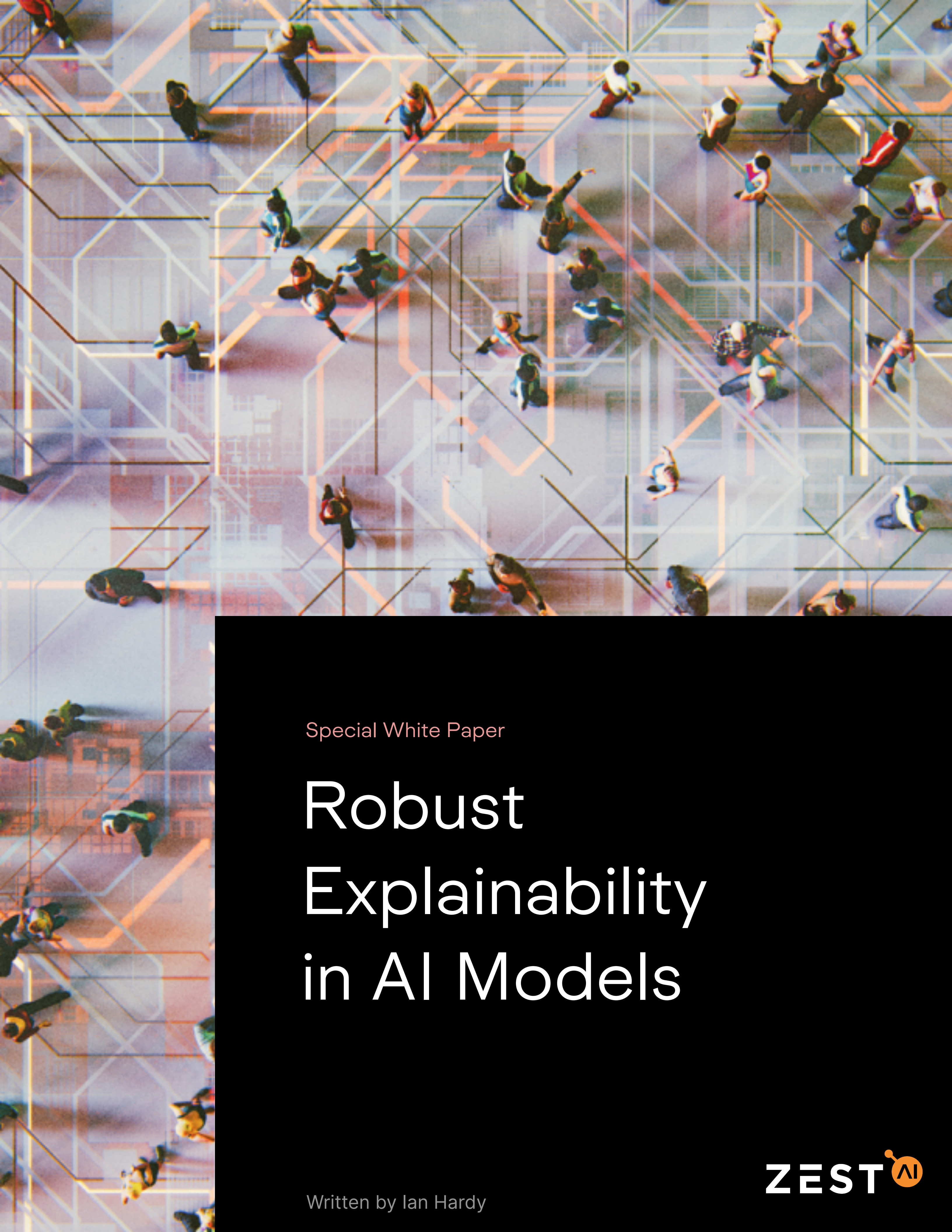
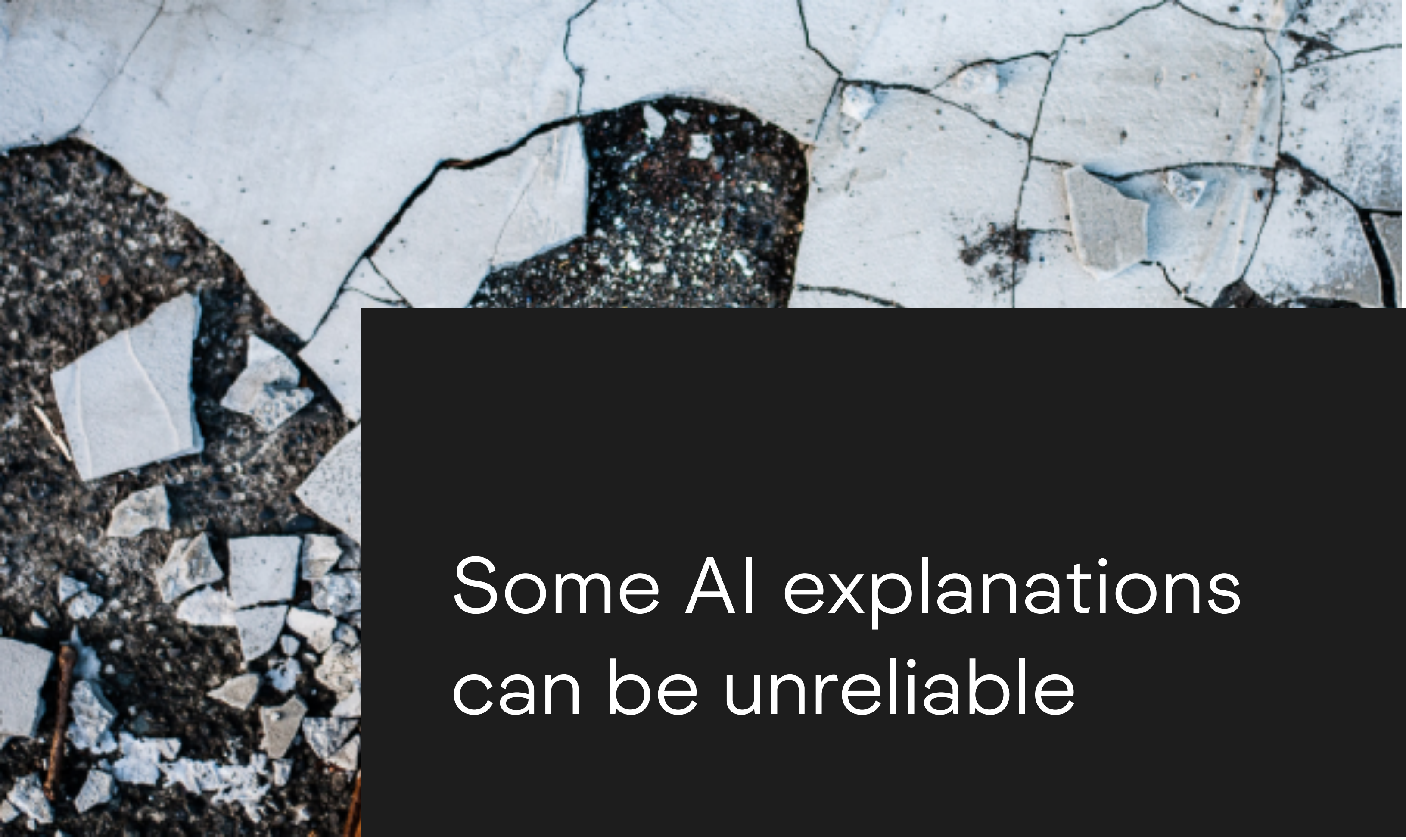


Special White Paper

Robust Explainability in AI Models

ZEST AI

Written by Ian Hardy

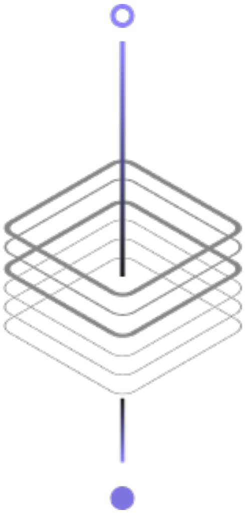


Some AI explanations can be unreliable

Ours aren’t.

Recent research points to weaknesses in some AI explanations due to oversensitivity. We run an experiment using our explainability approaches to demonstrate that with proper methods, AI explainability can be robust and reliable.

This is especially important when it comes to machine learning applications within highly regulated industries, such as credit underwriting. The following white paper provides a refresher on explainability, a discussion of selection and use of references, and an analysis of an experiment run to demonstrate robustness in explainability, including distributional referencing, results, and conclusion.

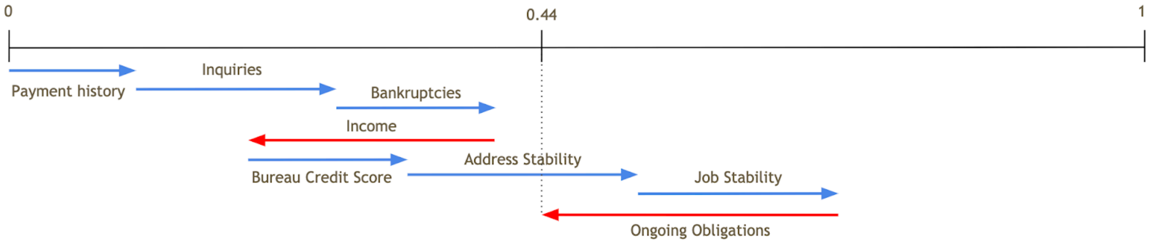


If we are going to reap the accuracy and automation benefits of algorithmic decision-making, we have to trust that our AI models are accurate and comprehensible. At Zest AI we are focused on making machine learning safe to use in many applications, especially those subject to stringent regulations, starting with credit underwriting. Machine learning models can produce far more predictive credit decisions, but they have to be trustworthy. By law, banks and credit unions have to provide accurate reasons why they denied any applicant -- so called adverse action reasons.

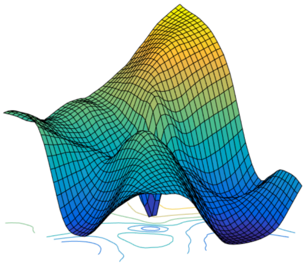
Further, fair lending regulation requires model developers identify drivers of algorithmic bias so they can mitigate them. Both require models to be highly explainable, and stakeholders must be assured that the explanations are correct. In this post we’re going to open the black box to “explain how we explain,” and then put our explainers to the test to show that they are robust and dependable.

The way we explain an applicant’s credit score generated by one of our machine learning credit models is through decomposition, or the measuring of the positive or negative impact of each variable used by the model. The variables associated with the largest changes to the model’s score are deemed the most influential. Below is an example of an ML model decomposition showing eight credit variables, or features, and how they’re assigned positive or negative influence values, with a magnitude corresponding to how influential that feature was.

Example of **efficient** decomposition of applicant with score of 0.44.



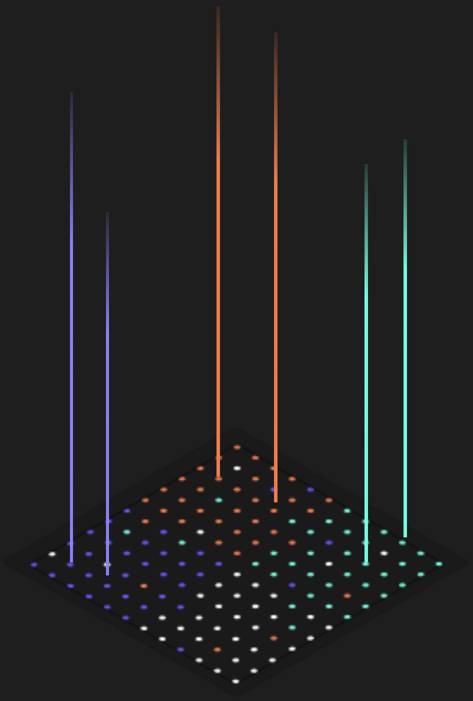
Recent research in the machine learning community has suggested that, due to the nonlinear nature of model decision surfaces, resulting explanations may be overly sensitive and non-robust. In other words, if two rejected credit card applicants had nearly identical input features, e.g., inquiries, bankruptcies, income, etc. and both received nearly identical model scores (default risk predictions), they could be provided with different adverse actions. This has led researchers to question the accuracy of the underlying explanations and calls into question how accurate explainability technology really is.



Pictured: the loss surface of a neural net with respect to the inputs

After all, shouldn't very similar people with very similar scores receive very similar explanations?

As a company, we know that building reliable and explainable AI is possible. Our customers enjoy the benefits of such models today. All of our models go through rigorous testing, including sensitivity testing of explanations, and those tests have never uncovered any issues with explanation stability.



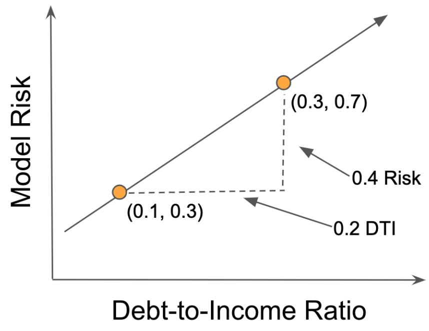
That being said, this new research generated a bit of buzz, and so we wanted to double-check our model explanations did not fall prey to the same kinds of attacks the researchers reported in their work.

What follows is the results of a series of experiments that tested the consistency of our explainability tools. The results show that Zest's explainability software defends against the kind of "explanation sensitivity" reported by academics.

To show how, we will first present a refresher on explainability and then demonstrate that our explanations are consistent and reliable.

A Refresher on Explainability

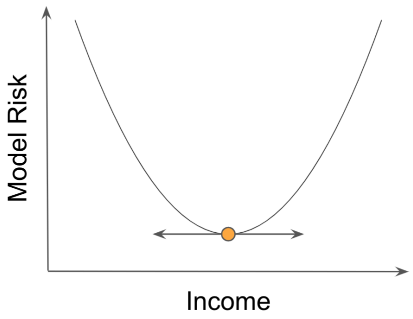
The simplified example to the right shows how we can calculate the rate at which a model's score changes when there's a change in a feature's value, in this case, debt-to-income (DTI). In this example, we know that for this model, any increase in DTI value above zero will correspond to twice the risk for the borrower. This ratio is known as the partial derivative of the score with respect to that feature; the individual feature derivatives together are collectively referred to as a gradient.



$$\frac{(0.4 * Risk)}{(0.2 * DTI)} = 2\frac{dR}{dDTI}$$

One might wonder whether measuring the gradient of a borrower's score is enough to explain the underlying decision: after all, the influence of an individual's features on a score could conceivably be measured by how that score changes as each feature is adjusted. We will show that it is not in fact sufficient.

ML models, including neural nets, generate a massive number of feature combinations that form a high dimensional surface with sharp spikes and dips. These in turn yield complex gradients that change across the range of the feature values in inconsistent ways, in other words, they are not simple and linear.

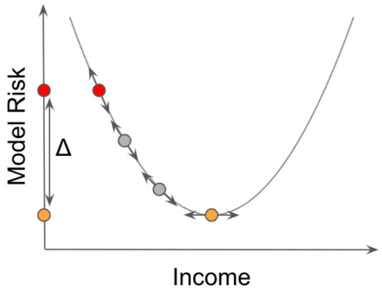


At the highlighted point, the derivative of the model's score w.r.t. income is zero, and thus it appears inconsequential

Consider, the case in which a borrower's income happened to lie on a point of the model's decision surface with a zero gradient, as can happen when the decision goes from a decreasing to increasing state (the slope has to cross zero at some point). When the derivative is taken with respect to income at that point, it will appear that the score does not change in that immediate neighborhood and consequently, the attribution to income will be zero. The explainer would ignore that feature in its analysis of the model, which is the wrong decision to make. Of course, income mattered.

A Refresher on Explainability

We, therefore, cannot simply measure the gradient of the model's score for a borrower and hope to produce an explanation that truly reflects the importance of their features in the outcome. However, by exploiting the second fundamental theorem of calculus and introducing a reference to compare, we can avoid this issue. We create a straight line path from a "baseline" or "reference" borrower to another borrower whose score we want to explain, and average the model's gradient along that path. The result is a breakdown of the difference in the baseline score and the point of interest's score, in terms of individual feature contributions. In the univariate example to our left, the average model gradient from our baseline borrower (red) to our borrower of interest (orange) is multiplied by the difference of income values, which will add up to the difference in their scores (Δ).



According to the F.T.C., we can sum the derivatives of the model's score between a referential borrower (red) and a borrower we're scoring (yellow) to explain the difference in their scores (Δ.)

In a multivariate setting, this tells you how much of the score difference was attributed to each variable (and whether it was a positive or negative contributor.)

This can be understood as a "contextual" explanation, where one borrower is explained "in the context" of another. The explanation is a breakdown of why their scores were different. This approach is particularly useful in the credit space, where we often want to explain why one borrower was denied when another was accepted.

A good analogy for these kinds of explanations is an evaluation of artwork. There is no such thing as an objective measure of quality in artwork, just like there is no such thing as one type of good or bad borrower. People, like art, are complex. Just as works of art are compared to other works of art to measure their relative quality, we can compare borrowers to other borrowers to explain why one is more or less risky than another.

This explainability method (and variants of it) are known as "path- attribution" methods, owing to the fact that they attribute importance to features by adding up the gradients of the model's score along a path from one borrower to another.

Selecting the Right Reference

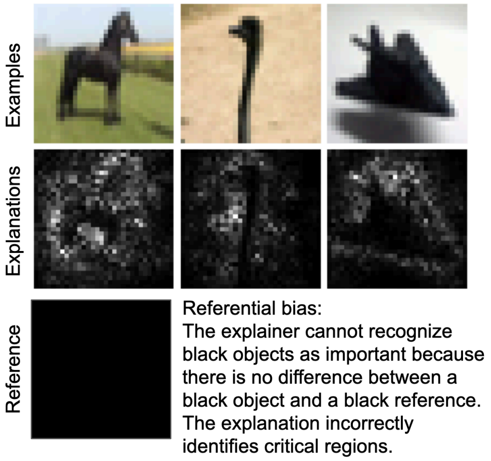
The choice of which referential borrower to compare others to may seem innocuous, but this important decision can introduce significant bias to a score’s explanation. Just like there is no such thing as a monolithically “risky” or “safe” borrower, there is no such thing as an “average” borrower, and this is important.

To illustrate, say a lender does decide to use a single baseline borrower as a reference in their explanation process, one who has the average value for each feature in the model. This borrower receives an average score from the model, so the lender believes they will be a suitable reference by which to explain rejected applications. Let’s say they make $50,000 a year.

Now, another applicant making $50,000 applies, is scored by the model and then explained using the method described above, comparing this new applicant with the “average” baseline.

Since the explainer is breaking down the score difference between the baseline applicant and this new applicant based on the difference in their feature values, and there is no difference in their income (both have $50,000 incomes), it cannot possibly attribute any importance to income. No matter what decision the model makes, or what other characteristics that borrower has, their income will receive zero importance. Importance of income only registers when the feature deviates from the baseline. In real life, this is similar to not “seeing” water when you are swimming beneath the water’s surface (or not noticing the beautiful warm weather because it’s always beautiful and warm). It probably goes without saying, but an output that says income isn’t important is problematic: income is probably one of the most important features in the model.

Selecting the Right References

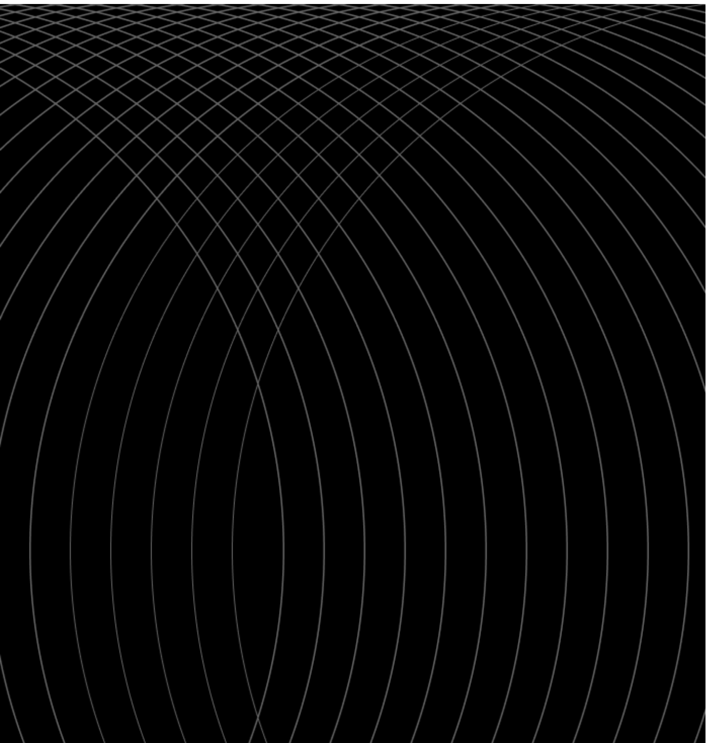


Referential bias:
The explainer cannot recognize black objects as important because there is no difference between a black object and a black reference. The explanation incorrectly identifies critical regions.

Some refer to this problem as referential biasing, because the reference creates blindspots in the explanation. In the computer vision example shown, the explanation of what the model “saw” ignores the subject of each photo because the subject is black, just like the reference.

For this reason, we at Zest choose to use *distributional referencing* when creating our explanations. Instead of using a single baseline borrower by which to judge other applicants, we draw a large sample of referential borrowers from the distribution on which the model learned. Recalling the art critic analogy, there is, of course, no single source of art by which others are judged. We compare each piece we see to the many we have seen before to recognize its merits and see where it is brilliant and where it falls short.

By averaging attributions with respect to many ‘referential’ borrowers, we mitigate potential bias from any one of the references and obtain an appropriate and coherent explanation.



Experimental Design

We tested our neural net explainability kit on Kaggle’s Lending Club Loan Data to ensure that the domain of the problem (image classification vs. credit analysis) did not have an impact on the sensitivity of our model. Furthermore, because two “applicants need to be nearly identical”, discrete variables (e.g., a loan term of 36 or 60 months) were dropped, as small perturbations do not exist for such features.

Experimental Design

We created a predictive neural net model, and implemented the path-attribution “attack” presented in the paper “Explanations Can be Manipulated and Geometry is to Blame.” In our case, this technique attempts to find the optimal (and slightest) adjustment to an applicant’s input features such that the model’s decision doesn’t change, but the explanation for that decision does.

The manipulated borrower that is generated from this process is referred to as “adversarial” to its original; its explanation is referred to as the “adversarial explanation” of the original. Technical details follow for those who are interested, though the next paragraph is not necessary for understanding the results of our experimentation.

The authors of the paper suggest a novel gradient-based approach for identifying an adversarial explanation, wherein one example (in their case, an image) is manipulated such that its explanation matches an arbitrary target (often the explanation of another image in the dataset) The loss function minimized in the attack combines the difference between the feature space of the manipulated example and the original example (ensuring that the original image and the perturbed image are similar) and the difference between the explanation of the manipulated example and the target explanation to be copied (ensuring that the resultant explanation is different than the original.) Below is the loss function in detail; x is an original input to the model, xadv is the manipulation of x, g() is the model, and h() is the explainer (ht is a target explanation, and γ is a tradeoff hyperparameter):

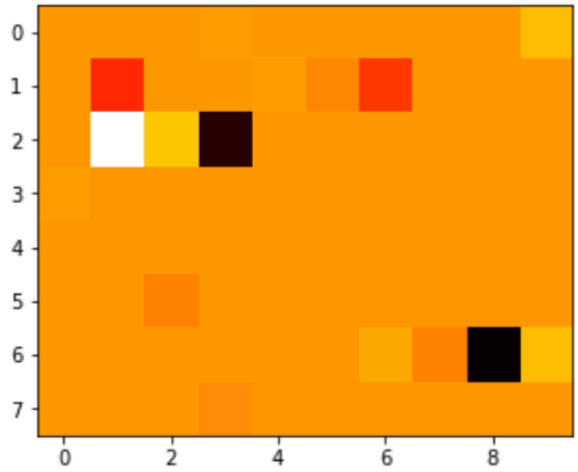
$$L = ||h(x_{adv}) - h^t||^2 + \gamma * ||g(x_{adv}) - g(x)||^2$$

An adverse explanation can then be found with a simple gradient-descent algorithm.

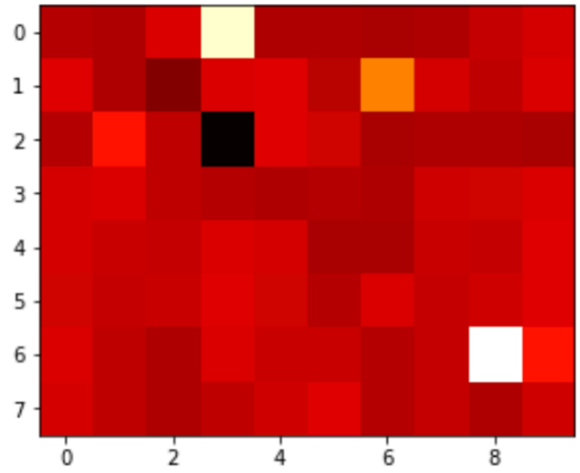
The key idea is that the original borrower and adverse borrower have to be inherently similar in their input features, they have to receive the same score from the model, and they have to have different explanations. That will point to inherent unreliability in the attribution. Our initial attack (the results of which are discussed below) only utilized a single reference point in the explanations for consistency with the original paper.

Results

In the figure below we compared the explanations of an original data point (an applicant) and its adversarial counterpart. For perspective, the original applicant's explanation is on the left and the adversarial explanation is on the right. Within each image, each rectangular block represents a different credit feature, with the color shade representing different explanation values (effectively a heatmap of each feature's attribution, arranged in a rectangular shape). Clearly, the adversarial explanation is different from the original. In other words, these applicants, if both denied, would have received very different adverse action notices.



Original Explanation
(g(x) = 0.298)

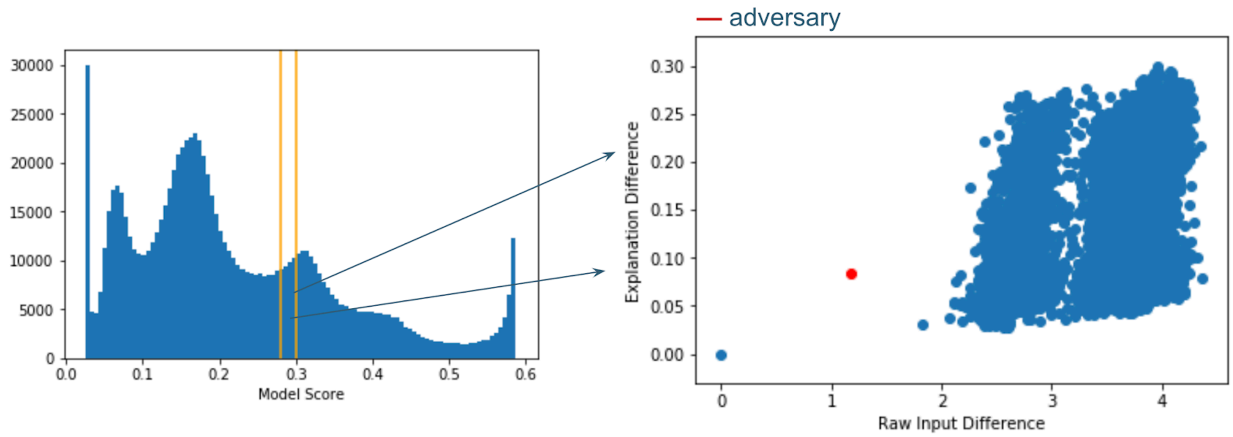


Adversarial Explanation
(g(x) = 0.290)

Also, notice that the model scores (predictions; g(x)) of each are similar too; both around 0.29~0.30.

To measure the similarity between our original borrower and its adverse counterpart, or in other words, to determine whether these two individuals were actually similar and should have received similar explanations, we took a look at how different a manipulated explanation is from our original with respect to the explanations of other points that scored similarly. We looked at a narrow band of borrowers that scored the same (0.29) as our original and adversary, and graph their difference in the score space and explanation space. The results are shown below:

Summary



In the plot on the right above, the Y-axis represents the distance in the explanation space (using an L2 norm) from the original point we manipulated to others that scored within 0.5% of it, and the X-axis represents the same distance but in the input (feature) space.

Essentially, it measures how “relatively similar” the points that scored 0.29 were in these two spaces. The blue dot on the lower left of the graph is our original point, and the red is its adversary.

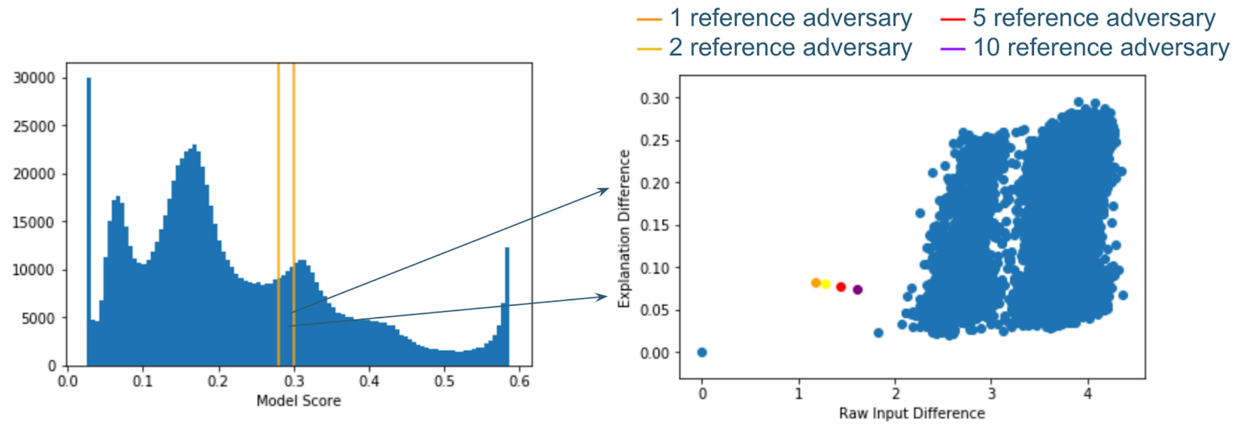
We can see that the adversarial point is closer in the feature space to our original than any other point in the training set with the same score, by a significant margin. However, in the explanation space, it’s fairly far -- certainly far enough that the original and adversary are clearly “different” borrowers from an explanation perspective. This is likely a result of the rich features available in credit underwriting which make it rather hard to fool the explainer.

The data used in the paper was image data, which only has pixel values representing color. By contrast, credit datasets include many de- correlated attributes from diverse data sources like credit reports, transaction history, alternative data, and application data. Nevertheless, we still note the adversary is much closer to the original point than all other points from the scores in this slice.

It is not impossible that for certain applicants in the dataset that this gap is further reduced, i.e., the adversary has a similar score, similar features, but a different explanation. This is the situation we are seeking to mitigate.

Summary

To prevent such situations from occurring, and because we already know that adding references helps avoid the referential bias problem, we increased the number of references and re-ran the experiment, to gauge the effect of distributional referencing on improving the robustness of the explanation. The results are displayed below:

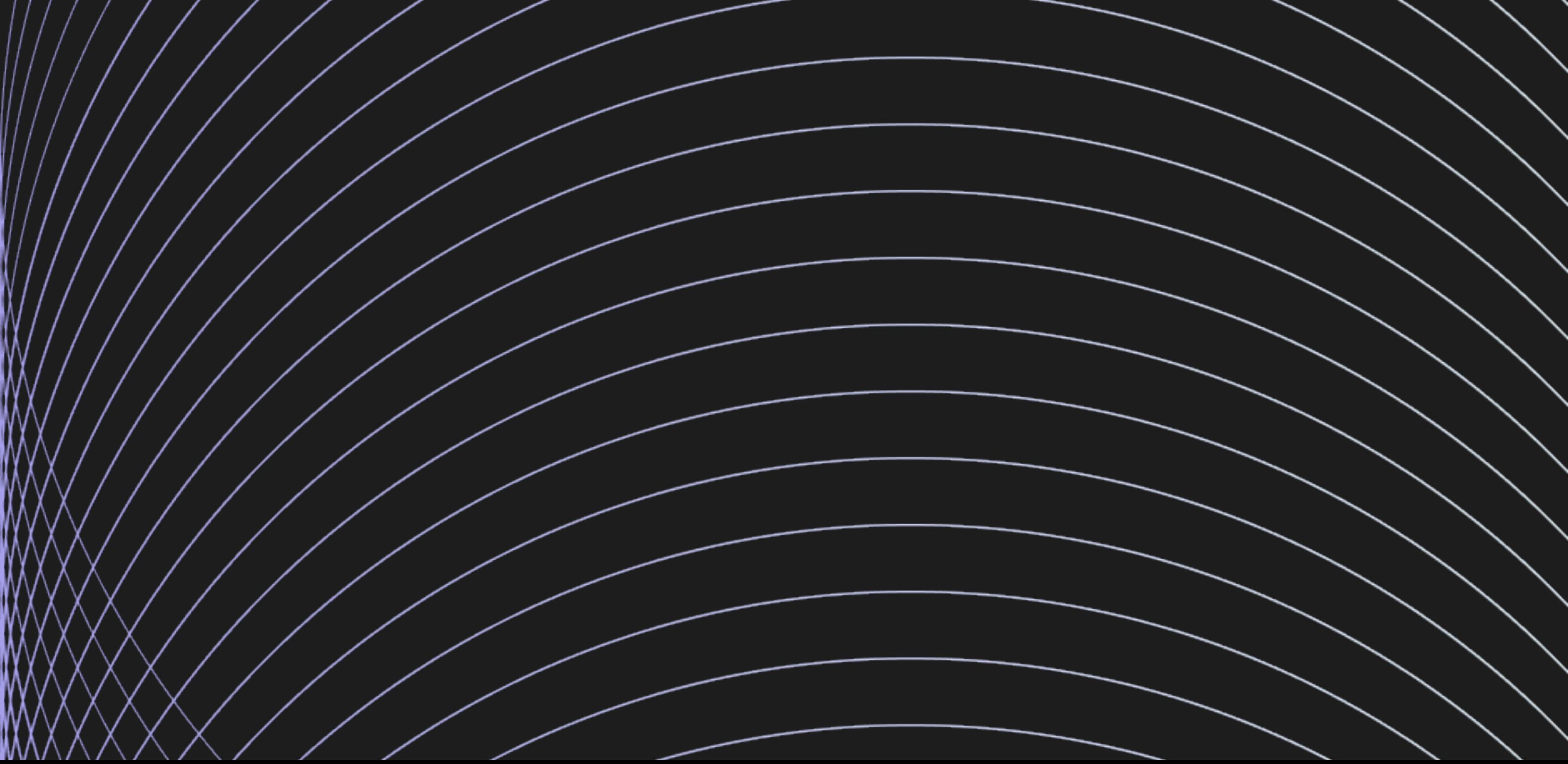


As the number of references increases, so too does the distance in the feature space between the original borrower and of the resulting manipulated borrower. It is harder and harder for the algorithm to find a borrower that scores similarly to our original and has a different explanation, without significantly changing the original feature space. This is the desired result.

It is fine that two borrowers have the same score and have different explanations. However, it ought to be because they have significantly different feature values, not that they merely lie in a part of the model's surface that is sensitive to noise. **Beyond ten references, we were unable to find such a borrower.**

In effect, distributional referencing prevents us from being able to "fool" our explainer, and helps ensure that Zest-powered explanations are stable, in addition to being accurate.

These experiments show that distributional referencing techniques, like the ones we employ here at Zest, are robust. The reference populations we use at Zest typically contain thousands of data points.



Special White Paper

Robust Explainability in AI Models

Conclusion

There has been a recent focus on ensuring the robustness of machine learning model predictions, and we hope to see this focus extend to the robustness of explainers, particularly in high-stakes applications where robust explainability is mandatory. In financial services, businesses rely on model analysis like we describe above to power consumer notices and to determine whether a model has bias and which features cause it. As such, it's important the methods used to explain model results are robust and accurate.

For more information, say **hello@zest.ai.**