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

Patrick Hall PHALL@H2O.AI
H2O.ai, Mountain View, CA

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.

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 (Lipton, 2016), (Doshi-Velez & Kim, 2017), (Gilpin et al., 2018), (Guidotti et al., 2018). Following Doshi-Velez and Kim (2017), 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 (Gilpin et al., 2018).

As in figure 1, the presented explanatory methods help practicioners make random forests, GBMs, and other types of popular supervised machine learning models more interpretable by enabling post-hoc explanations that are suitable for:

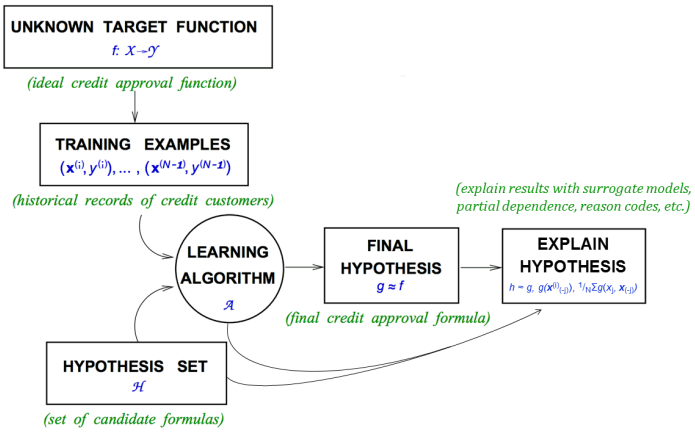

Figure 1. An augmented learning problem diagram in which several techniques create explanations for a credit scoring model. Adapted from **Learning From Data** (Abu-Mostafa et al., 2012).

- 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

Proceedings of the Joint Statistical Meetings, Vancouver, British Columbia, CA, 2018.

general interpretability recommendations for practicioners. Section 8 applies some of the techniques and recommendations to the well-known Kaggle diabetes dataset. 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 $X_j = [x_j^{(0)}, x_j^{(1)}, \ldots, x_j^{(N-1)}]^T$.

2.3. Models

- A type of machine learning model g, selected from a hypothesis set $\mathcal{H}$, is trained to represent an unknown target 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* (Craven & Shavlik, 1996), (Williams et al., 2017), (Bastani et al., 2017).

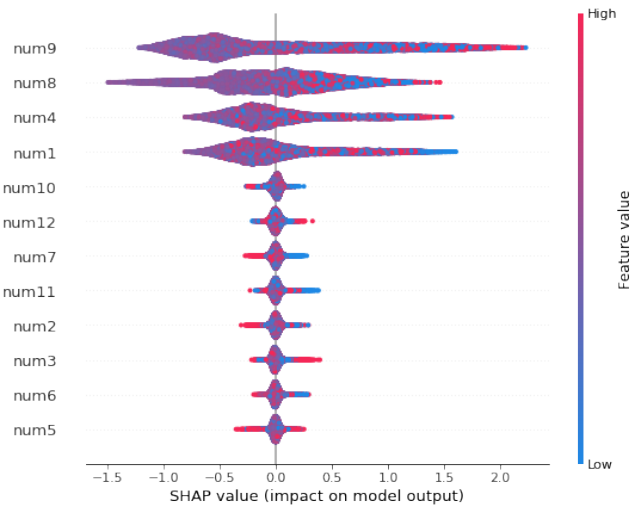

Figure 2. Shapley summary plot for known target function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$, and for a machine-learned GBM response function g_{GBM}.

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:

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

such that $h_{\text{tree}}(\mathbf{X}) \approx g(\mathbf{X})$.

Decision trees can be represented as directed graphs where the relative positions of input features can provide insight into their importance and interactions (Breiman et al., 1984). 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 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 target function depending on four input features with interactions:

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

and with eight noise features is simulated. 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 (Lundberg & Lee, 2017). 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 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.

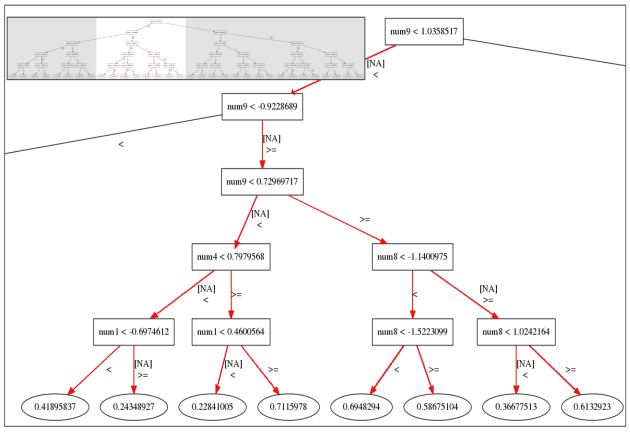

Figure 3. h_{tree} for previously defined known target function f and machine-learned GBM response function g_{GBM}.

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 (Craven & Shavlik, 1996); (Bastani et al., 2017) for training h_{tree} do exist. 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. (2018) 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 (Friedman et al., 2001). Individual conditional expectation (ICE) plots are a newer method that describes the local behavior of g for a single instance of $\mathbf{X}$, $\mathbf{x}^{(i)}$ (Goldstein et al., 2015). 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. (2001) 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 $\mathbf{x} \in \mathcal{X}$.

As in section 3, simulated data is used to highlight desirable characteristics of partial dependence and ICE plots. In figure 6 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

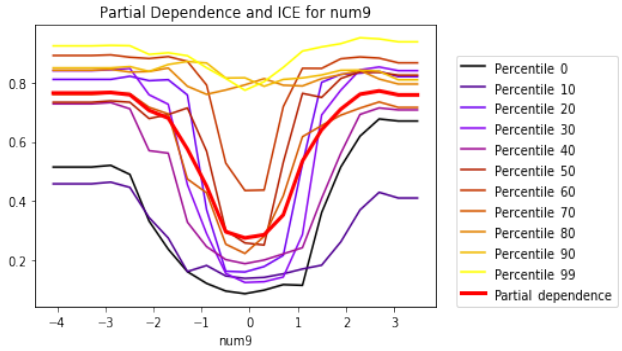

Figure 4. Partial dependence and ICE curves for previously defined known target function f and machine-learned GBM response function g_{GBM}.

$\sim -1 < \text{num}_9 <\sim 1$. 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 6 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 practically useful. LIME attempts to create more representative explanations by fitting a local surrogate model, h, in the local region of some data point 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. (2016) 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.

Following Ribeiro et al. (2016) 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 with 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} inducing sparsity in h_{GLM}.

Figure 5 displays local feature contribution values for the same g_{GBM} and simulated $\mathbf{X}$ with known target 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, $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9, before training each h_{GLM}. Table 1 contains prediction and fit information for g_{GBM} and h_{GLM}, critical information for analyzing LIMEs.

Table 1. $g_{\text{GBM}}(\mathbf{x})$ predictions, intercepts, and h_{GLM} R^2 for h_{GLM} models trained to explain $g_{\text{GBM}}(\mathbf{x})$ at the 10^{th}, median, and 90^{th} percentiles of previously defined $g_{\text{GBM}}(\mathbf{X})$ and known signal generating function f.

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

In table 1 it can be seen that LIME is not gauranteed to be 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 are not necessarily explaining all the variance of the g_{GBM} predictions in the local regions around the three $\mathbf{x}^{(i)}$ of interest. h_{GLM} intercepts are also important to

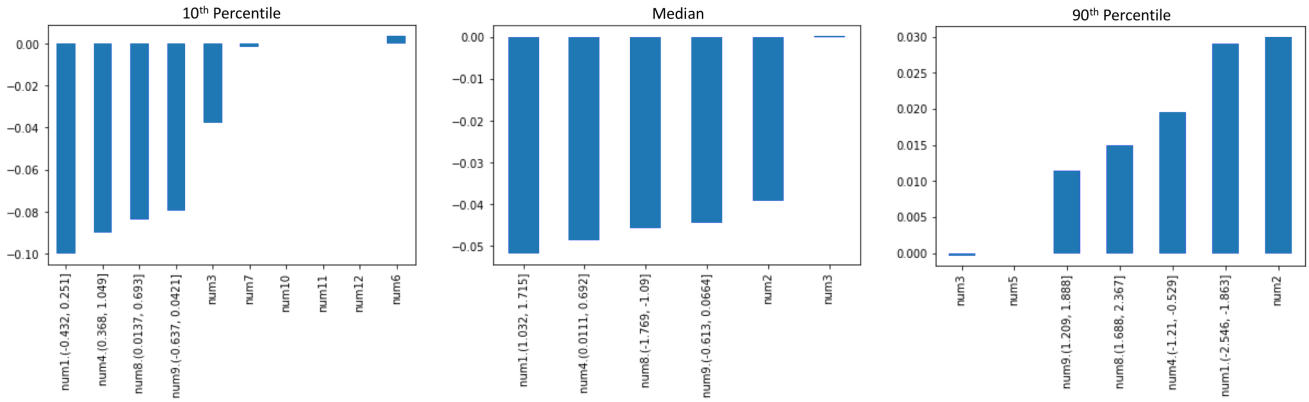

Figure 5. Sparse, low-fidelity local feature contributions found using LIME at three percentiles of $g_{\text{GBM}}(\mathbf{X})$ and for known signal generating function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$.

investigate, because local feature contribution values, $\beta_j x_j^{(i)}$, are offsets from the local h_{GLM} intercepts.

An immediately noticable characteristic of the local explanations displayed in figure 5 is their sparsity. The LASSO training procedure drives some h_{GLM} β_j coefficients to zero, ensuring that some $\beta_j x_j^{(i)}$ local feature contributions are also always 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 low value of $g_{\text{GBM}}(\mathbf{x})$.

For the median prediction of $g_{\text{GBM}}(\mathbf{x})$ it could be expected that some contributions from $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 would be positive but others would be negative. However, all local feature contributions are negative. Thic an be explained by the relatively high value of the h_{GLM} intercept for this decile. Because the h_{GLM} intercept is quite large compared to the $g_{\text{GBM}}(\mathbf{x})$, it is not alarming that all the $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 negative offsets with respect to the local intercept value. For the median $g_{\text{GBM}}(\mathbf{x})$ prediction the noise feature num_2 has a fairly large contribution and the local h_{GLM} R^2 could be considered to be less than adequate.

For the 90$^{\text{th}}$ decile $g_{\text{GBM}}(\mathbf{x})$ prediction the local contributions for $\text{num}_1, \text{num}_4, \text{num}_8$ and num_9 are positive as expected for the high value of $g_{\text{GBM}}(\mathbf{x})$, but the local h_{GLM} R^2 is somewhat poor and the noise feature *num*2 has the highest local feature contribution. 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, but 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 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, i.e. Hu et al 2018 and Hall et al. 2017, for real-time applications.
- Always investigate local h_{GLM} intercept values. Generated LIME samples can contain large proportions of out-of-range 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 construct standard deviations or even confidence intervals for local feature contribution values.

- When relying only on local linear models, note 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

Shapley explanations are a class of additive, consistent local feature contribution measures with long-standing theoretical support, (Lundberg & Lee, 2017). For some observation $\mathbf{x} \in \mathcal{X}$, Shapley explanations take the form:

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

Here $\mathbf{x}' \in \{0,1\}^{\mathcal{P}}$ is a binary representation of $\mathbf{x}$ where 0 indicates missingness. Each ϕ_j is the local feature contribution value associated with x_j.

- Calculating Shapley values directly is typically infeasible, but they can be estimated in different ways.
- Tree Shap is a specific implementation of Shapley explanations that leverages DT structures to disaggregrate the contribution of each x_j to $g(\mathbf{x})$ in a DT or DT-based ensemble model. (Lundberg et al., 2018)
- Tree Shap is ideal for high-fidelity explanations of DT-based models, perhaps even in regulated applications.
- Local feature contribution values are offsets from a global intercept.
- LIME can be constrained to become Shapley explanations, i.e. kernel shap.
- A similar, popular method known as *treeinterpreter* appears untrustworthy when applied to GBM models.

7. General Recommendations

- Monotonicity.
- Monotonically constrained XGBoost, Surrogate DT, PD and ICE plots, and Tree Shap are a direct and open source way to create an interpretable nonlinear model.
- Global and local explanatory techniques are often necessary to explain a model.
- Use simpler low-fidelity or sparse explanations to understand more accurate and complex high-fidelity explanations.
- Seek consistent results across multiple explanatory techniques.
- Methods relying on generated data are sometimes unpalatable to users. They want to understand *their* data.
- Surrogate models can provide low-fidelity explanations for model mechanisms in original feature spaces if g is defined to include feature extraction or engineering.
- To increase adoption, production deployment of explanatory methods must be straightforward.

8. Diabetes Data Use Case

9. Suggested Reading

- xNN derivatives, neural net-specific methods.
- Accurate and interpretable classifiers.
- Fairness.

10. Software Resources

Comparison of Explanatory Techniques on Simulated Data:
https://github.com/h2oai/mli-resources/tree/master/lime_shap_treeint_compare

In-depth Explanatory Technique Examples:
https://github.com/jphall663/interpretable_machine_learning_with_python

"Awesome" Machine Learning Interpretability Resource List:
https://github.com/jphall663/awesome-machine-learning-interpretability

11. Acknowledgements

References

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

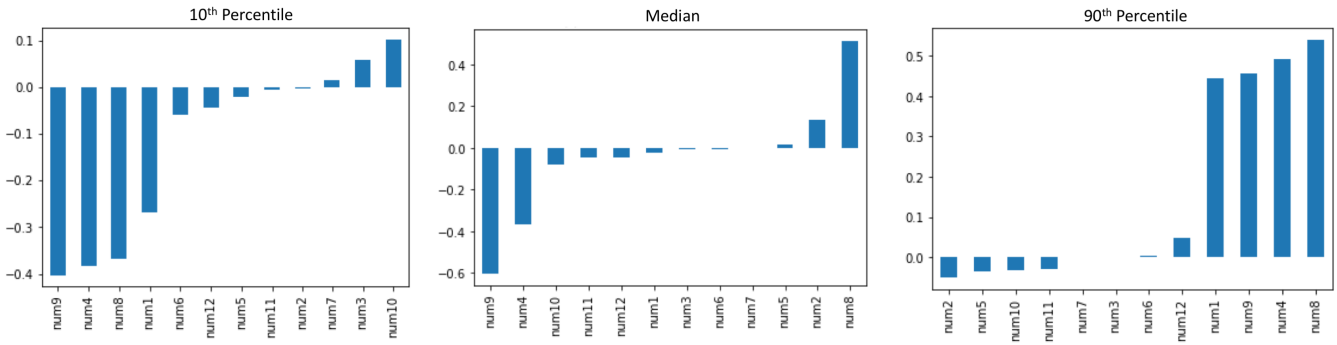

Figure 6. Complete, high-fidelity local feature contributions found using tree shap at three percentiles of $g_{\text{GBM}}(\mathbf{X})$ and for known signal generating function $f = \text{num}_1 * \text{num}_4 + |\text{num}_8| * \text{num}_9^2 + e$.

Bastani, Osbert, Kim, Carolyn, and Bastani, Hamsa. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017. URL https://arxiv.org/pdf/1705.08504.pdf.

Breiman, Leo, Friedman, Jerome H., Olshen, Richard A., and Stone, Charles J. *Classification and Regression tTrees*. Routledge, 1984.

Craven, Mark W. and Shavlik, Jude W. 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.

Doshi-Velez, Finale and Kim, Been. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. URL https://arxiv.org/pdf/1702.08608.pdf.

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

Gilpin, Leilani H, Bau, David, Yuan, Ben Z., Bajwa, Ayesha, Specter, Michael, and Kagal, Lalana. Explaining explanations: An approach to evaluating interpretability of machine learning. *arXiv preprint arXiv:1806.00069*, 2018. URL https://arxiv.org/pdf/1806.00069.pdf.

Goldstein, Alex, Kapelner, Adam, Bleich, Justin, and Pitkin, Emil. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1), 2015.

Guidotti, Riccardo, Monreale, Anna, Turini, Franco, Pedreschi, Dino, and Giannotti, Fosca. A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*, 2018. URL https://arxiv.org/pdf/1802.01933.pdf.

Hall, Patrick, Gill, Navdeep, Kurka, Megan, and Phan, Wen. Machine learning interpretability with h2o driverless ai, 2017. URL http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf.

Hu, Linwei, Chen, Jie, Nair, Vijayan N., and Sudjianto, Agus. 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.

Lipton, Zachary C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016. URL https://arxiv.org/pdf/1606.03490.pdf.

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

Lundberg, Scott M, Erion, Gabriel G, and Lee, Su-In. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018. URL https://arxiv.org/pdf/1706.06060.pdf.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. 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*, pp. 1135–1144. ACM, 2016. URL http://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf.

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