Machine Learning Interpretability with H2O Driverless AI

Patrick Hall, Navdeep Gill, Megan Kurka, & Wen Phan

Edited by: Angela Bartz

---

http://docs.h2o.ai

August 2020: First Edition

Machine Learning Interpretability with H2O Driverless AI
by Patrick Hall, Navdeep Gill, Megan Kurka, & Wen Phan
Edited by: Angela Bartz

Published by H2O.ai, Inc.
2307 Leghorn St.
Mountain View, CA 94043

August 2020: First Edition

Photos by ©H2O.ai, Inc.

Printed in the United States of America.

Contents

1 Introduction

For decades, common sense has deemed the complex, intricate formulas created by training machine learning algorithms to be uninterpretable. While it is unlikely that nonlinear, non-monotonic, and even non-continuous machine-learned response functions will ever be as directly interpretable as more traditional linear models, great advances have been made in recent years [1]. H2O Driverless AI incorporates a number of contemporary approaches to increase the transparency and accountability of complex models and to enable users to debug models for accuracy and fairness including:

- Decision tree surrogate models [2]
- Individual conditional expectation (ICE) plots [3]
- *K* local interpretable model-agnostic explanations (*K*-LIME)
- Leave-one-covariate-out (LOCO) local feature importance [4]
- Partial dependence plots [5]
- Random forest feature importance [5]

Before describing these techniques in detail, this booklet introduces fundamental concepts in machine learning interpretability (MLI) and puts forward a useful *global versus local* analysis motif. It also provides a brief, general justification for MLI and quickly examines a major practical challenge for the field: the multiplicity of good models [6]. It then presents the interpretability techniques in Driverless AI, puts forward expectations for explanation consistency across techniques, and finally, discusses several use cases.

1.1 About H2O Driverless AI

H2O Driverless AI is an artificial intelligence (AI) platform that automates some of the most difficult data science and machine learning workflows such as feature engineering, model validation, model tuning, model selection and model deployment. It aims to achieve highest predictive accuracy, comparable to expert data scientists, but in much shorter time thanks to end-to-end automation. Driverless AI also offers automatic visualizations and machine learning interpretability (MLI). Especially in regulated industries, model transparency and explanation are just as important as predictive performance.

Driverless AI runs on commodity hardware. It was also specifically designed to take advantage of graphical processing units (GPUs), including multi-GPU workstations and servers such as the NVIDIA DGX-1 for order-of-magnitude faster training.

For more information, see https://www.h2o.ai/driverless-ai/.

1.2 Machine Learning Interpretability Taxonomy

In the context of machine learning models and results, interpretability has been defined as *the ability to explain or to present in understandable terms to a human* [7]. Of course, interpretability and explanations are subjective and complicated subjects, and a previously defined taxonomy has proven useful for characterizing interpretability in greater detail for various explanatory techniques [1]. Following *Ideas on Interpreting Machine Learning*, presented approaches will be described in technical terms but also in terms of response function complexity, scope, application domain, understanding, and trust.

1.2.1 Response Function Complexity

The more complex a function, the more difficult it is to explain. Simple functions can be used to explain more complex functions, and not all explanatory techniques are a good match for all types of models. Hence, it's convenient to have a classification system for response function complexity.

Linear, monotonic functions: Response functions created by linear regression algorithms are probably the most popular, accountable, and transparent class of machine learning models. These models will be referred to here as linear and monotonic. They are transparent because changing any given input feature (or sometimes a combination or function of an input feature) changes the response function output at a defined rate, in only one direction, and at a magnitude represented by a readily available coefficient. Monotonicity also enables accountability through intuitive, and even automatic, reasoning about predictions. For instance, if a lender rejects a credit card application, they can say exactly why because their probability of default model often assumes that credit scores, account balances, and the length of credit history are linearly and monotonically related to the ability to pay a credit card bill. When these explanations are created automatically and listed in plain English, they are typically called *reason codes*. In Driverless AI, linear and monotonic functions are fit to very complex machine learning models to generate reason codes using a technique known as *K*-LIME discussed in section 2.3.

Nonlinear, monotonic functions: Although most machine learned response functions are nonlinear, some can be constrained to be monotonic with respect to any given input feature. While there is no single coefficient that represents the change in the response function induced by a change in a single input feature, nonlinear and monotonic functions are fairly transparent because their output always changes in one direction as a single input feature changes.

Nonlinear, monotonic response functions also enable accountability through the generation of both reason codes and feature importance measures. Moreover, nonlinear, monotonic response functions may even be suitable for use in regulated applications. In Driverless AI, users may soon be able to train nonlinear, monotonic models for additional interpretability.

Nonlinear, non-monotonic functions: Most machine learning algorithms create nonlinear, non-monotonic response functions. This class of functions are typically the least transparent and accountable of the three classes of functions discussed here. Their output can change in a positive or negative direction and at a varying rate for any change in an input feature. Typically, the only standard transparency measure these functions provide are global feature importance measures. By default, Driverless AI trains nonlinear, non-monotonic functions. Users may need to use a combination of techniques presented in sections 2.2 - 2.5 to interpret these extremely complex models.

1.2.2 Scope

Traditional linear models are globally interpretable because they exhibit the same functional behavior throughout their entire domain and range. Machine learning models learn local patterns in training data and represent these patterns through complex behavior in learned response functions. Therefore, machine-learned response functions may not be globally interpretable, or global interpretations of machine-learned functions may be approximate. In many cases, local explanations for complex functions may be more accurate or simply more desirable due to their ability to describe single predictions.

Global Interpretability: Some of the presented techniques facilitate global transparency in machine learning algorithms, their results, or the machine-learned relationship between the inputs and the target feature. Global interpretations help us understand the entire relationship modeled by the trained response function, but global interpretations can be approximate or based on averages.

Local Interpretability: Local interpretations promote understanding of small regions of the trained response function, such as clusters of input records and their corresponding predictions, deciles of predictions and their corresponding input observations, or even single predictions. Because small sections of the response function are more likely to be linear, monotonic, or otherwise well-behaved, local explanations can be more accurate than global explanations.

Global Versus Local Analysis Motif: Driverless AI provides both global and local explanations for complex, nonlinear, non-monotonic machine learning models. Reasoning about the accountability and trustworthiness of such complex functions can be difficult, but comparing global versus local behavior is often a

productive starting point. A few examples of *global versus local* investigation include:

- For observations with globally extreme predictions, determine if their local explanations justify their extreme predictions or probabilities.
- For observations with local explanations that differ drastically from global explanations, determine if their local explanations are reasonable.
- For observations with globally median predictions or probabilities, analyze whether their local behavior is similar to the model's global behavior.

1.2.3 Application Domain

Another important way to classify interpretability techniques is to determine whether they are model-agnostic (meaning they can be applied to different types of machine learning algorithms) or model-specific (meaning techniques that are only applicable for a single type or class of algorithms). In Driverless AI, decision tree surrogate, ICE, *K*-LIME, and partial dependence are all model-agnostic techniques, whereas LOCO and random forest feature importance are model-specific techniques.

1.2.4 Understanding and Trust

Machine learning algorithms and the functions they create during training are sophisticated, intricate, and opaque. Humans who would like to use these models have basic, emotional needs to understand and trust them because we rely on them for our livelihoods or because we need them to make important decisions for us. The techniques in Driverless AI enhance understanding and transparency by providing specific insights into the mechanisms and results of the generated model and its predictions. The techniques described here enhance trust, accountability, and fairness by enabling users to compare model mechanisms and results to domain expertise or reasonable expectations and by allowing users to observe or ensure the stability of the Driverless AI model.

1.3 Why Machine Learning for Interpretability?

Why consider machine learning approaches over linear models for explanatory or inferential purposes? In general, linear models focus on understanding and predicting average behavior, whereas machine-learned response functions can often make accurate, but more difficult to explain, predictions for subtler aspects of modeled phenomenon. In a sense, linear models are *approximate* but create very *exact explanations*, whereas machine learning can train more *exact models*

but enables only *approximate explanations*. As illustrated in figures 1 and 2, it is quite possible that an approximate explanation of an exact model may have as much or more value and meaning than an exact interpretation of an approximate model. In practice, this amounts to use cases such as more accurate financial risk assessments or better medical diagnoses that retain explainability while leveraging sophisticated machine learning approaches.

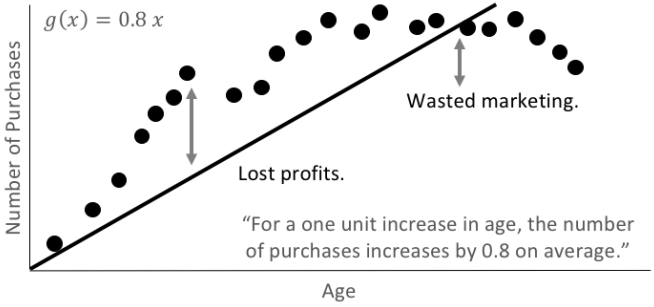

Figure 1: An illustration of *approximate* model with *exact* explanations.

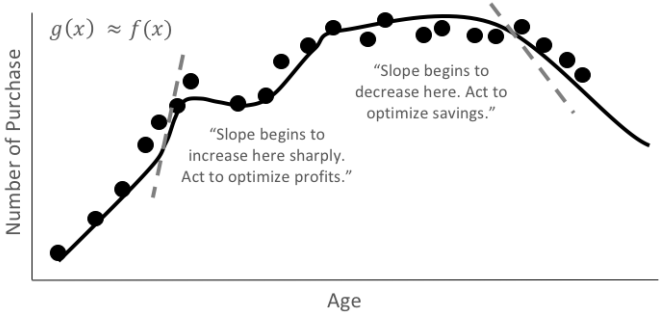

Figure 2: An illustration of an *exact* model with *approximate* explanations. Here $f(x)$ represents the true, unknown target function, which is approximated by training a machine learning algorithm on the pictured data points.

Moreover, the use of machine learning techniques for inferential or predictive purposes does not preclude using linear models for interpretation [8]. In fact, it is usually a heartening sign of stable and trustworthy results when two different predictive or inferential techniques produce similar results for the same problem.

1.4 The Multiplicity of Good Models

It is well understood that for the same set of input features and prediction targets, complex machine learning algorithms can produce multiple accurate models with very similar, but not the same, internal architectures [6]. This alone is an obstacle to interpretation, but when using these types of algorithms as interpretation tools or with interpretation tools, it is important to remember that details of explanations can change across multiple accurate models. This instability of explanations is a driving factor behind the presentation of multiple explanatory results in Driverless AI, enabling users to find explanatory information that is consistent across multiple modeling and interpretation techniques.

1.5 Citation

To cite this booklet, use the following: Hall, P., Gill, N., Kurka, M., Phan, W. (Aug 2020). *Machine Learning Interpretability with H2O Driverless AI*. http://docs.h2o.ai.

2 Interpretability Techniques

2.1 Notation for Interpretability Techniques

Spaces. Input features come from a P-dimensional input space $\mathcal{X}$ (i.e. $\mathcal{X} \in \mathbb{R}^P$). Output responses are in a C-dimensional output space $\mathcal{Y}$ (i.e. $\mathcal{Y} \in \mathbb{R}^C$).

Dataset. A dataset $\mathbf{D}$ consists of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})], \mathbf{x}^{(i)} \in \mathcal{X}, \mathbf{y}^{(i)} \in \mathcal{Y}$.

The input data can be represented as $\mathbf{X} = \left[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N-1)}\right]$. With each i-th observation denoted as an instance $\mathbf{x}^{(i)} = \left[x_0^{(i)}, x_1^{(i)}, \ldots, x_{P-1}^{(i)}\right]$ of a feature set $\mathcal{P} = \{X_0, X_1, \ldots, X_{P-1}\}$.

Learning Problem. We want to discover some *unknown target function* $f : \mathcal{X} \rightarrow \mathcal{Y}$ from our data $\mathbf{D}$. To do so, we explore a *hypothesis set* $\mathcal{H}$ and use a given learning algorithm $\mathcal{A}$ to find a function g that we hope sufficiently approximates our target function: $\mathbf{D} \xrightarrow{\mathcal{A}} g \approx f$. For a given observation $(\mathbf{x}, \mathbf{y})$, we hope that $g(\mathbf{x}) = \hat{\mathbf{y}} \approx \mathbf{y}$ and generalizes for unseen observations.

Explanation. To justify the predictions of $g(\mathbf{x})$, we may resort to a number of techniques. Some techniques will be global in scope and simply seek to generate an interpretable approximation for g itself, such that $h(\mathbf{x}) \approx g(\mathbf{x}) = \hat{\mathbf{y}}(\mathbf{x})$. Other techniques will be more local in scope and attempt to rank local contributions for each feature $X_j \in \mathcal{P}$ for some observation $\mathbf{x}^{(i)}$; this can create reason codes

for $g(\mathbf{x}^{(i)})$. Local contributions are often estimated by evaluating the product of a learned parameter β_j in g with a corresponding observed feature $x_j^{(i)}$ (i.e. $\beta_j x_j^{(i)}$), or by seeking to remove the contribution of some X_j in a prediction, $g(\mathbf{x}_{(-j)}^{(i)})$.

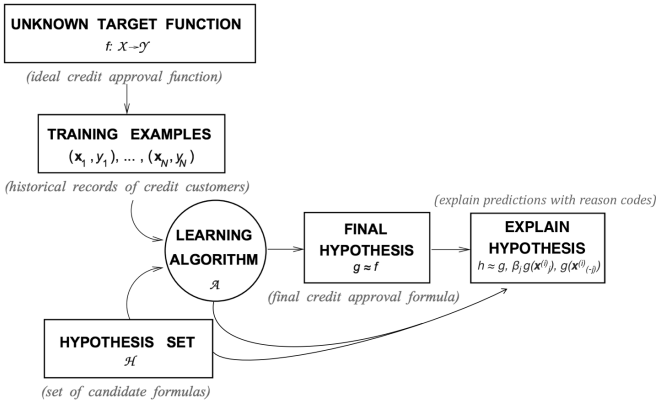

Figure 3: The learning problem. Adapted from **Learning From Data**. [9]

2.2 Decision Tree Surrogate Model

A *surrogate model* is a data mining and engineering technique in which a generally simpler model is used to explain another usually more complex model or phenomenon. Given our learned function g and set of predictions, $g(\mathbf{X}) = \hat{\mathbf{Y}}$, we can train a surrogate model h: $\mathbf{X}, \hat{\mathbf{Y}} \xrightarrow{\mathcal{A}_{\text{surrogate}}} h$, such that $h(\mathbf{X}) \approx g(\mathbf{X})$ [2]. To preserve interpretability, the hypothesis set for h is often restricted to linear models or decision trees.

For the purposes of interpretation in Driverless AI, g is considered to represent the entire pipeline, including both the feature transformations and model, and the surrogate model is a decision tree (h_{tree}). Users must also note that there exist few guarantees that h_{tree} accurately represents g. The RMSE for h_{tree} is displayed for assessing the fit between h_{tree} and g.

h_{tree} is used to increase the transparency of g by displaying an approximate flow chart of the decision making process of g as displayed in figure 4. h_{tree} also shows the likely important features and the most important interactions in g. h_{tree} can be used for visualizing, validating, and debugging g by comparing the displayed decision-process, important features, and important interactions to known standards, domain knowledge, and reasonable expectations.

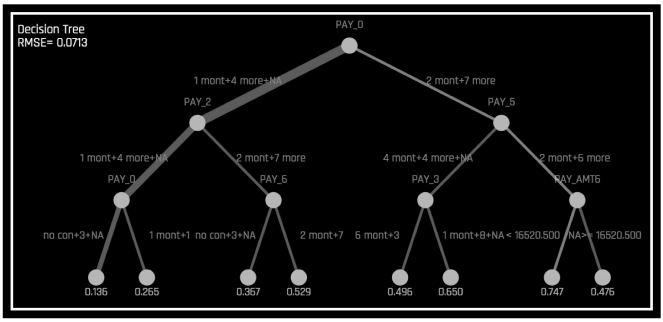

Figure 4: A summarization of a complex model's decision process as represented by a decision tree surrogate.

Figure 4 displays the decision tree surrogate, h_{tree}, for an example probability of default model, g, created with Driverless AI using the UCI repository credit card default data [10]. The PAY_0 feature is likely the most important feature in g due to its place in the initial split in h_{tree} and its second occurrence on the third level of h_{tree}. First level interactions between PAY_0 and PAY_2 and between PAY_0 and PAY_5 are visible along with several second level interactions. Following the decision path to the lowest probability leaf node in h_{tree} (figure 4 lower left) shows that customers who pay their first (PAY_0) and second (PAY_2) month bills on time are the least likely to default according to h_{tree}. The thickness of the edges in this path indicate that this is a very common decision path through h_{tree}. Following the decision path to the highest probability leaf node in h_{tree} (figure 4 second from right) shows that customers who are late on their first (PAY_0) and fifth (PAY_5) month bills and who pay less than 16520 in their sixth payment (PAY_AMT6) are the most likely to default according to h_{tree}. The thinness of the edges in this path indicate that this is a relatively rare decision path through h_{tree}. When an observation of data is selected using the *K*-LIME plot, discussed in section 2.3, h_{tree} can also provide a degree of local interpretability. When a single observation, $\mathbf{x}^{(i)}$, is selected, its path through h_{tree} is highlighted. The path of $\mathbf{x}^{(i)}$ through h_{tree} can be helpful when analyzing the logic or validity of $g(\mathbf{x}^{(i)})$.

MLI Taxonomy: Decision Tree Surrogate Models

- **Scope of Interpretability**. (1) Generally, decision tree surrogates provide global interpretability. (2) The attributes of a decision tree are used to explain global attributes of a complex Driverless AI model such as important features, interactions, and decision processes.
- **Appropriate Response Function Complexity**. Decision tree surrogate models can create explanations for models of nearly any complexity.

- **Understanding and Trust**. (1) Decision tree surrogate models foster understanding and transparency because they provide insight into the internal mechanisms of complex models. (2) They enhance trust, accountability, and fairness when their important features, interactions, and decision paths are in line with human domain knowledge and reasonable expectations.
- **Application Domain**. Decision tree surrogate models are model agnostic.

2.3 K-LIME

K-LIME is a variant of the LIME technique proposed by Ribeiro et al [8]. With *K*-LIME, local generalized linear model (GLM) surrogates are used to explain the predictions of complex response functions, and local regions are defined by *K* clusters or user-defined segments instead of simulated, perturbed observation samples. Currently in Driverless AI, local regions are segmented with K-means clustering, separating the input training data into K disjoint sets: $\{\mathbf{X}_0 \cup \mathbf{X}_1 \cup \ldots \mathbf{X}_{K-1}\} = \mathbf{X}$.

For each cluster, a local GLM model $h_{\text{GLM},k}$ is trained. K is chosen such that predictions from *all* the local GLM models would maximize R^2. This can be summarized mathematically as follows:

$$(\mathbf{X}_k, g(\mathbf{X}_k)) \xrightarrow{\mathcal{A}_{\text{GLM}}} h_{\text{GLM},k}, \forall k \in \{0, \ldots, K-1\}$$
$$\underset{K}{\text{argmax}} \; R^2(\hat{\mathbf{Y}}, h_{\text{GLM},k}(\mathbf{X}_k)), \forall k \in \{0, \ldots, K-1\}$$

K-LIME also trains one global surrogate GLM h_{global} on the *entire* input training dataset and global model predictions $g(\mathbf{X})$. If a given k-th cluster has less than 20 members, then h_{global} is used as a linear surrogate instead of $h_{\text{GLM},k}$. Intercepts, coefficients, R^2 values, accuracy, and predictions from all the surrogate K-LIME models (including the global surrogate) can be used to debug and increase transparency in g.

In Driverless AI, global *K*-LIME information is available in the global ranked predictions plot and the global section of the explanations dialog. The parameters of h_{global} give an indication of overall linear feature importance and the overall average direction in which an input feature influences g.

Figure 5 depicts a ranked predictions plot of $g(\mathbf{X})$, $h_{\text{global}}(\mathbf{X})$, and actual target values $\mathbf{Y}$ for the example probability of default model introduced in section 2.2. For N input training data observations ordered by index $i = \{0, \ldots, N-1\}$, let's *sort* the global model predictions $g(\mathbf{X})$ from smallest to largest and define

an index $\ell = \{0, \ldots, N-1\}$ for this ordering. The x-axis of the ranked prediction plot is ℓ, and the y-axis is the correspond predictions values: $g(\mathbf{x}^{(\ell)})$, $h_{\text{GLM},k}(\mathbf{x}^{(\ell)})$ (or $h_{\text{global}}(\mathbf{x}^{(\ell)})$), and $\mathbf{y}^{(\ell)}$.

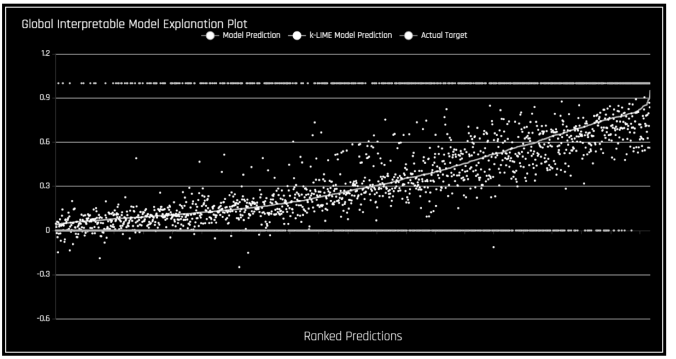

Figure 5: A ranked predictions plot of a global GLM surrogate model.

The global ranked predictions plot itself can be used as a rough diagnostic tool. In figure 5 it can be seen that g accurately models the original target, giving low probability predictions when most actual target values are 0 and giving high probability values when most actual target values are 1. Figure 5 also indicates that g behaves nonlinearly as the predictions of the global GLM surrogate, $h_{\text{global}}(\mathbf{x}^{(\ell)})$, are quite far from $g(\mathbf{x}^{(\ell)})$ in some cases. All displayed behavior of g is expected in the example use case. However, if this is not the case, users are encouraged to remove any potentially problematic features from the original data and retrain g or to retrain g on the same original features but adjust the settings in the new Driverless AI experiment to train a more acceptable model.

The coefficient parameters for each $h_{\text{GLM},k}$ can be used to profile a local region of g, to give an average description of the important features in the local region and to understand the average direction in which an input feature affects $g(\mathbf{x}^{(\ell)})$. In Driverless AI, this information is available in the ranked predictions plot for each cluster as in figure 6 or in the cluster section of the explanations dialog. While coefficient parameter values are useful, reason code values that provide the user with a feature's approximate local, linear contribution to $g(\mathbf{x}^{(\ell)})$ can be generated from *K*-LIME. Reason codes are powerful tools for accountability and fairness because they provide an explanation for each $g(\mathbf{x}^{(\ell)})$, enabling the user to understand the approximate magnitude and direction of an input feature's local contribution for $g(\mathbf{x}^{(\ell)})$. In *K*-LIME, reason code values are calculated by determining each coefficient-feature product. Reason code values are also written into automatically generated reason codes, available in the local reason code section of the explanations dialog (figure 7). A detailed example of calculating reason codes using *K*-LIME and the credit card default data

introduced in section 2.2 is explained in an upcoming sub-section.

Like all LIME explanations based on GLMs, the local explanations are linear in nature and are offsets from the baseline prediction, or intercept, which represents the average of the $h_{\text{GLM},k}$ model residuals. Of course, linear approximations to complex non-linear response functions will not always create suitable explanations, and users are urged to check the appropriate ranked predictions plot, the local GLM R^2 in the explanation dialog, and the accuracy of the $h_{\text{GLM},k}(\mathbf{x}^{(\ell)})$ prediction to understand the validity of the *K*-LIME reason codes. When $h_{\text{GLM},k}(\mathbf{x}^{(\ell)})$ accuracy for a given point or set of points is quite low, this can be an indication of extremely nonlinear behavior or the presence of strong or high-degree interactions g. In cases where $h_{\text{GLM},k}$ is not fitting g well, nonlinear LOCO feature importance values, discussed in section 2.5, may be a better explanatory tool for local behavior of g. As *K*-LIME reason codes rely on the creation of *K*-means clusters, extremely wide input data or strong correlation between input features may also degrade the quality of *K*-LIME local explanations.

K-LIME Reason Codes

For $h_{\text{GLM},k}$ and observation $\mathbf{x}^{(i)}$:

$$g(\mathbf{x}^{(i)}) \approx h_{\text{GLM},k}(\mathbf{x}^{(i)}) = \beta_0^{[k]} + \sum_{p=1}^{P} \beta_p^{[k]} x_p^{(i)} \tag{1}$$

By disaggregating the *K*-LIME predictions into individual coefficient-feature products, $\beta_p^{[k]} x_p^{(i)}$, the local, linear contribution of the feature can be determined. This coefficient-feature product is referred to as a reason code value and is used to create reason codes for each $g(\mathbf{x}^{(\ell)})$, as displayed in figures 6 and 7.

In this example, reason codes are generated by evaluating and disaggregating the local GLM presented in figure 6. The ranked predictions plot for the local GLM (cluster zero), $h_{\text{GLM},0}$, is highlighted for observation index $i = 62$ (not ranked ordered index ℓ) and displays a *K*-LIME prediction of 0.817 (i.e. $h_{\text{GLM},0}(\mathbf{x}^{(62)}) = 0.817$), a Driverless AI prediction of 0.740 (i.e. $g(\mathbf{x}^{(62)}) = 0.740$), and an actual target value of 1 (i.e. $\mathbf{y}^{(62)} = 1$). The five largest positive and negative reason code values, $\beta_p^{[0]} x_p^{(i)}$, are also displayed. Using the displayed reason code values in figure 6 and the automatically generated reason codes in figure 7 and following equation 1, it can be seen that $h_{\text{GLM},0}(\mathbf{x}^{(62)})$ is an acceptable approximation to the $g(\mathbf{x}^{(62)})$: $h_{\text{GLM},0}(\mathbf{x}^{(62)}) = 0.817 \approx g(\mathbf{x}^{(62)}) = 0.740$. This indicates displayed reason code values are likely to be accurate.

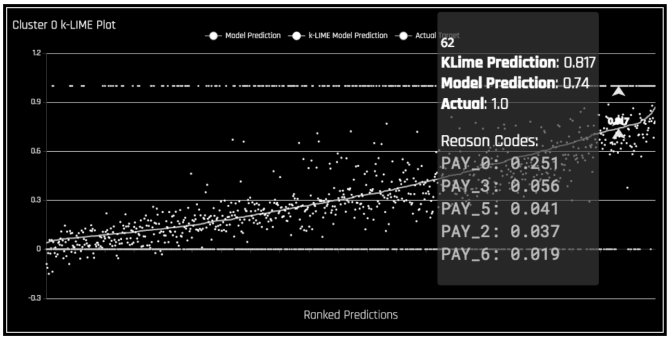

Figure 6: A ranked predictions plot of a local GLM surrogate model with selected observation and reason code values displayed.

Local Reason Codes

The baseline prediction for this cluster is: 0.418
For Row: 62 in Cluster: 0
Model Prediction Score: 0.740 k-LIME Prediction Score: 0.817
k-LIME predicts **"default.payment.next.month"** with 89.63% accuracy

Variable	with value	is associated with "default.payment.next.month" prediction's:
PAY_0	2 months delayed	increase of 25.06%
PAY_2	2 months delayed	increase of 3.66%
PAY_3	2 months delayed	increase of 5.62%
PAY_5	2 months delayed	increase of 4.15%
PAY_6	2 months delayed	increase of 1.86%
default.payment.next.month	1	target variable

Figure 7: Automatically generated reason codes.

A partial disaggregation of $h_{\text{GLM},0}$ into reason code values can also be derived from the displayed information in figures 6 and 7:

$$\begin{aligned} h_{\text{GLM},0}(\mathbf{x}^{(62)}) &= \beta_0^{[0]} + \beta_{\text{PAY\_0}}^{[0]} x_{\text{PAY\_0}}^{(62)} \\ &+ \beta_{\text{PAY\_2}}^{[0]} x_{\text{PAY\_2}}^{(62)} + \beta_{\text{PAY\_3}}^{[0]} x_{\text{PAY\_3}}^{(62)} + \beta_{\text{PAY\_5}} x_{\text{PAY\_5}}^{(62)} \\ &+ \beta_{\text{PAY\_6}}^{[0]} x_{\text{PAY\_6}}^{(62)} + ... + \beta_P^{[0]} x_P^{(62)} \end{aligned} \tag{2}$$

$$\begin{aligned} h_{\text{GLM},0}(\mathbf{x}^{(62)}) &= 0.418 + 0.251 \\ &+ 0.056 + 0.041 + 0.037 \\ &+ 0.019 + ... + \beta_P^{[0]} x_P^{(62)} \end{aligned} \tag{3}$$

where 0.418 is the intercept or baseline of $h_{\text{GLM},0}$ in figure 7, and the remaining numeric terms of equation 3 are taken from the reason code values in figure 6.

Other reason codes, whether large enough to be displayed by default or not, follow the same logic.

All of the largest reason code values in the example are positive, meaning they all contribute to the customer's high probability of default. The largest contributor to the customer's probability of default is $\beta^{[0]}_{\texttt{PAY\_0}} x^{(62)}_{\texttt{PAY\_0}}$, or in plainer terms, `PAY_0 = 2 months delayed` increases the customer's probability of default by approximately 0.25 or by 25%. This is the most important reason code in support the $g(\mathbf{x}^{(i)})$ probability for the customer defaulting next month. For this customer, according to *K*-LIME, the five most important reason codes contributing to their high $g(\mathbf{x}^{(i)})$ probability of default in ranked order are:

- `PAY_0 = 2 months delayed`
- `PAY_3 = 2 months delayed`
- `PAY_5 = 2 months delayed`
- `PAY_2 = 2 months delayed`
- `PAY_6 = 2 months delayed`

Using the *global versus local* analysis motif to reason about the example analysis results thus far, it could be seen as a sign of explanatory stability that several globally important features identified by the decision tree surrogate in section 2.2 are also appearing as locally important in *K*-LIME.

MLI Taxonomy: K-LIME

- **Scope of Interpretability**. *K*-LIME provides several different scales of interpretability: (1) coefficients of the global GLM surrogate provide information about global, average trends, (2) coefficients of in-segment GLM surrogates display average trends in local regions, and (3) when evaluated for specific in-segment observations, *K*-LIME provides reason codes on a per-observation basis.
- **Appropriate Response Function Complexity**. (1) *K*-LIME can create explanations for machine learning models of high complexity. (2) *K*-LIME accuracy can decrease when the Driverless AI model becomes too nonlinear.
- **Understanding and Trust**. (1) *K*-LIME increases transparency by revealing important input features and their linear trends. (2) *K*-LIME enhances accountability by creating explanations for each observation in a dataset. (3) *K*-LIME bolsters trust and fairness when the important features and their linear trends around specific records conform to human domain knowledge and reasonable expectations.
- **Application Domain**. *K*-LIME is model agnostic.

2.4 Partial Dependence and Individual Conditional Expectation

2.4.1 One-Dimensional Partial Dependence

For a P-dimensional feature space, we can consider a single feature $X_j \in \mathcal{P}$ and its complement set $X_{(-j)}$ (i.e $X_j \cup X_{(-j)} = \mathcal{P}$). The *one-dimensional partial dependence* of a function g on X_j is the marginal expectation:

$$\text{PD}(X_j, g) = \mathbb{E}_{X_{(-j)}}\left[g(X_j, X_{(-j)})\right] \tag{4}$$

Recall that the marginal expectation over $X_{(-j)}$ sums over the values of $X_{(-j)}$. Now we can explicitly write one-dimensional partial dependence as:

$$\begin{aligned} \text{PD}(X_j, g) &= \mathbb{E}_{X_{(-j)}}\left[g(X_j, X_{(-j)})\right] \\ &= \frac{1}{N}\sum_{i=0}^{N-1} g(X_j, \mathbf{x}_{(-j)}^{(i)}) \end{aligned} \tag{5}$$

Equation 5 essentially states that the partial dependence of a given feature X_j is the average of the response function g, setting the given feature $X_j = x_j$ and using all other existing feature vectors of the complement set $\mathbf{x}_{(-j)}^{(i)}$ as they exist in the dataset.

Partial dependence plots show the partial dependence as a function of *specific values* of our feature subset X_j. The plots show how machine-learned response functions change based on the values of an input feature of interest, while taking nonlinearity into consideration and averaging out the effects of all other input features. Partial dependence plots enable increased transparency in g and enable the ability to validate and debug g by comparing a feature's average predictions across its domain to known standards and reasonable expectations.

Figure 8 displays the one-dimensional partial dependence and ICE (see section 2.4.2) for a feature in the example credit card default data $X_j = \texttt{LIMIT\_BAL}$ for balance limits of $\mathbf{x}_j \in \{10,000, 114,200, 218,400, ..., 947,900\}$. The partial dependence (bottom Figure 8) gradually decreases for increasing values of LIMIT_BAL, indicating that the average predicted probability of default decreases as customer balance limits increase. The grey bands above and below the partial dependence curve are the standard deviations of the individual predictions, $g(\mathbf{x}^{(i)})$, across the domain of X_j. Wide standard deviation bands can indicate the average behavior of g (i.e., its partial dependence) is not highly representative of individual $g(\mathbf{x}^{(i)})$, which is often attributable to strong

interactions between X_j and some $X_{(-j)}$. As the displayed curve in figure 8 is aligned with well-known business practices in credit lending, and its standard deviation bands are relatively narrow, this result should bolster trust in g.

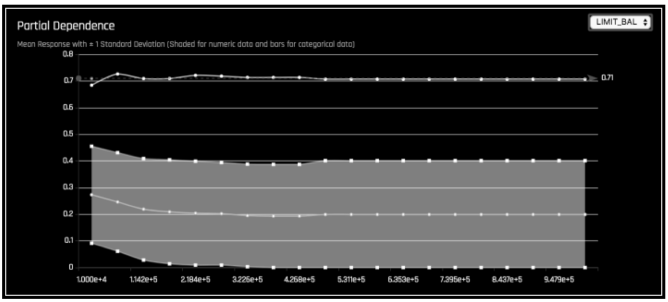

Figure 8: Partial dependence and ICE.

2.4.2 Individual Conditional Expectation

Individual conditional expectation (ICE) plots, a newer and less well-known adaptation of partial dependence plots, can be used to create more localized explanations for a single observation of data using the same basic ideas as partial dependence plots. ICE is also a type of nonlinear sensitivity analysis in which the model predictions for a single observation are measured while a feature of interest is varied over its domain.

Technically, ICE is a disaggregated partial dependence of the N responses $g(X_j, \mathbf{x}_{(-j)}^{(i)}), i \in \{1, \ldots, N\}$ (for a single feature X_j), instead of *averaging* the response across all observations of the training set [3]. An ICE plot for a single observation $\mathbf{x}^{(i)}$ is created by plotting $g(X_j = x_{j,q}, \mathbf{x}_{(-j)}^{(i)})$ versus $X_j = x_{j,q}$, $(q \in \{1, 2, \ldots\})$ while fixing $\mathbf{x}_{(-j)}^{(i)}$.

ICE plots enable a user to assess the Driverless AI model's prediction for an *individual observation of data*, $g(\mathbf{x}^{(i)})$:

1. Is it outside one standard deviation from the average model behavior represented by partial dependence?

2. Is the treatment of a specific observation valid in comparison to average model behavior, known standards, domain knowledge, and reasonable expectations?

3. How will the observation behave in hypothetical situations where one feature, X_j, in a selected observation is varied across its domain?

In Figure 8, the selected observation of interest and feature of interest are $\mathbf{x}^{(62)}$ and $X_{\texttt{LIMIT\_BAL}}$, respectively. The ICE curve's values, $g(X_{\texttt{LIMIT\_BAL}}, \mathbf{x}^{(62)}_{(-\texttt{LIMIT\_BAL})})$, are much larger than $\text{PD}(X_{\texttt{LIMIT\_BAL}}, g)$, clearly outside of the grey standard deviation regions. This result is parsimonious with previous findings, as the customer represented by $\mathbf{x}^{(62)}$ was shown to have a high probability of default due to late payments in section 2.3. Figure 8 also indicates that the prediction behavior for the customer represented by $\mathbf{x}^{(62)}$ is somewhat rare in the training data, and that no matter what balance limit the customer is assigned, they will still be very likely to default according the Driverless AI model.

MLI Taxonomy: Partial Dependence and ICE

- **Scope of Interpretability**. (1) Partial dependence is a global interpretability measure. (2) ICE is a local interpretability measure.
- **Appropriate Response Function Complexity**. Partial dependence and ICE can be used to explain response functions of nearly any complexity.
- **Understanding and Trust**. (1) Partial dependence and ICE increase understanding and transparency by describing the nonlinear behavior of complex response functions. (2) Partial dependence and ICE enhance trust, accountability, and fairness by enabling the comparison of described nonlinear behavior to human domain knowledge and reasonable expectations. (3) ICE, as a type of sensitivity analysis, can also engender trust when model behavior on simulated data is acceptable.
- **Application Domain**. Partial dependence and ICE are model-agnostic.

2.5 Feature Importance

Feature importance measures the effect that a feature has on the predictions of a model. Global feature importance measures the overall impact of an input feature on the Driverless AI model predictions while taking nonlinearity and interactions into consideration. Global feature importance values give an indication of the magnitude of a feature's contribution to model predictions for all observations. Unlike regression parameters, they are often unsigned and typically not directly related to the numerical predictions of the model. Local feature importance describes how the combination of the learned model rules or parameters and an individual observation's attributes affect a model's prediction for that observation while taking nonlinearity and interactions into effect.

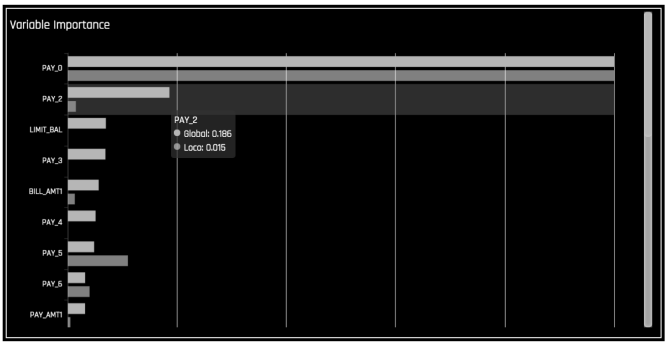

Figure 9: Global random forest and local LOCO feature importance.

2.5.1 Random Forest Feature Importance

Currently in Driverless AI, a random forest surrogate model h_{RF} consisting of B decision trees $h_{\text{tree},b}$ is trained on the predictions of the Driverless AI model.

$$h_{\text{RF}}(\mathbf{x}^{(i)}) = \frac{1}{B}\sum_{b=1}^{B} h_{\text{tree},b}\left(\mathbf{x}^{(i)}; \Theta_b\right), \tag{6}$$

Here Θ_b is the set of splitting rules for each tree $h_{\text{tree},b}$. As explained in [5], at each split in each tree $h_{\text{tree},b}$, the improvement in the split-criterion is the importance measure attributed to the splitting feature. The importance measure is accumulated over all trees seperately for each feature. The aggregated feature importance values are then scaled between 0 and 1, such that the most important feature has an importance value of 1.

Figure 9 displays the global and local feature importance values for the credit card default data, sorted in descending order from the globally most important feature to the globally least important feature. Local feature importance values are displayed under the global feature importance value for each feature. In figure 9, `PAY_0`, `PAY_2`, `LIMIT_BAL`, `PAY_3`, and `BILL_AMT1` are the top 5 most important features globally. As expected, this result is well aligned with the results of the decision tree surrogate model discussed in section 2.2. Taking the results of two interpretability techniques into consideration, it is extremely likely that timing of the customer's first 3 payments, `PAY_0`, `PAY_2`, and `PAY_3`, are the most important global features for any $g(\mathbf{x}^{(i)})$ prediction.

2.5.2 LOCO Feature Importance

Leave-one-covariate-out (LOCO) provides a mechanism for calculating feature importance values for any model g on a per-observation basis $\mathbf{x}^{(i)}$ by subtracting the model's prediction for an observation of data, $g(\mathbf{x}^{(i)})$, from the model's prediction for that observation of data *without* an input feature X_j of interest, $g(\mathbf{x}^{(i)}_{(-j)}) - g(\mathbf{x}^{(i)})$ [4]. LOCO is a model-agnostic idea, and $g(\mathbf{x}^{(i)}_{(-j)})$ can be calculated in various ways. However, in Driverless AI, $g(\mathbf{x}^{(i)}_{(-j)})$ is currently calculated using a model-specific technique in which the contribution X_j to $g(\mathbf{x}^{(i)})$ is approximated by using random forest surrogate model h_{RF}. Specifically, the prediction contribution of any rule $\theta_r^{[b]} \in \Theta_b$ containing X_j for tree $h_{\text{tree},b}$ is subtracted from the *original prediction* $h_{\text{tree},b}\left(\mathbf{x}^{(i)}; \Theta_b\right)$. For the random forest:

$$g(\mathbf{x}^{(i)}_{(-j)}) = h_{\text{RF}}(\mathbf{x}^{(i)}_{(-j)}) = \frac{1}{B}\sum_{b=1}^{B} h_{\text{tree},b}\left(\mathbf{x}^{(i)}; \Theta_{b,(-j)}\right), \tag{7}$$

where $\Theta_{b,(-j)}$ is the set of splitting rules for each tree $h_{\text{tree},b}$ with the *contributions of all rules involving feature X_j removed*. Although LOCO feature importance values can be signed quantities, they are scaled between 0 and 1 such that the most important feature for an observation of data, $\mathbf{x}^{(i)}$, has an importance value of 1 for direct *global versus local* comparison to random forest feature importance in Driverless AI.

In figure 9, LOCO local feature importance values are displayed under global random forest feature importance values for each X_j, and the global and local feature importance for `PAY_2` are highlighted for $\mathbf{x}^{(62)}$ in the credit card default data. From the nonlinear LOCO perspective, the most important local features for $g(\mathbf{x}^{(62)})$ are `PAY_0`, `PAY_5`, `PAY_6`, `PAY_2`, and `BILL_AMT1`. Because there is good alignment between the linear *K*-LIME reason codes and LOCO local feature importance values, it is extremely likely that `PAY_0`, `PAY_2`, `PAY_5`, and `PAY_6` are the most locally important features contributing to $g(\mathbf{x}^{(62)})$.

MLI Taxonomy: Fearture Importance

- **Scope of Interpretability**. (1) Random forest feature importance is a global interpretability measure. (2) LOCO feature importance is a local interpretability measure.
- **Appropriate Response Function Complexity**. Both random forest and LOCO feature importance can be used to explain tree-based response functions of nearly any complexity.
- **Understanding and Trust**. (1) Random forest feature importance increases transparency by reporting and ranking influential input features.

(2) LOCO feature importance enhances accountability by creating explanations for each model prediction. (3) Both global and local feature importance enhance trust and fairness when reported values conform to human domain knowledge and reasonable expectations.

- **Application Domain**. (1) Random forest feature importance is a model-specific explanatory technique. (2) LOCO is a model-agnostic concept, but its implementation in Driverless AI is model specific.

2.6 Expectations for Consistency Between Explanatory Techniques

Because machine learning models have intrinsically high variance, it is recommended that users look for explanatory themes that occur across multiple explanatory techniques and that are parsimonious with reasonable expectations or human domain knowledge. However, when looking for similarities between decision tree surrogates, *K*-LIME, partial dependence, ICE, sensitivity analysis, and feature importance, note that each technique is providing a different perspective into a complex, nonlinear, non-monotonic, and even noncontinuous response function.

The decision tree surrogate is a global, nonlinear description of the Driverless AI model behavior. Features that appear in the tree should have a direct relationship with features that appear in the global feature importance plot. For more linear Driverless AI models, features that appear in the decision tree surrogate model may also have large coefficients in the global *K*-LIME model.

K-LIME explanations are linear, do not consider interactions, and represent offsets from the local GLM intercept. LOCO importance values are nonlinear, do consider interactions, and do not explicitly consider a linear intercept or offset. *K*-LIME explanations and LOCO importance values are not expected to have a direct relationship but should align roughly as both are measures of a feature's local contribution on a model's predictions, especially in more linear regions of the Driverless AI model's learned response function.

ICE has a complex relationship with LOCO feature importance values. Comparing ICE to LOCO can only be done at the value of the selected feature that actually appears in the selected observation of the training data. When comparing ICE to LOCO, the total value of the prediction for the observation, the value of the feature in the selected observation, and the distance of the ICE value from the average prediction for the selected feature at the value in the selected observation must all be considered.

Partial dependence takes into consideration nonlinear, but average, behavior of the complex Driverless AI model. Strong interactions between input features

can cause ICE values to diverge from partial dependence values. ICE curves that are outside the standard deviation of partial dependence would be expected to fall into less populated decision paths of the decision tree surrogate; ICE curves that lie within the standard deviation of partial dependence would be expected to belong to more common decision paths.

2.7 Correcting Unreasonable Models

Once users have gained an approximate understanding of the Driverless AI response function using the tools described in sections 2.2 - 2.5, it is crucial to evaluate whether the displayed global and local explanations are reasonable. Explanations should inspire trust and confidence in the Driverless AI model. If this is not the case, users are encouraged to remove potentially problematic features from the original data and retrain the Driverless AI model. Users may also retrain the model on the same original features but change the settings in the Driverless AI experiment to create a more reasonable model.

3 Use Cases

3.1 Use Case: Titanic Survival Model Explanations

We have trained a Driverless AI model to predict survival on the well-known Titanic dataset. The goal of this use case is to explain and validate the mechanisms and predictions of the Driverless AI model using the techniques presented in sections 2.2 - 2.5.

3.1.1 Data

The Titanic dataset is available from: `https://s3.amazonaws.com/h2o-public-test-data/smalldata/gbm_test/titanic.csv`.

The data consist of passengers on the Titanic. The prediction target is whether or not a passenger survived (`survive`). The dataset contains 1,309 passengers, of which 500 survived. Several features were removed from the dataset including `name`, `boat`, `body`, `ticket`, and `home.dest` due to data leakage, as well as ambiguities that can hinder interpreting the Driverless AI model. The remaining input features are summarized in tables 1 and 2.

Table 1: Summary of numeric input features in the Titanic dataset.

Statistic	N	Mean	St. Dev.	Min	Max
age	1,309	29.88	14.41	0.17	80
sibsp	1,309	0.50	1.04	0	8
parch	1,309	0.39	0.87	0	9
fare	1,309	33.30	51.76	0	512.33

Table 2: Summary of categorical input features in the Titanic dataset.

	pclass	sex	cabin	embarked
1	1:323	female:466	:1014	: 2
2	2:277	male :843	C23 C25 C27 : 6	C:270
3	3:709		B57 B59 B63 B66: 5	Q:123
4			G6 : 5	S:914
5			B96 B98 : 4	
6			C22 C26 : 4	
7			(Other) : 271	

`Sex`, `pclass`, and `age` are expected to be globally important in the Driverless AI model. The summaries show that woman are much more likely to survive than men (73% vs 19%) and that first class passengers and children have a survival rate of over 50% compared with the overall survival rate of 38%.

3.1.2 K-LIME

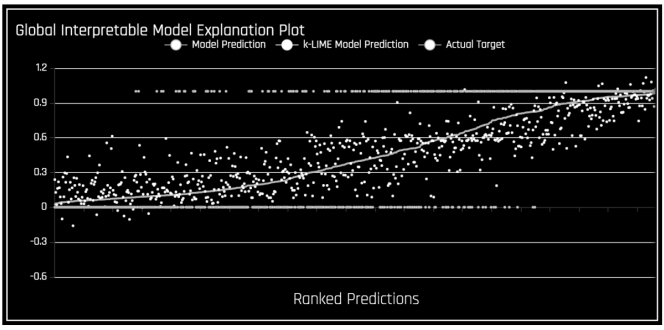

Figure 10: *K*-LIME plot for the Titanic survival model.

The *K*-LIME plot in figure 10 shows the Driverless AI model predictions as a continuous curve starting on the lower left and ending in the upper right. The *K*-LIME model predictions are the discontinuous points around the Driverless AI model predictions. Considering the global explanations in figure 11, we can also see that the *K*-LIME predictions generally follow the Driverless AI model's predictions, and the global *K*-LIME model explains about 75% of the variability in the Driverless AI model predictions, indicating that global explanations are approximate, but reasonably so.

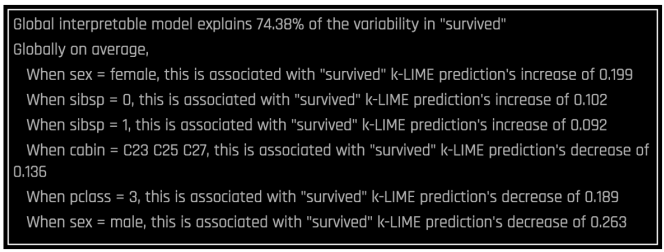

Figure 11: Global explanations for the Titanic survival model.

Figure 11 presents global explanations for the Driverless AI model. The explanations provide a linear understanding of input features and the outcome, `survive`, in plain English. As expected, they indicate that `sex` and `pclass` make the largest global, linear contributions to the Driverless AI model.

3.1.3 Feature Importance

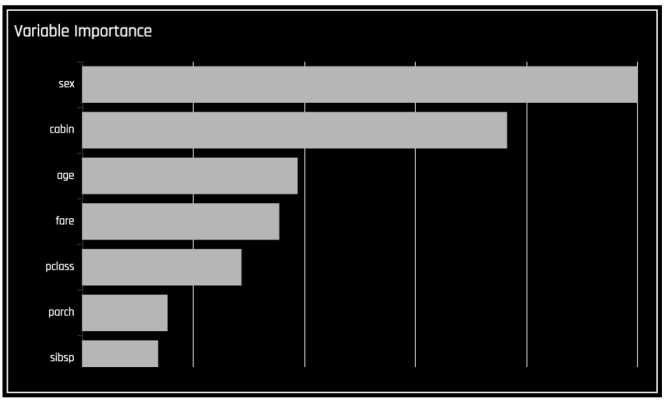

Figure 12: Feature importance for the Titanic survival model.

The features with the greatest importance values in the Driverless AI model are `sex`, `cabin`, and `age` as displayed in figure 12. Class is not ranked as a top feature, and instead cabin is assigned a high importance value. `cabin` denotes the cabin location of the passenger. The first letter of the cabin tells us the deck level of the passenger. For example, all cabins that start with `A` correspond to cabins in the upper promenade deck. Further data exploration indicates that first class passengers stayed in the top deck cabins, above second class passengers in the middle decks, and above third class passengers at the lowest deck levels. This correlation between `cabin` and `pclass` may explain why `cabin` is located higher than `pclass` in the global feature importance plot, especially if `cabin` contains similar but more granular information than `pclass`.

The feature importance figure matches hypotheses created during data exploration to a large extent. Feature importance, however, does not explain the relationship between a feature and the Driverless AI model's predictions. This is where we can examine partial dependence plots.

3.1.4 Partial Dependence Plots

The partial dependence plots show how different values of a feature affect the average prediction of the Driverless AI model. Figure 13 displays the partial dependence plot for `sex` and indicates that predicted survival increases dramatically for female passengers.

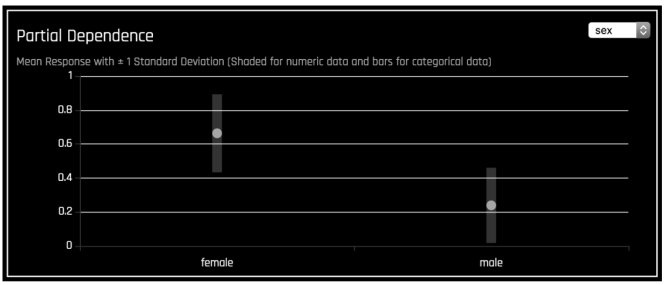

Figure 13: Partial dependence plot for `sex` for the Titanic survival model.

Figure 14 displays the partial dependence plot for `age`. The Driverless AI model predicts high probabilities for survival for passengers younger than 17. After the age of 17, increases in age do not result in large changes in the Driverless AI model's average predictions. This result is in agreement with previous findings in which children have a higher probability of survival.

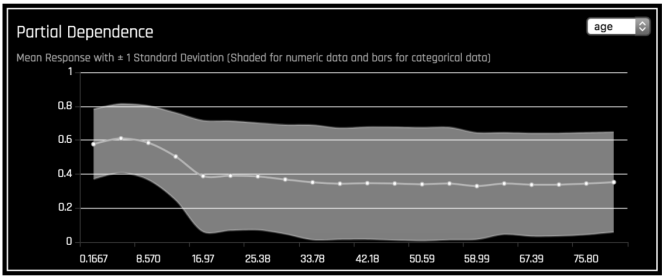

Figure 14: Partial dependence plot for `age` for the Titanic survival model.

3.1.5 Decision Tree Surrogate

In figure 15, the RMSE of 0.164 indicates the decision tree surrogate is able to approximate the Driverless AI model well. By following the decision paths down the decision tree surrogate, we can begin to see details in the Driverless AI model's decision processes. For example, it is not totally accurate to assume that all women will survive. Women in third class who paid a small amount for their fare actually have a lower prediction than the overall average survival rate. Likewise, there are some groups of men with high average predictions. Men with a cabin beginning in `A` (the top deck of the ship) have an average survival prediction greater than 75%.

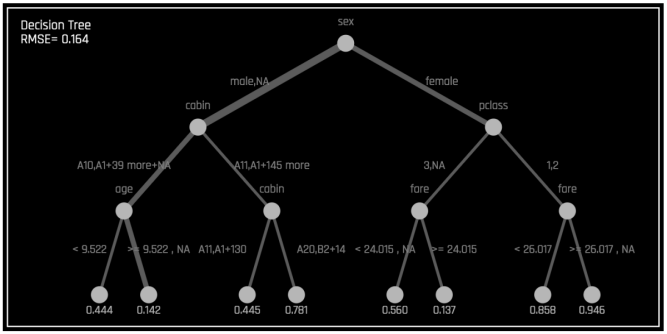

Figure 15: Decision tree surrogate for the Titanic survival model.

3.1.6 Local Explanations

Following the *global versus local* analysis motif, local contributions to model predictions for a single passenger are also analyzed and compared to global

explanations and reasonable expectations. Figure 16 shows the local dashboard after selecting a single passenger in the K-LIME plot. For this example use case, a female with a first class ticket is selected.

Figure 16: Local interpretability dashboard for a single female passenger.

In figure 16, the path of the selected individual through the far right decision path in the decision tree surrogate model is highlighted. This selected passenger falls into the leaf node with the greatest average model prediction for survival, which is nicely aligned with the Driverless AI model's predicted probability for survival of 0.89.

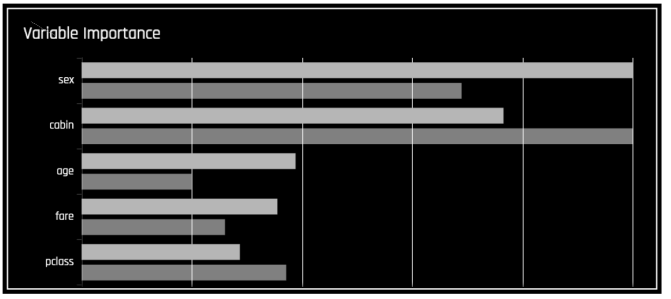

Figure 17: Local feature importance for a single female passenger.

When investigating observations locally, the feature importance has two bars per feature. The upper bar represents the global feature importance and the lower bar represents the local feature importance. In figure 17, the two features `sex` and `cabin` are the most important features both globally and locally for the selected individual.

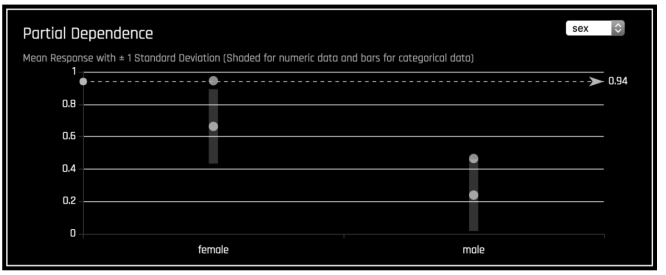

Figure 18: Partial dependence and ICE for a single female passenger.

The local dashboard also overlays ICE curves onto partial dependence plots. In figure 18, the lower points for partial dependence remain unchanged from figure 13 and show the average model prediction by `sex`. The upper points indicate how the selected passenger's prediction would change if their value for `sex` changed, and figure 18 indicates that her prediction for survival would decrease dramatically if her value for `sex` changed to `male`. Figure 18 also shows that the selected passenger is assigned a higher-than-average survival rate regardless of sex. This result is most likely due to the selected individual being a first class passenger.

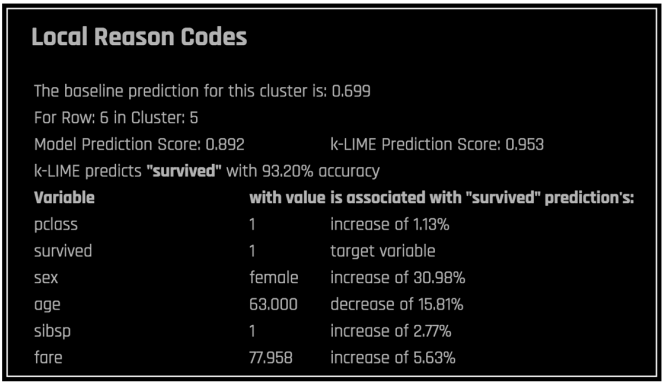

Figure 19: Local reason codes for a single female passenger.

The local English language explanations, or reason codes, from the *K*-LIME model in figure 19 parsimoniously indicate that the Driverless AI model's prediction increased for the selected passenger due to her value for `sex` and `fare` and decreased due to her relatively old age. For the selected passenger, global and local explanations are reasonable when compared to one-another

and to logical expectations. In practice, explanations for several different types of passengers, and especially for outliers and other anomalous observations, should be investigated and analyzed to enhance understanding and trust in the Driverless AI model.

3.2 Use Case: King County Housing Model Explanations

We have trained a Driverless AI model to predict housing prices in King County, Washington. The goal of this use case is to explain and validate the mechanisms and predictions of the trained Driverless AI model using the techniques presented in sections 2.2 - 2.5.

3.2.1 Data

The housing prices dataset is available from: https://www.kaggle.com/harlfoxem/housesalesprediction.

This dataset contains house sale prices for King County, Washington, which includes Seattle. It includes homes sold between May 2014 and May 2015. The prediction target is housing price, `price`. Several features were removed from the analysis including `ID`, `date`, `latitude`, `longitude`, `zipcode`, and several other ambiguous or multicollinear features that could hinder interpretability. The remaining input features are summarized in tables 3 and 4. The outcome `price` is right-skewed, requiring a log transform before training a Driverless AI model.

Table 3: Summary of numeric input features in the housing prices dataset.

Statistic	N	Mean	St. Dev.	Min	Max
price	21,613	540,088.100	367,127.200	75,000	7,700,000
sqft_living	21,613	2,079.900	918.441	290	13,540
sqft_lot	21,613	15,106.970	41,420.510	520	1,651,359
sqft_above	21,613	1,788.391	828.091	290	9,410
sqft_basement	21,613	291.509	442.575	0	4,820
yr_built	21,613	1,971.005	29.373	1,900	2,015
yr_renovated	21,613	84.402	401.679	0	2,015

Table 4: Summary of categorical input features in the housing prices dataset.

	bathrooms	bedrooms	floors	waterfront	view	condition
1	2.5 :5380	3 :9824	1.0:10680	0:21450	0:19489	1: 30
2	1.0 :3852	4 :6882	1.5: 1910	1: 163	1: 332	2: 172
3	1.75 :3048	2 :2760	2.0: 8241		2: 963	3:14031
4	2.25 :2047	5 :1601	2.5: 161		3: 510	4: 5679
5	2.0 :1930	6 : 272	3.0: 613		4: 319	5: 1701
6	1.5 :1446	1 : 199	3.5: 8			
7	(Other):3910	(Other): 75				

Living square footage, `sqft_living`, is linearly associated with increases in `price` with a correlation coefficient of greater than 0.6. There is also a linearly increasing trend between the number of `bathrooms` and the home `price`. The more `bathrooms`, the higher the home `price`. Hence, inputs related to square footage and the number of `bathrooms` are expected to be globally important in the Driverless AI model.

3.2.2 K-LIME

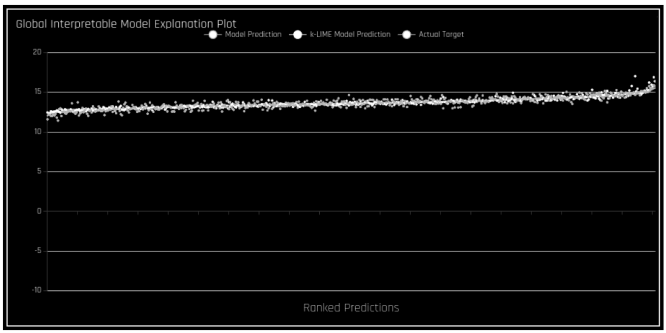

Figure 20: *K*-LIME plot for the King County home prices model.

The *K*-LIME plot in figure 20 shows the Driverless AI model predictions as a continuous curve starting at the middle left and ending in the upper right. The *K*-LIME model predictions are the discontinuous points around the Driverless AI model predictions. In figure 20, *K*-LIME accurately traces the original target, and according to figure 21, *K*-LIME explains 92 percent of the variability in the Driverless AI model predictions. The close fit of *K*-LIME to the Driverless AI model indicates that the global explanations in figure 21 very likely to be insightful and trustworthy.

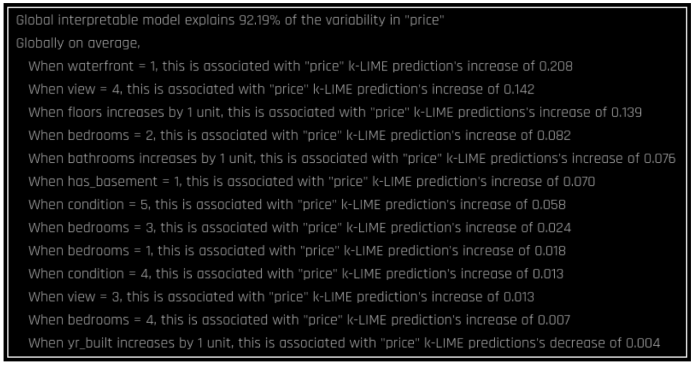

Figure 21: Global explantions for the King County home prices model.

The explanations provide a linear understanding of input features and the outcome, `price`. (Note, the *K*-LIME explanations are in the `log()` space for `price`.) For example:

> When `bathrooms` increase by 1 unit, this is associated with `price` *K*-LIME predictions increase of 0.076

This particular explanation falls in line with findings from data exploration in section 3.2.1 and with reasonable expectations. The more `bathrooms` a house has, the higher the `price`.

Another interesting global explanation relates to the year when a house was built.

> When `yr_built` increases by 1 unit, this is associated with `price` *K*-LIME predictions decrease of 0.004

This explanation indicates newer homes will have a lower `price` than older homes. We will explore this, perhaps counterintuitive, finding further with a partial dependence plot in section 3.2.4 and a decision tree surrogate model in section 3.2.5.

3.2.3 Feature Importance

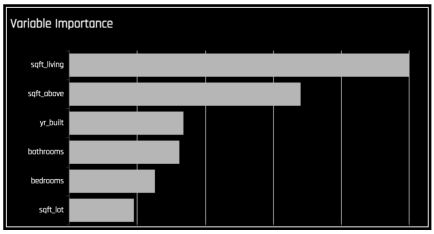

Figure 22: Feature Importance for the King County home prices model.

According to figure 22, the features with the largest global importance in the Driverless AI model are `sqft_living`, `sqft_above`, `yr_built`, and `bathrooms`. The high importance of `sqft_living` (total square footage of the home), `sqft_above` (total square footage minus square footage of the basement), and `bathrooms` follow trends observed during data exploration and are parsimonious with reasonable expectations. `yr_built` is also playing a large role in the predictions for `price`, beating out `bathrooms` just slightly.

3.2.4 Partial Dependence Plots

Partial dependence plots show how different values of a feature affect the average prediction for `price` in the Driverless AI model. Figure 23 indicates that as `sqft_living` increases, the average prediction of the Driverless AI model also increases.

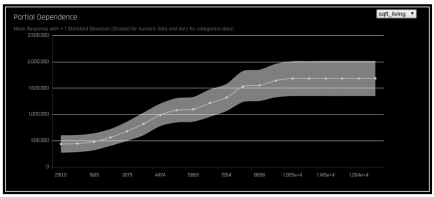

Figure 23: Partial dependence plot of `sqft_living` for the King County home prices model.

In figure 24, the partial dependence for `bathrooms` can be seen to increase slightly as the total number of `bathrooms` increases. Figures 23 and 24 are aligned with findings from data exploration and reasonable expectations.

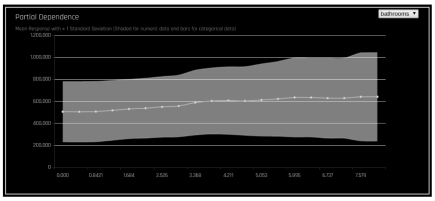

Figure 24: Partial dependence plot of `bathrooms` for the King County home prices model.

Figure 25 displays the partial dependence for `yr_built`, where Driverless AI model predictions for `price` decrease as the age of the house decreases. Figure 25 confirms the global *K*-LIME explanation for `yr_built` discussed in section 3.2.2. A relationship of this nature could indicate something unique about King County, Washington, but requires domain knowledge to decide whether the finding is valid or calls the Driverless AI model's behavior into question.

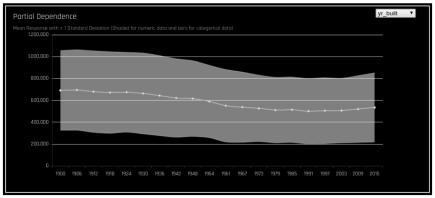

Figure 25: Partial dependence plot of `yr_built` for the King County home prices model.

3.2.5 Decision Tree Surrogate

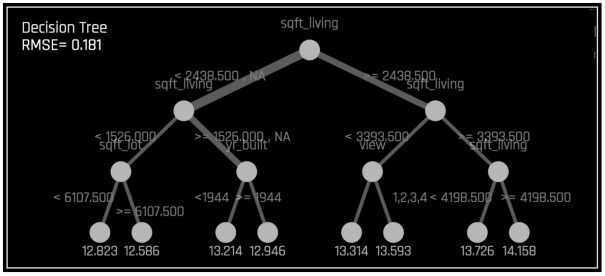

Figure 26: Decision tree surrogate for the King County home prices model.

In figure 26, the RMSE of 0.181 indicates the decision tree surrogate is able to approximate the Driverless AI model well. By following the decision paths down the decision tree surrogate, we can begin to see details in the Driverless AI model's decision processes. `sqft_living` is found in the first split and in several other splits of the decision tree surrogate, signifying its overall importance in the Driverless AI model and in agreement with several previous findings.

Moving down the left side of the decision tree surrogate, `yr_built` interacts with `sqft_living`. If `sqft_living` is greater than or equal to 1,526 square feet and the home was built before 1944, then the price is predicted to be `log(13.214)`, which is about $547,983, but if the home was built after 1944, then the price is predicted to be `log(12.946)`, which is about $419,156. This interaction, which is clearly expressed in the decision tree surrogate model, provides more insight into why older homes are predicted to cost more as discussed in sections 3.2.2 and 3.2.4.

3.2.6 Local Explanations

Following the *global versus local* analysis motif, local contributions to model predictions for a single home are also analyzed and compared to global explanations and reasonable expectations. Figure 27 shows the local dashboard after selecting a single home in the *K*-LIME plot. For this example use case, the least expensive home is selected.

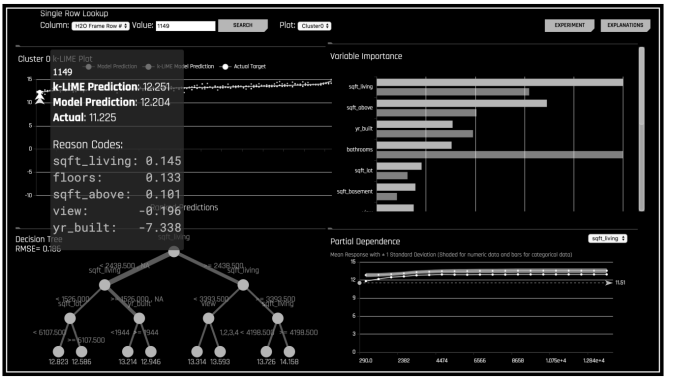

Figure 27: Local interpretability dashboard for the least expensive home.

Figure 28 shows the feature importance plot with two bars per feature. The top bar represents the global feature importance and the bottom bar represents the local feature importance. Locally for the least expensive home, `bathrooms` is the most important feature followed by `sqft_living`, `sqft_above`, and

yr_built. What might cause this difference between global and local feature importance values? It turns out the least expensive home has zero bathrooms!

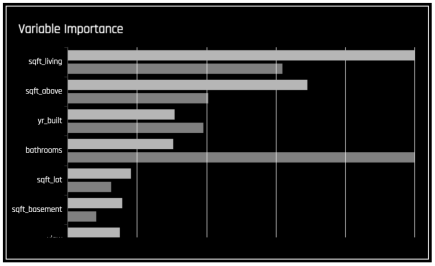

Figure 28: Local feature importance for the least expensive home.

The local dashboard also overlays ICE onto the partial dependence plot, as seen in figure 29. The upper partial dependence curve in figure 29 remains unchanged from figure 24 and shows the average Driverless AI model prediction by the number of bathrooms. The lower ICE curve in figure 29 indicates how the least expensive home's price prediction would change if its number of bathrooms changed, following the global trend of more bathrooms leading to higher predictions for price.

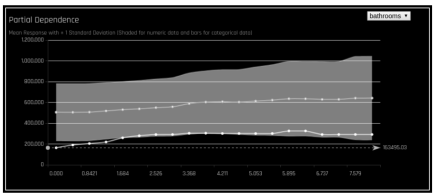

Figure 29: Partial dependence and ICE for the least expensive home.

The local English language explanations, i.e. *reason codes*, from *K*-LIME in figure 30 show that the Driverless AI model's prediction increased for this home due to its relatively high sqft_living and decreased due to its relatively recent yr_built, which is aligned with global explanations. Note that the number of bathrooms is not considered in *K*-LIME reason codes because this home does not have any bathrooms. Since this home's local value for bathrooms is zero, bathrooms cannot contribute to the *K*-LIME model.

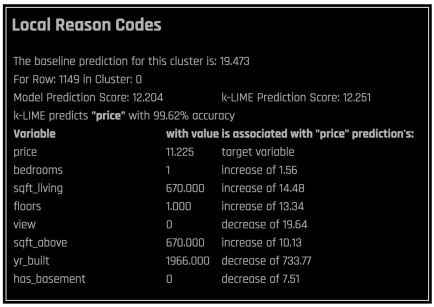

Figure 30: Reason codes for the least expensive home.

Continuing with the *global versus local* analysis, explanations for the most expensive home are considered briefly. In figure 31, the two features `sqft_living` and `sqft_above` are the most important features locally along with `bathrooms` and `yr_built`. The data indicate the most expensive home has eight `bathrooms`, 12050 square feet of total square footage in which 8570 is allocated for living space (not including the basement), and the home was built in 1910. Following global explanations and reasonable expectations, this most expensive home has characteristics that justify it's high prediction for `price`.

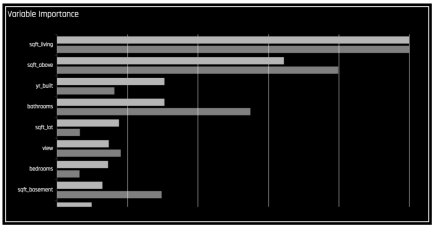

Figure 31: Local feature importance for the most expensive home.

For the selected homes, global and local explanations are reasonable when compared to one-another and to logical expectations. In practice, explanations for several different types of homes, and especially for outliers and other anomalous observations, should be investigated and analyzed to enhance understanding and trust in the Driverless AI model.

4 Acknowledgements

The authors would like to acknowledge the following individuals for their invaluable contributions to the software described in this booklet: Leland Wilkinson, Mark Chan, and Michal Kurka.

5 References

1. Patrick Hall, Wen Phan, and Sri Satish Ambati. Ideas on interpreting machine learning. *O'Reilly Ideas*, 2017. URL https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning

2. 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

3. 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

4. Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association just-accepted*, 2017. URL http://www.stat.cmu.edu/~ryantibs/papers/conformal.pdf

5. 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

6. Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3), 2001. URL https://projecteuclid.org/euclid.ss/1009213726

7. Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiV preprint*, 2017

8. 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

9. 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

10. M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml

6 Authors

Patrick Hall

Patrick Hall is senior director for data science products at H2O.ai where he focuses mainly on model interpretability. Patrick is also currently an adjunct professor in the Department of Decision Sciences at George Washington University, where he teaches graduate classes in data mining and machine learning. Prior to joining H2O.ai, Patrick held global customer facing roles and research and development roles at SAS Institute.

Follow him on Twitter: @jpatrickhall

Navdeep Gill

Navdeep is a software engineer and data scientist at H2O.ai. He graduated from California State University, East Bay with a M.S. degree in Computational Statistics, B.S. in Statistics, and a B.A. in Psychology (minor in Mathematics). During his education he gained interests in machine learning, time series analysis, statistical computing, data mining, and data visualization. Previous to H2O.ai, Navdeep worked at Cisco Systems, Inc. focusing on data science and software development, and before stepping into industry he worked in various Neuroscience labs as a researcher and analyst.

Follow him on Twitter: @Navdeep_Gill_

Megan Kurka

Megan is a customer data scientist at H2O.ai. Prior to working at H2O.ai, she worked as a data scientist building products driven by machine learning for B2B customers. Megan has experience working with customers across multiple industries, identifying common problems, and designing robust and automated solutions.

Wen Phan

Wen Phan is director of customer engineering and data science at H2O.ai. Wen works with customers and organizations to architect systems, smarter applications, and data products to make better decisions, achieve positive outcomes, and transform the way they do business. Wen holds a B.S. in electrical engineering and M.S. in analytics and decision sciences.

Follow him on Twitter: @wenphan