Interpretable Machine Learning

Patrick Hall

H2O.ai

You've probably heard by now that machine learning algorithms can use "big data" to predict whether a donor will give to a charity, whether an infant in a NICU will develop sepsis, whether a customer will respond to an ad, and on and on. Machine learning can even drive cars and predict elections! ... Err, wait. Can it? I actually think it can, but these recent high profile hiccups should leave everyone that works with data (big or not) and machine learning algorithms asking themselves some very hard questions: Do I really understand my data? Do I really understand the model and answers my machine learning algorithm is giving me? And do I actually trust these answers? Unfortunately, the complexity that bestows the extraordinary predictive abilities on machine learning algorithms also makes the answers the algorithms produce hard to understand, and maybe even hard to trust.

Although it is possible to enforce monotonicity constraints (a relationship that only changes in one direction) between inputs and a machine-learned response function, machine learning algorithms tend to create nonlinear, non-monotonic, non-polynomial, and even non-continuous functions that approximate the relationship between independent and dependent variables in a data set. (This relationship is also referred to as the conditional distribution of the dependent variable(s), given the values of the independent variable(s).) These functions can then make very specific predictions about the values of dependent variables for new data, like whether a donor will give to a charity, an infant in a NICU will develop sepsis, whether a customer will to respond to an ad, and on and on. Conversely, traditional linear models tend to create linear, monotonic, and continuous functions to approximate the very same relationships. Even though they're not always the most accurate predictors, the elegant simplicity of linear models makes the results they generate easy to interpret.

While understanding and trusting models and results is a general requirement for good (data) science, model interpretability is a serious legal mandate in the regulated verticals of banking, insurance, and other industries. Business analysts, doctors, and industry researchers simply have to understand and trust their models and modeling results. For this reason, linear models were the goto applied predictive modeling tool for decades even though it usually meant giving up a couple points on the accuracy scale. Today many organizations and individuals are embracing machine learning algorithms for predictive modeling tasks, but difficulties in interpretation still present a barrier for the widespread, practical use of machine learning algorithms.

Several approaches beyond the usual error measures and assessment plots are presented below for visualizing data and interpreting machine learning models and results. Users are encouraged to mix and match these techniques to best fit their own needs. Wherever possible interpretability is deconstructed into more basic components: complexity, scope, understanding and trust.

Contents

Part 1: Seeing your data

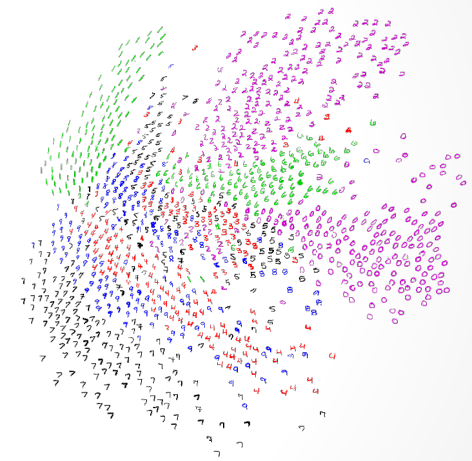
- Glyphs
- Correlation graphs
- 2-D projections
- Partial dependence plots
- Residual analysis

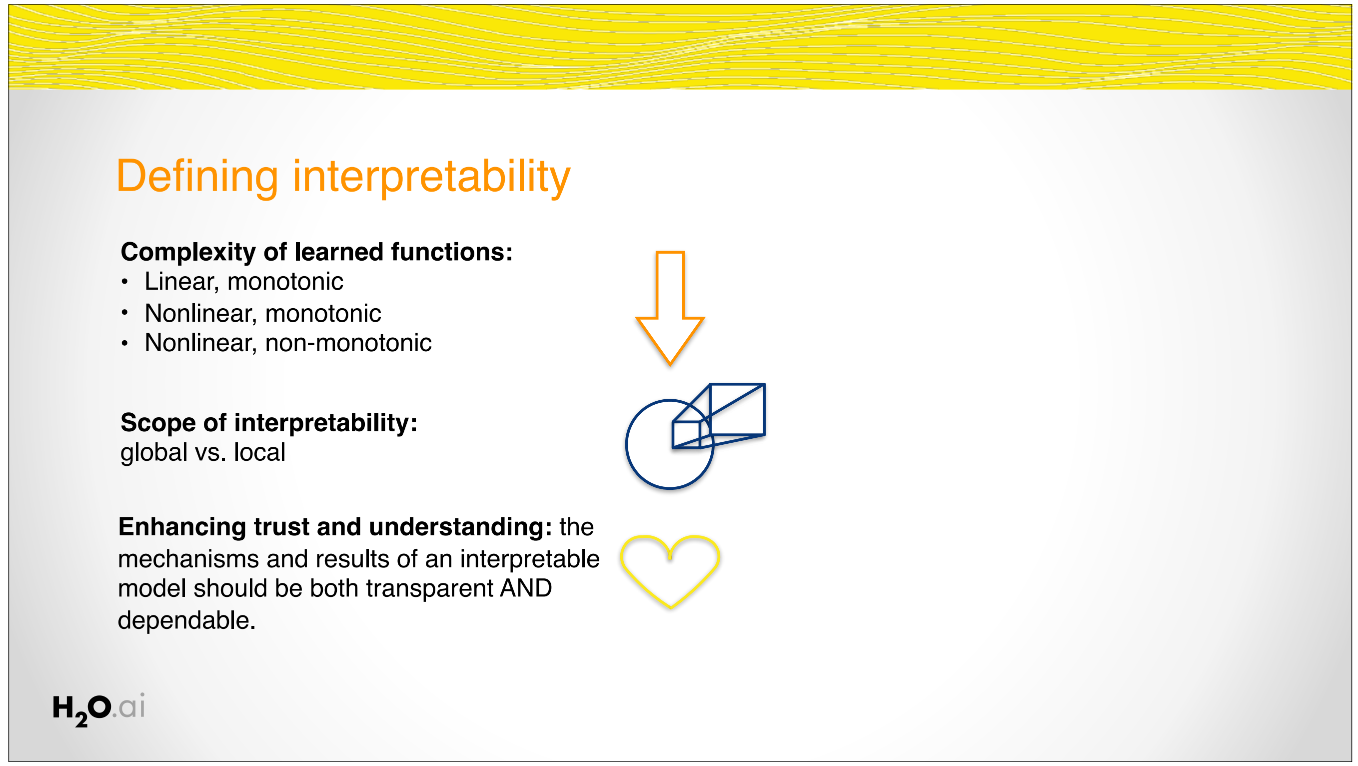
Part 2: Using machine learning in regulated industry

- OLS regression alternatives
- Build toward ML model benchmarks
- ML in traditional analytics processes
- Small, interpretable ensembles
- Monotonicity constraints

Part 3: Understanding complex ML models

- Surrogate models
- LIME
- Maximum activation analysis
- Sensitivity analysis
- Variable importance measures
- TreeInterpreter





Complexity of response function to be explained

Linear, monotonic functions: Functions created by linear regression algorithms are probably the most interpretable class of models. These models will be referred to here as linear and monotonic, meaning that for a change in any given independent variable (or sometimes combination or function of an independent variable) 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 lender rejects your credit card application, they can tell you why because their probability of default model 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 often called reason codes. Of course, linear and monotonic response functions enable the calculation of relative variable importance measures too. Linear and monotonic functions have several uses in machine learning interpretability. Part 2, surrogate models, and Local Interpretable Model-agnostic Explanations below all discuss the many ways linear, monotonic functions can be used to make machine learning interpretable.

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. While there is no single coefficient that represents the change in the response function induced by a change in a independent variable, nonlinear and monotonic functions do always change in one direction. Nonlinear, monotonic response functions usually allow for the generation of both reason codes and relative variable importance measures. Nonlinear, monotonic response functions are highly interpretable and often suitable for use in regulated applications.

(Of course there are linear, non-monotonic machine-learned response functions, for instance they can be created by the Multi-variate adaptive regression splines approach. These functions are not highlighted here because they tend to be less accurate predictors than purely nonlinear, non-monotonic functions while also lacking the high interpretability of their monotonic counterparts.)

Nonlinear, non-monotonic functions: The vast majority of machine learning algorithms create nonlinear, non-monotonic response functions. This class of functions are the most difficult to interpret. They can change in a positive and negative direction and at a varying rate for any change in an independent variable. Typically the only interpretability measure that can be provided directly from a nonlinear, non-monotonic function are relative variable importance measures. You may need to use a combination of the techniques presented below to interpret these extremely complex models.

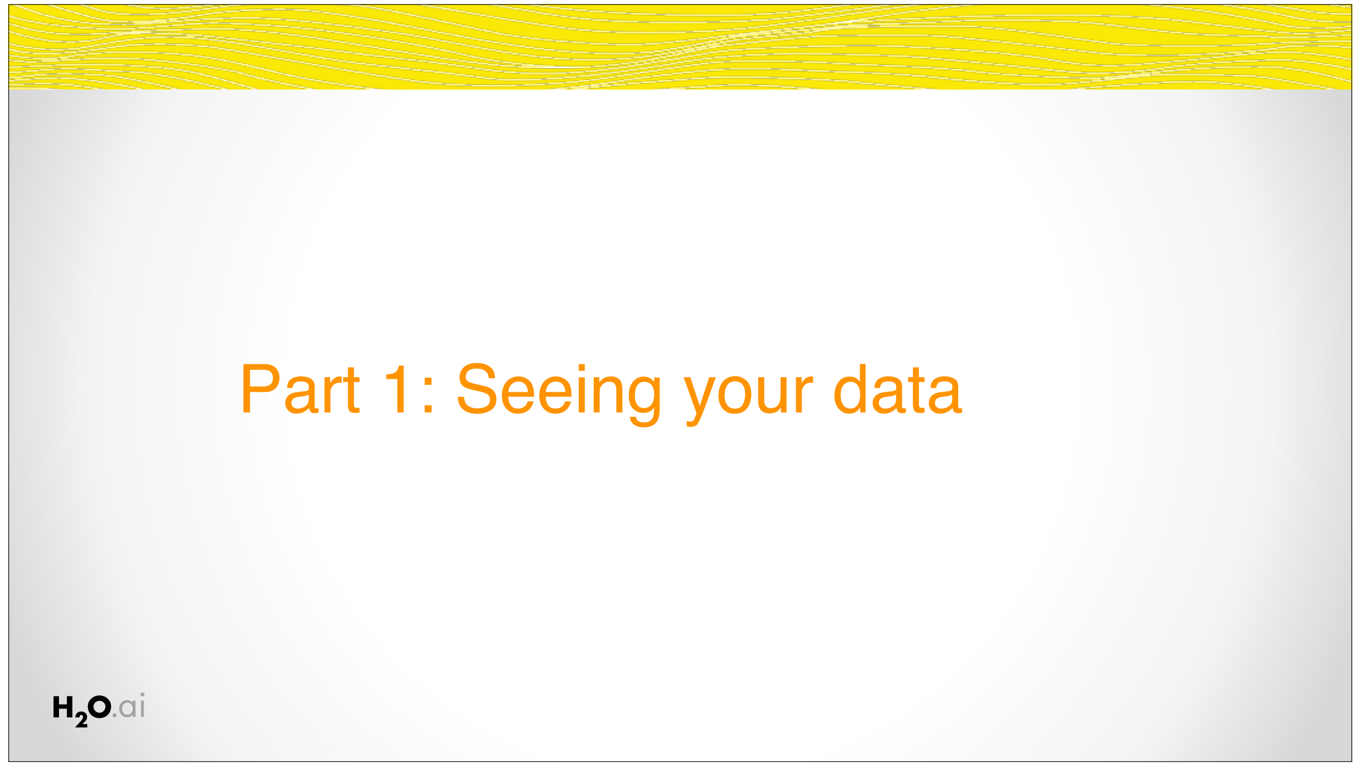
Scope of interpretability

Global interpretability: Some of the presented techniques facilitate global interpretations of machine learning algorithms, their results, or the machine-learned relationship between the inputs and the dependent variable(s), e.g. the model of the conditional distribution. Global interpretations help us understand the entire conditional distribution modeled by the trained response function but global interpretations can be approximate or based on average values.

Local interpretability: Local interpretations promote understanding of small regions of the conditional distribution, such as clusters of input records and their corresponding predictions, or deciles of predictions and their corresponding input rows. Because small sections of the conditional distribution are more likely to be linear, monotonic, or otherwise well-behaved, local explanations can be more accurate than global 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 us. For some users, technical descriptions of algorithms provide enough insight into machine learning models and cross-validation, error measures, and assessment plots provide enough information to trust a model. The techniques presented here go beyond these standard practices and measurements to enhance understanding and trust in machine learning models. These techniques enhance understanding when they provide specific insights into the mechanisms of the algorithms and the functions they create, or detailed information about the answers they provide. The techniques below enhance trust when they allow users to ensure the stability and dependability of machine learning algorithms, the functions they create, or the answers they generate.



Most real data sets are hard to see because they have many variables and many rows. Like most sighted people, I rely on my visual sense quite heavily for understanding information. For me, seeing data is basically tantamount to understanding data. Moreover, I can really only understand two or three visual dimensions, preferably two, and something called change blindness frustrates human attempts to reason analytically given information split across different pages or screens. So if a data set has more than two or three variables or more rows than can fit on a single page or screen, it's realistically going to be hard to understand what's going on in it without resulting to more advanced techniques than scrolling through countless rows of data.

Of course there are many, many ways to visualize data sets. The techniques highlighted below help illustrate all of a data set in just two dimensions, not just univariate or bivariate slices of a data set (meaning one or two variables at a time). This is important in machine learning because most machine learning algorithms automatically model high degree interactions between variables (meaning the effect of combining many, i.e. more than two or three, variables together). Traditional univariate and bivariate tables and plots are still important and you should use them, but they are more relevant in the context of traditional linear models and slightly less helpful in understanding nonlinear models that can pick up on arbitrarily high degree interactions between independent variables.

Do visualizations provide global or local interpretability?

Both. Most forms of visualizations can be used to see a courser view of the entire data set or they can provide granular views of local portions of the data set. Ideally advanced visualization tool kits enable users to pan, zoom, and drill-down easily. Otherwise, users can plot different parts of the data set at different scales themselves.

What complexity of functions can visualizations help interpret?

Visualizations can help explain functions of all complexities.

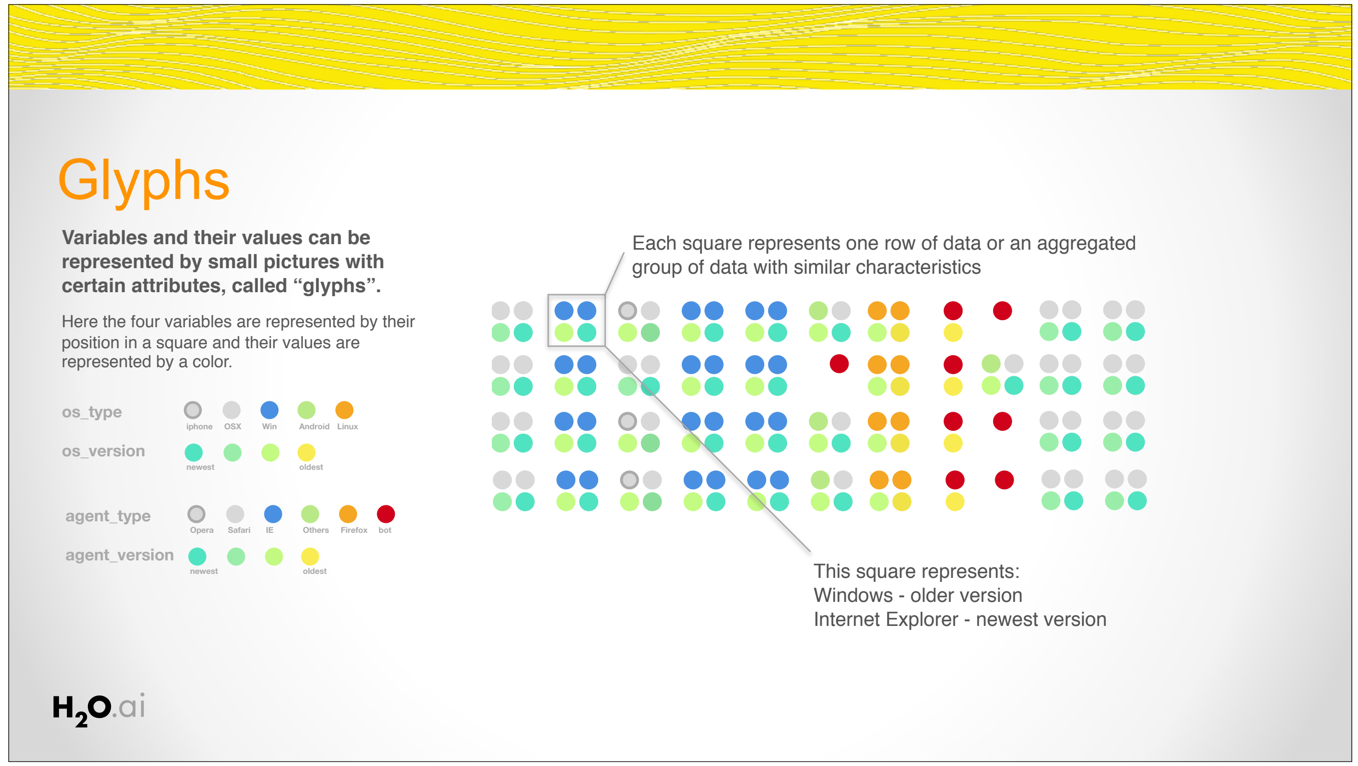
How do visualizations enhance understanding?

For most people, visual representations of structures (clusters, hierarchy, sparsity, outliers) and relationships (correlation) in a data set are easier to understand than scrolling through plain rows of data and looking at variable's values.

How do visualizations enhance trust?

Seeing structures and relationships in a data set usually makes those structures and relationships easier to understand. An accurate machine learning model should create answers that are representative of the structures and relationships in a data set. Understanding the structures and relationships in a data set is a first step to knowing if a model's answers are trustworthy.

In certain cases, visualizations can display the results of sensitivity analysis, which can also enhance trust in machine learning results. In general, visualizations themselves can sometimes be thought of as a type of sensitivity analysis when they are used to display data or models as they change over time or as data are intentionally changed to test stability or interesting scenarios.



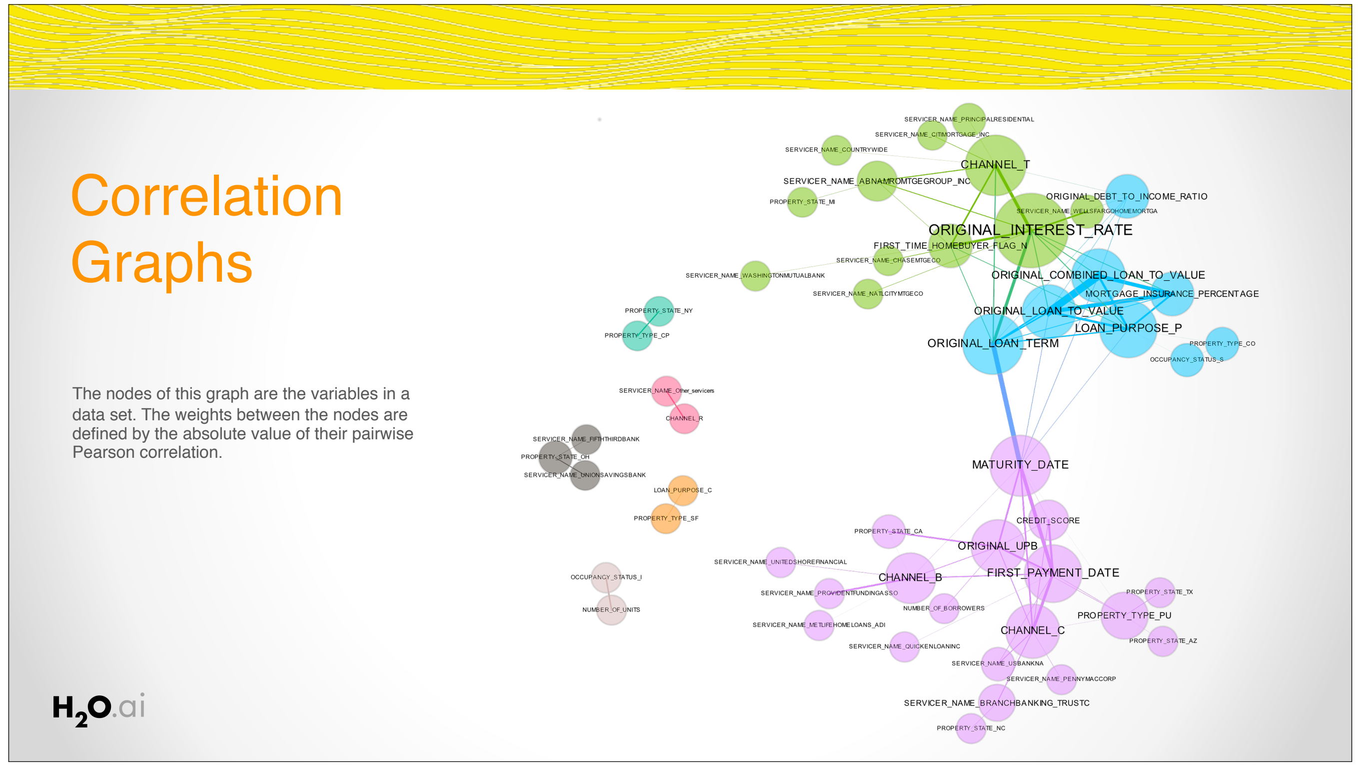
Glyphs

Figure 1: Glyphs representing operating systems and web browser (agent) types. Image courtesy of Ivy Wang and the H2o.ai team.

Glyphs are visual symbols used to represent a data. The color, texture, or alignment of a glyph can be used to represent different values or attributes of data. In figure 1, colored circles are defined to represent different types of operating systems and web browsers. When arranged in a certain way, these glyphs can be used to represent rows of a data set.

Figure 2: Glyphs arranged to represent many rows of a data set. Image courtesy of Ivy Wang and the H2o.ai team.

Figure two gives an example of how glyphs can be used to represent rows of a data set. Each grouping of four glyphs can be either a row of data or an aggregated group of rows in a data set. The highlighted Windows/Internet Explorer combination is very common in the data set and so is the OS X and Safari combination represented by two grey circles. It’s quite likely these two combinations are two compact and disjoint clusters of data. We can also see that in general operating system versions tend to be older than browser versions, and that using Windows and Safari is correlated with using newer operating system and browser versions whereas Linux users and bots are correlated with older operating system and browser versions. The red dots that represent queries from bots standout visually (unless you are red-green colorblind ...). Using bright colors or unique alignments for events of interest or outliers is a good method for making important or unusual data attributes readily apparent in a glyph representation.



Correlation Graphs

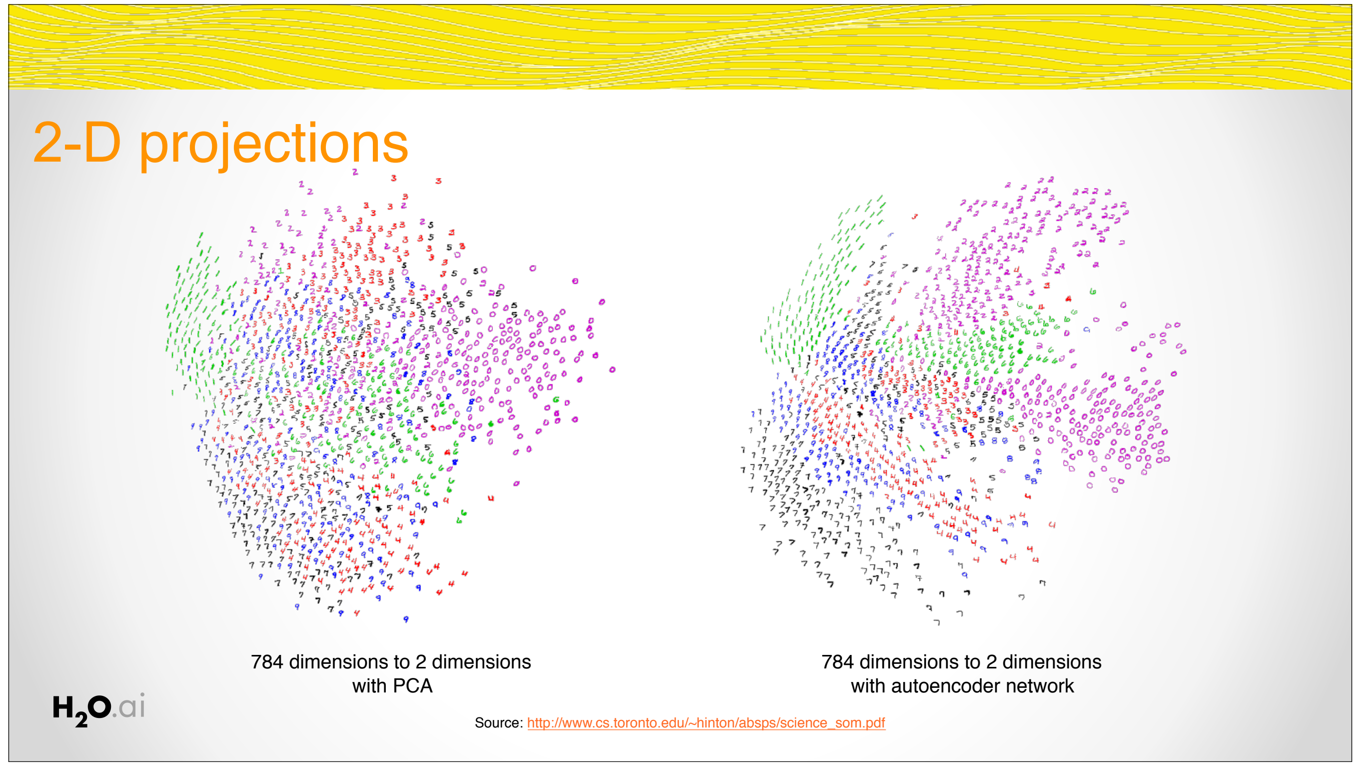
Figure 3: A correlation graph representing loans made by a large financial firm.

A correlation graph is a two dimensional representation of the relationships (correlation) in a data set. While many details regarding the display of a correlation graph are optional and could be improved beyond those chosen for figure 3, correlation graphs are a very powerful tool for seeing and understanding relationships (correlation) between variables in a data set. Even data sets with tens of thousands of variables can be displayed in two dimensions using this technique.

In figure 3, the nodes of the graph are the variables in a loan data set and the edge weights (thickness) between the nodes are defined by the absolute values of their pairwise Pearson correlation. For visual simplicity, 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 communities 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.

In a supervised model built for the data represented in figure 3, assuming one of the represented variables was an appropriate target, we would expect variable selection techniques to pick one or two variables from the light green, blue and purple groups, we would expect variables with thick connections to the target to be important variables in the model, and we would expect a model to learn that unconnected variables like CHANNEL_R are not very important. Figure 3 also illustrates common sense relationships such as that between FIRST_TIME_HOMEBUYER_FLAG_N and ORIGINAL_INTEREST_RATE that should be reflected in a trustworthy model.

The graph in figure 3 was created with Gephi.



2-D projections

Image: http://www.cs.toronto.edu/~hinton/absps/science_som.pdf
Figure 4: Two dimensional projections of the famous 784-dimensional MNIST data set using (left) Principal Components Analysis (PCA) and (right) a stacked denoising autoencoder.

There are many techniques for projecting the rows of a data set from a usually high-dimensional original space into a more visually understandable lower-dimensional space, ideally two or three dimensions. Popular techniques include:

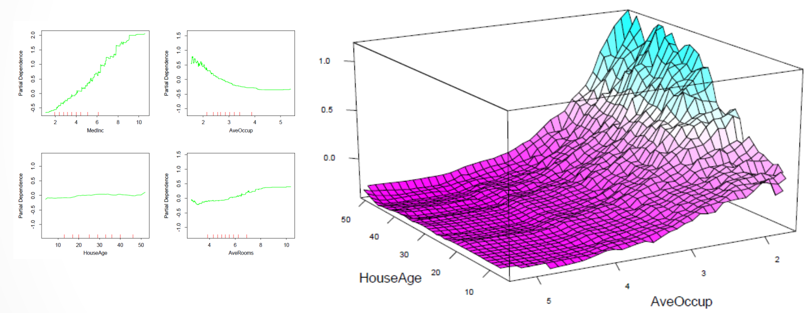
Principal Component Analysis (PCA)
Multidimensional Scaling (MDS)
t-SNE (t-distributed Stochastic Neighbor Embedding)
Autoencoder networks

Each of these techniques have strengths and weaknesses, but the key idea they all share is to represent the rows of a data set in a meaningful low dimensional space. When a data set has more than two or three dimensions, visualizing it with a scatter plot becomes essentially impossible, but these techniques enable even high-dimensional data sets to be projected into a representative low-dimensional space and visualized using the trusty, old scatter plot. A high quality projection visualized in a scatter plot should exhibit key structural elements of a data set such as clusters, hierarchy, sparsity, and outliers.

In figure 4, the famous MNIST data set is projected from its original 784 dimensions onto two dimensions using two different techniques, PCA and autoencoder networks. The quick and dirty PCA projection is able to separate digits labeled as zero from digits labeled as one very well. These two digit classes are projected into fairly compact clusters, but the other digit classes are generally overlapping. In the more sophisticated, but also more computationally expensive, autencoder projection all the digit classes appear as separate clusters with visually similar digits appearing close to one another in the reduced two-dimensional space. The autoencoder projection is capturing the clustered structure of the original high-dimensional space and the relative locations of those clusters. Interestingly, both plots are able to pick up on a few outlying digits.

Projections can add an extra and specific 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 data sets 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 relative 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 relative disjoint and compact in a projection, and relatively far from one another. Such results should also be stable under minor perturbations of the training or test data, and projections from perturbed vs. non-perturbed samples can be used to check for stability or for potential patterns of change over time.

Partial dependence plots



HomeValue ~ MedInc + AveOccup + HouseAge + AveRooms

Source: https://web.stanford.edu/~hastie/local.ftp/Springer/OLD/ESLII_print4.pdf

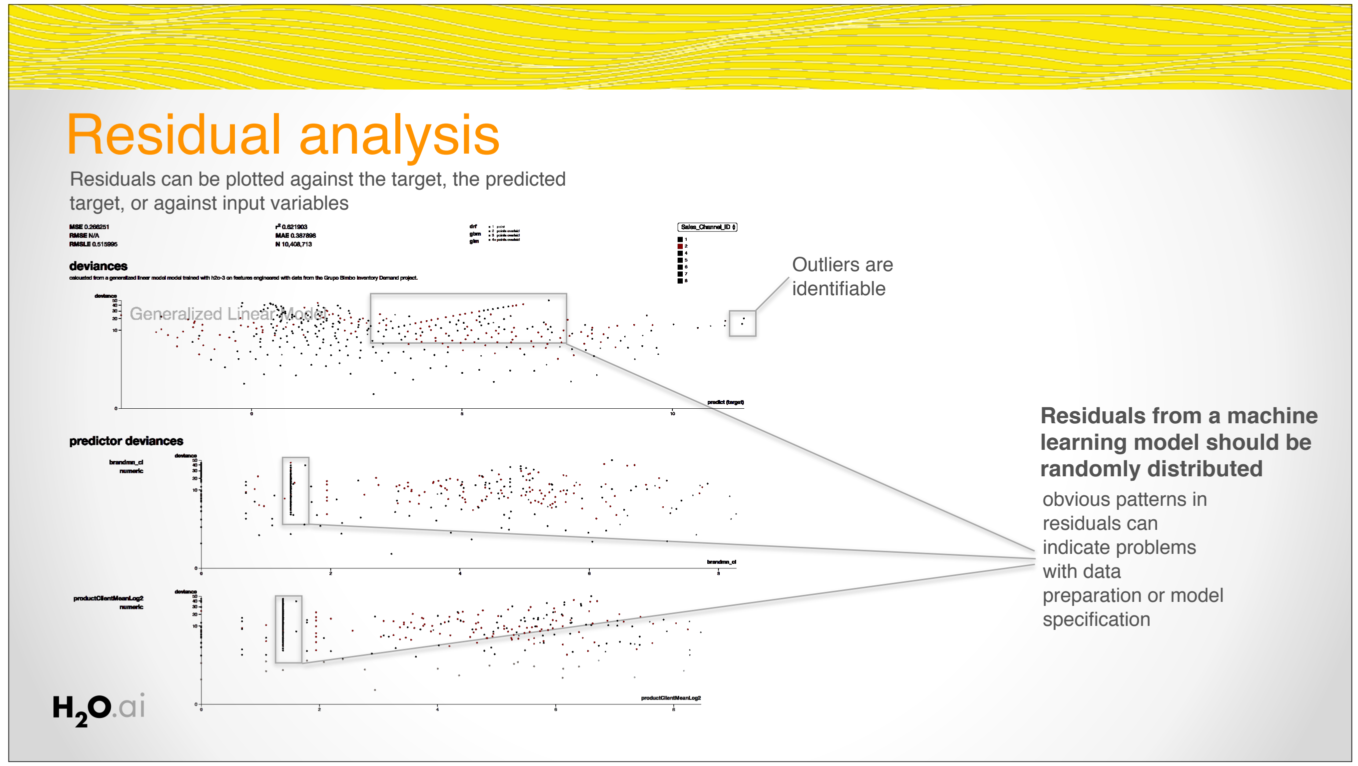
Partial dependence plots

Image: http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
Figure 5: One- and two- dimensional partial dependence plots from a gradient boosted tree ensemble model of the famous California housing data set.

Partial dependence plots show us the way machine-learned response functions change based on the values of one or two independent variables while holding all the other independent variables at their average value, e.g. the marginal average of the dependent variable by the displayed independent variable(s). Partial dependence plots with two independent variables are particularly useful for visualizing complex interactions.

Partial dependence plots are global in terms of the rows of a data set, but local in terms of the independent variables. They are used almost exclusively to show the relationship between one or two independent variables and the dependent variable over the entire domain of the independent variable(s). Individual conditional expectation plots, a newer and less well-known adaptation of partial dependence plots, can be used to create more localized partial dependence plots.

Partial dependence plots are superfluous for linear, monotonic response functions as they would be simply a straight line or a plane with the slope of the regression coefficient(s). Partial dependence plots can be used to verify monotonicity of response functions under monotonicity constraints, and they can be used to see the nonlinearity, non-monotonicity, and two-way interactions in very complex models. In fact, the way partial dependence plot enhance understanding is exactly by showing the nonlinearity, non-monotonicity, and two-way interactions 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.



Residual analysis

Figure 6: Screenshot from an example residual analysis application. Image courtesy of Micah Stubbs and the H2o.ai team.

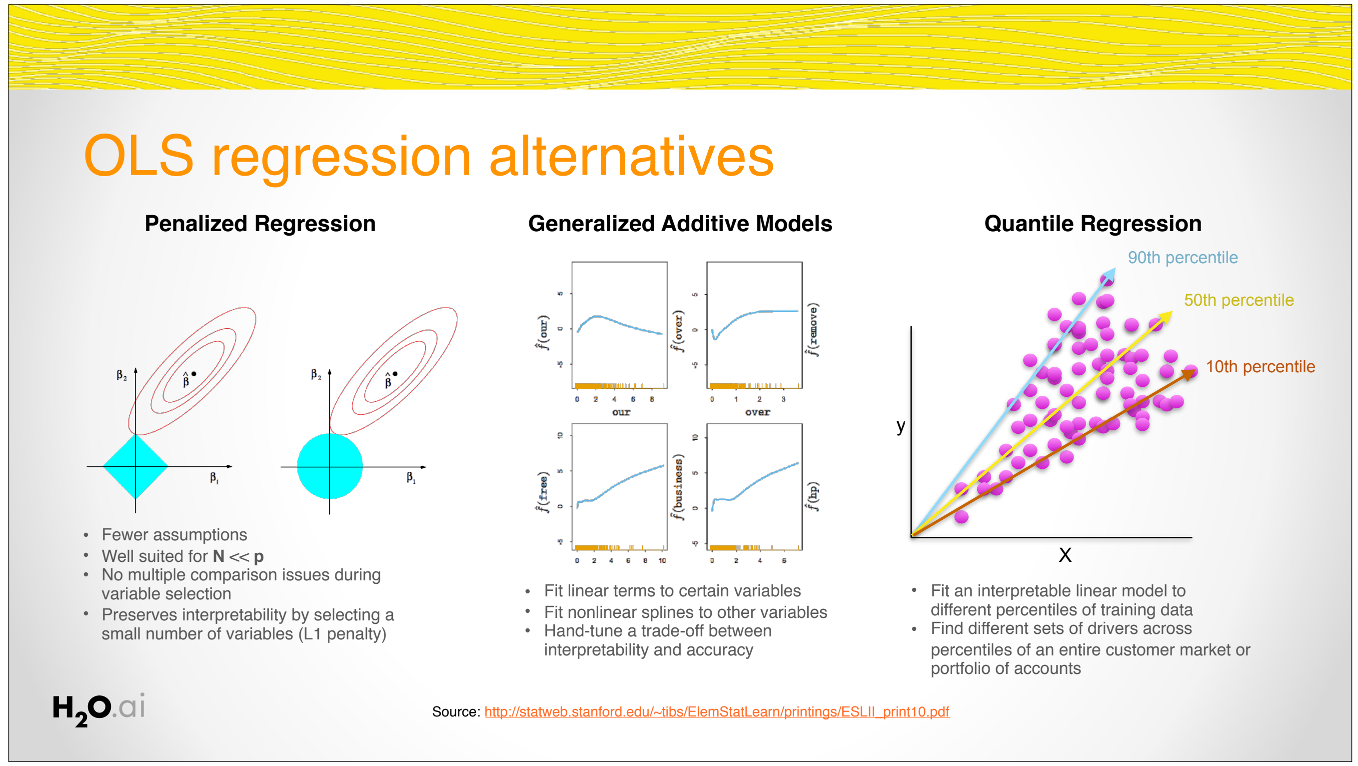
Residuals refer to the difference between the recorded value of a dependent variable and the predicted value of a dependent variable for every row in a data set. Generally, the residuals of a well-fit model should be randomly distributed because good models will account for most phenomena in a data set except for random error. Plotting the residual values, often the squared residual values - since those are always positive, against the predicted values is a time honored model assessment technique and a great way to see all of your modeling results in two dimensions. If strong patterns are visible in plotted residuals, this is a dead giveaway that there are problems with your data, your model, or with both. Vice versa, if models are producing random residuals this a strong indication of a well-fit, dependable, trustworthy model, especially if other fit statistics (i.e. R2, AUC, etc.) are in the appropriate ranges.

In figure 6, the callouts point to a strong linear pattern in the residuals. The plot shows the traditional residual plot and residuals plotted by certain independent variables. Breaking the residual plot out by independent variables can expose more granular information about residuals and assist in reasoning through the cause of non-random patterns. Figure 6 also points to outliers, which residual plots can help to identify. As many machine learning algorithms seek to minimize residuals, observations with high residual values will have a strong impact on most models and human analysis of the validity of these outliers can have a big impact on model accuracy.

Part 2: Using ML in regulated industry

For analysts and data scientists working in regulated industries, the potential boost in predictive accuracy provided by machine learning algorithms may not outweigh the current realities internal of documentation needs and external regulatory responsibilities. For these practitioners, traditional linear modeling techniques may be their only option for predictive modeling. However, the forces of innovation and competition don't stop because you work under a regulatory regime. Data scientists and analysts in the regulated verticals of banking, insurance, and other similar industries face a particular conundrum. They have to find ways to make more and more accurate predictions, but keep their models and modeling processes transparent and interpretable.

The techniques presented in this section are newer types of linear models or they use machine learning to augment traditional, linear modeling methods. They're meant for practitioners who just can't use machine learning algorithms to build predictive models because of interpretability concerns. They produce linear, monotonic response functions (or at least monotonic ones) with globally interpretable results similar to those of traditional linear models, but often with a boost in predictive accuracy provided by machine learning algorithms.



OLS regression alternatives

Penalized regression
Image: http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
Figure 7: Shrunken feasible regions for L1/LASSO penalized regression parameters (left) and L2/ridge penalized regression parameters (right).

Ordinary least squares (OLS) regression is about 200 years old. Maybe it's time to move on? If you're interested, penalized regression techniques can be a gentle introduction to machine learning. Contemporary penalized regression techniques usually combine L1/LASSO penalties for variable selection purposes and Tikhonov/L2/ridge penalties for robustness in a technique known as elastic net. They also make fewer assumptions about data than OLS regression. Instead of solving the classic normal equation or using statistical tests for variable selection, penalized regression minimizes constrained objective functions to find the best set of regression parameters for a given data set that also satisfy certain penalties for assigning correlated or meaningless variables high regression coefficients. You can learn all about penalized regression in Elements of Statistical Learning, but for our purposes here, its just important to know when you might want to try penalized regression.

Penalized regression is great for wide data, even data sets with more columns than rows, and for data sets with lots of correlated variables. L1/LASSO penalties drive unnecessary regression parameters to zero, avoiding potential multiple comparison problems that arise in forward, backward, and stepwise variable selection, but still picking a good, small subset of regression parameters for a data set. L2/ridge penalties help preserve parameter estimate stability, even when many correlated variables exist in a wide data set or important predictor variables are correlated, or when independent variables are highly correlated to the dependent variable. It's also important to know penalized regression techniques don't usually create confidence intervals or t-test p-values for regression parameters. These types of measures are typically only available through empirical bootstrapping experiments that require a lot of extra computing time.

Generalized Additive Models (GAMs)
Image: http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
Figure 8: Spline functions for several variables created by a generalized additive model.

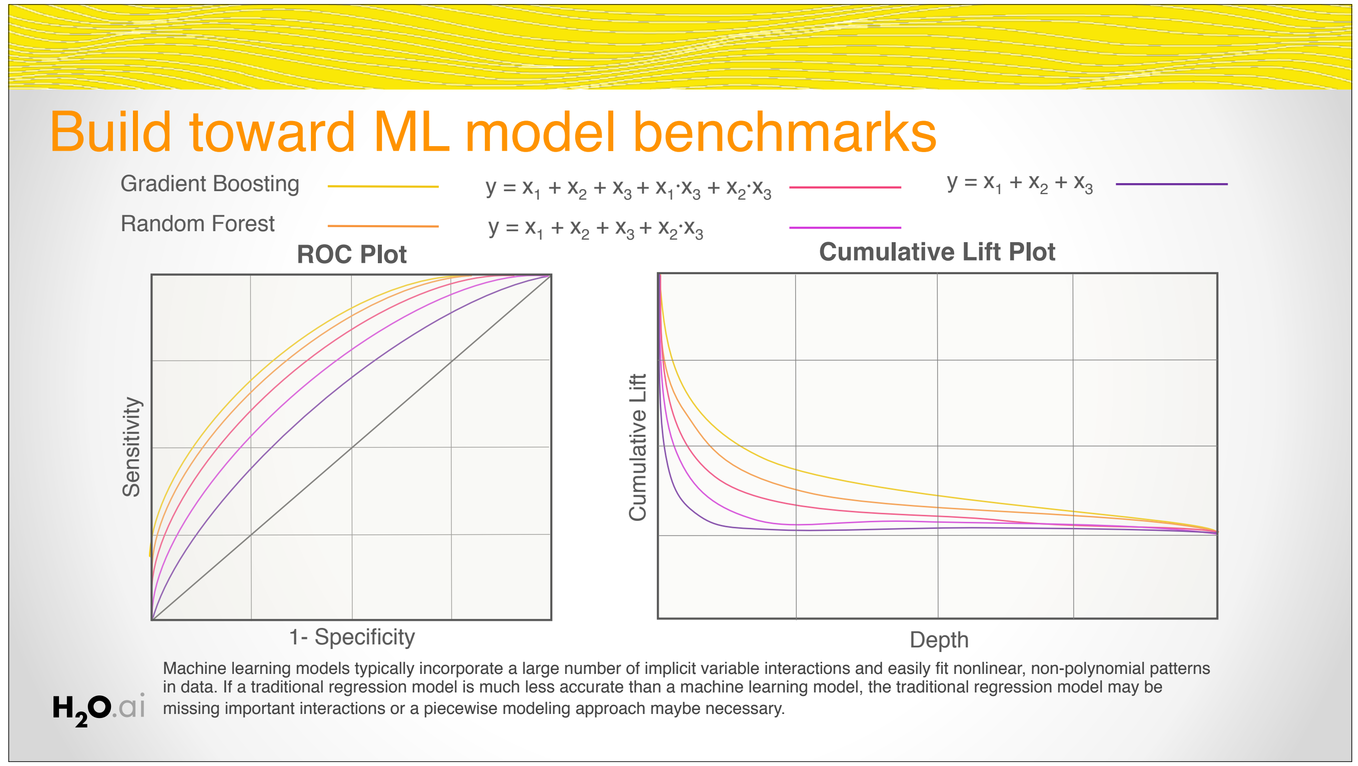
Generalized Additive Models (GAMs) enable you to hand-tune a tradeoff between accuracy and interpretability by fitting standard regression coefficients to certain variables and nonlinear spline functions to other variables. Also most implementations generate convenient plots of the the fitted splines. In many cases you may be able to eyeball the fitted spline and switch it out for a more interpretable polynomial, log, trigonometric or other simple function of the predictor variable. You can learn more about GAMs in Elements of Statistical Learning too.

Quantile regression
Figure 9: An illustration of quantile regression in two dimensions.

Quantile regression allows you to fit a traditional, interpretable, linear model to different percentiles of your training data, allowing you to find different sets of variables with different parameters for modeling different behaviors across a customer market or portfolio of accounts. It probably makes sense to model low value customers with different variables and different parameter values from those of high value customers, and quantile regression provides a statistical framework for doing so.

How do alternative regression techniques enhance understanding and trust?

Basically these techniques are plain old understandable, trusted linear models, but used in new and different ways. It's also quite possible that 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 models and more accurate understanding of modeled phenomena.



Build toward machine learning model benchmarks

Figure 10: Assessment plots that compare linear models with interactions to machine learning algorithms. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

Two of the main differences between machine learning algorithms and traditional linear models are:

Machine-learned response functions often incorporate a large number of implicit, high-degree variable interactions into their predictions while traditional, linear models typically use only single variables or two-way interaction terms.

Machine learning algorithms create nonlinear, non-polynomial, non-monotonic, and even non-continuous functions that can change drastically across an input variable's domain whereas traditional, linear models usually fit either linear functions which change at a constant rate across an input variable's domain or polynomial functions that change in smooth, well-defined ways across an input variable's domain.

Decision trees are a great way to see complex interactions in a data set. Fit a decision tree to your inputs and target and generate a plot of the tree. The variables that are under or over one-another in a given split typically have strong interactions. If a machine learning algorithm is seriously outperforming a traditional, linear model try adding some of these interactions into the linear model, including high-degree interactions that occur over several levels of the tree.

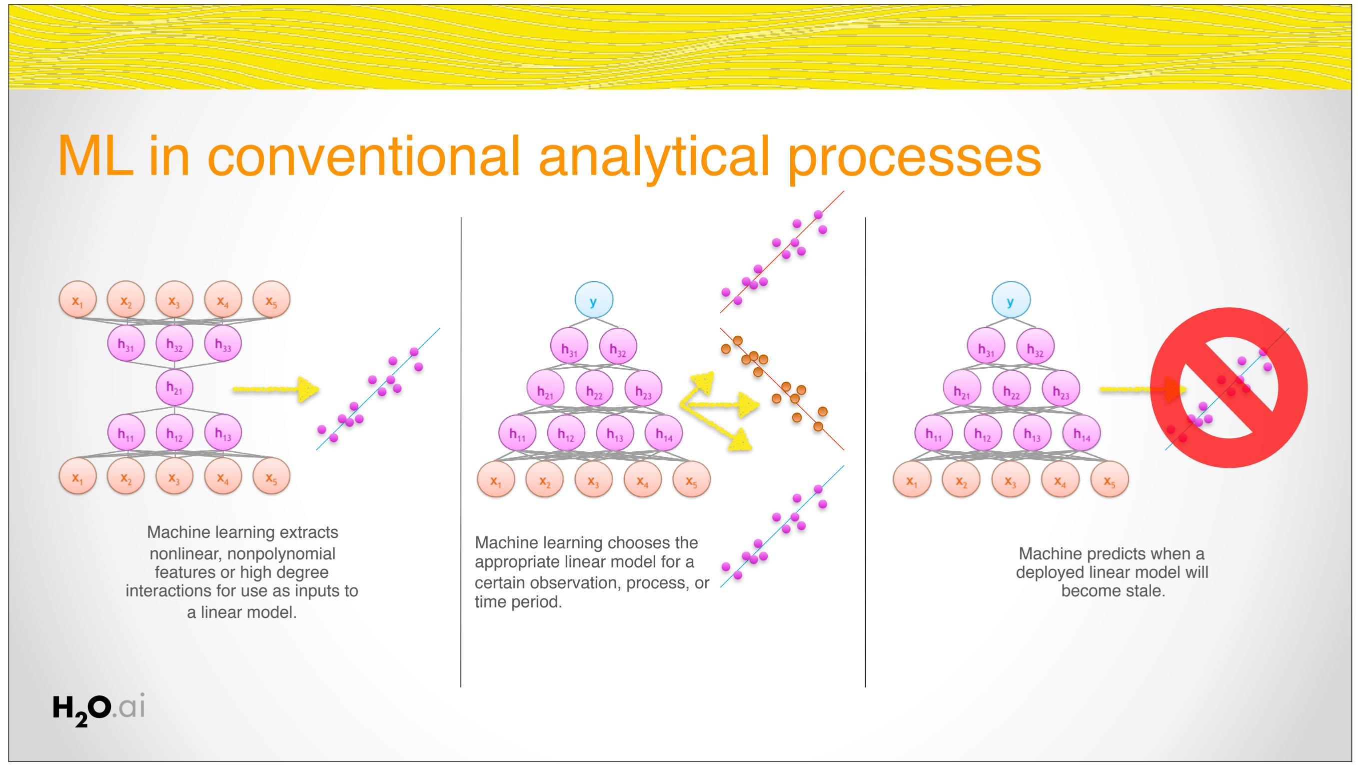
GAMs or partial dependence plots are ways to see how machine learned response functions treat a variable across it's domain and can give insight into where and how piecewise models could be used. Multi-variate adaptive regression splines is a statistical technique that can automatically discover and fit different linear functions to different parts of a complex, nonlinear conditional distribution. If a machine learning algorithm is vastly outperforming a traditional, linear model try breaking it into several piecewise linear models or try multi-variate adaptive regression splines.

How does building toward machine learning model benchmarks enhance understanding?

This process simply uses traditional, understandable models in a new way. Building toward machine learning model benchmarks could lead to greater understanding if more data exploration or techniques such as GAMs, partial dependence plots, or Multi-variate adaptive regression splines lead to deeper understanding of interactions and nonlinear phenomena in a data set.

How does building toward machine learning model benchmarks enhance trust?

This process simply uses traditional, trusted models in a new way. Building toward machine learning model benchmarks could lead to increased understanding and trust in models if additional data exploration or techniques such as GAMs, partial dependence plots, or Multi-variate adaptive regression splines create linear models that represent the phenomenon of interest in the data set more accurately.



Machine learning in traditional analytics processes

Figure 11: Diagrams of several potential uses for machine learning in traditional analytical processes. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

Instead of using machine learning predictions directly for analytical decisions, traditional analytical lifecycle processes such as data preparation and model deployment can be augmented with machine learning techniques leading to potentially more accurate predictions from regulator-approved linear, monotonic models. Figure 11 outlines three possible scenarios in which analytical processes can be augmented with machine learning:

Introduce complex predictors into traditional, linear models: Introducing interaction, polynomial, or simple functional transformations, i.e. log(), into linear models is a standard practice. Machine learning algorithms can be used to create different types of nonlinear and non-polynomial predictors that can also represent high-degree interactions between independent variables. There are many options for creating these predictors. Examples include the nonlinear features extracted by autoencoder networks or the optimal bins represented by the terminal node labels of a decision tree.

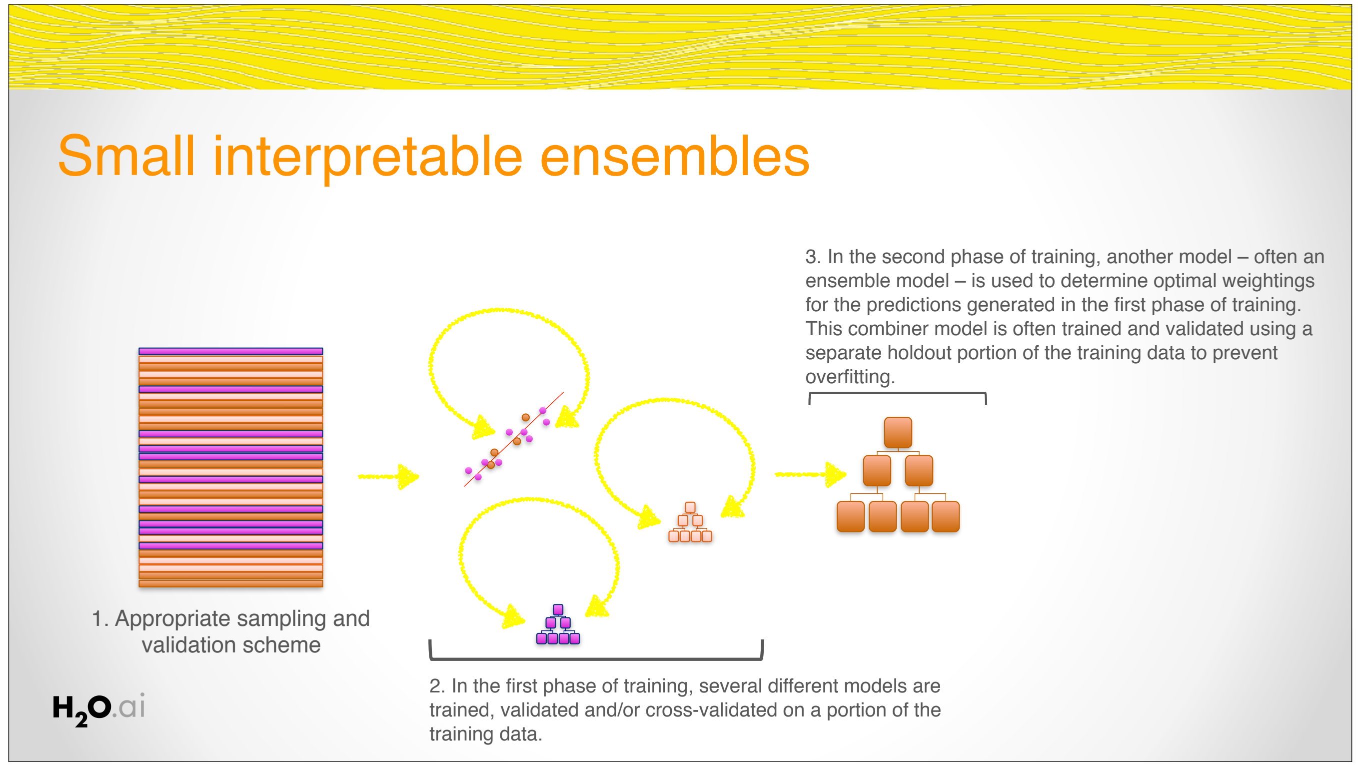
Use multiple, gated linear models: Very often segmenting data into smaller groups based on important data attributes or time period and building linear models for each segment can lead to more accurate results. It is not uncommon for organizations to use several deployed linear models to handle different market segments or different times of year. Deciding how to manually fuse the predictions of these different models can be a tedious task for analysts and data scientists. However, if data is collected about past model performance, this process can be automated by allowing a gate model to decide which linear model a particular observation should be delegated to for a decision.

Predict linear model degradation: In most cases, models are trained on static snapshots of data and then validated on later snapshots of similar data. Though an accepted practice, this process leads to model degradation when the real-world phenomena represented in the training and validation data start to change. Such degradation could occur when competitors enter or leave a market, when macroeconomic factors change, or when consumer fads change, and for many other common reasons. If data is collected about market and economic factors and about past model performance, another model can be used to predict when traditional deployed models need to be retrained or replaced. Like changing an expensive mechanical component before it actually requires maintenance, models can be retrained or replaced before their predictive power lessens.

Of course there are many other opportunities for incorporating machine learning into the lifecycle of a traditional model. You may have better ideas or implementations in place already!

How does incorporation of machine learning into traditional analytical processes enhance trust and understanding?

It can help make our understandable models more accurate, and if augmentation does lead to increased accuracy this is an indication that the pertinent phenomena in the data have been modeled in a more trustworthy, dependable fashion.



Small, interpretable ensembles

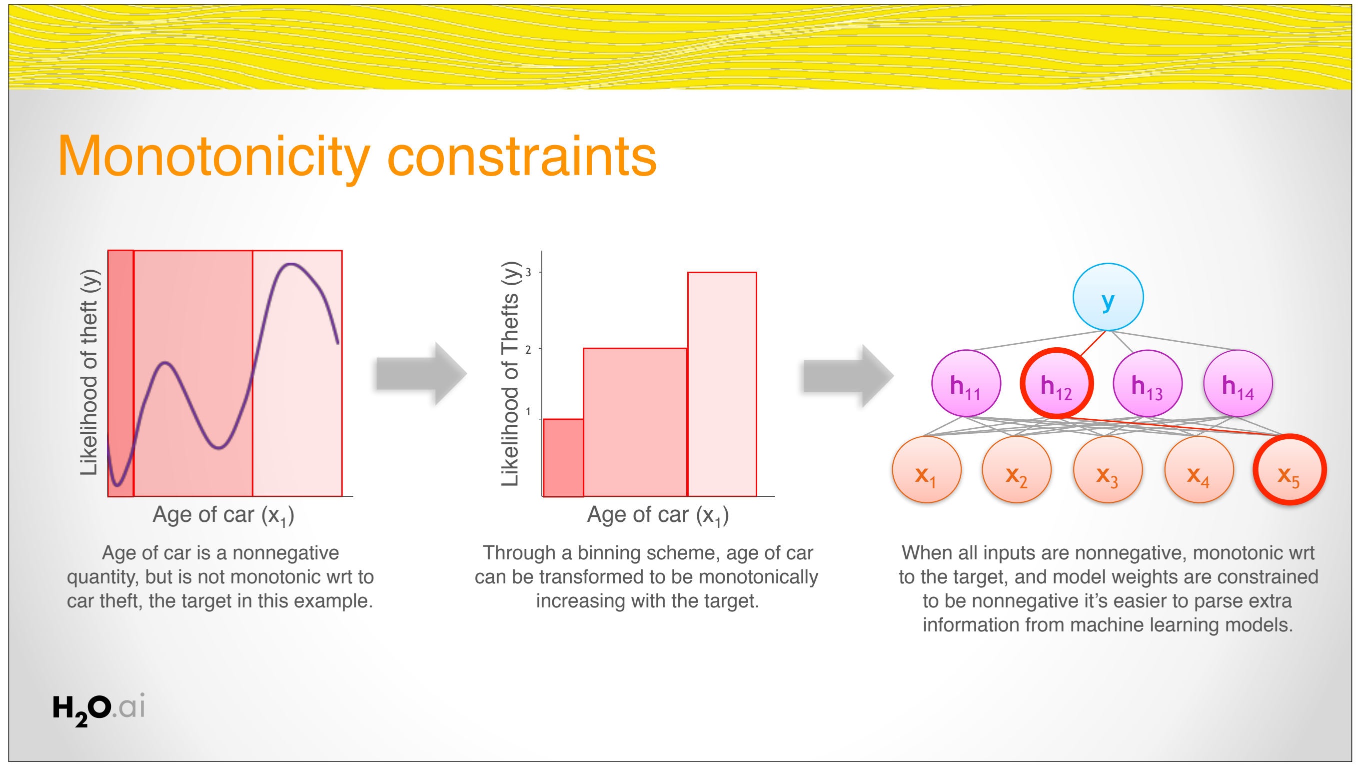
Figure 12: A diagram of a small, stacked ensemble. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

Many organizations are so adept at traditional linear modeling techniques that they simply cannot squeeze much more accuracy out of any single model. One potential way to increase accuracy without losing too much interpretability is to combine the predictions of a small number of well-understood models. The predictions can simply be averaged, manually weighted, or combined in more mathematically sophisticated ways. For instance, predictions from the best overall model for a certain purpose can be combined with another model for the same purpose that excels at rare event detection. An analyst or data scientist could do experiments to determine the best weighting for the predictions of each model in a simple ensemble and partial dependency plots could be used to ensure that the inputs still behave monotonically w.r.t. the ensemble model predictions.

If you prefer or require a more rigorous way to combine model predictions, then super learners are a great option. Super learners are a specific implementation of stacked generalization introduced by Wolpert in the early 1990s. Stacked generalization uses a model to decide the weighting for the constituent predictions in the ensemble. Over fitting is a serious concern when stacking models. Super learners prescribe an approach for cross-validation and add constraints on the prediction weights in the ensemble to limit overfitting and increase interpretability. Figure 12 is an illustration of cross-validated predictions from two decision trees and a linear regression being combined by another decision tree in a stacked ensemble.

How do small, interpretable ensembles enhance trust and understanding?

Small, interpretable ensembles allow us to boost the accuracy of traditional trustable models without sacrificing too much interpretability. Increased accuracy is an indication that the pertinent phenomena in the data have been modeled in a more trustworthy, dependable fashion. Trust can be further enhanced by small, interpretable ensembles when models compliment each other in ways that conform to human expectations and domain knowledge.



Monotonicity constraints

Figure 13: A illustration of monotonic data and model constraints for neural networks. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

Monotonicity constraints can turn difficult to interpret nonlinear, non-monotonic models into highly interpretable, and possibly regulator-approved, nonlinear, monotonic models. Monotonicity is very important for at least two reasons:

Monotonicity is expected by regulators: No matter what a training data sample says, regulators may still want to see monotonic behavior. Consider savings account balances in credit scoring. A high savings account balance should be an indication of creditworthiness, whereas a low savings account balance should be an indicator of potential default risk. If a certain batch of training data contains many examples of individuals with high savings account balances defaulting on loans or individuals with low savings account balances paying off loans, of course a machine-learned response function trained on this data would be non-monotonic w.r.t. savings account balance. This type of predictive function could be unsatisfactory to regulators because it defies decades of accumulated domain expertise and thus decreases trust in the model.

Monotonicity enables consistent reason code generation: Consistent reason code generation is considered a gold standard of model interpretability. If monotonicity is guaranteed by a credit scoring model, reasoning about credit applications is straightforward and automatic. If someone's savings account balance is low, their credit worthiness is also low. Once monotonicity is assured, reasons for credit decisions can then be reliably ranked using the max-points-lost method. The max-points-lost method places an individual on the monotonic, machine-learned response surface and measures their distance from the maximum point on the surface, i.e. the ideal most creditworthy possible customer. The axis (e.g. independent variable) on which an individual is the farthest from the ideal customer is the most important negative reason code for a credit decision. The axis (e.g. independent variable) on which an individual is the closest to the ideal customer is the least important negative reason code for a credit decision, and other independent variables are ranked as reason codes between these two given the position of the individual and in relation to the ideal customer. Monotonicity simply ensures clear, logical reasoning using the max-points-lost method: under a monotonic model, an individual who was granted a loan could never have a lower savings account balance than an individual who was denied a loan.

Monotonicity can arise from constraints on input data, constraints on generated models, or from both. Figure 13 represents a process where carefully chosen and processed non-negative, monotonic independent variables are used in conjunction with a single hidden layer neural network training algorithm that is constrained to produce only positive parameters. This training combination generates a nonlinear, monotonic response function from which reason codes can be calculated, and by analyzing model parameter values, high degree interactions can be identified. Finding and creating such non-negative, monotonic independent variables can be a tedious, time-consuming, trial and error task. Luckily, neural network and tree-based response functions can usually be constrained to be monotonic be w.r.t any given independent variable without burdensome data preprocessing requirements. Monotonic neural networks often entail a custom architecture and constraints on the values of the generated model parameters. For tree-based models, monotonicity constraints are usually enforced by a uniform splitting strategy, where splits of a variable in one direction always increase the average value of the dependent variable in the resultant child node, and splits of the variable in the other direction always decrease the average value of the dependent variable in resultant child node.

How do monotonicity constraints enhance understanding?

Nonlinear, monotonic models are globally interpretable functions. They are capable of automatically generating reason codes and for certain cases, i.e. single hidden layer neural networks and single decision trees, important, high-degree variable interactions can also be determined.

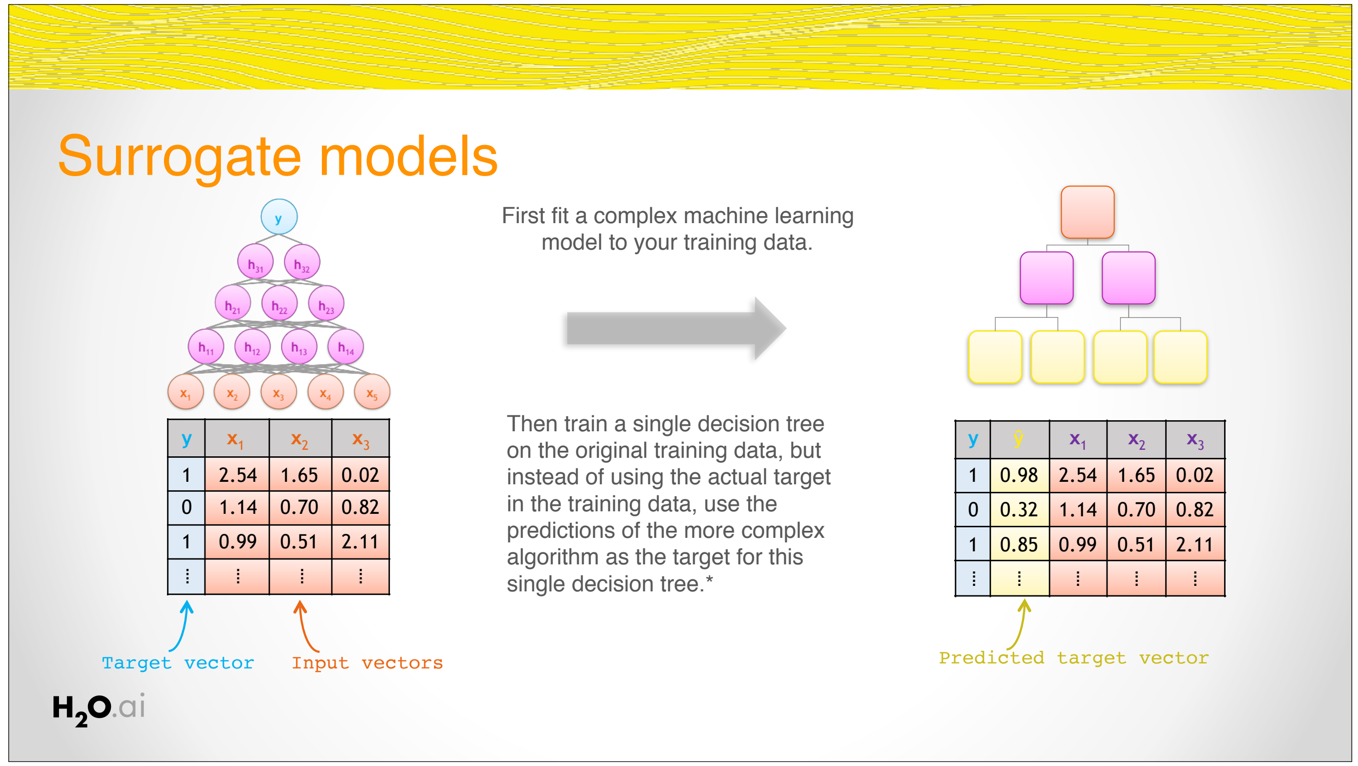
How do monotonicity constraints enhance trust?

Trust is increased when monotonic relationships, reason codes, and detected interactions are parsimonious with domain expertise or reasonable expectations. For neural networks, maximum activation analysis can be applied to determine if obviously different input observations activate different neurons in the network, and similar observations activate similar neurons. For decision tree ensembles maximum activation analysis is conducted by finding the lowest residual trees in the ensemble for a given input record. Such consistent activation patterns can increase trust. Sensitivity analysis can also increase trust in the model if results remain stable when input data undergoes minor perturbations and if the model changes in dependable, predictable ways over time.

Part 3: Understanding complex ML models

H2O.ai

The techniques presented in part 3 can create interpretations for nonlinear, non-monotonic response functions. They can be used alongside of techniques from part 1 and part 2 to increase the interpretability of all types of models. For the strictest internal documentation requirements and outside regulatory regimes, nonlinear, non-monotonic response functions may simply be too complex to explain. The techniques in part 3 are mostly meant to supplement the techniques in part 2 for increased interpretability in regulated models or to be used by analysts and data scientists working outside of highly regulated industry verticals. In either case, you might need to use more than one interpretability technique to derive satisfactory explanations for your models.



Surrogate models

Figure 14: An illustration of a decision tree surrogate model for explaining a complex neural network. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

A surrogate model is a simple model that is used to explain a complex model. Surrogate models are usually created by training a linear regression or decision tree on the original inputs of a complex model to be explained and the predictions of the model to be explained. Coefficients, variable importance, trends, and interactions displayed in the surrogate model are then assumed to be indicative of the internal mechanisms of the complex model to be explained. There are few, possibly no, theoretical guarantees that the simple surrogate model is highly representative of the more complex model.

What is the scope of interpretability for surrogate models?

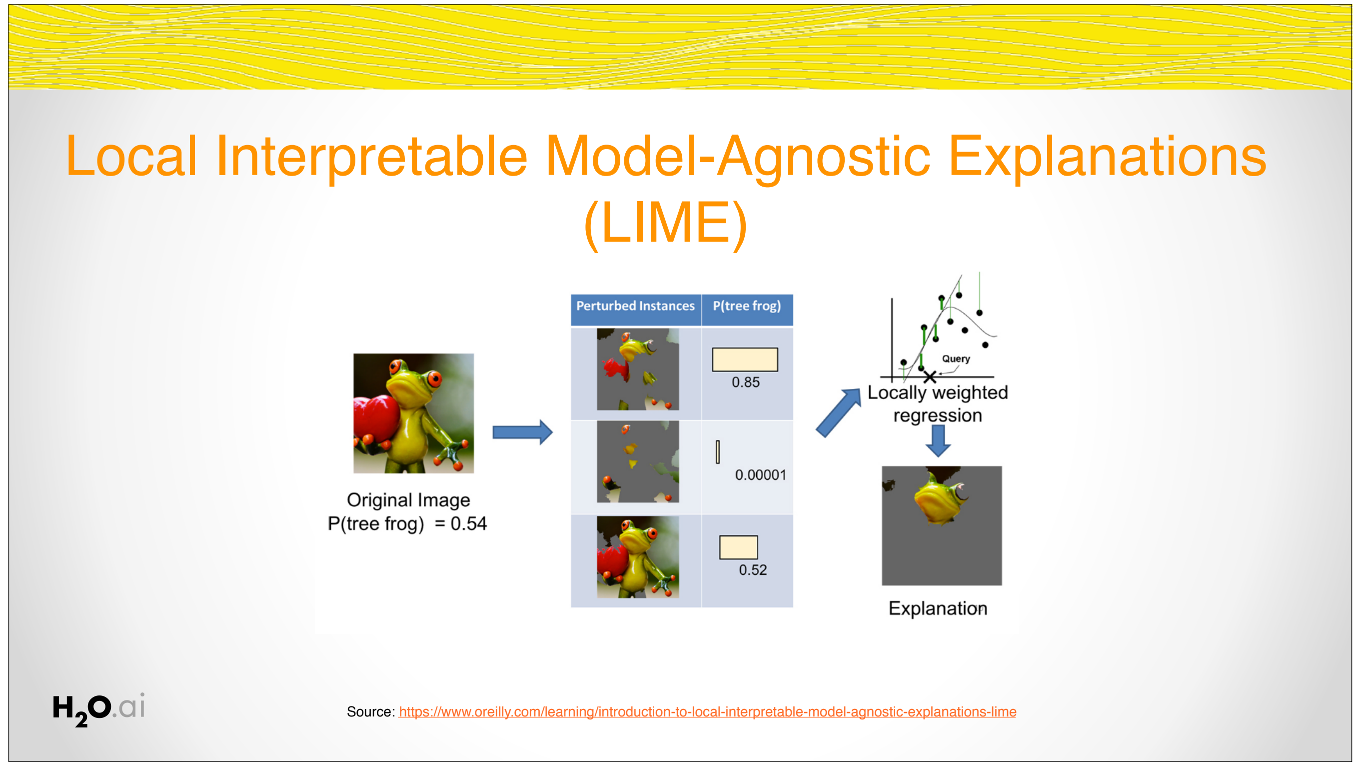
Generally surrogate models are global. The globally interpretable attributes of a simple model are used to explain global attributes of a more complex model. However there is nothing to preclude fitting surrogate models to more local regions of a complex model's conditional distribution, such as clusters of input records and their corresponding predictions, or deciles of predictions and their corresponding input rows. Because small sections of the conditional distribution are more likely to be linear, monotonic, or otherwise well-behaved, local surrogate models can be more accurate than global surrogate models. LIME is a formalized approach for local surrogate models. Of course, global and local surrogate models can be used together to foster both global and local interpretability.

How do surrogate models enhance understanding?

Surrogate models enhance understanding because they give insight into the internal mechanisms of complex models.

How do surrogate models enhance trust?

Surrogate models enhance trust when their coefficients, variable importance, trends, and interactions are in line with human domain knowledge and reasonable expectations of modeled phenomena. Surrogate models can increase trust when used in conjunction with sensitivity analysis to test that explanations remain stable and in line with human domain knowledge and reasonable expectations when data is lightly and purposefully perturbed, when interesting scenarios are simulated, or as data changes over time.



Local Interpretable Model-agnostic Explanations (LIME)

Image: https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime
Figure 15: An illustration of the LIME process in which a weighted linear model is used to explain a single prediction from a complex neural network.

LIME is a prescribed method for building local surrogate models around single observations. It is meant to shed light on how decisions are made for specific observations. LIME requires that a set of explainable records be found or simulated and these records are scored using the complex model whose decisions are to be explained. For a decision about a given record to be interpreted, the explanatory records are weighted by their closeness to that record, and an L1 regularized linear model is trained on this weighted explanatory set. The parameters of the linear model then help explain the prediction for the selected record.

LIME has been described most often in the context of explaining image or text classification decisions. It could certainly also be applied to business or customer data, for instance by explaining customers at every decile of predicted probabilities for default or churn or by explaining representative customers from well-known market segments.

What is the scope of interpretability for LIME?

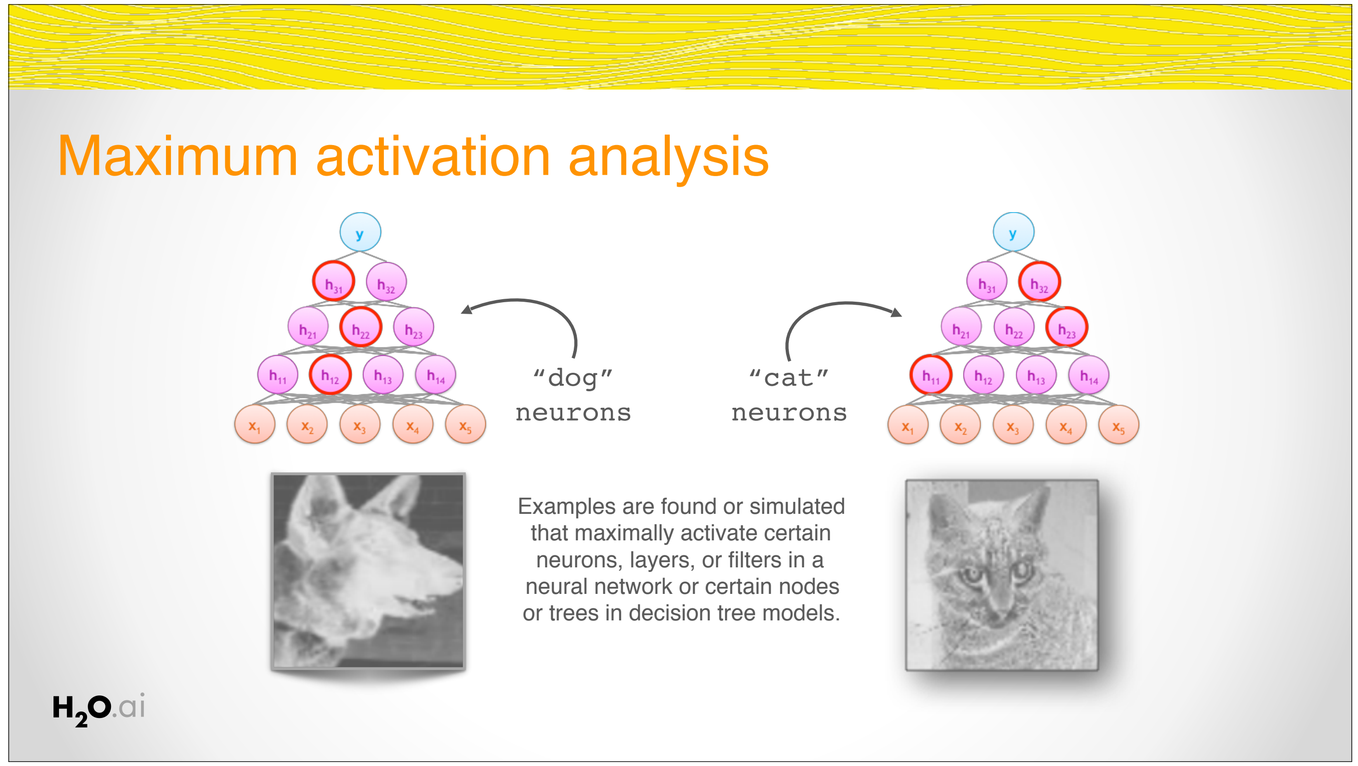
LIME is a technique for local interpretability.

How does LIME enhance understanding?

LIME provides insight into the important variables and their linear trends around specific, important observations, even for extremely complex response functions.

How does LIME enhance trust?

LIME increases trust when the important variables and their linear trends around specific records conform to human domain knowledge and reasonable expectations of modeled phenomena. LIME can also enhance trust when used in conjunction with maximum activation analysis to see that a model treats obviously different records using different internal mechanisms and obviously similar records using similar internal mechanisms. LIME can even be used as a type of sensitivity analysis to determine whether explanations remain stable and in line with human domain knowledge and expectations when data is intentionally and subtly perturbed, when pertinent scenarios are simulated, or as data changes over time.



Maximum activation analysis

Figure 16: Illustration of different inputs activating different neurons in a neural network. Image courtesy of Patrick Hall, Eight Tips for Making Machine Learning More Interpretable, 2016 Analytics Experience conference presentation.

In maximum activation analysis examples are found or simulated that maximally activate certain neurons, layers, or filters in a neural network or certain terminal nodes (or trees) in decision tree ensembles. Figure 16 is an illustration of a dog image causing high magnitude outputs from a different set of neurons than a cat image. For the purposes of maximum activation analysis, low residuals for a certain tree is analogous to high magnitude neuron output in a neural network. Maximum activation analysis elucidates internal mechanisms of complex models by determining the parts of the response function that specific observations or groups of similar observations excite to the highest degree, either by high magnitude output from neurons or by low residual output from trees. Maximum activation analysis can also find interactions when different types of observations consistently activate the same internal structures of a model.

What is the scope of interpretability for maximum activation analysis?

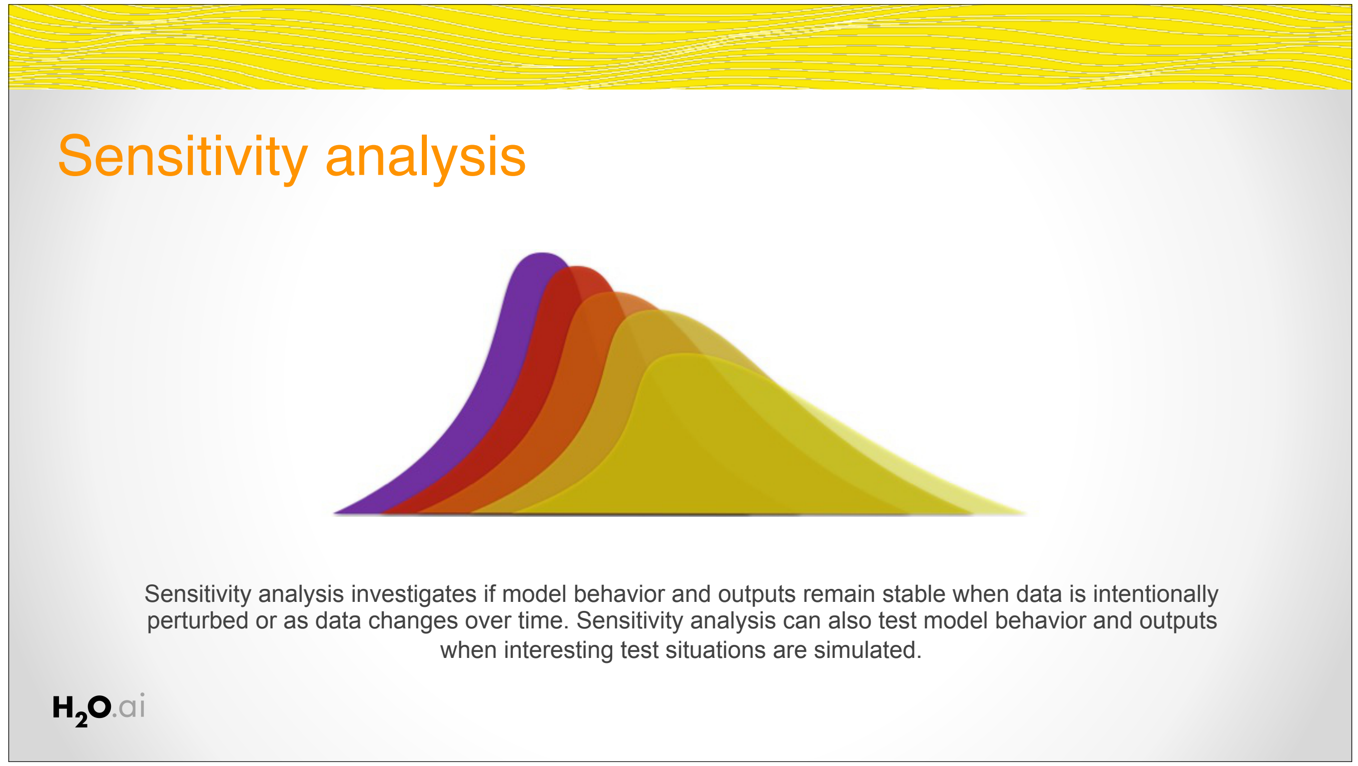
Maximum activation analysis is local in scope because it illustrates how certain observations or groups of observations are treated by discernible aspects of a complex response function.

How does maximum activation analysis enhance understanding?

It increases understanding because it elucidates the structure of complex models.

How does maximum activation analysis enhance trust?

Maximum activation analysis enhances trust when a complex model treats obviously different records using different internal mechanisms and obviously similar records using similar internal mechanisms. It enhances trust when found interactions match human domain knowledge or expectations. Maximum activation analysis also increases trust when used as a type of sensitivity analysis to investigate if internal treatment of observations remains stable when data is lightly perturbed, when interesting situations are simulated, or as data changes over time.



Sensitivity analysis

Figure 17: A representation of distributions changing over time.

Sensitivity analysis investigates if model behavior and outputs remain stable when data is intentionally perturbed or as data changes over time. Sensitivity analysis can also test model behavior and outputs when interesting test situations are simulated. Different techniques, including those presented in this essay and many others, can be used to conduct sensitivity analysis. Output distributions, error measurements, plots, and interpretation techniques can be used to explore the way models behave in important scenarios, how they change over time, or if models remain stable when data is subtly and intentionally corrupted.

What is the scope of interpretability for sensitivity analysis?

Sensitivity analysis is a global interpretation technique 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.

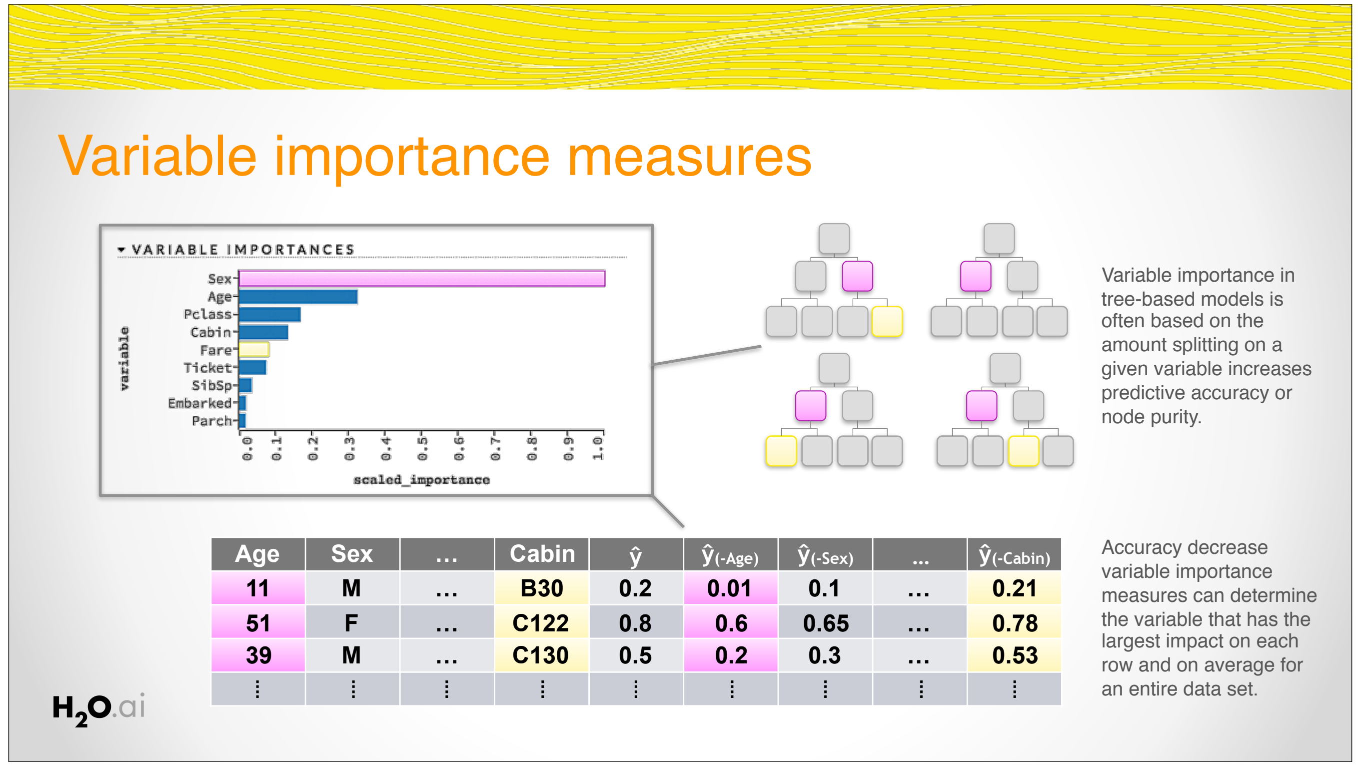
Sensitivity analysis is a local interpretation technique when local interpretation techniques are used, for instance using LIME to determine if the important variables in a credit allocation decision remain stable for a given customer segment under macroeconomic stress testing.

How does sensitivity analysis enhance 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 changes over time.

How does sensitivity analysis enhance trust?

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.



Variable importance measures

For nonlinear, non-monotonic response functions, variable importance measures are often the only commonly available quantitative measure of the machine-learned relationships between independent variables and the dependent variable 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.

Tree-based variable importance measures
alt text
Figure 18: An illustration of variable importance in a decision tree ensemble 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 a single decision tree, a variable's importance is quantitatively determined by the cumulative change in the splitting criterion for every node in which that variable was chosen as the best splitting candidate. For a gradient boosted tree ensemble, variable importance is calculated as it is for a single tree but aggregated for the ensemble. For a random forest, variable importance is also calculated as it is for a single tree and aggregated, but an additional measure of variable importance is provided by the change in out-of-bag accuracy caused by shuffling the independent variable of interest, where larger decreases in accuracy are taken as larger indications of importance. (Shuffling is seen as zeroing-out the effect of the given independent variable in the model, because other variables are not shuffled.) For neural networks, variable importance measures are typically associated with the aggregated, absolute magnitude of model parameters for a given variable of interest. 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.

Leave-One-Covariate-Out (LOCO)
alt text
Figure 19: An illustration of the general LOCO approach.

A recent preprint article put forward a model-agnostic generalization of mean accuracy decrease variable importance measures called Leave-One-Covariate-Out or LOCO. LOCO creates local interpretations for each row in a training or unlabeled score set by scoring the row of data one time for each input variable (e.g. covariate) in the row. In each scoring run, one input variable is set to missing, zero, it's 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. LOCO creates global variable importance measures by estimating the mean change in accuracy for each variable over an entire data set and even provides confidence intervals for these global estimates of variable importance.

What is the scope of interpretability for variable importance?

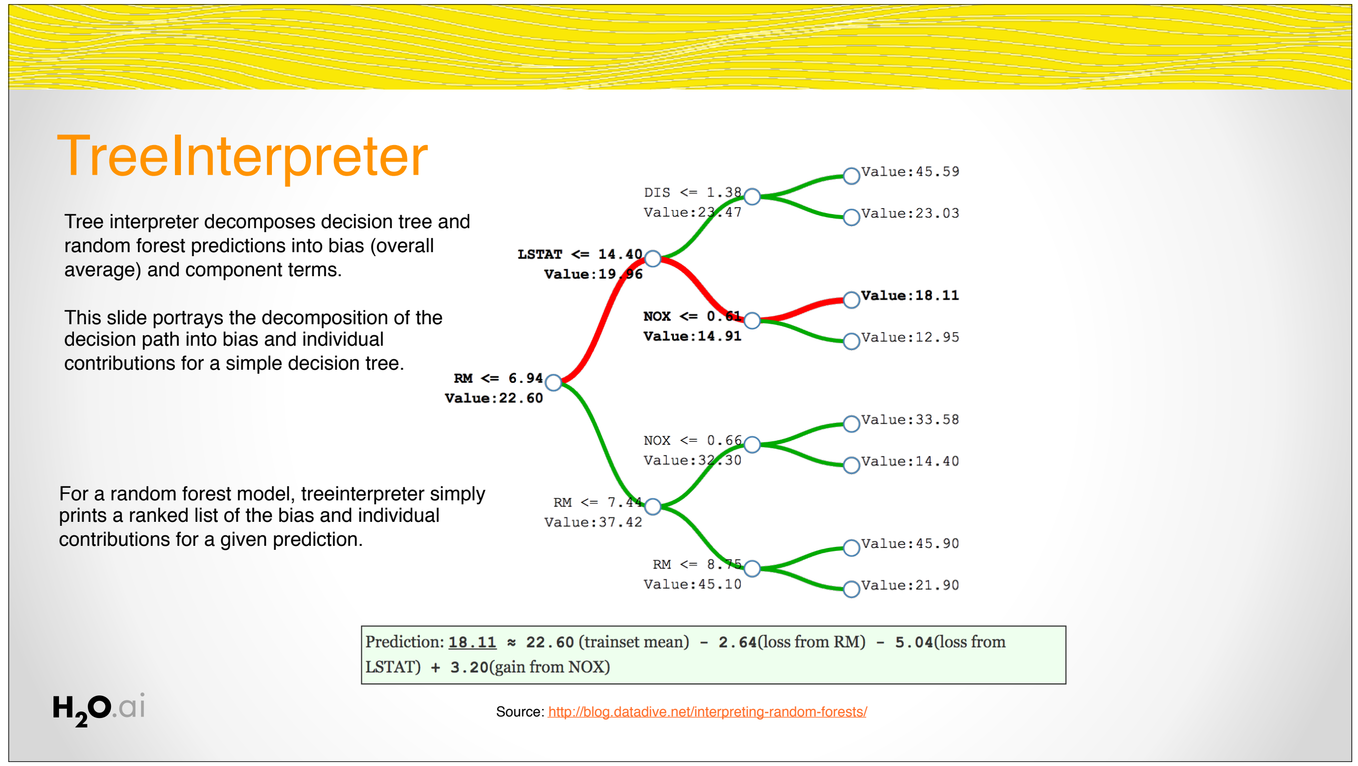
Variable importance measures are usually global in scope, however the LOCO approach offers local variable importance measures for each row in a training set or in new data.

How does variable importance enhance understanding?

Variable importance measures increase understanding because they tell us the most influential variables in a model and their relative rank.

How does variable importance enhance trust?

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.



TreeInterpreter

Image: http://blog.datadive.net/interpreting-random-forests/
Figure 20: A single decision tree with a highlighted decision path.

Several average tree interpretation approaches have been proposed over the years, but the simple, open source package known as treeinterpreter has become popular in recent months. Treeinterpreter decomposes decision tree and random forest predictions into bias (overall training data average) and component terms from each independent variable used in the model. Figure 19 portrays the decomposition of the decision path into bias and individual contributions for a simple decision tree. For a random forest model, treeinterpreter simply prints a ranked list of the bias and individual contributions for a given prediction.

What is the scope of interpretability for treeinterpreter?

Treeinterpreter is global in scope because it represents average contributions of independent variables to an overall decision tree or random forest model prediction.

How does treeinterpreter enhance understanding?

Treeinterpreter increases understanding by displaying average, ranked contributions of independent variables to the predictions of decision tree and random forest models.

How does treeinterpreter enhance trust?

Treeinterpreter enhances trust when displayed model contributions conform to human domain knowledge or reasonable expectations. Tree interpreter 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.

Questions?