI with marginal effects, the adverse impact ratio (AIR), and the standardized mean difference (SMD) [3], [24], [25].[7] Accuracy and other confusion matrix metrics are also reported by demographic segment[26]. All outlined materials and methods are implemented in open source Python code, and are made available in the software resources associated with this text.

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 $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \dots, x_{P-1}^{(i)}]$, 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: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- Each j-th input column vector of $\mathbf{X}$ is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \dots, x_j^{(N-1)}]^T$.

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 testing is denoted as g.

1.2. Simulated Data

1.3. Mortgage Data

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

1.4. Monotonically Constrained Gradient Boosting Machine

MGBMs constrain typical GBM training to consider only tree splits that obey user-defined positive and negative monotonicity constraints. The MGBM remains an additive combination of B trees

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

trained by gradient boosting, T_b, but each tree learns a set of splitting rules that respect monotonicity constraints, Θ_b^{mono}.

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

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:

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

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, $\{w_{b,j,k,L}, w_{b,j,kR}\}$. For terminal nodes, $\{w_{b,j,k,L}, w_{b,j,kR}\}$ can be direct numeric components of some g^{MGBM} prediction. For two values of some feature X_j, $x_j^{\alpha} \leq x_j^{\beta}$, where the prediction for each value results in $T_b(x_j^{\alpha}; \Theta_b) = w_{\alpha}$ and $T_b(x_j^{\beta}; \Theta_b) = w_{\beta}$, Θ_b is restricted to be positive monotonic w.r.t. X_j by the following rules and constraints.

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, $T(x_j; \theta_{b,j,0}) = \{w_{b,j,0,L}, w_{j,0,R}\}$ 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 parent set of node weights, $\{w_{b,j,k-1,L}, w_{b,j,k-1,R}\}$, such that $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} \leq \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.
4. (1) and (2) are also applied to all right child nodes, except that for right child nodes $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} \geq \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.

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, $T(x_j^{\alpha}; \theta_{b,0}) = T(x_j^{\beta}; \theta_{b,j,0}) = w_{b,0}$. 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.

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 the sign of the Pearson correlation between each X_j and y, 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 MGBM models, see the software resources referenced in Subsection 1.9. Also consider that MGBM models with one-dimensional monotonicity constraints between some X_j and $\hat{Y}$ 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.

1.5. Explainable Neural Network

XNNs are an alternative formulation of additive index models in which the ridge functions are neural networks [19]. 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 incorporate boosting in their training [12], [27]. 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, $\sum_{j=0}^{J} \beta_{k,j} x_j$.

$$g^{\text{XNN}}(\mathbf{x}) = \mu_0 + \sum_{k=1}^{K} \gamma_k n_k \Big(\sum_{j=1}^{J} \beta_{k,j} x_j\Big) \tag{3}$$

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.
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 suite a given modeling task. To facilitate direct visualization, the input to each subnetwork is the 1-dimensional output of its associated projection layer hidden unit, $\sum_j \beta_{k,j} x_j$.
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 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 are highly sensitive to weight initialization and manual feature selection is informed by Shapley feature importance from the g^{MGBM}. For more details regarding g^{XNN} training, see the software resources in Subsection 1.9. Be aware that g^{XNN} model architectures may restrict important interactions, appear highly sensitive to initialization routines, and require manual and judicious feature selection due to burdensome training times.

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 [12]. ICE plots are a newer method that describes the local behavior of g for a single instance $\mathbf{x} \in \mathcal{X}$ [21]. 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 [21], [28]), to identify interactions modeled by g, and to create a holistic global and local portrait of the predictions of a complex model for some X_j [21].

Following Friedman *et al.* [12] a single feature $X_j \in \mathbf{X}$ and its complement set $\mathbf{X}_{(-j)} \in \mathbf{X}$ (where $X_j \cup \mathbf{X}_{(-j)} = \mathbf{X}$) is considered. 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 $\mathbf{X}_{(-j)}$ is left unchanged. 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 $\mathbf{x} \in \mathbf{X}_{(-j)}$ 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 always be paired with ICE or be replaced with accumulated local effect (ALE) plots [21], [28].

1.7. Shapley Values

Shapley explanations are a class of additive, locally accurate feature contribution measures with long-standing theoretical support [22], [29]. 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 can never decrease in magnitude for some x_j when g is changed such that x_j truly makes a stronger contribution to $g(\mathbf{x})$ [22], [23]. For some instance $\mathbf{x} \in \mathcal{X}$, Shapley explanations take the form:

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

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 local feature contribution value associated with x_j and ϕ_0 is the average of $g(\mathbf{X})$. Local, per-instance explanations using Shapley values tend to involve ranking of 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 $\mathbf{x}$. Global explanations are typically the absolute mean of the ϕ_j associated with a given X_j across all of the observations in some partition of data $\mathbf{X}$.

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 $\mathbf{z}$.

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

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 $g(\mathbf{x})$ [23]. 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 this manuscript. 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 the same model with augmented training data [30]. Consistency and accuracy guarantees are perhaps a factor in the growing momentum behind Shapley values as a candidate technique for generating consumer-specific and personalized adverse action notices for automated ML-based decisions in highly-regulated settings such as credit lending [31].

1.8. Discrimination Metrics and Test Description

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` and `seaborn` 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 interpretabilily, post-hoc explanation, and discrimination testing results are presented.

2.1. Simulated Data Results

2.1.1. Constrained vs. Unconstrained Model Fit Assessment

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

Model	Accuracy	AUC	Logloss	RMSE
g^{GBM}				
g^{MGBM}				
g^{ANN}				
g^{XNN}				

2.1.2. Interpretability and Post-hoc Explanation Results

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

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

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

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

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

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

2.1.3. Discrimination Testing Results

2.2. Mortgage Data Results

2.2.1. Constrained vs. Unconstrained Model Fit Assessment

Table 2. Accuracy metrics for g^{GBM}, g^{MGBM}, g^{ANN}, and g^{XNN} on mortgage test data.

Model	Accuracy	AUC	Logloss	RMSE
g^{GBM}				
g^{MGBM}				
g^{ANN}				
g^{XNN}				

2.2.2. Interpretability and Post-hoc Explanation Results

Figure 7. Global Tree SHAP feature importance for g^{MGBM} on the test partition of the simulated data.

Figure 8. PD and ICE across deciles for the three most important input features for g^{MGBM} on the test partition of the mortgage data.

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

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

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

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

2.2.3. Discrimination Testing Results

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 workflows 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 workflow blueprint that illustrates some of the additional steps that may be required to build a fully understandable and trustworthy machine learning 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 here as further reading.

[8] See: `https://github.com/jphall663/hc_ml` for details regarding the workflow in Figure 13.

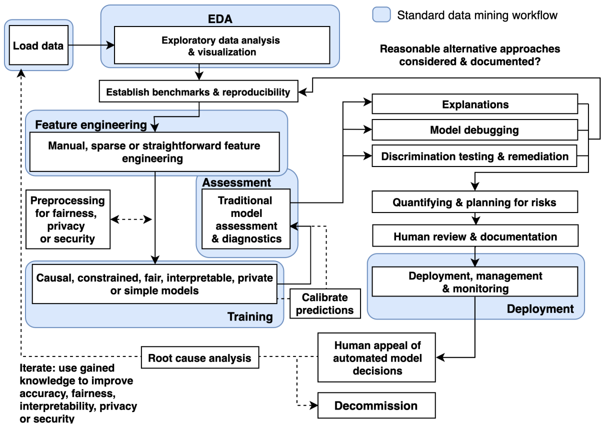

Figure 13. A diagram of a proposed holistic ML workflow in which explanations (highlighted in red) are used along with interpretable models, DI analysis and remediation techniques, and other review and appeal mechanisms to create an understandable and trustworthy ML system.

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 Microsoft and Cornell and led by Rich Caruana. 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 includes:

- **Exploratory data visualization (EDA)**: `H2OAggregatorEstimator` in `h2o` [36].
- **Sparse feature extraction**: `H2OGeneralizedLowRankEstimator` in `h2o` [37].
- **Privacy preprocessing and private models**: differential privacy for ML in `diffprivlib` [38], [39]; private models in `tensorflow/privacy` [40], [41].
- **Post-hoc explanation**: structured data explanations with `alibi` and `PDPbox`; image classification explanations with `DeepExplainer`; 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. `https://github.com/openai/gpt-2`, `https://modelcards.withgoogle.com/object-detection`.

See: `https://github.com/jphall663/awesome-machine-learning-interpretability` for a longer, curated list of related software packages and resources.

[9] Optimal sparse decision trees: https://github.com/xiyanghu/OSDT.
[10] "This looks like that" interpretable deep learning: https://github.com/cfchen-duke/ProtoPNet.

3.2. Impact of Discrimination Testing on Model Use and Adoption

3.3. Viable Discrimination Remediation Approaches

3.4. Intersectionality of Interpretability, Fairness, 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 [17], [18], [54].[11] (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 [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.[12,13]) Again, this text makes no claims that the opacity, discrimination, or security problems in ML have been solved, even treated as independent problems. Instead, it aims to highlight these issues as both singular entities and intersectional phenomena. Practitioners should of course consider the discussed interpretable modeling, post-hoc explanation, and discrimination testing approaches as 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. Of course, accuracy, discrimination, and security characteristics of a system can change over time, and simply testing for these problems at training time, as presented herein, 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.

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 machine learning workflow 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 this manuscript 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 others illustrate numerous ways for ML practitioners to become part of the solution to these problems, instead of perpetuating and exacerbating them.

---

[11] See: *Microsoft deletes 'teen girl' AI after it became a Hitler-loving sex robot within 24 hours*.
[12] See: *The Philosopher Whose Fingerprints Are All Over the FTC's New Approach to Privacy*.
[13] See: Andrew Burt HBR article - upcoming.

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. BLDS editor for editing.

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

Abbreviations

The following abbreviations are used in this text: AI – artificial intelligence, ANN – artificial neural network, DI – disparate impact, EBM or GA^{2}M – explainable boosting machine, i.e. variants GAMs that consider two-way interactions and incorporate boosting into training, GAM – generalized additive model, GBM – gradient boosting machine, ICE – individual conditional expectation, MGBM – monotonic gradient boosting machine, ML – machine learning, PD – partial dependence, SGD – stochastic gradient descent, US – United States, XNN – explainable neural network.

References

1. Rudin, C. Please Stop Explaining Black Box Models for High Stakes Decisions. *arXiv preprint arXiv:1811.10154* **2018**. URL: https://arxiv.org/pdf/1811.10154.pdf.
2. Barocas, S.; Hardt, M.; Narayanan, A. *Fairness and Machine Learning*; fairmlbook.org, 2019. URL: http://www.fairmlbook.org.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. Williams, M.; others. *Interpretability*; Fast Forward Labs, 2017. URL: https://www.cloudera.com/products/fast-forward-labs-research.html.
10. 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.
11. 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.
12. 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.
13. Recht, B.; Re, C.; Wright, S.; Niu, F. HOGWILD: A Lock-free Approach to Parallelizing Stochastic Gradient Descent. Advances in neural information processing systems, 2011, pp. 693–701. URL: https://papers.nips.cc/paper/4390-hogwild-a-lock-free-approach-to-parallelizing-stochastic-gradient-descent.pdf.

14. 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.
15. 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.
16. Zeiler, M.D. ADADELTA: an Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701* **2012**. URL: https://arxiv.org/pdf/1212.5701.pdf.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; 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.
23. 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-.
24. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Routledge, 2013.
25. Cohen, J. A Power Primer. *Psychological bulletin* **1992**, *112*, 155. URL: https://www.ime.usp.br/~abe/lista/pdfn45sGokvRe.pdf.
26. 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.
27. 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.
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.
29. 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.
30. Molnar, C. ***Interpretable Machine Learning***; christophm.github.io, 2018. URL: https://christophm.github.io/interpretable-ml-book/.
31. 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.
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.
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: .
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. 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.
55. Hoare, C.A.R. The 1980 ACM Turing Award Lecture. *Communications* **1981**. URL: http://www.cs.fsu.edu/~engelen/courses/COP4610/hoare.pdf.