WHITE PAPER

Explainable Artificial Intelligence for Anti-Money Laundering

Contents

Artificial intelligence is driving innovation in all sorts of industries, the anti-financial crime industry being no exception. Specifically, machine learning (ML) is a subfield of applied AI that is driving most of this innovation. However, anti-money laundering (AML) efforts have been somewhat more cautious in the adoption of these techniques relative to anti-fraud efforts. To those outside the anti-financial crime industry, it might not be intuitive why this is. After all, both ultimately involve sniffing out efforts to disguise illegal financial activity. The critical difference between the two is the level of governance that drives AML efforts. Financial institutions have a natural profit motive to protect themselves and their customers from fraud, whereas government regulation is the primary driver of AML initiatives.

This presents an additional hurdle for practitioners: An automated AML system must not only catch suspicious activity; it must also be able to explain why it did so in a way that is satisfying to a regulator. ML techniques have promise to drive significant innovation in the AML industry, but are often opaque black boxes, and thereby hard to justify to regulators. This obstacle has historically led financial institutions to understandably hesitate from investing in ML for their AML programs.

This problem is not unique to the AML industry. As ML techniques have become mainstream across multiple industries, there has become a subsequent realization of the need to explain these models better in order to trust them enough to be implemented. This field is often referred as “explainable artificial intelligence” (XAI). Advances in XAI are worth laying out, because it opens up new opportunities in AML that may have previously been disregarded due to these concerns of explainability.

On Dec. 3, 2018, in a statement titled *Joint Statement on Innovative Efforts to Combat Money Laundering and Terrorist Financing*, the various US government agencies responsible for regulating AML offered a cautious endorsement for financial institutions to “responsibly implement innovative approaches” to augment their AML efforts. This statement is generally understood across the AML industry to signal that regulators intend to be more open minded on the use of AI/ML technologies in the future. Although not directly stated, it is inferred that these XAI techniques are a critical component of the governance aspect of implementing AI/ML models in AML.

In this paper, we explore some of the XAI techniques that are already in use or currently trending, and help build a reader's understanding of some of the options that could be used in a real implementation of an ML model for an AML use case. The options explored in this paper assume a use case of building either a transaction monitoring alert generation model or an alert prioritization/scoring/hibernation model for AML. In data science terms, we are essentially assuming that we are working to explain a supervised classification model.

What Is Meant That Machine-Learning Models Are a Black Box?

Most of the popular machine-learning algorithms driving headlines, such as random forests, gradient boosting and neural networks, are often casually referred as being "black boxes." Despite what the phrase may imply, it is not to say that the inner workings of the models cannot be seen, so much as that the logic produced by these models are so complicated that they might as well be a black box. Just as it is impractical to fully understand a large table of data by reading every row, it is similarly impractical to understand many machine-learning models by reading their scoring code. Ironically, the models we develop to make sense of big data become big data themselves in the sense that the workings of these models are too much information to understand outright.[1]

Additionally, there are statistical techniques (sometimes called ML models, sometimes not) that are useful for making predictions, but whose inner machinations are simple enough that we do not need to reach into the XAI toolbox to understand their output. This paper will refer to these as "white box statistical techniques" and they will be explained first because some of them are even used in certain XAI methods. These models fall somewhere in between rule-based scenarios and black box models in both their predictive power and their complexity.

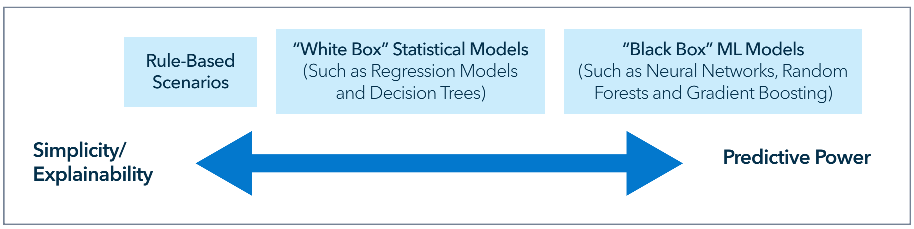


Figure 1: A visualization of the spectrum of AML modeling approaches between simplicity and predictive power.

The solutions offered by XAI typically work by systematically testing different inputs to a model, recording the changes in output, and then summarizing it either visually or statistically. XAI techniques summarize the workings of a model much in the same way various charts and descriptive statistics (like bar charts and averages) can be used to summarize the content of a data table. This analogy extends to why this paper recommends using a toolbox approach to explaining models (as opposed to a standardized approach), as the right time to use each is situational. XAI techniques can be broken up into two categories, though some of the techniques this paper will discuss cross-over into both:

1. **Global Explanations.** These tell you what variables (known as features) affect the model the most and generally what the impact is when determining a score.
2. **Local Explanations.** These narrow in on a specific observation and attempt to explain why a particular decision was made for that instance.

[1] SAS® products used for developing machine-learning models, such as SAS® Enterprise Miner™ and SAS Visual Data Mining and Machine Learning, are transparent about the parameters and methodologies used to generate their models. That is, the model development, comparison and selection processes in these products are NOT a black box. However, the models they generate may still be a black box (depending on the algorithm being applied in the final model) for the reasons mentioned here.

From within white box statistical techniques, this paper will review regression models and decision trees. From global explanations, this paper will discuss global feature importance, partial dependence (PD) plots and surrogate models. From local explanations, this paper will discuss risk factor reporting, individual condition expectation (ICE) plots, locally interpretable model-agnostic explanations (LIME) and SHapley Additive exPlanations (SHAP).

White Box Statistical Techniques

While more complex models (such as random forests and neural networks) are seen as more cutting edge, it is perfectly acceptable to choose a less complex model in the right situation. Regression models and decision trees are powerful, traditional statistical models that generate output that is human readable.

The disadvantage of these models is they generally do not capture the same level of complexity as black box models. Specifically, white box models may sometimes struggle to adequately represent complex nonlinear effects and interactions (when the effects of input variables depend on other variables). If the underlying pattern of the data is simple, then white box models are recommended to capture it due to their superior transparency.

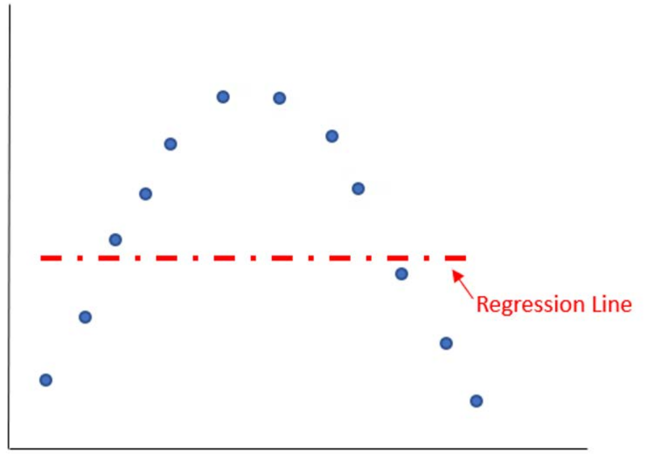


Figure 2: To demonstrate a known limitation of regression models, a type of white box model, the above illustration is an example of a regression model that has been misspecified (i.e., there is an error in the model design). When capturing a relationship between two variables, a linear regression model will draw a straight line to minimize the distance between the line and the data points. As shown above, this approach will not work as intended if the actual relationship between the variables is quadratic. A regression model can be programmed to account for this, but only if this problem is correctly identified, and this is just one example of how a regression model can be misspecified.

Decision Trees

How it works

Using the inputs provided, decision trees identify splitting rules that most efficiently separate a target variable (in this case a decision to alert or not).

How to interpret it

To understand why a decision was made for a specific observation, one can follow the rules shown in the tree. Alternatively, a decision tree can be thought of as a sequence of conditional (if-then) statements.

Why to use it/why not to use it

Decision trees are generally regarded as easily understandable by users of varying backgrounds. However, sometimes this model type does not fit the pattern of the data and other models typically outperform it.

Example of this technique in SAS® software

Figure 3: The middle section of a decision tree generated by SAS Visual Data Mining and Machine Learning. The “tree” is essentially a visualization of a set of rules. If we start from the top right, we can see we have a rule based on transaction medium. In this example, cash or check transactions are much more likely to be alerted, whereas wires are more muddled and require us to follow rules further down the tree to make a decision.

Regression Models

How it works

There are many different varieties of regression models, but they all work by attempting to create a line of best fit; minimizing the distance between individual data points and the line. Logistic regression is another type of regression model relevant for AML, specifically because it is used to estimate the likelihood of an event (such as productive alert vs. false positive alert).

How to interpret it
These models output a coefficient for every variable in the model (often called beta parameters), which we can use to weight each variable value in our calculation of a score (plus an intercept) for an estimate, with some additional adjustments needed for different variations of regression models. For example, a logistic regression attempts to predict the log of the odds (or log-odds) of an event, as opposed to a continuous number as in a linear regression, so that changes the interpretation of the coefficients somewhat. To illustrate this, below are some example formulas used to translate the coefficients into the model's prediction for an observation.

Linear regression coefficient interpretation formula (the estimate, ŷ, is a continuous number):

$$\hat{y} = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

Logistic regression coefficient interpretation formula (the estimate, P, is a probability between 0 and 1):

$$P = \exp(\beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k) \ / (1 + \exp(\beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k))$$

Why to use it/why not to use it
In their raw form, regression coefficients are not intuitive for someone who is not already versed in statistics, but their relationship with a model's decision can be broken down in a way that would be satisfactory to a regulator or reshaped into something more interpretable if desired for end users (like a credit score). As we will see later, regressions are utilized a lot in XAI techniques.

Example of this technique in SAS® software

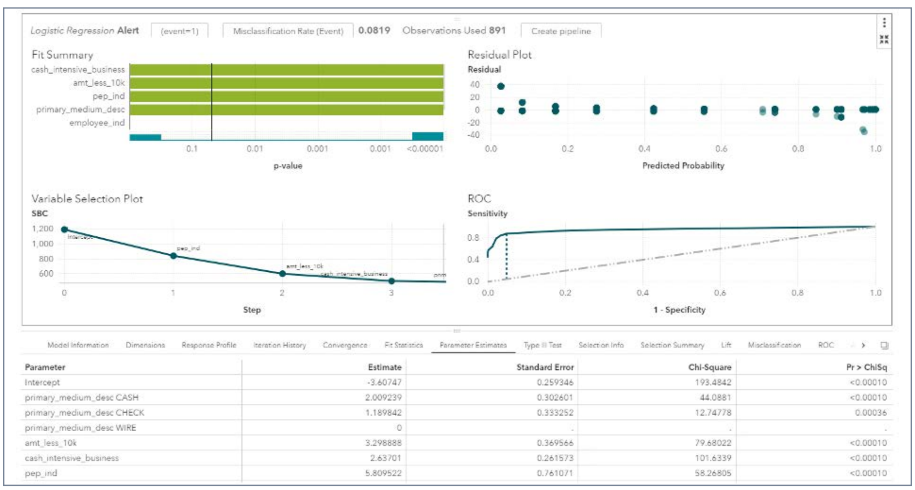


Figure 4: The summary statistics of a logistic regression model in SAS Visual Data Mining and Machine Learning. The column Estimate in the bottom table shows us the coefficient estimates for each variable in the model. From the above section, you can apply the equation given for logistic regression to calculate what the predicted probability would be for an individual observation. P = exp(-3.6 + 2X1 + 1.2X2 + 3.3X3 + 2.6X4 + 5.8X5) / (1 + exp((-3.6 + 2X1 + 1.2X2 + 3.3X3 + 2.6X4 + 5.8X5)). Note that there are two coefficients used to represent primary_medium_desc because this is a categorical variable with three different possible values, so the model uses a pair of binary "dummy" variables to represent which category the observation belongs to.

Black Box Explanations

Global Explanations

Global explanations give us a sense of the big picture. They tell us which features contribute the most and what their impact is to model predictions in general.

Global Feature Importance

How it works

Global feature importance refers to a family of different methods that seek to rank input features by their contribution to the model's predictions. The formula to be used depends in part on the ML algorithm being applied. For example, one method that SAS provides for a random forest (an ML model made of multiple decision trees) is called random branch assignments, which evaluates the importance of a feature on how often samples traverse a splitting rule based on that feature.

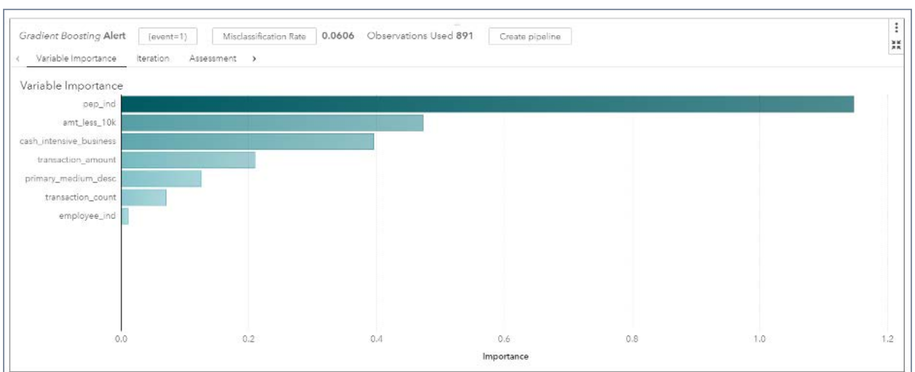
How to interpret it

In a relative sense, feature importance indicates which features had the biggest impact on the model's predictions. Provided importance values are not intended to have any interpretation beyond as a comparison of the importance of the features in the same model relative to each other.

Why to use it/why not to use it

This metric provides a quick way to tell which features the model is most reliant upon. However, it does not describe how a specific feature affects the model's estimates. In the below example, we can tell that pep_ind is the most important feature, but we cannot tell whether it increases or decreases the calculated prediction, or by how much.

Example of this technique in SAS® software



Variable	Importance	Standard Deviation
pep_ind	1.1473	12.2516
amt_less_10k	0.4731	7.1733
cash_intensive_business	0.3958	5.1575
transaction_amount	0.2097	0.5804
primary_medium_desc	0.1255	0.5511

Figure 5: the Variable Importance plot showing the most important features in a gradient boosting model in SAS Visual Data Mining and Machine Learning. In this example, we can see pep_ind is the most important feature in this model and would likely reduce the model's accuracy if removed from the model. On the other hand, employee_ind is not as important, and would be less likely to reduce the model's accuracy if removed.

Partial Dependence (PD) Plots

How it works

Generally speaking, XAI techniques work by testing different inputs into a model, and drawing conclusions from the changes in output. Similarly, PD plots are generated by taking a sample of data, permuting by a variable of interest, and plotting the average at each possible value that the variable could take.

How to interpret it

PD plots display the average prediction of the model given a certain value for a feature.

Why to use it/why not to use it

At present, PD plots are the best way to get a global explanation that gives a specific answer as to how changing a value for a variable affects model output. We should caution that PD plots are still generalizing the way the underlying model works. PD plots are great for showing nonlinear effects, but interaction effects aren't always shown unless specifically accounted for. We can attempt to account for this by creating PD plots that visualize the effect of altering multiple features at once, although there is always some level of generalization of the full model's behavior and a limit to how much can be displayed in a manner that is easy to understand. Interpretation may become challenging for interaction complexity that exceeds two inputs.

Example of this technique in SAS® software

Figure 6: PD plot from SAS Visual Data Mining and Machine Learning showing the average prediction when pep_ind is flagged or not. In this example, we can see that the model on average predicted the probability of the event to be around .8 when pep_ind = 1 and around .3 when pep_ind = 0, demonstrating the marginal impact of this variable on predictions.

Surrogate Models

How it works
Surrogate models are white box models (as explained in an earlier section of this paper) built from the same data as the prediction model but using the prediction model's output as its target. In this setup, a more complex black box style model is used to generate the predictions, and a simpler white box statistical model sits on top to explain what the prediction model is doing.

How to interpret it
Surrogate models are interpreted identically to the white box models that serve as their foundations (see the earlier section on white box models). The only difference is the surrogate is explaining the outputs of the prediction model.

Why to use it/why not to use it
The surrogate model will not completely emulate the behavior of the prediction model it explains (if it does then it begs the question why the surrogate is not used as the underlying prediction model in the first place) but it can help draw some generalizations on the underlying logic that the prediction model is using. There is a lot of flexibility to this approach, as you can use whatever white box model the end users are most comfortable with. Surrogate models provide both global and local explanations.

Example of this technique in SAS® software

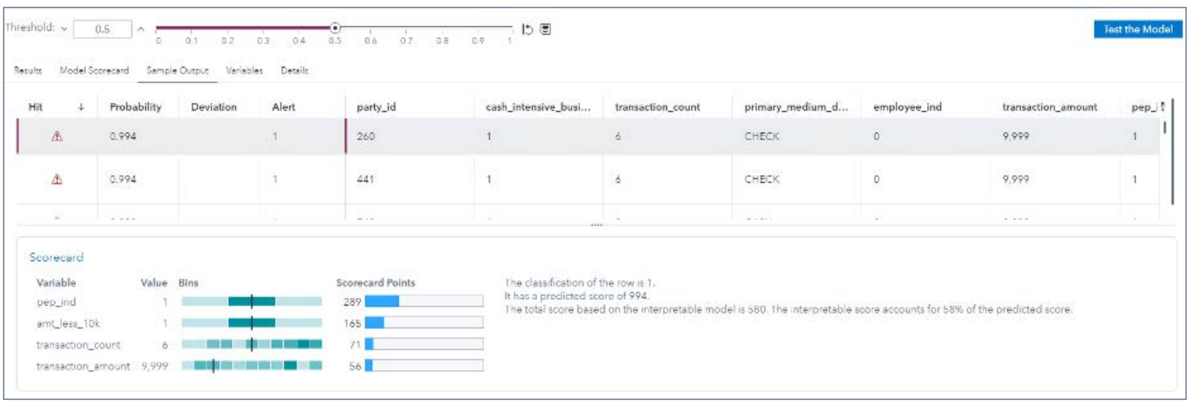


Figure 7: Out of the box, SAS Adaptive Learning and Intelligent Agent System uses a logistic regression surrogate model to explain the prediction model, with a scorecard used to explain the coefficients of the surrogate regression model. This scorecard works the same as a credit score, where different points are assigned for a feature's value belonging to a certain bin. As previously stated, some nuance of the prediction model's inner workings are lost in the translation to the simpler regression model, with an estimate of the amount explained also shown (58% in this observation).

Local Explanations

While global explanations are helpful for documentation and general understanding of the model, investigators usually desire hints as to why the model determined a certain score for a specific case assigned to them (i.e., why did the model determine this activity as suspicious). For this, it is not quite good enough to have global explanations.

Risk Factor Reporting

How it works

Rule-based scenarios have been an industry standard in AML for a long time. The "features" in a machine-learning model are something akin to a scenario. Therefore, explaining why a specific decision was made can be as simple as reporting which features, or risk factors, were triggered. If there are a lot of related risk factors, some additional aggregation might be done as well.

How to interpret it

Most organizations already have their own ideas of what risk factors they want to monitor. Given that this is not a statistics-based approach, this is more about having a conversation about what types of activity the organization is interested in monitoring and codifying them into rules.

Why to use it/why not to use it

The advantage of this approach is that it's straightforward, and its disadvantage is the lack of quantified detail it provides in breaking down how much each feature contributed to a decision. From a compliance perspective, this might be "good enough," although some of the other approaches listed here might add more value to different groups of users looking for more information.

Individual Conditional Expectation (ICE) Plots

How it works

ICE plots are a lot like partial dependence plots, in that they show how adjusting an individual variable affects the score in a model. The key difference is partial dependence plots show us the average effect of adjusting a feature, whereas the ICE plots are totally within the context of an individual data point. ICE plots assume that all other features of the model are held constant, with only a feature of interest being adjusted.

How to interpret it

We can infer that if all the other features are held constant, the ICE plot shows us the expected change in the prediction if we altered that feature.

Why to use it/why not to use it

ICE plots are useful primarily in situations when we want to know how an individual prediction would have changed if a feature were altered. ICE plots share many of the same limitations as PD plots. For example, it may be hard to capture all of the interaction effects impacting the model's decision.

Example of this technique in SAS® software

Figure 8: In blue is the ICE plot for a single transaction, in yellow is the PD plot as shown in SAS Visual Data Mining and Machine Learning. We can tell from our ICE plot that pep_ind had an impact on this observation's decision (around .6 probability when pep_ind = 0 vs .8 when pep_ind = 1). The PD plot shows a much bigger impact for pep_ind. This demonstrates how the global explainer (PD) can diverge somewhat with the observation-specific local explainer (ICE).

Locally Interpretable Model-Agnostic Explanations (LIME)

How it works

LIME is another technique using a linear regression as a surrogate model. Its innovation is that it permutes different values to test the scores of the model, while weighting examples closer to the focal observation higher.

How to interpret it

Since LIME is a linear regression surrogate, the output are regression coefficients. Its interpretation works essentially the same as that of a regression-based surrogate model, just with different assumptions on what the sample data looks like.

Why to use it/why not to use it

A common criticism of regression surrogates is that attempting to explain a nonlinear model with a linear one is not a reliable way to capture important relationships in the model. LIME tries to get around this by assuming that explanations are still linear at a local level.

LIME has been shown to be useful for identifying unintended model behavior, but there are a lot of problems with it if the intention is to meet a compliance requirement. Primarily, it is difficult to define what should count as the local space, and not entirely clear if that space can truly be defined by a linear model, as is assumed by LIME. Explanations given by LIME for two points that are seemingly close together can also be wildly different, which makes it hard to fully trust the explanations given. With that said, we would have been remiss not to mention it, and it has demonstrated practical usefulness for data scientists.

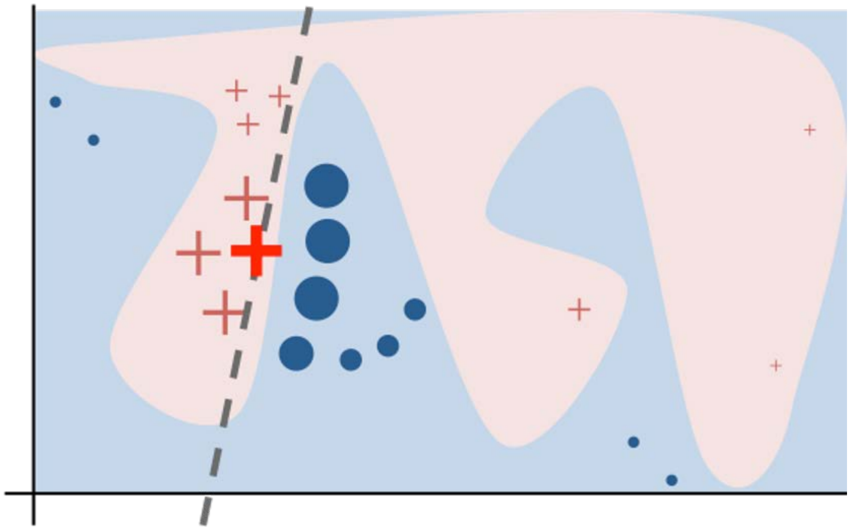


Figure 9: This illustrates the local effect LIME attempts to model. The pink/blue sections represent the decisions made by the complex black box model. The bold red cross is the observation being explained by LIME. LIME will sample other observations that are close by and use that to train a linear regression model that draws the local boundary.

Illustration from kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf

Example of this technique in SAS® software

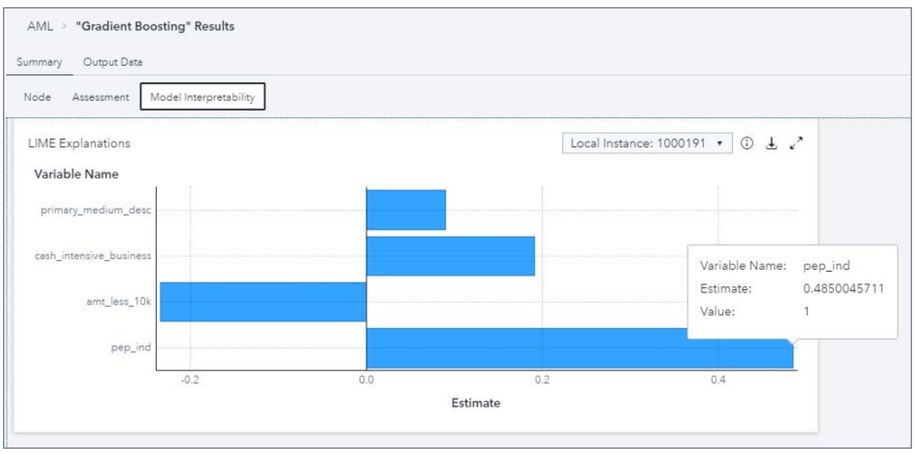


Figure 10: In this sample LIME output, shown in SAS Visual Data Mining and Machine Learning, pep_ind increased the estimate of the probability of a prediction by 48.5% in the local space.

SHapley Additive exPlanations (SHAP)

How it works

The goal of SHAP is to find the contribution of each feature to the final prediction of an observation. This is done by testing the effect of removing a feature from the model on the model's output. To account for different interaction effects, many or all possible subsets of the different features may be tested and their output is summarized with a weighted average.

How to interpret it

An estimate provided by SHAP should be thought of as the change in the prediction provided by the information given to us by the inclusion of a feature. If we added the estimates for all the features together plus the estimated average predicted value (an intercept), this will result in the prediction in the focal observation. Positive estimates contributed to the conclusion that the event occurred, and negative estimates contributed to the conclusion that the event did not occur.

Why to use it/why not to use it

SHAP is a cutting-edge explanation technique and it ties together concepts from other techniques previously mentioned in this paper. SHAP is generally thought to be a more robust option compared to some other local explainers, such as LIME. It is primarily intended for use in local explanations, but it can be used for global explanations as well. Unfortunately, SHAP's glaring disadvantage is that running all these different tests for different subsets of features is computationally intensive, and it gets exponentially worse for each feature added to the model. There are different implementations of this technique that help mitigate this problem, but this is an issue that could be a limitation for a model with a large number of features.

Example of this technique in SAS®

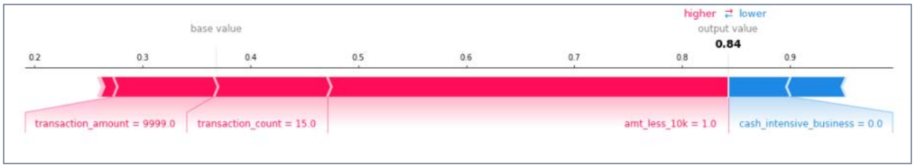


Figure 11: SAS provides integration to open source languages such as Python. This figure is a visualization of SHAP provided by Python's shap package. In this example, we can see that the model assumes a 37% probability (given by "base value") before considering the features in the model, and arriving at 84% as its final estimate (output value). This example seems to be an example of structuring, where the combination of transaction_amount, amt_less_10k and transaction_count were important features in the model for determining a high probability of alerting this observation for money laundering. Conversely, cash_intensive_business was a feature that led the model to estimate a lower probability.

Summary

Although ML models have traditionally not been used in AML, that has been changing recently due to loosening restrictions by regulatory agencies and advancements in the ability to adequately explain their output. This paper reviewed many of the prominent techniques either in popular use or gaining traction. After reading this paper, a reader should have built a high-level understanding of the options for interpreting an ML model used in an AML implementation. Below is a table briefly reviewing the techniques discussed.

Method Discussed	White Box Model or Black Box Model Explainer	Global Explainer or Local Explainer
Regression	White Box Model	Both
Decision Tree	White Box Model	Both
Global Feature Importance	Black Box Model Explainer	Global
Partial Dependence Plots	Black Box Model Explainer	Global
Surrogate Models	Black Box Model Explainer	Both
Risk Factor Reporting	Black Box Model Explainer	Local
Individual Conditional Expectation (ICE) Plots	Black Box Model Explainer	Local
Locally Interpretable Model-Agnostic Explanations (LIME)	Black Box Model Explainer	Local
SHapley Additive exPlanations (SHAP)	Black Box Model Explainer	Both

Figure 12: A review of methods discussed in this paper. Note that for White Box Models, they can be thought of has having both global and local explanations, considering they don’t need any XAI techniques applied to them to be explained.

To learn more about SAS solutions, please visit: sas.com