MOST AI EXPLAINABILITY IS SNAKE OIL. OURS ISN'T.

By Jay Budzik, Chief Technology Officer, ZestFinance

Advanced machine learning (ML) is a subset of AI that uses more data and sophisticated math to make better predictions and decisions. It's what powers self-driving cars, Netflix recommendations, and a lot of bank fraud detection. Banks and lenders could make a lot more money using ML on top of legacy credit scoring techniques to fnd better borrowers and reject more bad ones. But adoption of ML has been held back by the technology's "black-box" nature. ML models are exceedingly complex. You can't run a credit model safely or accurately if you can't explain its decisions.

What we really want from AI explainability are three things: consistency, accuracy, and performance. A handful of techniques are being promoted in the market claiming to solve this "black-box" problem. Caveat emptor. The techniques are often inconsistent, inaccurate, computationally expensive, and/or fail to spot unacceptable outcomes such as race- and gender-based discrimination. Some of these techniques have been in use for a long time, and are quite effective on old modeling approaches like logistic regression. But they don't work well with ML.

✕

SOME BIG COMPANIES' EXPLAINABILITY TECHNIQUES HAVE SERIOUS FLAWS: THEY'RE INACCURATE, INCONSISTENT, AND SLOW

At ZestFinance, we've built new kinds of explainability math into our ZAML software tools that quickly render the inner workings of ML models transparent from creation through deployment. You can use these tools to monitor model health in run-time. You can trust the results are fair and accurate. And we've automated the reporting so all you have to do is push a button and produce all the documentation required to comply with regulations.

In this paper, we compare a few popular explainability techniques with our ZAML technology, describe their relative strengths and weaknesses, and show how each technique stacks up on a simple modeling problem. The test we ran showed that ZAML is more accurate, consistent, and faster than any of these other approaches.

FOUR KINDS OF EXPLAINABILITY TECHNIQUES EXPLAINED

Three common machine learning explainability techniques being championed today are known as LOCO, permutation impact, and LIME. LOCO, which stands for "leave one column out," substitutes "missing" for a variable and recomputes the model's prediction. The idea is that if the score changes a lot, the variable that was left out must be really important. Permutation impact (PI), also called permutation importance, substitutes a variable with a randomly selected value and recomputes the model's prediction. As with LOCO, the idea is that if the score changes a lot, the variable that was scrambled must be really important. LIME, which stands for local interpretable model-agnostic explanations, fits a new linear model in a local neighborhood around a given applicant's real data values and the real model's score for those synthetic values. It then uses this new, linear approximation of the actual model to explain how the more complex model behaves. Essentially, you're taking a very complex model and pretending it's simple so you can explain it. ZestFinance's ZAML software uses a proprietary explanation method derived from game theory and multivariate calculus that works on the actual underlying model. ZAML explainability determines the relative importance of each variable to the final score by looking at how it interacts with other variables. The checklist below compares four explainability techniques on seven important capabilities. We also added scores for each of the techniques from a test we ran on their accuracy, consistency and speed. Accuracy is measured by average rank correlation, or how far off the explanations are from the ground truth baselines. Consistency is measured by variance, or how much the difference in explanations varies across all applicants. Speed was the time it took to generate explanations for all the rows in the dataset.

This chart shows how common explainability methods (defined below) compare with ZAML on core attributes, along with the results of a side-by-side explainability test we ran on two common ML models.

Does Your Explainability Method...?	Why it Matters	LOCO	LIME	PI	ZAML
Run just as fast as the number of model variables rises.	Speed	✗	✓	✗	✓
Analyze the inner workings of the model, not just inputs and outputs'	Accuracy	✗	✗	✗	✓
Analyze the final model, not a refitted or proxy model.	Accuracy	✗	✗	✓	✓
Analyze in a multivariate manner, not just univariate.	Accuracy	✗	✓	✗	✓
Reflect global not local feature importance.	Accuracy	✓	✗	✓	✓
Make itself user-friendly and not require expert tuning	Consistency	✓	✗	✓	✓
Use actual data, not synthetic or randomly shuffled data?	Consistency	✓	✗	✗	✓
Accuracy in explaining: * XGBoost Neural Network		90% 96%	82% 92%	84% 93%	**99%** **100%**
Variance (consistency at explaining): XGBoost Neural Network		0.02 0.004	0.07 0.01	0.03 0.009	**0.002** **0.0002**
Explanation time (sec) * XGBoost Neural Network		0.003 0.005	35 49	0.02 0.04	**0.006** **0.26**

* Based on an evaluation of the performance of explainability techniques on two popular ML models trained on a simple, two-variable "Moons" dataset. Results may not be universally generalizable. Accuracy = correlation error. Consistency = variance.

LET'S UNPACK THIS CHECKLIST

Explanations of ML models have to meet a variety of requirements to be trustworthy for high stakes business applications such as credit risk modeling, chief among them accuracy, consistency, and performance, i.e., prediction speed. The chart above summarizes how the four techniques you may be hearing about stack up on an extremely simple model. For real-world explainability problems, such as those endemic to credit risk modeling, the performance differences are even more pronounced. We used a simple model to demonstrate these differences to make a point - even a model with just two variables can't be accurately explained accurately by LOCO, PI, or LIME. ZAML explainability methods can. If an explainability technique can't even explain the simplest of "toy" models, is it wise to trust it on a real-world application where large dollar amounts are at risk?

What we really want from AI explainability are three things: consistency, accuracy, and performance. LOCO, LIME, and PI all have shortcomings on one or more of these attributes. LOCO and PI work by analyzing one feature one at a time. This means as the number of model variables goes up, the algorithms become slower and more expensive to run. LOCO, LIME, and PI look only at the inputs and outputs of the model, which means they have access to much less information than explainers such as ZAML's that look at the internal structure of the model. Probing the model externally (i.e. the inputs and outputs) is an imperfect process leading to potential mistakes and inaccuracies. So is analyzing refitted and/or proxy models, as LOCO and LIME require, instead of analyzing your final model. We believe that analyzing the real model is really important - if you don't you're opening yourself up to lots of risks.

ANALYZING THE REAL MODEL IS REALLY IMPORTANT - IF YOU DON'T YOU'RE OPENING YOURSELF UP TO LOTS OF RISKS.

When attempting to explain advanced ML models, it's important that your explanations capture the effect of a feature holistically, or in relation to other features. Univariate analysis, as is typically employed with LOCO and PI, will not properly capture these feature interactions and correlation effects. Again, accuracy suffers. Explainers should also calculate feature explanations from a global, and not just local perspective. Let's say you built a model to understand why Kobe Bryant was a great basketball player. Explaining it using only local feature importance will suggest that his height was not that big of a deal because, from the model's perspective, all the NBA players clustered around Kobe are also very tall. But you can't deny that Kobe's 6'6" height has nothing to do with why he's so good at basketball. Again, inaccuracies crop up.

Most modeling techniques also have tuning knobs known as hyperparameters that can be automatically dialed in to make a model more accurate. There is no practical equivalent to this "automated tuning" operation for explainers. This makes it really difficult to be confident in the results you get from an explainer like LIME, because there's no way to tell whether you correctly set a hyperparameter such as the data-area size. Set it too small and the data will be super sparse in the high-dimensional space of all the features in the model. Set it too wide and you will violate the local linear assumption of LIME.

A consistent explainer uses real data. LIME and PI probe the model with non-real data. For example, let’s say you built a model with income, debt, and debt to income ratio. PI will shuffle each column one at a time independently. This means when PI analyzes debt to income ratio by shuffling, it will no longer be the debt feature divided by the income feature. Since all of the data in the train data maintained this relationship, it should be considered undefined behavior to query a model this way.

Even slight inaccuracies in explanations can lead to devastating outcomes, like releasing models that discriminate based on age, gender, race, or ethnicity, or releasing models that are unstable and produce high default rates that can cause lending businesses to hemorrhage money. A model explainability technique should provide the same answer given the same inputs and model. Inaccurate or inconsistent model explainability techniques might appear to provide some assurance of risk management, but if the results aren’t accurate, they provide little value or assurance to business stakeholders and regulators.

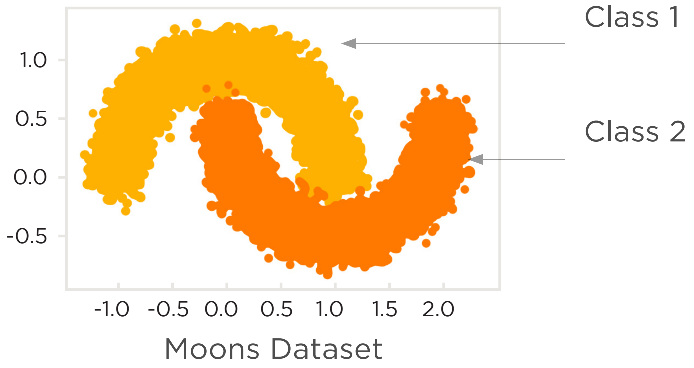
Speed and efficiency are also important. Model explainability should be used often during the model build process and as the model is applied in order to generate explanations for each model-based decision. Slow and inefficient techniques limit the usefulness of explainability: If you have to wait overnight for results, you’re going to run the analysis less often, and get less frequent insights about how your model is working. That means less time working on better models.

PUTTING EXPLAINABILITY TECHNIQUES TO THE TEST

We took our comparison of explainability techniques one step further and ran an experiment to evaluate the various approaches, we conducted an experiment on a sample data set. Any good experiment needs a control or ground truth for comparison. We wanted to compare how different the explainability approaches performed in relation to ground truth on a simple modeling problem to see if any differences emerged. For this study, we used as ground truth a reference value computed using math derived from competitive game theory. The reference value measures the average marginal benefit introduced by a given model's input variables. It does this by exhaustively enumerating all possible combinations of variables.

For machine learning models, this requires training an exponential number of model variations, each with a different combination of variables included. By understanding the comparative performance of the models, we can quantify the value of each variable used in the model. For our experiment, we evaluated the performance of explainability techniques on the "moons" dataset. This is a simple, two variable data set, which is often used in undergraduate machine learning courses to demonstrate the limitations of linear models.

Binary Classification Problem



The task is to build a model that separates the orange data points from the yellow data points using only the X and Y coordinates. A linear model simply can't do this very well. To see this, just draw any straight line through the above diagram and notice that on either side of the line you have both yellow and green data points. There is no straight line that can separate the greens from the yellows. The moons dataset provides a simple example showing that machine learning methods that model variable interactions and that model the real shape of the data are more predictive than old methods like logistic regression that can only create linear models. (This is why you have so many models, but this is a topic for another post.)

We built two machine learning classification models trained on this data set. One using XGBoost, a popular tree ensemble induction algorithm, and Tensorflow, a popular neural network package. At the time of writing, both were considered best in class machine learning modeling techniques. We then applied the various explainability approaches to the resulting tree and neural network models. The results are provided below.

EXPLAINING GRADIENT BOOSTED TREES

	Correlation Error	Variance	Runtime
ZAML	**1.1%**	**0.002004**	**0.006s**
LOCO	10.1%	0.015283	0.003s
PI	15.6%	0.029044	0.022s
LIME	17.9%	0.067037	34.45s

Comparison of explainability techniques for a simple Gradient Boosted Tree model

For Gradient Boosted Trees, LIME, PI, and LOCO all produce high error rates and much higher variance even on a very simple modeling problem, when compared with ZAML. Also important to note is that LIME is much slower to compute. While PI and LOCO appear to be more reasonable in this test on a 2 variable model with 100,000 rows, these methods get much slower very quickly as the number of rows and columns increase, so for larger datasets, like those we see every day in our credit risk modeling work, they become impractical. We've waited hours, even days for explanations from LOCO, PI, and LIME. In our experience, the ZAML approach remains reasonable to compute on a single host, even for models with thousands of variables. With ZAML, we no longer have to wait more than a few seconds for an answer.

EXPLAINING NEURAL NETWORKS

	Correlation Error	Variance	Runtime
ZAML	**0.1%**	**0.000253**	**0.258s**
LOCO	4.1%	0.003953	0.005s
PI	6.8%	0.009088	0.039s
LIME	7.7%	0.010679	48.52s

Comparison of explainability techniques for a Neural Network Model

For neural networks, LIME, PI and LOCO also generate much higher error rates and higher variance. Consistent with its performance on gradient boosted trees, LIME is very slow. For those interested, the scatter plots showing the comparison of each method with the reference values are included in the appendix below. We think the differences in accuracy from the baseline speak for themselves.

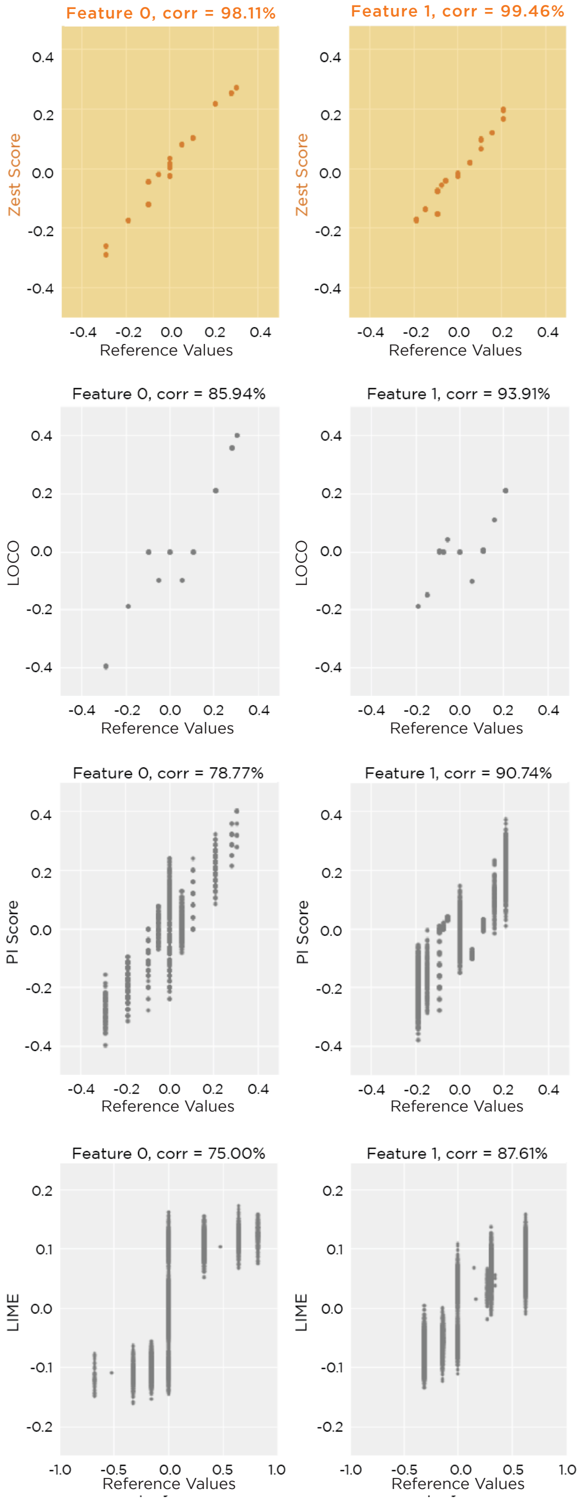
CONCLUSION

Not all explainability techniques are created equal. Even on a simple modeling problem like the two-dimensional “moons” dataset, with only 100,000 rows, we have shown that ZAML explainability is more accurate, consistent, and often faster than the other approaches. When applying machine learning on high-stakes business problems like credit risk modeling, it’s critical to get the core explainability method right so that you truly do know what your model is doing and can manage the risk associated with its application. If the core explainability math you use generates the wrong answer, you might get your business into serious trouble.

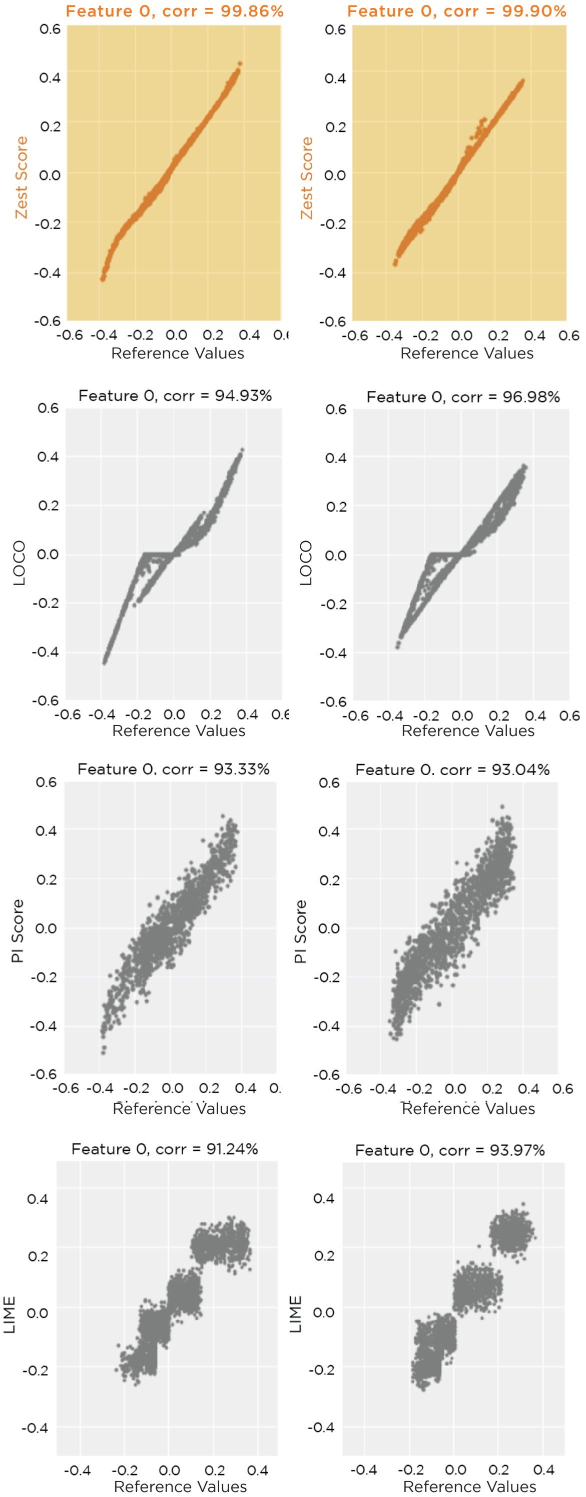
APPENDIX COMPARISON OF METHODS

The charts below show another view comparing the performance of four explainability methods LOCO (Drop One), Permutation Impact (PI), LIME and ZAML. For each method, we created a chart for each of the two features in the “moons dataset” being explained. The dots represent samples from the dataset. The vertical axis is the contribution of that feature to the score generated by the explainability method. The horizontal axis is the contribution of that feature to the score generated by the reference value technique, a widely accepted way of quantifying the contribution of individual elements in a complex system. **The ideal distribution should be on a tight diagonal line, which would reflect as close to 100% correlation as possible.**

GRADIENT BOOSTED TREES



NEURAL NETWORK


Feature 0, corr = 99.86%
Feature 0, corr = 99.90%
Zest Score
Feature 0, corr = 94.93%
Feature 0, corr = 96.98%
LOCO
Feature 0, corr = 93.33%
Feature 0. corr = 93.04%
PI Score
Feature 0, corr = 91.24%
Feature 0, corr = 93.97%
LIME
Reference Values