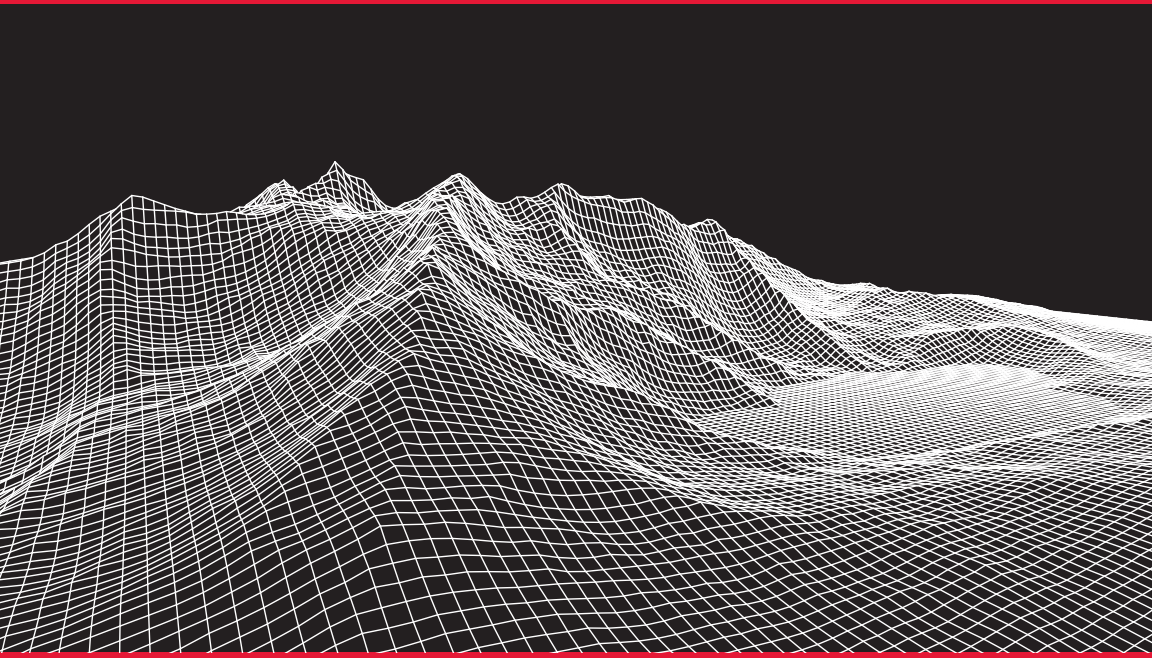
O'REILLY®

Compliments of dataiku

An Introduction to Machine Learning Interpretability

An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI



Patrick Hall & Navdeep Gill

"With machine learning and AI gaining momentum in the enterprise, model management has become more critical than ever to build trust. Trust in systems, trust in data, and trust in results - but most of all, trust in people."

Florian Douetteau, Dataiku CEO

dataiku

Your Path to [Explainable] Enterprise AI

Dataiku is the centralized platform that helps companies bring transparency around data and data processes to empower everyone throughout the enterprise - not just a siloed group - to draw and use insights from data on their own.

200+
CUSTOMERS

20,000+
ACTIVE USERS
*data scientists, analysts, engineers, & more

GE · SEPHORA · Unilever · KUKA · FOX NETWORKS GROUP · BNP PARIBAS

www.dataiku.com

An Introduction to Machine Learning Interpretability

An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI

Patrick Hall and Navdeep Gill

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

An Introduction to Machine Learning Interpretability
by Patrick Hall and Navdeep Gill

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com/safari*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Nicole Tache
Production Editor: Nicholas Adams
Copyeditor: Octal Publishing, Inc.
Interior Designer: David Futato
Cover Designer: Randy Comer
Illustrator: Rebecca Demarest

April 2018: First Edition

Revision History for the First Edition
2018-03-28: First Release

978-1-492-05029-2

[LSI]

Table of Contents

An Introduction to Machine Learning Interpretability

Understanding and trusting models and their results is a hallmark of good science. Scientists, engineers, physicians, researchers, and humans in general have the need to understand and trust models and modeling results that affect their work and their lives. However, the forces of innovation and competition are now driving analysts and data scientists to try ever-more complex predictive modeling and machine learning algorithms. Such algorithms for machine learning include gradient-boosted ensembles (GBM), artificial neural networks (ANN), and random forests, among many others. Many machine learning algorithms have been labeled "black box" models because of their inscrutable inner-workings. What makes these models accurate is what makes their predictions difficult to understand: they are very complex. This is a fundamental trade-off. These algorithms are typically more accurate for predicting nonlinear, faint, or rare phenomena. Unfortunately, more accuracy almost always comes at the expense of interpretability, and interpretability is crucial for business adoption, model documentation, regulatory oversight, and human acceptance and trust.

The inherent trade-off between accuracy and interpretability in predictive modeling can be a particularly vexing catch-22 for analysts and data scientists working in regulated industries. Due to strenuous regulatory and documentation requirements, data science professionals in the regulated verticals of banking, insurance, healthcare, and other industries often feel locked into using traditional, linear modeling techniques to create their predictive models. So, how can you use machine learning to improve the accuracy of your predictive

models and increase the value they provide to your organization while still retaining some degree of interpretability?

This report provides some answers to this question by introducing interpretable machine learning techniques, algorithms, and models. It discusses predictive modeling and machine learning from an applied perspective and puts forward social and commercial motivations for interpretability, fairness, accountability, and transparency in machine learning. It defines interpretability, examines some of the major theoretical difficulties in the burgeoning field, and provides a taxonomy for classifying and describing interpretable machine learning techniques. We then discuss many credible and practical machine learning interpretability techniques, consider testing of these interpretability techniques themselves, and, finally, we present a set of open source code examples for interpretability techniques.

Machine Learning and Predictive Modeling in Practice

Companies and organizations use machine learning and predictive models for a very wide variety of revenue- or value-generating applications. A tiny sample of such applications includes deciding whether to award someone a credit card or loan, deciding whether to release someone from a hospital, or generating custom recommendations for new products or services. Although many principles of applied machine learning are shared across industries, the practice of machine learning at banks, insurance companies, healthcare providers and in other regulated industries is often quite different from machine learning as conceptualized in popular blogs, the news and technology media, and academia. It's also somewhat different from the practice of machine learning in the technologically advanced and generally unregulated digital, ecommerce, FinTech, and internet verticals. Teaching and research in machine learning tend to put a central focus on algorithms, and the computer science, mathematics, and statistics of learning from data. Personal blogs and media outlets also tend to focus on algorithms and often with more hype and less rigor than in academia. In commercial practice, talent acquisition, data engineering, data security, hardened deployment of machine learning apps and systems, managing and monitoring an ever-increasing number of predictive models, modeling process

documentation, and regulatory compliance often take precedence over more academic concerns regarding machine learning algorithms[1].

Successful entities in both traditional enterprise and in digital, ecommerce, FinTech, and internet verticals have developed processes for recruiting and retaining analytical talent, amassed vast amounts of data, and engineered massive flows of data through corporate IT systems. Both types of entities have faced data security challenges; both have learned to deploy the complex logic that defines machine learning models into operational, public-facing IT systems; and both are learning to manage the large number of predictive and machine learning models required to stay competitive in today's data-driven commercial landscape. However, larger, more established companies tend to practice statistics, analytics, and data mining at the margins of their business to optimize revenue or allocation of other valuable assets. Digital, ecommerce, FinTech, and internet companies, operating outside of most regulatory oversight, and often with direct access to huge data stores and world-class talent pools, have often made web-based data and machine learning products central to their business.

In the context of applied machine learning, more regulated, and often more traditional, companies tend to face a unique challenge. They must use techniques, algorithms, and models that are simple and transparent enough to allow for detailed documentation of internal system mechanisms and in-depth analysis by government regulators. Interpretable, fair, and transparent models are a serious legal mandate in banking, insurance, healthcare, and other industries. Some of the major regulatory statutes currently governing these industries include the Civil Rights Acts of 1964 and 1991, the Americans with Disabilities Act, the Genetic Information Nondiscrimination Act, the Health Insurance Portability and Accountability Act, the Equal Credit Opportunity Act, the Fair Credit Reporting Act, the Fair Housing Act, Federal Reserve SR 11-7, and European Union (EU) Greater Data Privacy Regulation (GDPR) Article 22[2]. Moreover, regulatory regimes are continuously changing, and these regulatory regimes are key drivers of what constitutes interpretability in applied machine learning.

Social and Commercial Motivations for Machine Learning Interpretability

> *The now-contemplated field of data science amounts to a superset of the fields of statistics and machine learning, which adds some technology for "scaling up" to "big data." This chosen superset is motivated by commercial rather than intellectual developments. Choosing in this way is likely to miss out on the really important intellectual event of the next 50 years.*
>
> —David Donoho[3]

Usage of AI and machine learning models is likely to become more commonplace as larger swaths of the economy embrace automation and data-driven decision making. Even though these predictive systems can be quite accurate, they have been treated as inscrutable black boxes in the past, that produce only numeric or categorical predictions with no accompanying explanations. Unfortunately, recent studies and recent events have drawn attention to mathematical and sociological flaws in prominent machine learning systems, but practitioners usually don't have the appropriate tools to pry open machine learning black boxes to debug and troubleshoot them[4][5].

Although this report focuses mainly on the commercial aspects of interpretable machine learning, it is always crucially important to consider social motivations and impacts of data science, including interpretability, fairness, accountability, and transparency in machine learning. One of the greatest hopes for data science and machine learning is simply increased convenience, automation, and organization in our day-to-day lives. Even today, I am beginning to see fully automated baggage scanners at airports and my phone is constantly recommending new music that I actually like. As these types of automation and conveniences grow more common, machine learning engineers will need more and better tools to debug these ever-more present, decision-making systems. As machine learning begins to make a larger impact on everyday human life, whether it's just additional convenience or assisting in serious, impactful, or historically fraught and life-altering decisions, people will likely want to know how these automated decisions are being made. This might be the most fundamental application of machine learning interpretability, and some argue the EU GDPR is

already legislating a "right to explanation" for EU citizens impacted by algorithmic decisions[6].

Machine learning also promises quick, accurate, and unbiased decision making in life-changing scenarios. Computers can theoretically use machine learning to make objective, data-driven decisions in critical situations like criminal convictions, medical diagnoses, and college admissions, but interpretability, among other technological advances, is needed to guarantee the promises of correctness and objectivity. Without interpretability, accountability, and transparency in machine learning decisions, there is no certainty that a machine learning system is not simply relearning and reapplying long-held, regrettable, and erroneous human biases. Nor are there any assurances that human operators have not designed a machine learning system to make intentionally prejudicial decisions.

Hacking and adversarial attacks on machine learning systems are also a serious concern. Without real insight into a complex machine learning system's operational mechanisms, it can be very difficult to determine whether its outputs have been altered by malicious hacking or whether its inputs can be changed to create unwanted or unpredictable decisions. Researchers recently discovered that slight changes, such as applying stickers, can prevent machine learning systems from recognizing street signs[7]. Such adversarial attacks, which require almost no software engineering expertise, can obviously have severe consequences.

For traditional and often more-regulated commercial applications, machine learning can enhance established analytical practices (typically by increasing prediction accuracy over conventional but highly interpretable linear models) or it can enable the incorporation of unstructured data into analytical pursuits. In many industries, linear models have long been the preferred tools for predictive modeling, and many practitioners and decision-makers are simply suspicious of machine learning. If nonlinear models—generated by training machine learning algorithms—make more accurate predictions on previously unseen data, this typically translates into improved financial margins but only if the model is accepted by internal validation teams and business partners and approved by external regulators. Interpretability can increase transparency and trust in complex machine learning models, and it can allow more sophisticated and potentially more accurate nonlinear models to be used in place of traditional linear models, even in some regulated dealings. Equifax's

NeuroDecision is a great example of modifying a machine learning technique (an ANN) to be interpretable and using it to make measurably more accurate predictions than a linear model in a regulated application. To make automated credit-lending decisions, NeuroDecision uses ANNs with simple constraints, which are somewhat more accurate than conventional regression models and also produce the regulator-mandated reason codes that explain the logic behind a credit-lending decision. NeuroDecision's increased accuracy could lead to credit lending in a broader portion of the market, such as new-to-credit consumers, than previously possible[1][8].

Less-traditional and typically less-regulated companies currently face a greatly reduced burden when it comes to creating fair, accountable, and transparent machine learning systems. For these companies, interpretability is often an important but secondary concern. Even though transparency into complex data and machine learning products might be necessary for internal debugging, validation, or business adoption purposes, the world has been using Google's search engine and Netflix's movie recommendations for years without widespread demands to know why or how these machine learning systems generate their results. However, as the apps and systems that digital, ecommerce, FinTech, and internet companies create (often based on machine learning) continue to change from occasional conveniences or novelties into day-to-day necessities, consumer and public demand for interpretability, fairness, accountability, and transparency in these products will likely increase.

The Multiplicity of Good Models and Model Locality

If machine learning can lead to more accurate models and eventually financial gains, why isn't everyone using *interpretable* machine learning? Simple answer: it's fundamentally difficult and it's a very new field of research. One of the most difficult mathematical problems in interpretable machine learning goes by several names. In his seminal 2001 paper, Professor Leo Breiman of UC, Berkeley, coined the phrase: *the multiplicity of good models*[9]. Some in credit scoring refer to this phenomenon as *model locality*. It is well understood that for the same set of input variables and prediction targets, complex machine learning algorithms can produce multiple accurate models with very similar, but not the same, internal architectures. 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. Because of this systematic instability, multiple interpretability techniques should be used to derive explanations for a single model, and practitioners are urged to seek consistent results across multiple modeling and interpretation techniques.

Figures 1-1 and 1-2 are cartoon illustrations of the surfaces defined by error functions for two fictitious predictive models. In Figure 1-1 the error function is representative of a traditional linear model's error function. The surface created by the error function in Figure 1-1 is convex. It has a clear global minimum in three dimensions, meaning that given two input variables, such as a customer's income and a customer's interest rate, the most accurate model trained to predict loan defaults (or any other outcome) would almost always give the same weight to each input in the prediction, and the location of the minimum of the error function and the weights for the inputs would be unlikely to change very much if the model was retrained, even if the input data about customer's income and interest rate changed a little bit. (The actual numeric values for the weights could be ascertained by tracing a straight line from minimum of the error function pictured in Figure 1-1 to the interest rate axis [the X axis] and income axis [the Y axis].)

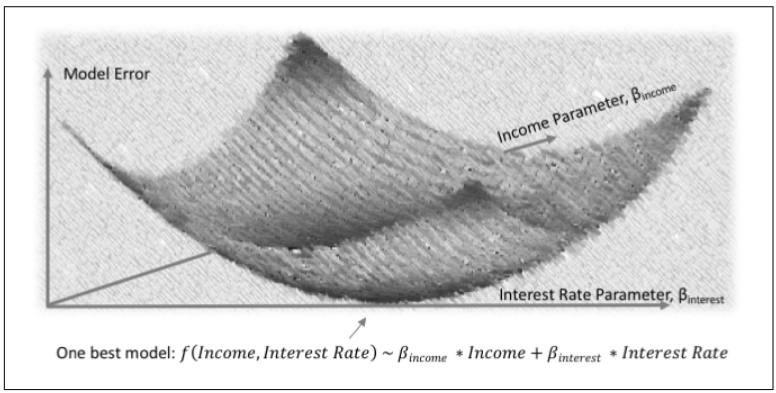


Figure 1-1. An illustration of the error surface of a traditional linear model. (Figure courtesy of H2O.ai.)

Because of the convex nature of the error surface for linear models, there is basically only one *best* model, given some relatively stable

set of inputs and a prediction target. The model associated with the error surface displayed in Figure 1-1 would be said to have strong model locality. Moreover, because the weighting of income versus interest rate is highly stable in the pictured error function and its associated linear model, explanations about how the function made decisions about loan defaults based on those two inputs would also be stable. More stable explanations are often considered more trustworthy explanations.

Figure 1-2 depicts a nonconvex error surface that is representative of the error function for a machine learning function with two inputs—for example, a customer's income and a customer's interest rate—and an output, such as the same customer's probability of defaulting on a loan. This nonconvex error surface with no obvious global minimum implies there are many different ways a complex machine learning algorithm could learn to weigh a customer's income and a customer's interest rate to make a good decision about when they might default. Each of these different weightings would create a different function for making loan default decisions, and each of these different functions would have different explanations. Less-stable explanations feel less trustworthy, but are less-stable explanations actually valuable and useful? The answer to this question is central to the value proposition of interpretable machine learning and is examined in the next section.

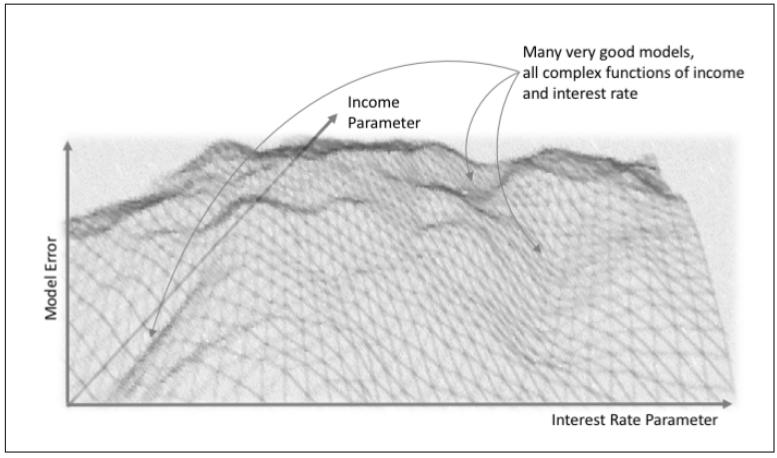


Figure 1-2. An illustration of the error surface of a machine learning model. (Figure courtesy of H2O.ai.)

Accurate Models with Approximate Explanations

Due to many valid concerns, including the multiplicity of good models, many researchers and practitioners deemed the complex, intricate formulas created by training machine learning algorithms to be uninterpretable for many years. Although great advances have been made in recent years to make these often nonlinear, nonmonotonic, and noncontinuous machine-learned response functions more understandable[10][11], it is likely that such functions will never be as directly or universally interpretable as more traditional linear models.

Why consider machine learning approaches for inferential or explanatory 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 create very exact interpretations for approximate models (see Figure 1-3).

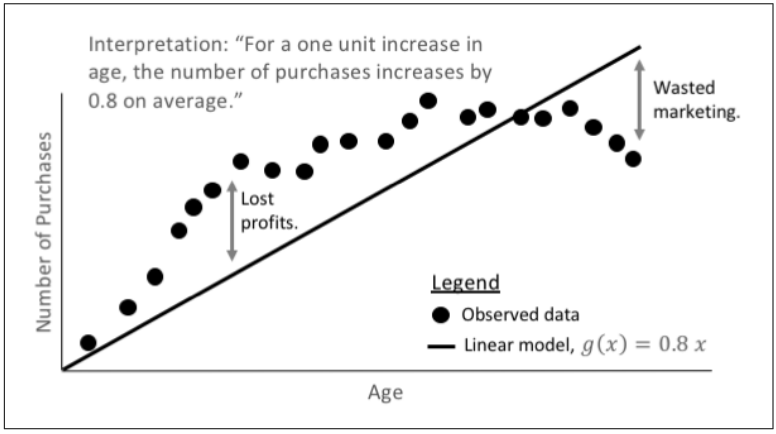


Figure 1-3. A linear model, g(x), predicts the average number of purchases, given a customer's age. The predictions can be inaccurate but the explanations are straightforward and stable. (Figure courtesy of H2O.ai.)

Whereas linear models account for global, average phenomena in a dataset, machine learning models attempt to learn about the local and nonlinear characteristics of a dataset and also tend to be evalu-

ated in terms of predictive accuracy. The machine learning interpretability approach seeks to make approximate interpretations for these types of more exact models. After an accurate predictive model has been trained, it should then be examined from many different viewpoints, including its ability to generate approximate explanations. As illustrated in Figure 1-4, it is possible that an approximate interpretation of a more exact model can have as much, or more, value and meaning than the exact interpretations provided by an approximate model.

Additionally, the use of machine learning techniques for inferential or predictive purposes shouldn't prevent us from using linear models for interpretation. In fact, using local linear approximations of more complex machine-learned functions to derive explanations, as depicted in Figure 1-4, is one of the most popular current approaches. This technique has become known as local interpretable model-agnostic explanations (LIME), and several free and open source implementations of LIME are available for practitioners to evaluate[12].

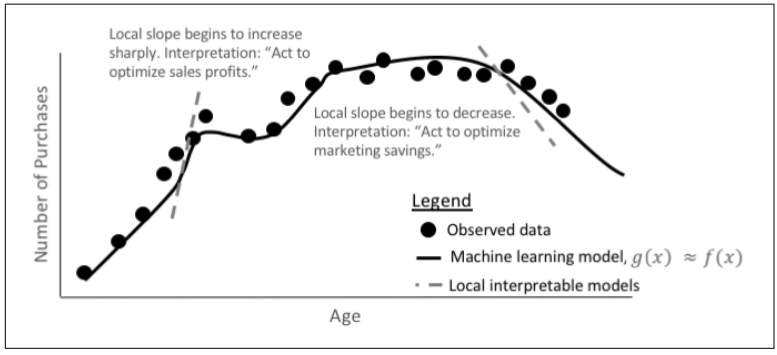


Figure 1-4. A machine learning model, g(x), predicts the number of purchases, given a customer's age, very accurately, nearly replicating the true, unknown signal-generating function, f(x). Although the explanations for this function are approximate, they are at least as useful, if not more so, than the linear model explanations in Figure 1-3. (Figure courtesy of H2O.ai.)

Defining Interpretability

Let's take a step back now and offer a definition of *interpretability*, and also briefly introduce those groups at the forefront of machine learning interpretability research today. 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." [13]. The latter might be the simplest definition of machine learning interpretability, but there are several communities with different and sophisticated notions of what interpretability is today and should be in the future. Two of the most prominent groups pursuing interpretability research are a group of academics operating under the acronym FAT* and civilian and military researchers funded by the Defense Advanced Research Projects Agency (DARPA). FAT* academics (meaning *fairness*, *accountability*, *and transparency* in multiple artificial intelligence, machine learning, computer science, legal, social science, and policy applications) are primarily focused on promoting and enabling interpretability and fairness in algorithmic decision-making systems with social and commercial impact. DARPA-funded researchers seem primarily interested in increasing interpretability in sophisticated pattern recognition models needed for security applications. They tend to label their work *explainable AI*, or XAI.

A Machine Learning Interpretability Taxonomy for Applied Practitioners

Technical challenges as well as the needs and perspectives of different user communities make machine learning interpretability a subjective and complicated subject. Luckily, a previously defined taxonomy has proven useful for characterizing the interpretability of various popular explanatory techniques used in commercial data mining, analytics, data science, and machine learning applications[10]. The taxonomy describes models in terms of their complexity, and categorizes interpretability techniques by the global or local scope of explanations they generate, the family of algorithms to which they can be applied, and their ability to promote trust and understanding.

A Scale for Interpretability

The complexity of a machine learning model is directly related to its interpretability. Generally, the more complex the model, the more difficult it is to interpret and explain. The number of weights or rules in a model—or its *Vapnik–Chervonenkis dimension*, a more formal measure—are good ways to quantify a model's complexity.

However, analyzing the functional form of a model is particularly useful for commercial applications such as credit scoring. The following list describes the functional forms of models and discusses their degree of interpretability in various use cases.

High interpretability—linear, monotonic functions
: Functions created by traditional regression algorithms are probably the most interpretable class of models. We refer to these models here as "linear and monotonic," meaning that for a change in any given input variable (or sometimes combination or function of an input variable), the output of the response function changes at a defined rate, in only one direction, and at a magnitude represented by a readily available coefficient. Monotonicity also enables intuitive and even automatic reasoning about predictions. For instance, if a credit lender rejects your credit card application, it can easily tell you why because its probability-of-default model often assumes your credit score, your account balances, and the length of your credit history are monotonically related to your ability to pay your credit card bill. When these explanations are created automatically, they are typically called *reason codes*. Linear, monotonic functions play another important role in machine learning interpretability. Besides being highly interpretable themselves, linear and monotonic functions are also used in explanatory techniques, including the popular LIME approach.

Medium interpretability—nonlinear, monotonic functions
: Although most machine-learned response functions are nonlinear, some can be constrained to be monotonic with respect to any given independent variable. Although there is no single coefficient that represents the change in the response function output induced by a change in a single input variable, nonlinear and monotonic functions do always change in one direction as a single input variable changes. Nonlinear, monotonic response functions usually allow for the generation of both reason codes and relative variable importance measures. Nonlinear, monotonic response functions are therefore interpretable and potentially suitable for use in regulated applications.

  Of course, there are linear, nonmonotonic machine-learned response functions that can, for instance, be created by the multivariate adaptive regression splines (MARS) approach. We do not highlight these functions here. They tend to be less accurate

predictors than purely nonlinear, nonmonotonic functions and less directly interpretable than their completely monotonic counterparts.

Low interpretability—nonlinear, nonmonotonic functions
: Most machine learning algorithms create nonlinear, nonmonotonic response functions. This class of functions is the most difficult to interpret, as they can change in a positive and negative direction and at a varying rate for any change in an input variable. Typically, the only standard interpretability measures these functions provide are relative variable importance measures. You should use a combination of several techniques, which we present in the sections that follow, to interpret these extremely complex models.

Global and Local Interpretability

It's often important to understand the entire model that you've trained on a global scale, and also to zoom into local regions of your data or your predictions and derive local explanations. Global interpretations help us understand the inputs and their entire modeled relationship with the prediction target, but global interpretations can be highly approximate in some cases. Local interpretations help us understand model predictions for a single row of data or a group of similar rows. Because small sections of a machine-learned response function are more likely to be linear, monotonic, or otherwise well-behaved, local explanations can be more accurate than global explanations. It's also very likely that the best explanations of a machine learning model will come from combining the results of global and local interpretation techniques. In subsequent sections we will use the following descriptors to classify the scope of an interpretable machine learning approach:

Global interpretability
: Some machine learning interpretability techniques facilitate global explanations of machine learning algorithms, their results, or the machine-learned relationship between the prediction target and the input variables.

Local interpretability
: Local interpretations promote understanding of small regions of the machine-learned relationship between the prediction target and the input variables, such as clusters of input records and

their corresponding predictions, or deciles of predictions and their corresponding input rows, or even single rows of data.

Model-Agnostic and Model-Specific Interpretability

Another important way to classify model interpretability techniques is whether they are *model agnostic*, meaning they can be applied to different types of machine learning algorithms, or *model specific*, meaning techniques that are applicable only for a single type or class of algorithm. For instance, the LIME technique is model agnostic and can be used to interpret nearly any set of machine learning inputs and machine learning predictions. On the other hand, the technique known as *treeinterpreter* is model specific and can be applied only to decision tree models. Although model-agnostic interpretability techniques are convenient, and in some ways ideal, they often rely on surrogate models or other approximations that can degrade the accuracy of the explanations they provide. Model-specific interpretation techniques tend to use the model to be interpreted directly, leading to potentially more accurate explanations.

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 some users, technical descriptions of algorithms in textbooks and journals provide enough insight to fully understand machine learning models. For these users, cross-validation, error measures, and assessment plots probably also provide enough information to trust a model. Unfortunately, for many applied practitioners, the usual definitions and assessments don't often inspire full trust and understanding in machine learning models and their results.

Trust and understanding are different phenomena, and both are important. The techniques presented in the next section go beyond standard assessment and diagnostic practices to engender greater understanding and trust in complex models. These techniques enhance understanding by either providing transparency and specific insights into the mechanisms of the algorithms and the functions they create or by providing detailed information and accountability for the answers they provide. The techniques that fol-

low enhance trust by enabling users to observe or ensure the fairness, stability, and dependability of machine learning algorithms, the functions they create, and the answers they generate.

Common Interpretability Techniques

Many credible techniques for training interpretable models and gaining insights into model behavior and mechanisms have existed for years. Many others have been put forward in a recent flurry of research. This section of the report discusses many such interpretability techniques in terms of the proposed machine learning interpretability taxonomy. The section begins by discussing data visualization approaches because having a strong understanding of a dataset is a first step toward validating, explaining, and trusting models. We then present white-box modeling techniques, or models with directly transparent inner workings, followed by techniques that can generate explanations for the most complex types of predictive models such as model visualizations, reason codes, and variable importance measures. We conclude the section by discussing approaches for testing machine learning models for fairness, stability, and trustworthiness.

Seeing and Understanding Your Data

Seeing and understanding data is important for interpretable machine learning because models represent data, and understanding the contents of that data helps set reasonable expectations for model behavior and output. Unfortunately, most real datasets are difficult to see and understand because they have many variables and many rows. Even though plotting many dimensions is technically possible, doing so often detracts from, instead of enhances, human understanding of complex datasets. Of course, there are many, many ways to visualize datasets. We chose the techniques highlighted in Tables 1-1 and 1-2 and in Figure 1-5 because they help illustrate many important aspects of a dataset in just two dimensions.

Table 1-1. A description of 2-D projection data visualization approaches

Technique: 2-D projections
Description: Projecting rows of a dataset from a usually high-dimensional original space into a more visually understandable lower-dimensional space, ideally two or three dimensions. Techniques to achieve this include Principal Components Analysis (PCA), Multidimensional Scaling (MDS), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Autoencoder Networks.

Suggested usage: The key idea is to represent the rows of a dataset in a meaningful low-dimensional space. Datasets containing images, text, or even business data with many variables can be difficult to visualize as a whole. These projection techniques enable high-dimensional datasets to be projected into representative low-dimensional spaces and visualized using the trusty old scatter plot technique. A high-quality projection visualized in a scatter plot should exhibit key structural elements of a dataset, such as clusters, hierarchy, sparsity, and outliers. 2-D projections are often used in fraud or anomaly detection to find outlying entities, like people, transactions, or computers, or unusual clusters of entities.
References: *Visualizing Data using t-SNE* MDS, Cox, T.F., Cox, M.A.A. *Multidimensional Scaling*. Chapman and Hall. 2001. *The Elements of Statistical Learning* *Reducing the Dimensionality of Data with Neural Networks*
OSS: h2o.ai R (various packages) scikit-learn (various functions)
Global or local scope: Global and local. You can use most forms of visualizations to see a courser view of the entire dataset, or they can provide granular views of local portions of the dataset. Ideally, advanced visualization tool kits enable users to pan, zoom, and drill-down easily. Otherwise, users can plot different parts of the dataset at different scales themselves.
Best-suited complexity: 2-D projections can help us to understand very complex relationships in datasets. / **Model specific or model agnostic:** Model agnostic; visualizing complex datasets with many variables.
Trust and understanding: Projections add a degree of trust if they are used to confirm machine learning modeling results. For instance, if known hierarchies, classes, or clusters exist in training or test datasets and these structures are visible in 2-D projections, it is possible to confirm that a machine learning model is labeling these structures correctly. A secondary check is to confirm that similar attributes of structures are projected relatively near one another and different attributes of structures are projected relatively far from one another. Consider a model used to classify or cluster marketing segments. It is reasonable to expect a machine learning model to label older, richer customers differently than younger, less affluent customers, and moreover to expect that these different groups should be relatively disjointed and compact in a projection, and relatively far from one another.

Table 1-2. A description of the correlation graph data visualization approach

Technique: Correlation graphs

Description: A correlation graph is a two-dimensional representation of the relationships (correlation) in a dataset. The authors create correlation graphs in which the nodes of the graph are the variables in a dataset and the edge weights (thickness) between the nodes are defined by the absolute values of their pairwise Pearson correlation. For visual simplicity, absolute weights below a certain threshold are not displayed, the node size is determined by a node's number of connections (node degree), node color is determined by a graph community calculation, and node position is defined by a graph force field algorithm. The correlation graph allows us to see groups of correlated variables, identify irrelevant variables, and discover or verify important relationships that machine learning models should incorporate, all in two dimensions.

Suggested usage: Correlation graphs are a very powerful tool for seeing and understanding relationships (correlation) between variables in a dataset. They are especially powerful in text mining or topic modeling to see the relationships between entities and ideas. Traditional network graphs—a similar approach—are also popular for finding relationships between customers or products in transactional data and for use in fraud detection to find unusual interactions between entities like people or computers.

OSS:
Gephi
https://github.com/jphall663/corr_graph

Global or local scope: Global and local. You can use most forms of visualizations to see a courser view of the entire dataset, or they can provide granular views of local portions of the dataset. Ideally, advanced visualization tool kits enable users to pan, zoom, and drill-down easily. Otherwise, users can plot different parts of the dataset at different scales themselves.

Best-suited complexity: Correlation graphs can help us understand complex relationships but can become difficult to understand with more than several thousand variables.	**Model specific or model agnostic:** Model agnostic; visualizing complex datasets with many variables.

Trust and understanding: Correlation graphs promote understanding by displaying important and complex relationships in a dataset. They can enhance trust in a model if variables with thick connections to the target are important variables in the model, and we would expect a model to learn that unconnected variables are not very important. Also, common sense relationships displayed in the correlation graph should be reflected in a trustworthy model.

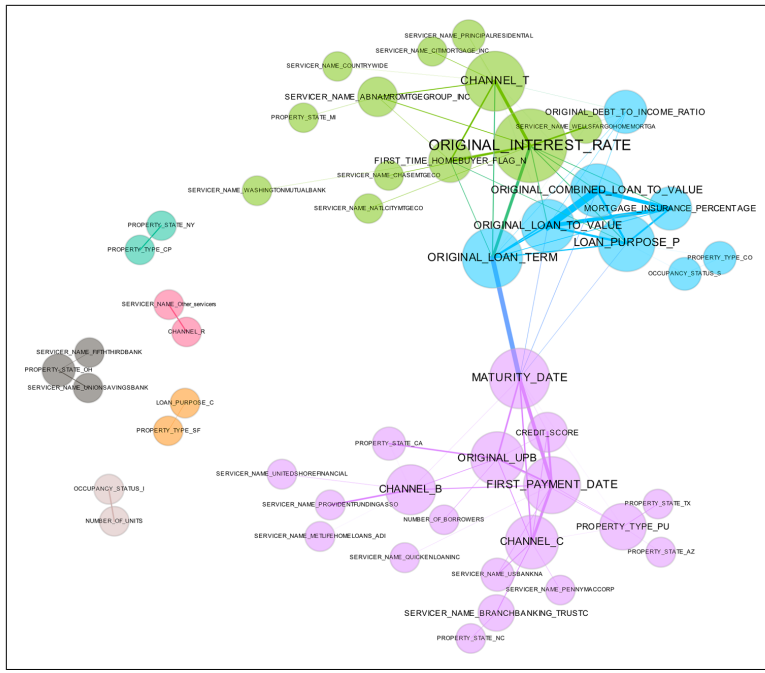


Figure 1-5. A data visualization, a correlation graph, that is helpful for enhancing trust and understanding in machine learning models because it displays important, complex relationships between variables in a dataset as edges and nodes in an undirected graph. (Figure courtesy of H2O.ai.)

Techniques for Creating White-Box Models

When starting a machine learning endeavor, it's a best practice to determine to what degree interpretability is an important criterion for success. If interpretability is of paramount importance to the project's outcome, it's best to use an interpretable modeling technique from the beginning. The techniques described in Tables 1-3 through 1-8 will enable you to create highly transparent models, well-suited for regulated industry or other applications in which interpretability is of extreme importance.

Table 1-3. A description of the decision tree white-box modeling approach

Technique: Decision trees	
Description: Decision trees create a model that predicts the value of a target variable based on several input variables. Decision trees are directed graphs in which each interior node corresponds to an input variable; there are edges to child nodes for values of that input variable that create the highest target purity in each child. Each terminal node or leaf node represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf. These paths can be visualized or explained with simple if-then rules.	
Suggested usage: Decision trees are great for training simple, transparent models on I.I.D. data —data where a unique customer, patient, product, or other entity is represented in each row. They are beneficial when the goal is to understand relationships between the input and target variable with "Boolean-like" logic. Decision trees can also be displayed graphically in a way that is easy for non-experts to interpret.	
References: Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984. *The Elements of Statistical Learning*	
OSS: rpart scikit-learn (various functions)	
Global or local scope: Global.	
Best-suited complexity: Decision trees can create very complex nonlinear, nonmonotonic functions. For best interpretability, restrict to shallow depth and binary splits. Predictions can also be restricted to be monotonic with respect to input variable values.	**Model specific or model agnostic:** Model specific; interpretability is a key motivating factor for using decision-tree models.
Trust and understanding: Increases trust and understanding because input to target mappings follow a decision structure that can be easily visualized and interpreted and compared to domain knowledge and reasonable expectations.	

Table 1-4. A description of the XNN modeling approach

Technique: eXplainable Neural Networks	
Description: Often considered the least transparent of black-box models, recent work in XNN implementation and explaining artificial neural network (ANN) predictions may render that notion of ANNs obsolete. Many of the breakthroughs in ANN explanation stem from the straightforward calculation of derivatives of the trained ANN response function with regard to input variables made possible by the proliferation of deep learning toolkits such as Tensorflow. These derivatives allow for the disaggregation of the trained ANN response function prediction into input variable contributions for any observation.	
Suggested usage: Explaining ANN predictions is impactful for at least two major reasons. While most users will be familiar with the widespread use of ANNs in pattern recognition, they are also used for more traditional data mining applications such as fraud detection, and even for regulated applications such as credit scoring. Moreover, ANNs can now be used as accurate and explainable surrogate models, potentially increasing the fidelity of both global and local surrogate model techniques.	
References: Ancona, M., E. Ceolini, C. Öztireli, and M. Gross. *Towards Better Understanding of Gradient-based Attribution Methods for Deep Neural Networks*, ICLR 2018. Vaughan, Joel, et al. *Explainable Neural Networks Based on Additive Index Models* .	
OSS: Skater (integrated gradients and layerwise relevance propagation) DeepLift	
Global or local scope: Typically local but can be both.	
Best-suited complexity: XNNs can be used to directly model extremely nonlinear, non-monotonic phenomena or they can be used as surrogate models to explain other nonlinear, non-monotonic models.	**Model specific or model agnostic:** As directly interpretable models, XNNs rely on model specific mechanisms. Used as surrogate models, XNNs are model agnostic.
Trust and understanding: XNN techniques are typically used to make ANN models themselves more understandable or as surrogate models to make other nonlinear models more understandable.	

Table 1-5. A description of the monotonic GBM white-box modeling approach

Technique: Monotonic gradient-boosted machines (GBMs)	
Description: Monotonicity constraints can turn difficult-to-interpret nonlinear, nonmonotonic models into highly interpretable, and possibly regulator-approved, nonlinear, monotonic models. One application of this can be achieved with monotonicity constraints in GBMs by enforcing a uniform splitting strategy in constituent decision trees, where binary splits of a variable in one direction always increase the average value of the dependent variable in the resultant child node, and binary splits of the variable in the other direction always decrease the average value of the dependent variable in the other resultant child node.	
Suggested usage: Potentially appropriate for most traditional data mining and predictive modeling tasks, even in regulated industries (no matter what a training data sample says, regulators might still want to see monotonic behavior) and for consistent reason code generation (consistent reason code generation is generally considered a gold standard of model interpretability).	
Reference: XGBoost Documentation	
OSS: XGBoost *http://bit.ly/2IlRhh5* *https://github.com/h2oai/mli-resources/blob/master/notebooks/mono_xgboost.ipynb*	
Global or local scope: Monotonicity constraints create globally interpretable response functions.	**Best-suited complexity:** Monotonic GBM's create nonlinear, monotonic response functions.
Model specific or model agnostic: As implementations of monotonicity constraints vary for different types of models in practice, they are a model-specific interpretation technique.	**Trust and understanding:** Understanding is increased by enforcing straightforward relationships between input variables and the prediction target. Trust is increased when monotonic relationships, reason codes, and detected interactions are parsimonious with domain expertise or reasonable expectations.

Table 1-6. A description of alternative regression white-box modeling approaches

Technique: Logistic, elastic net, GAM, and quantile regression
Description: These techniques use contemporary methods to augment traditional, linear modeling methods. Linear model interpretation techniques are highly sophisticated, typically model specific, and the inferential features and capabilities of linear models are rarely found in other classes of models. These types of models usually produce linear, monotonic response functions with globally interpretable results like those of traditional linear models but often with a boost in predictive accuracy.
Suggested usage: Interpretability for regulated industries; these techniques are meant for practitioners who just can't use complex machine learning algorithms to build predictive models because of interpretability concerns or who seek the most interpretable possible modeling results.

References: *The Elements of Statistical Learning* Koenker, R. *Quantile regression (No. 38)*. Cambridge University Press, 2005.
OSS: gam glmnet h2o.ai quantreg scikit-learn (various functions)

Global or local scope: Alternative regression techniques often produce globally interpretable linear, monotonic functions that can be interpreted using coefficient values or other traditional regression measures and statistics.	**Best-suited complexity:** Alternative regression functions are generally linear, monotonic functions. However, GAM approaches can create quite complex nonlinear functions.

Model specific or model agnostic: Model specific.
Trust and understanding: The lessened assumption burden, the ability to select variables without potentially problematic multiple statistical significance tests, the ability to incorporate important but correlated predictors, the ability to fit nonlinear phenomena, or the ability to fit different quantiles of the data's conditional distribution (and not just the mean of the conditional distribution) could lead to more accurate understanding of modeled phenomena. Basically, these techniques are trusted linear models but used in new, different, and typically more robust ways.

Table 1-7. A description of rule-based white-box modeling approaches

Technique: Rule-based models
Description: A rule-based model is a type of model that is composed of many simple Boolean statements that can be built by using expert knowledge or learning from real data.
Suggested usage: Useful in predictive modeling and fraud and anomaly detection when interpretability is a priority and simple explanations for relationships between inputs and targets are desired, but a linear model is not necessary. Often used in transactional data to find simple, frequently occurring pairs or triplets of products in purchases.
Reference: *An Introduction to Data Mining*, Chapter 6
OSS: RuleFit arules FP-growth Scalable Bayesian Rule Lists

Global or local scope: Rule-based models can be both globally and locally interpretable.	**Best-suited complexity:** Most rule-based models are easy to follow for users because they obey Boolean logic ("if, then"), but they can model extremely complex nonlinear, nonmonotonic phenomena.

Model specific or model agnostic: Model specific; can be highly interpretable if rules are restricted to simple combinations of input variable values.	**Trust and understanding:** Rule-based models increase understanding by creating straightforward, Boolean rules that can be understood easily by users. Rule-based models increase trust when the generated rules match domain knowledge or reasonable expectations.

Table 1-8. A description of SLIM white-box modeling approaches

Technique: Supersparse Linear Integer Models (SLIMs)	
Description: SLIMs create predictive models that require users to only add, subtract, or multiply values associated with a handful of input variables to generate accurate predictions.	
Suggested usage: SLIMs are perfect for serious situations in which interpretability and simplicity are critical, similar to diagnosing newborn infant health using the well-known Agpar scale.	
Reference: *Supersparse Linear Integer Models for Optimized Medical Scoring Systems*	
Software: slim-python	
Global or local scope: SLIMs are globally interpretable.	**Best-suited complexity:** SLIMs are simple, linear models.
Model specific or model agnostic: Model specific; interpretability for SLIMs is intrinsically linked to their linear nature and several model-specific optimization routines.	**Trust and understanding:** SLIMs enhance understanding by breaking complex scenarios into simple rules for handling system inputs. They increase trust when their predictions are accurate and their rules reflect human domain knowledge or reasonable expectations.

Techniques for Enhancing Interpretability in Complex Machine Learning Models

In some machine learning projects, accuracy is more important than interpretability, but some level of transparency is still desirable. In other projects, dirty or unstructured input data rules out the use of highly interpretable classical regression models, even if explainability is a necessary outcome of the project. The techniques described here are meant to be used in these situations or other scenarios in which explanations must be extracted from complex, nonlinear, black-box models or decisioning systems. Many of these techniques can also be used on more transparent white-box models to further enhance interpretability.

Seeing model mechanisms with model visualizations

Model visualization techniques can provide graphical insights into the prediction behavior of nearly any black-box model and into the prediction mistakes they might make. A few popular model visualizations, including decision-tree surrogate models, individual conditional expectation plots, partial-dependence plots, and residual plots are presented in Tables 1-9 through 1-12 and in Figures 1-6 and 1-7. Surrogate models are simple models of more complex models, and decision-tree surrogate models (Figure 1-6) create an approximate overall flow chart of a complex model's decision-making processes. Individual conditional expectation (ICE) plots and partial-dependence plots (Figure 1-7) provide a local and global view, respectively, into how a model's predictions change based on certain input variables. Residual analysis provides a mechanism to investigate how black-box models make errors in their predictions while also highlighting anomalous data and outliers that might have undue influence on a model's predictions.

Table 1-9. A description of the decision-tree surrogate model visualization technique

Technique: Decision-tree surrogates
Description: A decision-tree surrogate model is a simple model that is used to explain a complex model. Decision-tree surrogate models are usually created by training a decision tree on the original inputs and predictions of a complex model. Variable importance, trends, and interactions displayed in the surrogate model are then assumed to be indicative of the internal mechanisms of the complex model. There are few, possibly no, theoretical guarantees that the simple surrogate model is highly representative of the more complex model.
Suggested usage: Use decision-tree surrogate models to create approximate flow charts of a more complex model's decision-making processes.
References: *Extracting Tree-Structured Representations of Trained Networks* *Interpreting Blackbox Models via Model Extraction*
OSS: *http://bit.ly/2DL3Ip3* *https://github.com/h2oai/mli-resources/blob/master/notebooks/dt_surrogate.ipynb*
Global or local scope: Generally, decision-tree surrogate models are global. The globally interpretable attributes of a simple model are used to explain global attributes of a more complex model. However, there is nothing to preclude fitting decision-tree surrogate models to more local regions of a complex model's predictions and their corresponding input rows.

Best-suited complexity: Surrogate models can help explain machine learning models of medium to high complexity, including nonlinear, monotonic or nonmonotonic models.	**Model specific or model agnostic:** Model agnostic.
Trust and understanding: Decision-tree surrogate models enhance trust when their variable importance, trends, and interactions are aligned with human domain knowledge and reasonable expectations of modeled phenomena. Decision-tree surrogate models enhance understanding because they provide insight into the internal mechanisms of complex models.	

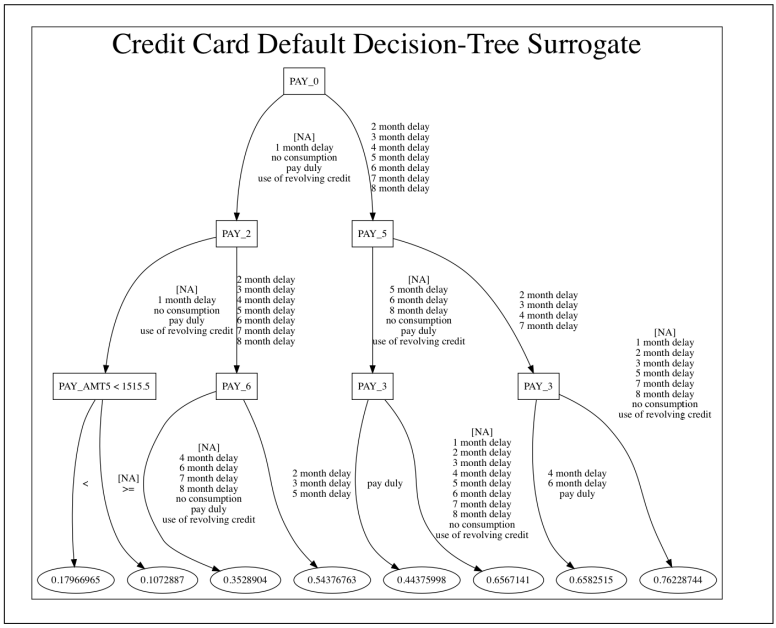


Figure 1-6. A model visualization, a decision-tree surrogate that illustrates an approximate overall flowchart of the decision processes learned by a more complex machine learning model. (Figure courtesy of H2O.ai.)

Table 1-10. A description of the ICE plot model visualization technique

Technique: Individual Conditional Expectation (ICE) plots
Description: ICE plots, a newer and less well-known adaptation of partial-dependence plots, can be used to create local explanations using the same ideas as partial-dependence plots.

Suggested usage: ICE plots depict how a model behaves for a single row of data and can be used to validate monotonicity constraints. ICE pairs nicely with partial dependence in the same plot to provide local information to augment the global information provided by partial dependence. ICE can detect when partial dependence fails in the presence of strong interactions among input variables. Some practitioners feel that ICE can be misleading in the presence of strong correlations between input variables.
Reference: *Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation*
OSS: ICEbox *http://bit.ly/2lIRhh5* *https://github.com/h2oai/mli-resources/blob/master/notebooks/pdp_ice.ipynb*

Global or local scope: ICE plots are local because they apply to one observation at a time.	**Best-suited complexity:** Can be used to describe nearly any function, including nonlinear, nonmonotonic functions.

Model specific or model agnostic: Model agnostic.
Trust and understanding: ICE plots enhance understanding by showing the nonlinearity, nonmonotonicity, and two-way interactions between input variables and a target variable in complex models, per observation. They can also enhance trust when displayed relationships conform to domain knowledge expectations, when the plots remain stable or change in expected ways over time, or when displayed relationships remain stable under minor perturbations of the input data.

Table 1-11. A description of the partial dependence plot model visualization technique

Technique: Partial-dependence plots
Description: Partial-dependence plots show us the average manner in which machine-learned response functions change based on the values of one or two input variables of interest, while averaging out the effects of all other input variables.
Suggested usage: Partial-dependence plots show the nonlinearity, nonmonotonicity, and two-way interactions in very complex models and can be used to verify monotonicity of response functions under monotonicity constraints. They pair nicely with ICE plots, and ICE plots can expose when partial dependence becomes inaccurate in the presence of strong interactions.
Reference: *The Elements of Statistical Learning*
OSS: h2o.ai R (various packages) scikit-learn (various functions) *http://bit.ly/2lIRhh5* *https://github.com/h2oai/mli-resources/blob/master/notebooks/pdp_ice.ipynb*

Global or local scope: Partial-dependence plots are global in terms of the rows of a dataset but local in terms of the input variables.	**Best-suited complexity:** Can be used to describe almost any function, including complex nonlinear, nonmonotonic functions.
Model specific or model agnostic: Model agnostic.	
Trust and understanding: Partial-dependence plots enhance understanding by showing the nonlinearity, nonmonotonicity, and two-way interactions between input variables and a dependent variable in complex models. They can also enhance trust when displayed relationships conform to domain knowledge expectations, when the plots remain stable or change in expected ways over time, or when displayed relationships remain stable under minor perturbations of the input data.	

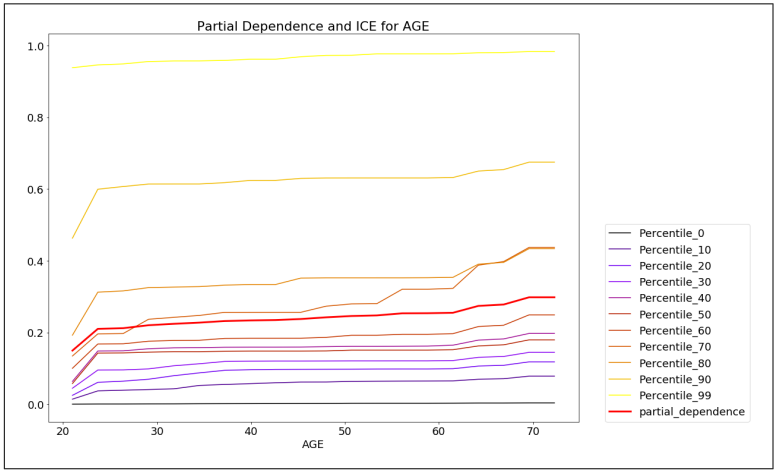


Figure 1-7. A model visualization in which partial dependence is displayed with ICE for the input variable AGE across several percentiles of predicted probabilities to explain and validate the behavior of a monotonic GBM model. (Figure courtesy of H2O.ai.)

Table 1-12. A description of the residual plot model visualization technique

Technique: Residual plots
Suggested usage: Diagnostic for any machine learning model. Plotting the residual values against the predicted values is a time-honored model assessment technique and a great way to find outliers and see all of your modeling results in two dimensions.
Description: Residuals refer to the difference between the actual value of a target variable and the predicted value of a target variable for every row in a data set. Residuals can be plotted in 2-D to analyze predictive models.
OSS: *http://bit.ly/2FQ7X8E*

Global or local scope: Residual analysis can be global in scope when used to assess the goodness-of-fit for a model over an entire dataset. It can be local in scope when used to diagnose how a model treats a single row or small group of rows of data.	**Best-suited complexity:** Can be used to assess machine learning models of varying complexity, including linear, nonlinear and nonmonotonic functions.
Model specific or model agnostic: Model agnostic.	**Trust and understanding:** Residual analysis can promote understanding by guiding users toward problematic predictions and enabling users to debug such problems. It can enhance trust when residuals are appropriately distributed and other fit statistics (i.e., R^2, AUC, etc.) are in the appropriate ranges.

Deriving reason codes for enhanced transparency and accountability

Reason codes (or *turn-down codes*) are plain-text explanations of a model prediction in terms of a model's input variables. The latter phrases come from credit scoring. Credit lenders in the US must provide reasons for automatically rejecting a credit application. If a lender rejects your credit card application, it must tell you why based on the values of input variables to its credit risk models such as your credit score, your account balances, and the length of your credit history.

Reason codes are crucially important for machine learning interpretability in applied settings because they tell practitioners why a model makes a decision in terms of the model's input variables, and they can help practitioners understand if high weight is being given to potentially problematic inputs including gender, age, marital status, or disability status. Of course, generating reason codes for linear models is nothing new to banks, credit bureaus, and other entities. The techniques described in Tables 1-13 through 1-17 are interesting because you can apply them to generate approximate reason codes for potentially more accurate machine learning models.

Like global surrogate models, local surrogate models are simple models of complex models, but they are trained only for certain, interesting rows of data (for instance, the best customers in a dataset or most-likely-to-fail pieces of equipment according to some model's predictions). LIME is a prescribed method for building local linear surrogate models around single observations. Decision trees or other rule-based models can also be used in local regions. Both can shed light on how decisions are made for specific observations, and reason codes can be derived by sorting the contributions of input

variables in these local models. Newer, related, and highly anticipated work from the creators of LIME, called *anchors*, uses rules to explain the local prediction behavior of complex models. Other promising techniques for generating reason codes for machine learning models include treeinterpreter, leave-one-covariate-out (LOCO) local variable importance, and Shapley explanations.

Table 1-13. A description of the anchors local variable importance, or reason code, technique

Technique: Anchors	
Description: A newer approach from the inventors of LIME that generates high-precision sets of plain-language rules to describe a machine learning model prediction in terms of the model's input variable values.	
Suggested usage: Anchors is currently most applicable to classification problems in both traditional data mining and pattern-recognition domains.	
Reference: *Anchors: High-Precision Model-Agnostic Explanations*	
OSS: anchor	
Global or local scope: Local.	**Best-suited complexity:** Anchors can create explanations for very complex functions, but the rule set needed to describe the prediction can become large.
Model specific or model agnostic: Model agnostic.	**Trust and understanding:** Anchor explanations increase understanding by creating explanations for each prediction in a dataset. They enhance trust when the important features for specific records conform to human domain knowledge and reasonable expectations.

Table 1-14. A description of the LOCO local variable importance, or reason code, technique

Technique: Leave-One-Covariate-Out (LOCO) variable importance
Description: A general implementation of LOCO might proceed as follows. LOCO creates local interpretations for each row in a training or unlabeled score set by scoring the row of data once and then again for each input variable (e.g., covariate) in the row. In each additional scoring run, one input variable is set to missing, zero, its mean value, or another appropriate value for leaving it out of the prediction. The input variable with the largest absolute impact on the prediction for that row is taken to be the most important variable for that row's prediction. Variables can also be ranked by their impact on the prediction on a per-row basis.
Suggested usage: You can use LOCO to build reason codes for each row of data on which a complex model makes a prediction. LOCO can deteriorate in accuracy when complex nonlinear dependencies exist in a model. Shapley explanations might be a better technique in this case.
Reference: *Distribution-Free Predictive Inference for Regression*

OSS: conformal *http://bit.ly/2DL3Ip3* *https://github.com/h2oai/mli-resources/blob/master/notebooks/loco.ipynb*	
Global or local scope: Typically local, but LOCO also creates global variable importance measures by estimating the mean change in accuracy for each variable over an entire dataset and can even provide confidence intervals for these global estimates of variable importance.	**Best-suited complexity:** LOCO measures are most useful for nonlinear, nonmonotonic response functions but can be applied to many types of machine-learned response functions.
Model specific or model agnostic: Model agnostic.	
Trust and understanding: LOCO measures increase understanding because they tell us the most influential variables in a model for a particular observation and their relative rank. LOCO measures increase trust if they are in line with human domain knowledge and reasonable expectations. They also increase trust if they remain stable when data is lightly and intentionally perturbed and whether they change in acceptable ways as data changes over time or when pertinent scenarios are simulated.	

Table 1-15. A description of the LIME local variable importance, or reason code, technique

Technique: Local Interpretable Model-Agnostic Explanations (LIME)	
Description: Uses local linear surrogate models to explain regions in a complex machine-learned response function around an observation of interest.	
Suggested usage: Local linear model parameters can be used to describe the average behavior of a complex machine-learned response function around an observation of interest and to construct reason codes. Appropriate for pattern recognition applications, as well. Potentially inappropriate for generating explanations in real time on unseen data.	
Reference: *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*	
OSS: eli5 lime (Python) lime (R) *http://bit.ly/2u4Ychs*	
Best-suited complexity: Suited for response functions of high complexity but can fail in regions of extreme nonlinearity or high-degree interactions.	**Global or local scope:** Local.
Model specific or model agnostic: Model agnostic.	
Trust and understanding: LIME increases transparency by revealing important input features and their linear trends. LIME enhances accountability by creating explanations for each observation in a dataset. LIME bolsters trust and fairness when the important features and their linear trends around specific records conform to human domain knowledge and reasonable expectations.	

Table 1-16. A description of the treeinterpreter local variable importance, or reason code, technique

Technique: Treeinterpreter	
Description: Treeinterpreter decomposes decision tree, random forest, and gradient-boosting machine (GBM) predictions into bias (overall training data average) and component terms for each variable used in a model. Treeinterpreter simply outputs a list of the bias and individual variable contributions globally and for each record.	
Suggested usage: You can use Treeinterpreter to interpret complex tree-based models, and to create reason codes for each prediction. However, local contributions do not sum to the model prediction in some cases and in some implementations, which is an unnerving level of approximation for such a simple technique.	
Reference: *Random forest interpretation with scikit-learn*	
OSS: eli treeinterpreter	
Global or local scope: Treeinterpreter is global in scope when it represents average contributions of input variables to overall model predictions. It is local in scope when used to explain single predictions.	**Best-suited complexity:** Treeinterpreter is meant to explain the usually nonlinear, nonmonotonic response functions created by decision tree, random forest, and GBM algorithms.
Model specific or model agnostic: Treeinterpreter is model specific to algorithms based on decision trees.	
Trust and understanding: Treeinterpreter increases understanding by displaying ranked contributions of input variables to the predictions of decision tree models. Treeinterpreter enhances trust when displayed variable contributions conform to human domain knowledge or reasonable expectations. Treeinterpreter also enhances trust if displayed explanations remain stable when data is subtly and intentionally corrupted and if explanations change in appropriate ways as data changes over time or when interesting scenarios are simulated.	

Table 1-17. A description of the Shapley local variable importance, or reason code, technique

Technique: Shapley explanations
Description: Shapely explanations are a promising newer technique with credible theoretical support that unifies approaches such as LIME, LOCO, and treeinterpreter for deriving consistent local variable contributions to black-box model predictions.
Suggested usage: Shapely explanations are based on accurate, local contributions of input variables and can be rank-ordered to generate reason codes. Shapley explanations have theoretical support, which might make them more suitable for use in regulated industry, but they can be time consuming to calculate, especially outside of XGBoost.
Reference: *A Unified Approach to Interpreting Model Predictions*

OSS: shap XGBoost	
Global or local scope: Shapley explanations are local but can be aggregated to create global explanations.	**Best-suited complexity:** This method applies to any machine learning model, including nonlinear and nonmonotonic models.
Model specific or model agnostic: Can be both. Uses a variant of LIME for model-agnostic explanations. Takes advantage of tree structures for decision tree models.	**Trust and understanding:** Shapely explanations enhance understanding by creating explanations for each observation in a dataset. They bolster trust when the important features for specific records conform to human domain knowledge and reasonable expectations.

Variable importance measures

Variable importance quantifies the global contribution of each input variable to the predictions of a complex machine learning model. For nonlinear, nonmonotonic response functions, variable importance measures are often the only commonly available quantitative measure of the machine-learned relationships between input variables and the prediction target in a model. Variable importance measures rarely give insight into even the average direction that a variable affects a response function. They simply state the magnitude of a variable's relationship with the response as compared to other variables used in the model.

Variable importance measures are typically seen in tree-based models but are sometimes also reported for other models. A simple heuristic rule for variable importance in a decision tree is related to the depth and frequency at which a variable is split on in a tree, where variables used higher in the tree and more frequently in the tree are more important. For artificial neural networks, variable importance measures are typically associated with the aggregated, absolute magnitude of model parameters associated with a given variable of interest.

Table 1-18. A description of global variable importance techniques

Technique: Global variable importance
Suggested usage: Understanding an input variable's global contribution to model predictions. Practitioners should be aware that unsophisticated measures of variable importance can be biased toward larger scale variables or variables with a high number of categories. Global feature importance measures are typically not appropriate for creating reason codes.

References: *Greedy Function Approximation: A Gradient Boosting Machine* *Random Forests*	
OSS: h2o.ai R (various packages) scikit-learn (various functions)	
Global or local scope: Global.	**Best-suited complexity:** Variable importance measures are most useful for nonlinear, nonmonotonic response functions but can be applied to many types of machine-learned response functions.
Model specific or model agnostic: Global variable importance techniques are typically model specific.	
Trust and understanding: Variable importance measures increase understanding because they tell us the most influential variables in a model and their relative rank. Variable importance measures increase trust if they are in line with human domain knowledge and reasonable expectations. They also increase trust if they remain stable when data is lightly and intentionally perturbed, and if they change in acceptable ways as data changes over time or when pertinent scenarios are simulated.	

Fairness

Fairness is yet another important facet of interpretability, and an admirable goal for any machine learning project whose outcome will affect human lives. Traditional checks for fairness, often called disparate impact analysis, typically include assessing model predictions across sensitive demographic segments of ethnicity or gender. Today the study of fairness in machine learning is widening and progressing rapidly, including the development of techniques to remove unfairness, or bias, from model predictions and models that learn to make fair predictions.

Table 1-19. A description of fairness

Technique: Fairness (various techniques)
Description: Fairness means that models treat segments within training data and new unseen data roughly equally in terms of predictions, accuracy, variance, or error.
Suggested usage: Different types of contemporary fairness techniques can detect bias, can correct bias in model predictions, and can learn to make fair predictions.
Reference: Barocas, S., M. Hardt, and A. Narayanan. *Fairness and Machine Learning*.
OSS: AIF360

Global or local scope: Most straightforward fairness techniques check or achieve group fairness (i.e., segments of interest receive similar treatment by a model as the entire population). Some fairness techniques can check or achieve individual fairness (i.e., similar individuals are treated similarly by a model).	**Best-suited complexity:** Fairness is best paired with transparent, understandable models.
Model specific or model agnostic: Many bias detection and correction techniques involve post-processing of predictions and can be model agnostic. However, some fairness techniques do involve model-specific information.	
Trust and understanding: Fairness is crucial for trusting machine learning models.	

Sensitivity Analysis: Testing Models for Stability and Trustworthiness

Sensitivity analysis investigates whether model behavior and outputs remain stable when data is intentionally perturbed or other changes are simulated in data. Beyond traditional assessment practices, sensitivity analysis of machine learning model predictions is perhaps the most important validation technique for machine learning models. Machine learning model predictions can change dramatically due to only minor changes in input variable values. In practice, many linear model validation techniques focus on the numerical instability of regression parameters due to correlation between input variables or between input variables and the target variable. It can be prudent for those switching from linear modeling techniques to machine learning techniques to focus less on numerical instability of model parameters and to focus more on the potential instability of model predictions. One of the main thrusts of linear model validation is sniffing out correlation in the training data that could lead to model parameter instability and low-quality predictions on new data. The regularization built into most machine learning algorithms makes their parameters and rules more accurate in the presence of correlated inputs, but as discussed repeatedly, machine learning algorithms can produce very complex nonlinear, nonmonotonic response functions that can produce wildly varying predictions for only minor changes in input variable values. Hence, in the context of machine learning, directly testing a model's predictions on simulated, unseen data is likely a better use of time than digging through static training data looking for hidden correlations.

Sensitivity analysis can also test model behavior and outputs when interesting situations or known corner cases are simulated. For

instance, test your model's predictions on negative incomes or ages, use character values instead of numeric values for certain variables, or try input variable values 10 to 20% larger in magnitude than would ever be expected to be encountered in new data. If you can't think of any interesting situations or corner cases, simply try a *random data attack*: score many samples of random data with your machine learning model and analyze the resulting predictions. You will likely be surprised by what you find.

Table 1-20. A description of sensitivity analysis

Technique: Sensitivity analysis
Suggested usage: Testing machine learning model predictions for accuracy and stability using simulated data. *If you are using a machine learning model, you should probably be conducting sensitivity analysis.*
OSS: *http://bit.ly/2FQ7X8E* *https://github.com/h2oai/mli-resources/blob/master/notebooks/sensitivity_analysis.ipynb*
Global or local scope: Sensitivity analysis can be a global interpretation technique when many input rows to a model are perturbed, scored, and checked for problems, or when global interpretation techniques are used, such as using a single, global surrogate model to ensure major interactions remain stable when data is lightly and purposely corrupted. Sensitivity analysis can be a local interpretation technique when a single row is perturbed, scored, and checked or when local interpretation techniques are used, for instance using LIME to determine if the important variables in a credit allocation decision remain stable for a given customer segment under macroeconomic stress testing.
Best-suited complexity: Sensitivity analysis can help explain the predictions of nearly any type of response function, but it is probably most appropriate for nonlinear response functions and response functions that model high degree variable interactions. For both cases, small changes in input variable values can result in large changes in a predicted response.
Model specific or model agnostic: Model agnostic.
Trust and understanding: Sensitivity analysis enhances understanding because it shows a model's likely behavior and output in important situations, and how a model's behavior and output may change over time. Sensitivity analysis enhances trust when a model's behavior and outputs remain stable when data is subtly and intentionally corrupted. It also increases trust if models adhere to human domain knowledge and expectations when interesting situations are simulated, or as data changes over time.

Testing Interpretability

The approximate nature of machine learning explanations can, and often should, call into question the trustworthiness of model explanations themselves. Don't fret! You can test explanations for accuracy. Originally, researchers proposed testing machine learning model

explanations by their capacity to enable humans to correctly determine the outcome of a model prediction based on input data values.[13] Very recent research has highlighted the potential bias of human practitioners toward simpler explanations, even when simple explanations are inaccurate[14]. Given that human evaluation studies are likely impractical for most commercial data science or machine learning groups anyway, several more automated approaches for testing model explanations are proposed here.

Simulated data
: You can use simulated data with known characteristics to test explanations. For instance, models trained on totally random data with no relationship between a number of input variables and a prediction target should not give strong weight to any input variable nor generate compelling local explanations or reason codes. Conversely, you can use simulated data with a known signal generating function to test that explanations accurately represent that known function.

Explanation stability with increased prediction accuracy
: If previously known, accurate explanations or reason codes from a simpler linear model are available, you can use them as a reference for the accuracy of explanations from a related, but more complex and hopefully more accurate, model. You can perform tests to see how accurate a model can become before its prediction's reason codes veer away from known standards.

Explanation stability under data perturbation
: Trustworthy explanations likely should not change drastically for minor changes in input data. You can set and test thresholds for allowable explanation value changes automatically by perturbing input data. Explanations or reason code values can also be averaged across a number of models to create more stable explanations.

Machine Learning Interpretability in Action

To see how some of the interpretability techniques discussed in this report might look and feel in action, a public, open source repository has been provided.

This repository contains examples of white-box models, model visualizations, reason code generation, and sensitivity analysis applied to

the well-known Taiwanese credit card customer dataset[15] using the popular XGBoost and H2O libraries in Python.

Conclusion

FAT/ML, explainable AI, and machine learning interpretability are new, rapidly changing, and expanding fields. The widespread acceptance of machine learning interpretability techniques will likely be a contributing factor in the increasing adoption of machine learning and artificial intelligence in both commercial applications and in our day-to-day lives. At this juncture, training interpretable machine learning models is still a difficult process, and yet practitioners should probably begin considering accountability, explainability, interpretability, and transparency from the beginning of any serious applied machine learning project. Moreover, new explanatory techniques are bubbling up frequently. Some are rigorously tested before being publicized, others are not. Some recognize the approximate nature of the explanations they generate. Some do not. Practitioners should be cognizant of the source of any explainability software, consider testing any explanations they plan to use in mission-critical applications, try out multiple local and global explanation-generating techniques, and seek consistent results across these multiple techniques.

Along with interpretability, automated machine learning is another important new trend in artificial intelligence. Several open source and proprietary software packages now build machine learning models automatically with minimal human intervention. These new automatic systems tend to be even more complex, and therefore black box in nature, than today's more human-oriented data science workflows. For automation of machine learning to take hold across a broad cross-section of industries, these cutting-edge predictive modeling systems will need to be accountable, interpretable, and transparent to their users.

References

[1] Hall, Patrick, Wen Phan, and Katie Whitson. *The Evolution of Analytics: Opportunities and Challenges for Machine Learning in Business.* Sebastopol, CA: O'Reilly Media, 2016. *http://oreil.ly/2DIBefK*

[2] *Interpretability.* Fast Forward Labs, 2017. *https://www.fastforwardlabs.com/research/ff06*

[3] Donoho, David. "50 years of Data Science." Tukey Centennial Workshop, 2015. *http://bit.ly/2GQOh1J*

[4] Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015. *http://www.evolvingai.org/fooling*

[5] Angwin, Julia et al. "Machine bias: there's software used across the country to predict future criminals. and it's biased against blacks." *ProPublica*. 2016. *https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*

[6] Goodman, Bryce, and Seth Flaxman. "EU regulations on algorithmic decision-making and a 'right to explanation'" *ICML workshop on human interpretability in machine learning (WHI 2016)*. 2016. URL *https://arxiv.org/pdf/1606.08813.pdf*

[7] Evtimov, Ivan et al. "Robust Physical-World Attacks on Deep Learning Models." arXiv preprint. 2017. *https://iotsecurity.eecs.umich.edu/#roadsigns*

[8] Bob Crutchfield. "Approve More Business Customers," *Equifax Insights Blog*. 2017. *https://insight.equifax.com/approve-business-customers/*

[9] Breiman, Leo. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)." *Statistical Science*. 2001. *http://bit.ly/2pwz6m5*

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

[11] Patrick Hall et al. *Machine Learning Interpretability with H2O Driverless AI*. H2O.ai. 2017. *http://bit.ly/2FRqKAL*

[12] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should I trust you?: Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. *http://bit.ly/2FROG6X*

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

[14] Herman, Bernease. "The Promise and Peril of Human Evaluation for Model Interpretability." arXiv preprint. 2017. *https://arxiv.org/pdf/1711.07414.pdf*

[15] Lichman, M. UCI machine learning repository, 2013. *http://archive.ics.uci.edu/ml*

Acknowledgments

The authors are thankful to colleagues past and present whose comments, thoughts, and tips undoubtedly shaped our own thinking on the topic of interpretable machine learning. In particular, we thank Mark Chan, Leland Wilkinson, Michal and Megan Kurka, Wen Phan, Lingyao Meng, and Sri Satish Ambati at H2O.ai, Andrew Burt at Immuta, and Lisa Song at George Washington University.

About the Authors

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.

Navdeep Gill is a software engineer and data scientist at H2O.ai where he focuses on model interpretability, GPU accelerated machine learning, and automated machine learning. 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.