A Discussion of Machine Learning Explanation Tools with Practical Recommendations and a Use Case

Patrick Hall*

Abstract

This paper discusses several explanatory methods that go beyond the error measurements and plots traditionally used to assess machine learning models. The approaches, decision tree surrogate models, individual conditional expectation (ICE) plots, local interpretable model-agnostic explanations (LIME), partial dependence plots, and Shapley explanations, vary in terms of scope, fidelity, and suitable application domain. Along with descriptions of these methods, practical recommendations, a use case, and in-depth software examples are also presented.

Key Words: Machine learning, interpretability, explanations, transparency, FATML, XAI.

1. Introduction

Interpretability of statistical and machine learning models is a multifaceted, complex, and evolving subject. This paper focuses mostly on just one aspect of model interpretability: explaining the mechanisms and predictions of models trained using supervised decision tree ensemble algorithms, like gradient boosting machines (GBMs) and random forests.

Others have defined key terms and put forward general motivations for better intepretability of machine learning models [8], [10], [12], [17]. Following Doshi-Velez and Kim, this discussion uses "the ability to explain or to present in understandable terms to a human," as the definition of *interpretable*. "When you can no longer keep asking why," will serve as the working definition for a *good explanation* of model mechanisms or predictions [10].

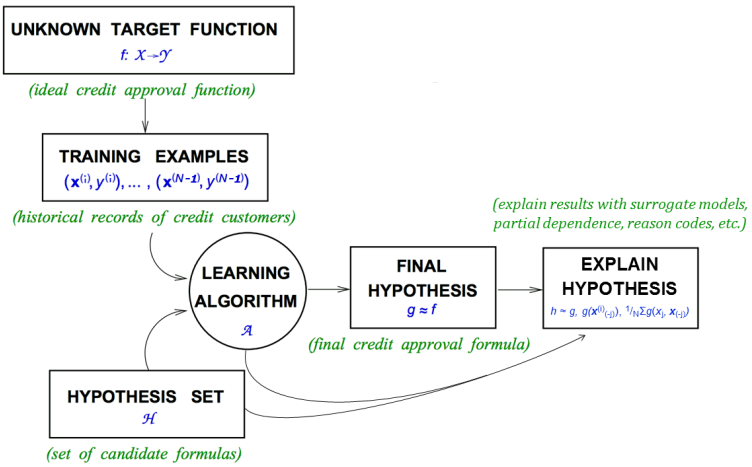

Figure 1: An augmented learning problem diagram in which several techniques create explanations for a credit scoring model. Adapted from **Learning From Data** [1].

As in Figure 1, the presented explanatory methods help practicioners make random forests, GBMs, and other types of popular supervised machine learning models more inter-

*H2O.ai, Mountain View, CA

pretable by enabling post-hoc explanations that are suitable for:

- Facilitating regulatory compliance.
- Understanding or debugging model mechanisms and predictions.
- Preventing or debugging accidental or intentional discrimination in model predictions.
- Preventing or debugging malicious hacking of models or adversarial attacks on models.

Detailed discussions of the explanatory methods begin below by defining notation. Then Sections 3 – 6 discuss explanatory methods and present recommendations for each method. Section 7 presents some general interpretability recommendations for practicioners. Section 8 applies some of the techniques and recommendations to the well-known UCI credit card dataset [16]. Section 9 discusses several additional interpretability subjects that are likely important for practicioners, and finally, Section 10 highlights a few software resources that accompany this paper.

2. Notation

To facilitate technical descriptions of explanatory techniques, notation for input and output spaces, datasets, and models is defined.

2.1 Spaces

- Input features come from a set $\mathcal{X}$ contained in a *P*-dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$.
- Known labels corresponding to instances of $\mathcal{X}$ come from the set $\mathcal{Y}$ and are contained in a C-dimensional label space, $\mathcal{Y} \subset \mathbb{R}^C$.
- Learned output responses come from a set $\hat{\mathcal{Y}}$.

2.2 Datasets

- The input dataset $\mathbf{X}$ is composed of observed instances of the set $\mathcal{X}$ with a corresponding dataset of labels $\mathbf{Y}$, observed instances of the set $\mathcal{Y}$.
- Each i-th observation of $\mathbf{X}$ is denoted as $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \ldots, x_{P-1}^{(i)}]$, with corresponding i-th labels in $\mathbf{Y}$, $\mathbf{y}^{(i)} = [y_0^{(i)}, y_1^{(i)}, \ldots, y_{C-1}^{(i)}]$.
- $\mathbf{X}$ and $\mathbf{Y}$ consists of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- Each j-th input column vector of $\mathbf{X}$ is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \ldots, x_j^{(N-1)}]^T$.

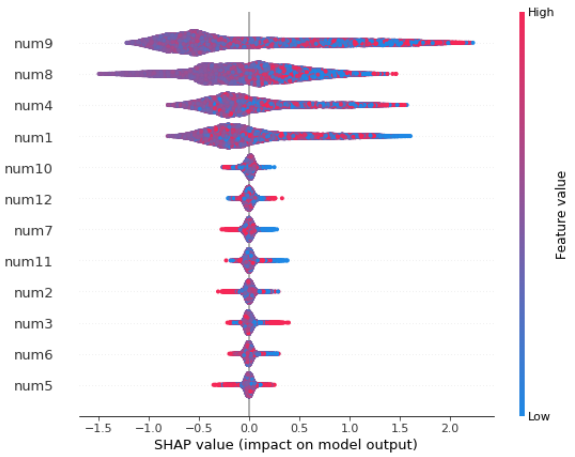

Figure 2: Shapley summary plot for known signal-generating function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$, and for a machine-learned GBM response function g_{GBM}.

2.3 Models

- A type of machine learning 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 $\mathcal{A}$: $\mathbf{X}, \mathbf{Y} \xrightarrow{\mathcal{A}} g$.
- 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 is denoted as g.

3. Surrogate Decision Trees

The phrase *surrogate model* is used here to refer to a simple model, h, of a complex model, g. This type of model is referred to by various other names, such as *proxy* or *shadow* models and the process of training surrogate models is often referred to as *model extraction* [7], [25], [4].

3.1 Description

Given a learned function g and set of learned output responses $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and a tree splitting and pruning approach $\mathcal{A}$, a global – or over all $\mathbf{X}$ – surrogate decision tree h_{tree}, can be extracted such that $h_{\text{tree}}(\mathbf{X}) \approx g(\mathbf{X})$:

$$\mathbf{X}, g(\mathbf{X}) \xrightarrow{\mathcal{A}} h_{\text{tree}} \tag{1}$$

Decision trees can be represented as directed graphs where the relative positions of input features can provide insight into their importance and interactions [5]. This makes decision trees useful surrogate models. Input features that appear high and often in the directed graph representation of h_{tree} are assumed to have high importance in g. Input features directly above or below one-another in h_{tree} are assumed to have strong interactions in g. These relative relationships between input features in h_{tree} can be used to verify and debug the feature importance, interactions, and predictions of g.

Figures 2 and 3 use simulated data to emperically demonstrate the desired relationships between input feature importance and interactions in the input space $\mathbf{X}$, the label space $f(\mathbf{X}) = \mathbf{Y}$, a GBM model to be explained g_{GBM}, and a decision tree surrogate h_{tree}. Data with a known signal-generating function depending on four input features with interactions and with eight noise features is simulated.

$$f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e \tag{2}$$

g_{GBM} is trained: $\mathbf{X}, \mathbf{f}(\mathbf{X}) \xrightarrow{\mathcal{A}} g_{\text{GBM}}$ such that $g_{\text{GBM}} \approx f$. Then h_{tree} is extracted by $\mathbf{X}, \mathbf{g}_{\text{GBM}}(\mathbf{X}) \xrightarrow{\mathcal{A}} h_{\text{tree}}$, such that $h_{\text{tree}}(\mathbf{X}) \approx g_{\text{GBM}}(\mathbf{X}) \approx f(\mathbf{X})$.

Figure 2 displays the local Shapley importance values for an input feature's impact on each $g_{\text{GBM}}(\mathbf{X})$ prediction. Plotting Shapley values can be a more wholistic and consistent feature importance metric than traditional single-value quantities [18]. As expected, Figure 2 shows that num_9 and num_8 tend to make the largest contributions to $g_{\text{GBM}}(\mathbf{X})$ followed by num_4 and num_1. Also as expected, noise features make minimal contributions to $g_{\text{GBM}}(\mathbf{X})$. Shapley values are discussed in detail in Section 6.

Figure 3 is a directed graph representation of h_{tree} that prominently displays the importance of input features num_9 and num_8 along with num_4 and num_1. Figure 3 also visually highlights the interactions between these inputs. URLs to the data and software used to generate Figures 2 and 3 are available in Section 10.

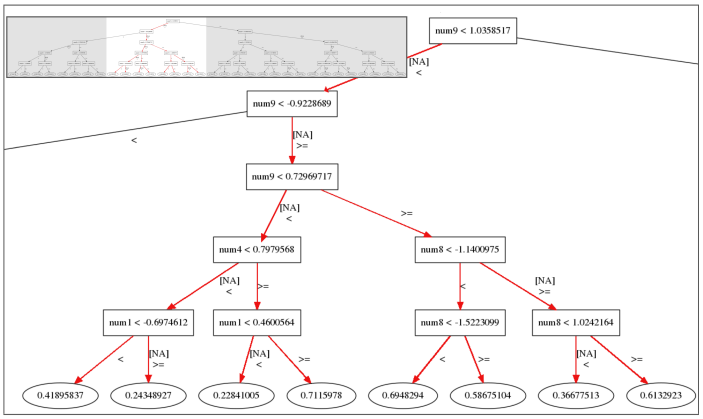

Figure 3: h_{tree} for previously defined known signal-generating function f and machine-learned GBM response function g_{GBM}. An image of the entire h_{tree} directed graph is available in the supplementary materials described in Section 10.

3.2 Recommendations

- A shallow-depth h_{tree} displays a global, low-fidelity, high-interpretability flow chart of important features and interactions in g. Because there are few theoretical gaurantees that h_{tree} truly represents g, always use error measures to assess the trustworthiness of h_{tree}.

- Prescribed methods for training h_{tree} do exist [7] [4]. In practice, straightforward cross-validation approaches are typically sufficient. Moreover, comparing cross-validated training error to traditional training error can give an indication of the stability of the single tree, h_{tree}.

- Hu et al. use local linear surrogate models, h_{GLM}, in h_{tree} leaf nodes to increase overall surrogate model fidelity while also retaining a high degree of interpretability.

4. Partial Dependence and Individual Conditional Expectation plots

Partial dependence (PD) plots are a well-known method for describing the average predictions of a complex model, g, across some partition of data, $\mathbf{X}$, for some interesting input feature, X_j [9]. Individual conditional expectation (ICE) plots are a newer method that describes the local behavior of g for a single instance of $\mathcal{X}$, $\mathbf{x}^{(i)}$ [11]. Partial dependence and ICE can be combined in the same plot to identify strong interactions modeled by g and to create a wholistic portrait of the predictions of a complex model for some interesting input feature, X_j.

4.1 Description

Following Friedman et al. 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. $\text{PD}(X_j, g)$ for a given feature X_j is estimated as the average output of the learned function, $g(\mathbf{X})$, when all the components of X_j are set to a constant $x \in \mathcal{X}$ and $\mathbf{X}_{(-j)}$ is left untouched. $\text{ICE}(X_j, \mathbf{x}^{(i)}, g)$ for a given observation $\mathbf{x}^{(i)}$ and feature X_j is estimated as the output of the learned function, $g(\mathbf{x}^{(i)})$, when $x_j^{(i)}$ is set to a constant $x \in \mathcal{X}$ and all $\mathbf{x}^{(i)} \in \mathbf{X}_{(-j)}$ are left untouched. Partial dependence and ICE curves are usually plotted over some set of interesting constants $\mathbf{x} \in \mathcal{X}$.

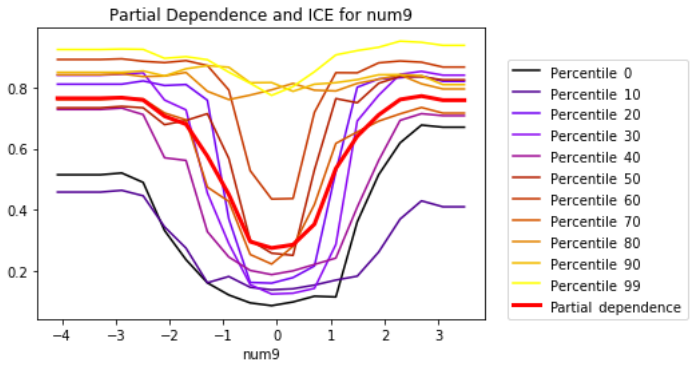

Figure 4: Partial dependence and ICE curves for previously defined known signal-generating function f and machine-learned GBM response function g_{GBM}.

As in Section 3, simulated data is used to highlight desirable characteristics of partial dependence and ICE plots. In Figure 4 partial dependence and ICE at the minimum, maximum, and each decile of $g_{\text{GBM}}(\mathbf{X})$ are plotted. The known quadratic behavior of num_9 is plainly visible, except for high value predictions, the 80^{th} percentiles of $g_{\text{GBM}}(\mathbf{X})$ and above and for $\sim -1 < \text{num}_9 <\sim 1$. When partial dependence and ICE curves diverge, this often points to an interaction that is being averaged out of the partial dependence. Given the form of Equation 2, there is a known interaction between num_9 and num_8. Combining the information from partial dependence and ICE plots with h_{tree} can help elucidate more detailed information about modeled interactions in g. For the simulated example, h_{tree} shows an interaction between num_9 and num_8 and additional modeled interactions between num_9, num_4, and num_1 for $\sim -0.92 \leq \text{num}_9 <\sim 1.04$. URLs to the data and software used to generate Figure 4 are available in Section 10.

4.2 Recommendations

- Combining h_{tree} with partial dependence and ICE curves is a convenient method for detecting, confirming, and understanding important interactions in g.
- As monotonicity is often a desired trait for interpretable models, partial dependence and ICE plots can be used to verify the monotonicity of g on average and across deciles of $g(\mathbf{X})$ w.r.t. some input feature X_j.

5. Local Interpretable Model-agnostic Explanations (LIME)

Global and local scope is a key concept in explaining machine learning models and predictions. Section 3 presents decision trees as a global – or over all $\mathbf{X}$ – surrogate model. As machine-learned response functions, g, can be complex, simple global surrogate models can sometimes be too approximate to be trustworthy. LIME attempts to create more representative explanations by fitting a local surrogate model, h, in the local region of some observation of interest $\mathbf{x} \in \mathcal{X}$. Both h and local regions can be defined to suite the needs of users.

5.1 Description

Ribeiro et al. defines LIME for some observation $\mathbf{x} \in \mathcal{X}$ as:

$$\underset{h \in \mathcal{H}}{\arg\max}\, \mathcal{L}(g, h, \pi_{\mathbf{X}}) + \Omega(h) \tag{3}$$

where h is an interpretable surrogate model of g, often a linear model h_{GLM}, $\pi_{\mathbf{X}}$ is a weighting function over the domain of g, and $\Omega(h)$ limits the complexity of h [20]. Following Ribeiro et al. h_{GLM} is often trained by:

$$\mathbf{X}', g(\mathbf{X}') \xrightarrow{\mathcal{A}_{\text{LASSO}}} h_{\text{GLM}} \tag{4}$$

where $\mathbf{X}'$ is sampled from $\mathcal{X}$, $\pi_{\mathbf{X}}$ weighs $\mathbf{X}'$ samples by their Euclidean similarity to $\mathbf{x}$ to enforce locality, local feature contributions are estimated as the product of h_{GLM} coefficients and local row values $\beta_j x_j^{(i)}$, and $\Omega(h)$ is defined as a LASSO, or L1, penalty on h_{GLM} coefficients inducing sparsity in h_{GLM}.

Figure 5 displays estimated local feature contribution values for the same g_{GBM} and simulated $\mathbf{X}$ with known signal-generating function f used in previous sections. To increase the nonlinear capacity of the three h_{GLM} models, information from the Shapley summary plot in Figure 2 is used to select inputs to discretize before training each h_{GLM}: , $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9. Table 1 contains prediction and fit information for g_{GBM} and h_{GLM}. This is critical information for analyzing LIMEs.

Table 1: g_{GBM} and h_{GLM} predictions and h_{GLM} intercepts and fit measurements for the h_{GLM} models trained to explain $g_{\text{GBM}}(\mathbf{x})$ at the 10$^{\text{th}}$, median, and 90$^{\text{th}}$ percentiles of previously defined $g_{\text{GBM}}(\mathbf{X})$ and known signal-generating function f.

$g_{\text{GBM}}(\mathbf{X})$ Percentile	$g_{\text{GBM}}(\mathbf{x})$ Prediction	$h_{GLM}(\mathbf{x})$ Prediction	h_{GLM} Intercept	h_{GLM} R^2
10$^{\text{th}}$	0.16	0.13	0.53	0.72
Median	0.30	0.47	0.70	0.57
90$^{\text{th}}$	0.82	0.86	0.76	0.40

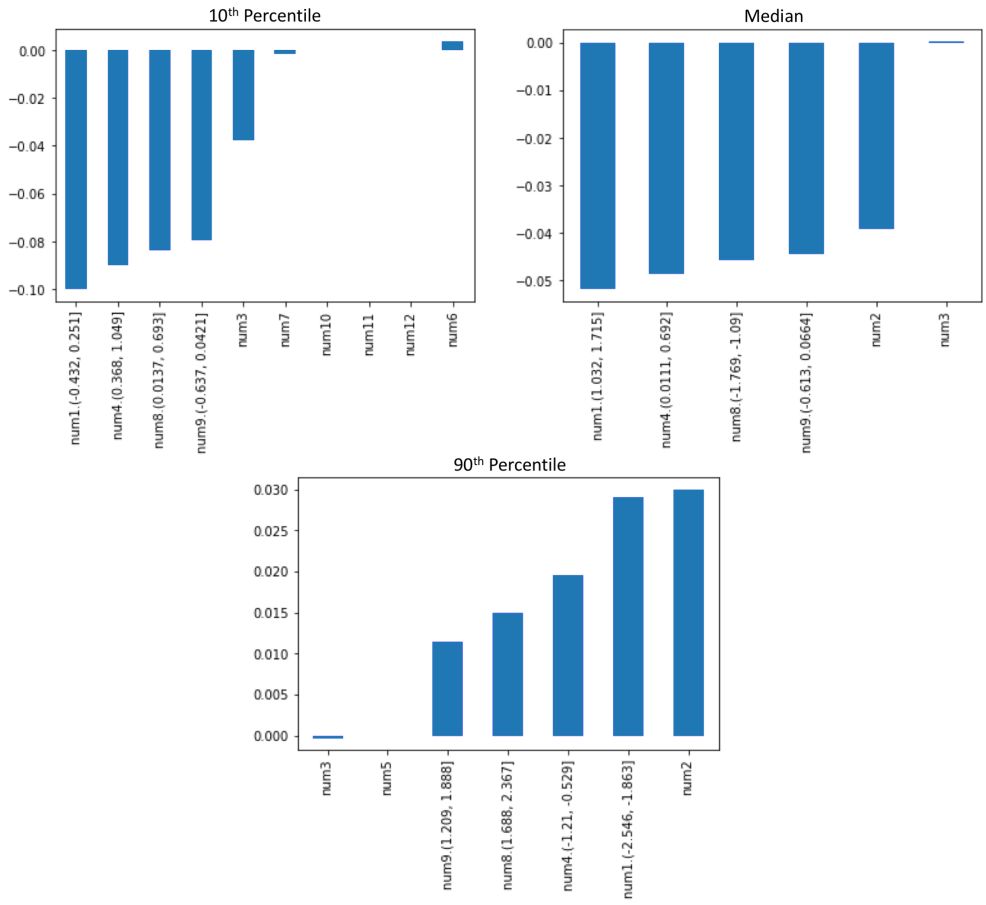

Figure 5: Sparse, low-fidelity local feature contributions found using LIME at three percentiles of $g_{\text{GBM}}(\mathbf{X})$ for known signal-generating function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$.

Table 1 shows that LIME is not necessarily locally accurate, meaning that the predictions of $h_{GLM}(\mathbf{x})$ are not always equal to the prediction of $g_{\text{GBM}}(\mathbf{x})$. Moreover, the three h_{GLM} models do not necessarily explain all of the variance of g_{GBM} predictions in the local regions around the three $\mathbf{x}^{(i)}$ of interest. h_{GLM} intercepts are also displayed because local feature contribution values, $\beta_j x_j^{(i)}$, are offsets from the local h_{GLM} intercepts.

An immediately noticable characteristic of the estimated local contributions in Figure 5 is their sparsity. LASSO input feature selection drives some h_{GLM} β_j coefficients to zero so that some $\beta_j x_j^{(i)}$ local feature contributions are also zero. For the 10$^{\text{th}}$ decile $g_{\text{GBM}}(\mathbf{x})$ prediction, the local h_{GLM} R^2 is adequate and the LIME values appear parsimonious with reasonable expectations. The contributions from discretized $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 outweigh all other noise feature contributions and the $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 contributions are all negative, as expected for the relatively low value of $g_{\text{GBM}}(\mathbf{x})$.

For the median prediction of $g_{\text{GBM}}(\mathbf{x})$, it could be expected that some estimated contributions for $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 should be positive and others should be negative. However, all local feature contributions are negative due to the relatively high value of the h_{GLM} intercept at the median percentile of $g_{\text{GBM}}(\mathbf{x})$. Because the h_{GLM} intercept is quite large compared to the $g_{\text{GBM}}(\mathbf{x})$ prediction, it is not alarming that all the $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 contributions are negative offsets w.r.t. the local h_{GLM} intercept value. For

the median $g_{\text{GBM}}(\mathbf{x})$ prediction, h_{GLM} estimates that the noise feature num_2 has a fairly large contribution and the local h_{GLM} R^2 is also probably less than adequate to generate trustworthy explanations.

For the 90$^{\text{th}}$ decile $g_{\text{GBM}}(\mathbf{x})$ predictions, the local contributions for $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 are positive as expected for the relatively high value of $g_{\text{GBM}}(\mathbf{x})$, but the local h_{GLM} R^2 is somewhat poor and the noise feature $num2$ has the highest local feature contribtion. This large attribution to the noise feature num2 could stem from problems in the LIME procedure or in the fit of g_{GBM} to f. Further investigation, or model debugging, is conducted in Section 6.

Generally the LIMEs in section 5 would be considered to be sparse or high-interpretability but also low-fidelity explanations. This is not always the case with LIME and the fit of some h_{GLM} to a local region around some $g(\mathbf{x})$ will vary in accuracy. URLs to the data and software used to generate Table 1 and Figure 5 are available in section 10.

5.2 Recommendations

- Always use fit measures to assess the trustworthiness of LIMEs.
- Local feature contribution values are often offsets from a local h_{GLM} intercept. Note that this intercept can sometimes account for the most important local phenomena.
- Some LIME methods can be difficult to deploy for explaining predictions in real-time. Consider highly deployable variants for real-time applications [14], [15].
- Always investigate local h_{GLM} intercept values. Generated LIME samples can contain large proportions of out-of-domain data that can lead to unrealistic intercept values.
- To increase the fidelity of LIMEs, try LIME on discretized input features and on manually constructed interactions.
- Use cross-validation to estimate standard deviations or even confidence intervals for local feature contribution values.
- When relying only on local linear models, note that LIME can fail to create acceptable explanations, particularly in the presence of extreme nonlinearity or high-degree interactions. Other types of local models with model-specific explanatory mechanisms, such as decision trees or neural networks, can be used in these cases.

6. Tree Shap

7. General Recommendations

The following recommendations apply to several or all of the described explanatory techniques or to the practice of applied interpretable machine learning in general.

- Less complex models are typically easier to explain. Section 9 contains information about directly interpretable white-box machine learning models.
- Monotonicity is a desirable characteristic in interpretable models. White-box, monotonically constrained XGBoost models along with the explanatory techiques described in this paper are a direct and open source way to train and explain an interpretable machine learning model. A monotonically constrained XGBoost GBM is trained and explained in Section 8.

- Several explanatory techniques are usually required to create good explanations. Users should apply a combination global and local and low- and high-fidelity explanatory techniques to a machine learning model and seek consistent results across multiple explanatory techniques. Simpler low-fidelity or sparse explanations can be used to understand more accurate, and sometimes more sophisticated, high-fidelity explanations.

- Methods relying on surrogate models or generated data are sometimes unpalatable to users. User sometimes *need* to understand *their* model on *their* data.

- Surrogate models can provide low-fidelity explanations for an entire machine learning pipeline in the original feature space if g is defined to include feature extraction or feature engineering.

- Consider production deployment of explanatory methods carefully. Currently, the deployment of some open source software packages is not straightforward, especially for the generation of explanations on new data in real-time.

8. Credit Card Data Use Case

9. Suggested Reading

9.1 White-box Models

The application of post-hoc explanatory techniques is convenient for previously existing machine learning models, workflows, or pipelines. However, a more direct approach may be to train an interpretable white-box machine learning model which may or may not require additional post-hoc explanatory analysis. Monotonic XGBoost is an excellent option to evaluate because the software is open source, readily available, easily installable and deployable, and highly scalable [6]. Acclaimed work by the Rudin group at Duke Univeristy is also likely of interest to many users. They have developed several types of rule-based models [3], [26], linear model variants [23], and many other novel algorithms suitable for use in high stakes, mission-critical prediction and decision-making scenarios.

9.2 Explainable Neural Networks (xNNs)

Often considered the darkest of black-box models, recent work in xNN implementation and explaining artificial neural network (ANN) predictions may render that notion of ANNs completely obsolete. Many of the breakthroughs in ANN explanation stem from the straightfoward calculation of accurate derivatives of the trained ANN response function w.r.t. to input variables made possible by the proliferation of deep learning toolkits such as tensorflow [19]. These derivatives allow for the disaggregation of the trained ANN response function prediction, $g_{ANN}(\mathbf{X})$, into input feature contributions for any observation in the domain of $\mathcal{X}$. Popular techniques have names like DeepLIFT and integrated gradients [21], [22], [2]. Explaining ANN predictions is impactful for at least two major reasons. While most users will be familiar with the wide-spead use of ANNs in pattern recognition, they are also used for more traditional data mining applications such as fraud detection, and even for regulated applications such as credit scoring [13]. Moreover, ANNs can now be used as accurate and explainable surrogate models, potentially increasing the fidelity of both global and local surrogate model techniques. For an excellent discussion of xNNs in a practical setting see *Explainable Neural Networks based on Additive Index Models* by the Wells Fargo Corporate Model Risk group [24].

9.3 Fairness

10. Software Resources and Supplementary Materials

11. Acknowledgements

References

[1] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. ***Learning from Data***. AMLBook, New York, 2012. URL: https://work.caltech.edu/textbook.html.

[2] Marco Ancona, Enea Ceolini, Cengiz Oztireli, and Markus Gross. Towards Better Understanding of Gradient-based Attribution Methods for Deep Neural Networks. In *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.

[3] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning Certifiably Optimal Rule Lists for Categorical Data. *Journal of Machine Learning Research*, 18(234):1–78, 2018. URL: https://corels.eecs.harvard.edu/.

[4] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting Blackbox Models via Model Extraction. *arXiv preprint arXiv:1705.08504*, 2017. URL: https://arxiv.org/pdf/1705.08504.pdf.

[5] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Routledge, 1984.

[6] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. URL: https://arxiv.org/pdf/1603.02754.pdf.

[7] Mark W. Craven and Jude W. Shavlik. Extracting Tree-structured Representations of Trained Networks. *Advances in Neural Information Processing Systems*, 1996. URL: http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf.

[8] Finale Doshi-Velez and Been Kim. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017. URL: https://arxiv.org/pdf/1702.08608.pdf.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. ***The Elements of Statistical Learning***. Springer, New York, 2001. URL: https://web.stanford.edu/∼hastie/ElemStatLearn/printings/ESLII_print12.pdf.

[10] Leilani H Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning. *arXiv preprint arXiv:1806.00069*, 2018. URL: https://arxiv.org/pdf/1806.00069.pdf.

[11] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1), 2015. URL: https://arxiv.org/pdf/1309.6392.pdf.

[12] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A Survey of Methods for Explaining Black Box Models. *arXiv preprint arXiv:1802.01933*, 2018. URL: https://arxiv.org/pdf/1802.01933.pdf.

[13] Patrick Hall and Navdeep Gill. *An Introduction to Machine Learning Interpretability*. O'Reilly Media, 2018. URL: https://www.safaribooksonline.com/library/view/an-introduction-to/9781492033158/.

[14] Patrick Hall, Navdeep Gill, Megan Kurka, and Wen Phan. *Machine Learning Interpretability with H2O Driverless AI*. H2O.ai, 2017. URL: http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf.

[15] Linwei Hu, Jie Chen, Vijayan N. Nair, and Agus Sudjianto. Locally Interpretable Models and Effects Based on Supervised Partitioning (LIME-SUP). *arXiv preprint arXiv:1806.00663*, 2018. URL: https://arxiv.org/ftp/arxiv/papers/1806/1806.00663.pdf.

[16] M. Lichman. UCI Machine Learning Repository, 2013. URL: http://archive.ics.uci.edu/ml.

[17] Zachary C. Lipton. The Mythos of Model Interpretability. *arXiv preprint arXiv:1606.03490*, 2016. URL: https://arxiv.org/pdf/1606.03490.pdf.

[18] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[19] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part i): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10561*, 2017. URL: https://arxiv.org/pdf/1711.10561.pdf.

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust you?: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. URL: http://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf.

[21] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. *arXiv preprint arXiv:1704.02685*, 2017. URL: https://arxiv.org/pdf/1704.02685.pdf.

[22] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. *arXiv preprint arXiv:1703.01365*, 2017. URL: https://arxiv.org/pdf/1703.01365.pdf.

[23] Berk Ustun and Cynthia Rudin. Supersparse Linear Integer Models for Optimized Medical Scoring Systems. *Machine Learning*, 102(3):349–391, 2016. URL: https://users.cs.duke.edu/∼cynthia/docs/UstunTrRuAAAI13.pdf.

[24] Joel Vaughan, Agus Sudjianto, Erind Brahimi, Jie Chen, and Vijayan N Nair. Explainable Neural Networks Based on Additive Index Models. *arXiv preprint arXiv:1806.01933*, 2018. URL: https://arxiv.org/pdf/1806.01933.pdf.

[25] Mike Williams et al. *Interpretability*. Fast Forward Labs, 2017. URL: https://www.fastforwardlabs.com/research/FF06.

[26] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable Bayesian Rule Lists. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. URL: https://arxiv.org/pdf/1602.08610.pdf.