WHITE PAPER

XEMP Prediction Explanations with DataRobot

For Data Scientists and Model Validators

DataRobot

XEMP Prediction Explanations with DataRobot

Predictive models make decisions for scoring bank loans, ranking care for patients, and identifying which equipment may fail. As their importance has grown, so has the desire to understand the predictions that the models are making.

Introduction

An understanding of predictions is necessary as legal and regulatory requirements such as the EU's General Data Protection Regulation (GDPR) "right to explanation" is enforced, as well as concerns over inequality and bias in predictions, and safety critical applications play a more prominent role in the decision-making processes of many organizations.

One example of how lack of transparency can erode trust in predictive models is the case of Tammy Dobbs, as noted in the AI Now 2018 report. As part of the Arkansas state disability program, Tammy was allocated 56 hours of home care to help her with her cerebral palsy. Years later, a state assessor used a proprietary algorithm to readjust Tammy's allocated hours to 32 hours per week, offering no explanation for the decision. Legal Aid of Arkansas sued the State of Arkansas, eventually winning a ruling that the new algorithmic allocation was both erroneous and unconstitutional. The case cemented the distrust that many people harbored towards AI and its role in making the important decisions that affect people's day-to-day lives. It is also a good example of why transparency is so important when explaining the decisions behind predictive models.

When explaining important predictions or decisions to a non-technical stakeholder, it is so important to provide straightforward and people-oriented answers to such questions as:

- Which input feature values in this data point caused the prediction to be so high (or so low)?
- Which input feature values in this data point caused the decision to be negative (or positive)?
- Was my sensitive feature (e.g. gender, race, age etc.) a factor in the prediction or decision?
- Which input feature values for this data point were the most important in determining the result?
- Which feature values made the prediction lower? Which feature values made the prediction higher?

The table below shows an example of a prediction by DataRobot. The model predicts that hospital patient Lester's probability readmission rate is 63.7%—quite high compared to a typical patient. Naturally, doctors and administrators have questions about why his readmission rate is so high. By adding explanations, it is possible to get a ranked list of the key factors for the prediction, and we can see that the top factor is the primary diagnosis feature or variable. In this example, the code of abdominal pain at an unspecified site has high positive strength as a factor in the prediction. The positive and negative effect of the factor strength is indicated by orange (positive) or blue (negative) background color. A secondary factor is the number of prior inpatient stays Lester had in the last year, since patients that have had previous admissions tend to readmit at a higher rate. Because this is Lester's first visit in over a year, this factor has a moderate negative effect on the readmission probability. We can see that these are the two most important factors for Lester's high readmission score, and they provide the necessary context for understanding the prediction.

Factors Contributing to the Readmission Probability for Lester Briones

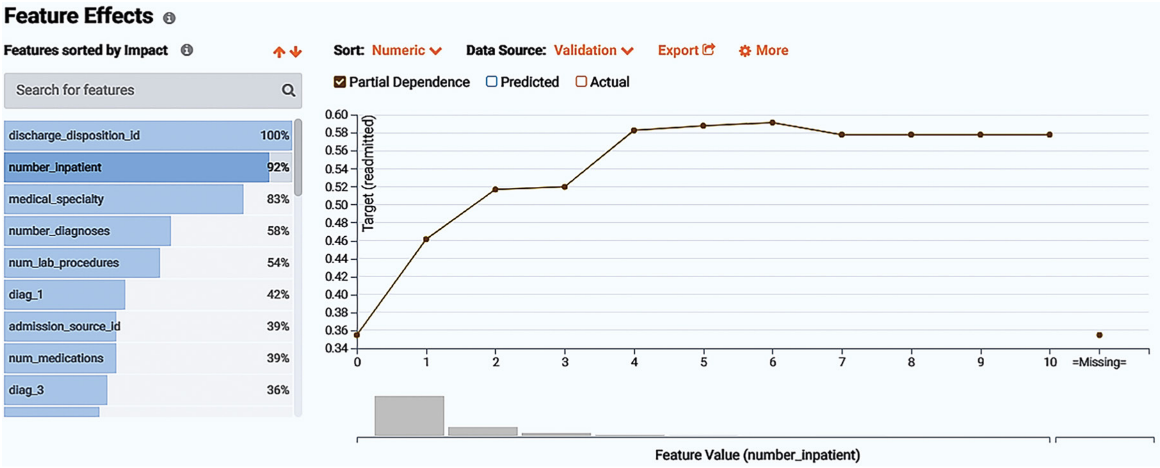
Primary Factor	Primary Diagnosis: **Abdominal pain, unspecified site**	**High** Factor Strength	**63.7%** Readmission Probability
Secondary Factor	Number of Inpatient Stays Last 12 Months: **0**	**Moderate** Factor Strength	

Prediction explanations can be communicated more powerfully and intuitively by combining them with existing interpretability tools, such as partial dependence plots. As we saw above, Lester had not had any inpatient stays in the previous 12 months, and this had a negative effect upon his predicted probability of readmission. The partial dependence plot shows that patients with zero inpatient stays, the most common situation, had only 36% probability of readmission on average, compared to 48% to 60% average readmission rates for patients with one or more inpatient stays in the previous 12 months.

Explanations offer clear value, yet there is not widespread use of them within enterprises. We have found three factors that are limiting widespread use of explanations.

1. Explanations are a new technique in data science and there is a clear lack of criteria on what constitutes a good explanation.
2. While data scientists have access to many open source explanation methodologies, such as LIME, Shap, and Breakdown, there is a lack of benchmarks to evaluate which of these approaches offer suitable accuracy. Without a way to evaluate multiple approaches, it's not clear which approach an enterprise should widely adopt.

3. Enterprises require reliable scalable algorithms that run across large complex datasets. Most explanation approaches have not been designed to account for scalability, and some approaches can take hours to return a single prediction explanation, something that is simply not scalable for a modern enterprise.



This white paper will begin with identifying the key criteria for a good explanation. After establishing the criteria, the paper highlights some of the shortcomings of the most widely used explanation approach, LIME. Next, we introduce and review DataRobot's eXemplar-based Explanations of Model Predictions (XEMP). The last section of the paper provides several benchmark test results in order to quantify the advantages of XEMP over Local Interpretable Model-Agnostic Explanations (LIME).

How to Compare Explanation Approaches

A starting point for what constitutes a good explanation comes from the Montréal Declaration for Responsible Development of Artificial Intelligence. This framework was developed by hundreds of experts and citizens as a framework to guide development of Artificial Intelligence. Section 5-2 states:

> The decisions made by AIs affecting a person's life, quality of life, or reputation should always be justifiable in a language that is understood by the people who use them or who are subjected to the consequences of their use. Justification consists in making transparent the most important factors and parameters shaping the decision, and should take the same form as the justification we would demand of a human making the same kind of decision.

Using this framework, there are at least three different properties that a good explanation should have:

1. The explanations need to be readable and suitable for all the stakeholders, including those that are non-technical. At a practical level, this suggests that explanations should be stated in the language understandable by the people affected by the model,

and that concepts used in explanations must be intuitive to the mathematical background of stakeholders. For example, if age or gender is a factor in the explanation, then they should just be plainly stated in the explanation. This may seem obvious, but data scientists typically do creative transformations and many also generate new features. For example, instead of incorporating a category of race in an explanation, a technical explanation may use an encoded version of race using weight of evidence transformation. If the explanation is given based on this transformed feature, it becomes difficult for a non-data scientist to understand the impact of this feature.

2. The explanation should contain the most important factors for the decision. This can be evaluated in several ways, overall strength, fidelity to the prediction, and consistency. A good explanation should highlight just a handful of most important factors. Those factors should have a strong influence on the prediction. Later in the quantitative testing section of this document, we compare different explanation methodologies based on how strong of an influence the explanation has on its prediction using the concept of explanation accuracy.

> The explanation should contain the most important factors for the decision. This can be evaluated in several ways, overall strength, fidelity to the prediction, and consistency.

An explanation should have fidelity to the prediction and the model. When it comes to prediction, fidelity means that two very different predictions should have two different explanations. In a similar fashion, we expect that if an explanation has fidelity to the data and predictions, we will see a variation of explanations across many examples. Later in the quantitative testing, we evaluate fidelity using the concept of explanation diversity. Similarly, an explanation should have fidelity to the model at hand. It should faithfully follow the gradients and discontinuities found in the original model's partial dependence, include all interaction effects, and faithfully follow the original model's decision boundaries. An explanation based on a surrogate or secondary model will not have fidelity to the original model. A lack of fidelity can also raise concerns by model validators and regulators that require explanations based on the actual model (not a surrogate).

3. Explanations should be identical and replicable given the same model, same input feature values, and same prediction to be explained. This type of consistency is important to have confidence in explanations. After all, a prediction that can generate two different explanations is problematic. If you asked a human their reason for a specific decision, and each time you asked they told you a different story, then you wouldn't give much credibility to those explanations. You might even be suspicious that the person was

just inventing the decisions! It will not be possible to trust an explanation if stakeholders don't get identical explanations for the same model.

Why not just use LIME...

To calculate an explanation, LIME starts by generating synthetic data around a record to be explained (X). The concept is to create a local cloud of data points and their predictions. The next step is building a surrogate or secondary linear model on the data points, which is chosen because it is considered easier to explain. The explanation is then based on the coefficients of the linear model. This is sometimes called a local explanation, because the explanation is only valid for the local region around X.

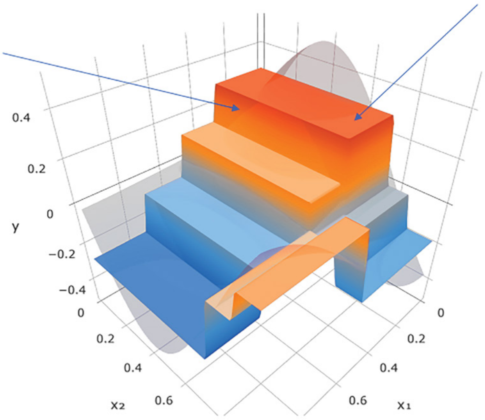
A key concern with LIME is the use of a surrogate model as the basis for the explanation. Cynthia Rudin, a professor in computer science at Duke University who specializes in interpretable models, points out that surrogate models are not correct 100% of the time. Her argument against surrogate models is that even if a surrogate model is only correct 80% of the time (which she notes is generally considered good for a data science model), then 20% of the time it doesn't even understand the underlying model. In essence, Rudin points out that the weakness of the surrogate model is that it does not fully capture the behavior of the underlying deployed model.

LIME uses a linear model as a surrogate model, but using a linear model to approximate a complex model will not be realistic in many situations. While the original LIME paper assumes that the local region is linear, there is no empirical support for that idea. Consider how tree-based methods operate; they partition in ways that are non-linear and discontinuous, e.g. step functions. Figure 1 illustrates how this can occur in a simple decision tree. While some regions have a flat surface, there are discontinuities or steep curves that cannot be approximated by the smooth planes assumed by LIME. In a subsequent paper, the authors of LIME recognize that there is little reason to believe that local behavior is linear, and that LIME can lead "users to being potentially misled as to how the model will behave".

LIME

Local Interpretable Model-Agnostic Explanations (LIME) is the most popular explanation technology. LIME was developed at the University of Washington in 2016 and is implemented in several open source packages and by several commercial software vendors. In this section, we explain how LIME operates and some of its limitations.

Discontinuity, poorly approximated by a local linear model



Flat region, well approximated by a local linear model

Figure 1. Image source: http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

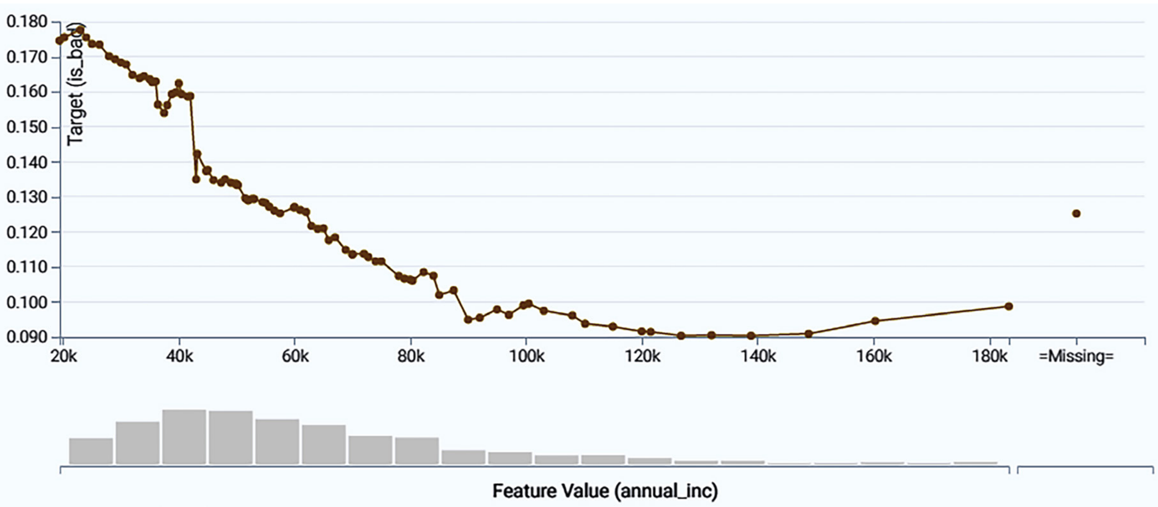
The inability of a linear surrogate model, such as used by LIME, to capture the behavior of the underlying model raises doubt as to whether it is an appropriate technique. After all, most interpretability methods, such as feature importance and partial dependence, are based on directly querying the underlying model. Moreover, legal requirements about explanations typically require explanations based on the actual model (not a surrogate). For example, the Equal Credit Opportunity Act (ECOA), as implemented by Regulation B, and the Fair Credit Reporting Act (FCRA) were to designed to protect consumers and businesses against potential discrimination by requiring that creditors support their credit decisioning processes and to alert the consumer that negative information was the basis for “adverse action.” Specifically, in the event that a creditor has denied a consumer credit then the creditor must provide an adverse action notice to the consumer that includes the top four factors that adversely affected the credit score; if one of the key factors was the number of inquiries to the consumer’s report, you must list five key factors. There is no option for approximation, nor surrogate models, in this regulatory requirement. Instead, the lenders must be able to accurately interpret explain the actual model (not a surrogate) to ensure alignment with regulatory requirements.

In the process of creating an explanation, LIME generates synthetic data upon which the surrogate model is built. This leads to a myriad of problems. First, LIME generates synthetic data randomly. This means if LIME is run twice, you can have two different sets of synthetic data generated, and hence two different explanations. The result is that LIME can supply multiple contradictory explanations. The table below shows an example from the Boston Housing dataset where LIME provides two different explanations for the same data point. The only commonality is that both RM (average number of rooms per dwelling) and LSTAT (percentage of the population with lower education and working statuses) appear near the top in both explanations.

Value	Explanation 1	Explanation 2
Predicted Value	21.56	21.56
Top Explanation	LSTAT	RM
	RM	LSTAT
	DIS	TAX
	B	PRATIO
	ZN	CRIM

This lack of consistency is built into the LIME approach. LIME’s reliance on generating random data means that given the same dataset and underlying model, that two independent implementations running explanations may get different explanations for a given data point. According to LIME's author, "LIME explanations are the result of a random sampling process, so we shouldn't expect to get the exact same explanation every time."[1] Consequently, LIME has significant issues when it comes to reproducibility and confidence in the explanations.

A related issue is ensuring the generated synthetic data accurately represents the actual data. In the widely used Python implementation of LIME created by LIME's authors, there is an assumption of normality when generating synthetic data for continuous features. However, real-world datasets often include zero-inflated, skewed, and truncated features that are not normally distributed. Commonly found examples of these features include: how many purchases a customer has made; how much a customer spent on purchases; how long a person has been married; number of criminal convictions; and years of education. The inability of LIME to handle non-normal distributions can lead to inaccurate explanations. In the quantitative benchmarking section of this document, one of the examples will illustrate this point.



Once LIME generates the data, it builds a model around those data points. A critical issue is the size of the local region. If the region is too big, then the explanations mirror the overall global feature importance and there is not fidelity to the data point being explained. However, if the local region is too small, LIME may provide counterintuitive explanations. Consider the below example of income for a probability of defaulting on a loan. Overall, there is a general trend that the higher the income, the less likely a person will be to default. However, there are some areas, say between 90K and 100K or between 140K to 200K, where the model actually increases the probability of default as income goes up. If our data point falls into this range and the local area is so small to only encompass this area, the explanations that come out of LIME may say that one of the risks for defaulting is that the person has 100K of income. So, while it is strictly true compared to someone that has 90K for this dataset, it really isn't a very useful or fair comparison. As we see later, other methods are more explicit about their comparison and don't suffer from this issue of local minima.

[1] https://github.com/marcotcr/lime/issues/119

The LIME approach incorporates a number of hyperparameters which need to be selected when getting explanations. These can include setting the kernel size or the number of clusters in KLIME, a faster variation of LIME. A quick perusal of the discussion about LIME or published papers on KLIME highlights differing approaches. As a practical matter, any explanation approach that requires such tuning is troublesome. How do you know if you have properly set the kernel size? Or the number of clusters? In the case of LIME, their authors acknowledge that it isn't clear what a "local region" it is, and there is no simple metric for deciding on these boundaries. Many users end up experimenting to find explanations that seem to make sense. Such an subjective approach leads to the issue that the same prediction may lead to different explanations by two different users of LIME on the same dataset. This again leads to problems of consistency in explanations and should create doubt in the certainty of explanations from LIME.

LIME suffers from a number of limitations, including dependence on a surrogate linear model, generating of synthetic data, and an inability to scale.

Finally, LIME is too computationally intensive for production use. One explanation requires generating synthetic data and then conducting thousands of predictions. The scale of one explanation to thousands of predictions is too computationally intensive for production use that require millions of explanations. This is why variations of LIME, such as KLIME, are employed.

In summary, LIME suffers from a number of limitations, including dependence on a surrogate linear model, generating of synthetic data, and an inability to scale.

Introducing XEMP

XEMP was designed within DataRobot and introduced to our customers in 2016. XEMP had several design objectives. First, the approach needed to be model agnostic. In a modern enterprise, data scientists use a variety of algorithms and ensemble methods. Any approach that is limited to a particular model structure will be very limited in its usage. This is known as the "no free lunch" concept in machine learning, meaning that there is no one best model or approach for every problem and that you cannot know in advance which algorithm will work best on your data. Consequently, there is a need for diversity of model types and ensembling of models. As a result, an explanation approach needs to work with different types of models and combinations of models, whether it is Random Forest, SVM, or deep learning. This standard is no different to how we treat other explanatory tools like partial dependence.

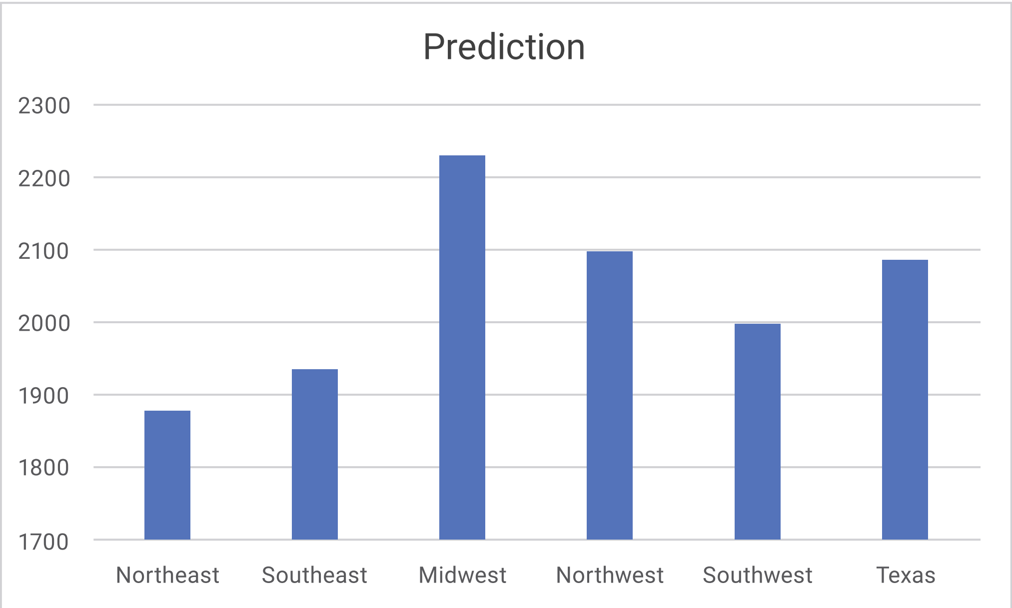
Second, explanations need to be based on underlying model rather than a surrogate model. As noted in the section on LIME, any approach that does not explain the underlying model would never meet the regulatory criteria that requires an explanation of the decision the deployed model has made. Besides the regulatory reasons, it is just common sense that the best way to understand a model is to look at the model, not a second hand approximation.

Finally, XEMP would need to be flexible for handling diverse data types, while maintaining computationally efficiency. Our customers must calculate millions of explanations in a reasonable amount of time. This requires some creativity in the design process. While there are many explanation approaches, most of them have not considered computational efficiency in their design. As an example, one approach—breakdown—can take 3.5 hours for just one explanation.

Since XEMP explains predictions by how they differ from a typical value, it uses exemplars to represent a baseline "typical" data value. No single input feature value can truly represent the full range of the data, so an exemplar uses a series of synthetic data rows comprising the range of values that an input feature can take, along with the consequent variations in predicted values. Unlike LIME, the synthetic data created by XEMP takes actual values from the entire range of values found in the population, and these synthetic data are consistent each time an explanation is calculated, not varying randomly.

To illustrate XEMP, let's start with a simple insurance model that predicts Sally should pay $2,000 for car insurance. Sally wants to know, what data points about her are leading to that price? For background, Sally is 43, female, and lives in the Southwest. XEMP is going to compare each of the features or variables about Sally to a baseline exemplar value.

The first step for XEMP is to evaluate each feature to calculate its exemplar values. For Sally, we get predictions for all the possible values of locations as shown below.



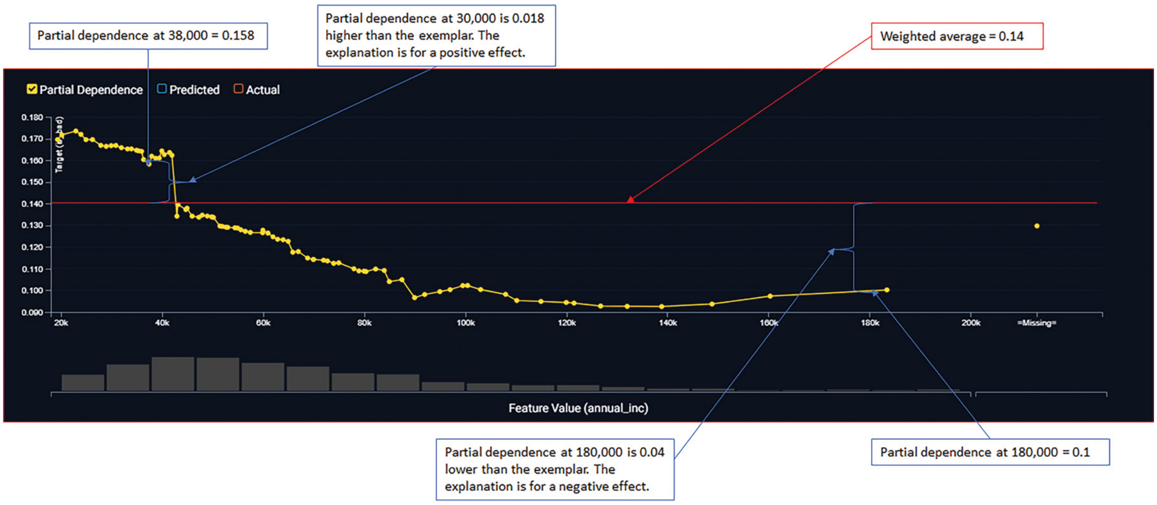
The next step is to average all of these features to generate a baseline exemplar prediction. The value of $2,040 represents the baseline price for Sally across all the possible locations. This value is used as our baseline exemplar prediction. The table below shows all the predictions for Sally across all the locations, including "Southwest," the actual value we are evaluating.

Driver Age	Gender	Location	Prediction
43	F	Northeast	1880
43	F	Southeast	1940
43	F	Midwest	2230
43	F	Northwest	2100
43	F	Southwest	2000
43	F	Texas	2090
Baseline Prediction			**2040**

The next step is identifying how much of a factor location is for Sally. In this case, the exemplar value of $2,040 is subtracted from the actual prediction of $2,000, leading to a difference of -$40, meaning that the influence of location is reducing the prediction by $40 from the exemplar value. The same step can be repeated for other other features. In this case, let's say the effect of gender was 100, and age was 20. The final explanation for Sally's $2,000 prediction is that gender was the strongest explanation for the prediction, followed by location and age.

The key to understanding how the XEMP methodology produces an explanation is to know what the prediction is being compared to.

The key to understanding how the XEMP methodology produces an explanation is to know what the prediction is being compared to. Consider the example below that was discussed in the context of LIME. With LIME, there were issues with local minima. In contrast, XEMP uses the concept of an exemplar value for comparison. This means XEMP is comparing a data point with an exemplar data point. The partial dependence plot below shows the probability of a Lending Club loan going bad versus the applicant's income.



The exemplar prediction is the weighted average of the partial dependencies, in this case approximately 0.14. The explanation for an income with a partial dependence lower than 0.14 should show a negative effect upon the prediction, while the explanation for an income with a partial dependence higher than 0.14 should show a positive effect upon the prediction. The prediction explanation for a person with $180,000 income should show that income has a negative effect upon the prediction, because the partial dependence for high incomes is substantially lower than for most other incomes. Similarly, the prediction explanation with a $38,000 income shows a positive effect upon the prediction. This is true even though the predicted value is lower than at $36,000 or $40,000, because the partial dependence at this income is higher than the exemplar.

In order to have fidelity with the model, a prediction explanation should be consistent with the partial dependence or, more pedantically, the individual conditional expectation (ICE).

In order to have fidelity with the model, a prediction explanation should be consistent with the partial dependence or, more pedantically, the individual conditional expectation (ICE). The explanation should show a positive effect for feature values where the ICE value is high, and a negative effect for feature values where the ICE value is low.

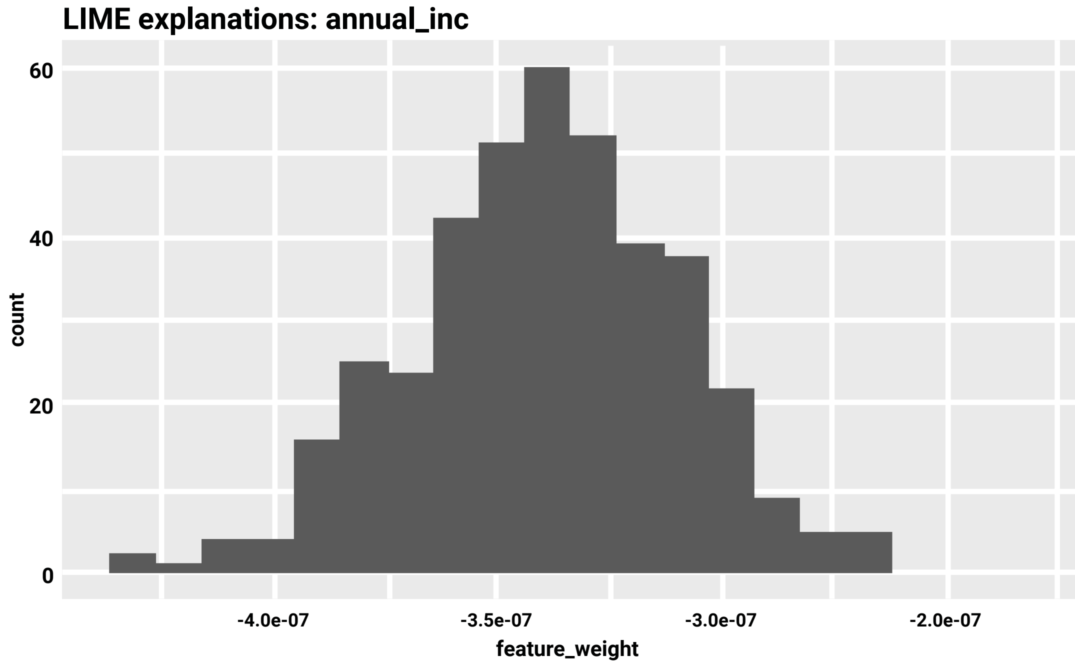


Figure 2

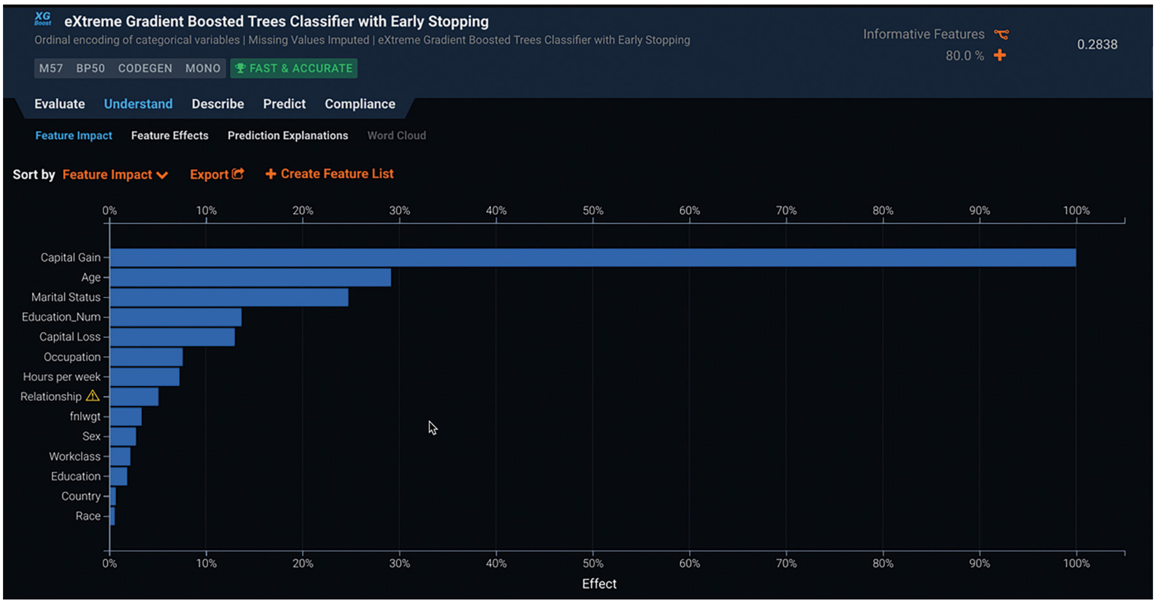
While XEMP creates both positive and negative reasons, when we run LIME on this same data, it does not always retain fidelity to the model. If the hyperparameters were not tuned to use binning for numeric features, we found that LIME always produced small negative explanations for annual income, no matter what the value of annual income. This is counterintuitive. An input feature cannot always have a negative effect upon the prediction, no matter what value it takes. There must be some values that give a better result. This is a reminder that methods that require hyperparameter tuning can be problematic.

Benchmarking the Performance of XEMP against LIME

This section will quantify the benefits of XEMP compared to LIME in three areas. First is a measure of explanation diversity. This test recognizes that it is important to have explanations which have diversity. If every explanation was just a mirror of the overall feature importance, that would not be useful. Users want to understand how this particular prediction is different. Second is measuring the accuracy of explanations. To evaluate the accuracy of explanations, we use a permutation method introduced by Lundeberg. Finally, we benchmark the computational performance of these approaches across several datasets.

EXPLANATION DIVERSITY

An important trait for explanations is that they maintain fidelity to the data being explained. A common issue for the fidelity of explanations is that relationship to feature importance. Feature importance identifies what the most important features are that affect predictions in the model. Typically, it would look like a ranked list in Figure 2. However, when it comes to explanations for individual predictions, we expect them to deviate from the overall feature importance. After all, if all explanations mirror the overall feature importance, they would not be showing fidelity to the individual feature values and predicted value for this data point examples.

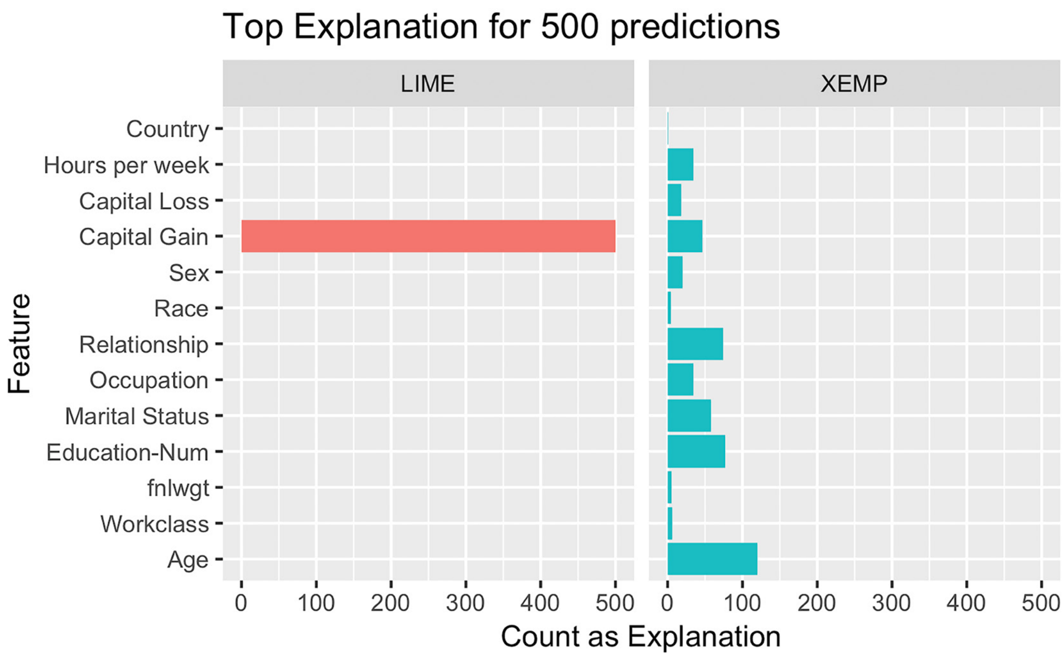


In contrast, the purpose of explanations is to help clarify particular predictions. For example, what are the key factors that lead to Zhang being approved for a loan? While the explanations would be influenced by the overall feature impact, it wouldn't be helpful if every explanation was a mirror of feature importance. Instead, we want our explanation to maintain fidelity, and this is termed as explanation diversity.

Explanation diversity is quantified by identifying how many different types of explanations are used.

Explanation diversity is quantified by identifying how many different types of explanations are used. To evaluate LIME and XEMP, explanations were run on the Adult dataset, a widely used benchmarking dataset in interpretability research. In fact, the LIME python implementation includes a tutorial showing how to get explanations from the Adult dataset.

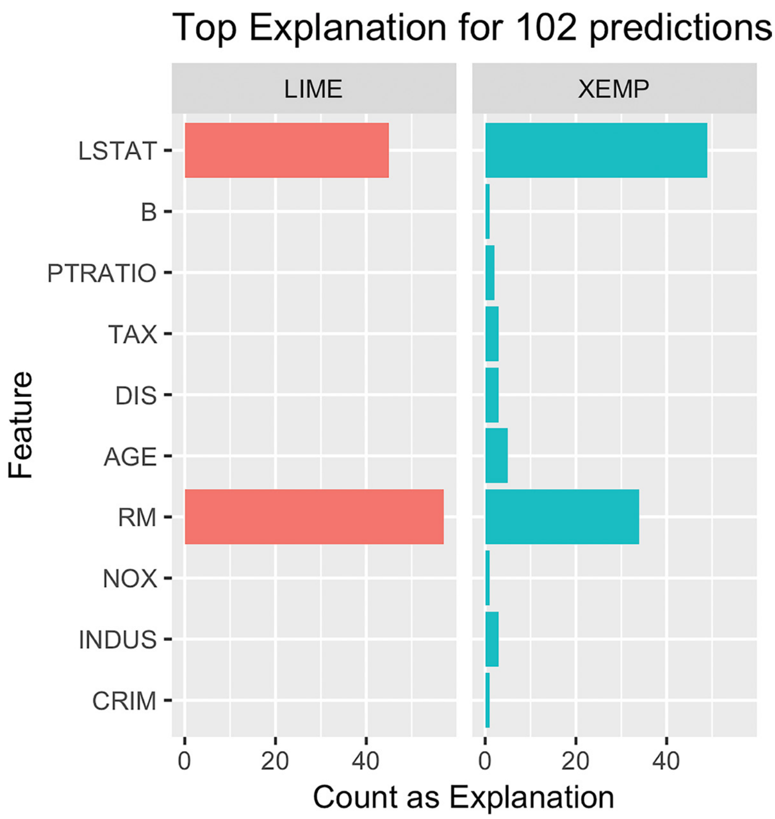
To benchmark both implementations, explanations were run on five hundred examples in the test set. The figures below show the top explanations using both approaches. In the case of LIME, every explanation, all 500 of them, had the feature "capital_gain" as their top explanation. In contrast, for XEMP, we had thirteen different features that sometimes came up as the top explanation. Age and education were the top two explanations.



Clearly, LIME is lacking in diversity of prediction explanations. LIME is favoring the top feature in feature impact for each and every explanation. The lack of diversity draws concerns when using an explanation approach. After all, the goal is to understand why a prediction is being made for a certain example. To have the same explanation for every example isn't very useful. One reason for the lack of diversity could be that capital gain is a zero-inflated feature and LIME synthetic data generation step cannot adequately model zero-inflated distributions.

Clearly, LIME is lacking in diversity of prediction explanations. LIME is favoring the top feature in feature impact for each and every explanation.

This leads us to consider a dataset with no zero-inflated features next. This way, LIME's defects around synthetic data generation are not emphasized. The Boston Housing dataset is another popular benchmarking dataset and the results are shown below. LIME still suffers from a lack of explanation diversity and focuses on a handful of features, while XEMP delivers a much more diverse set of features.

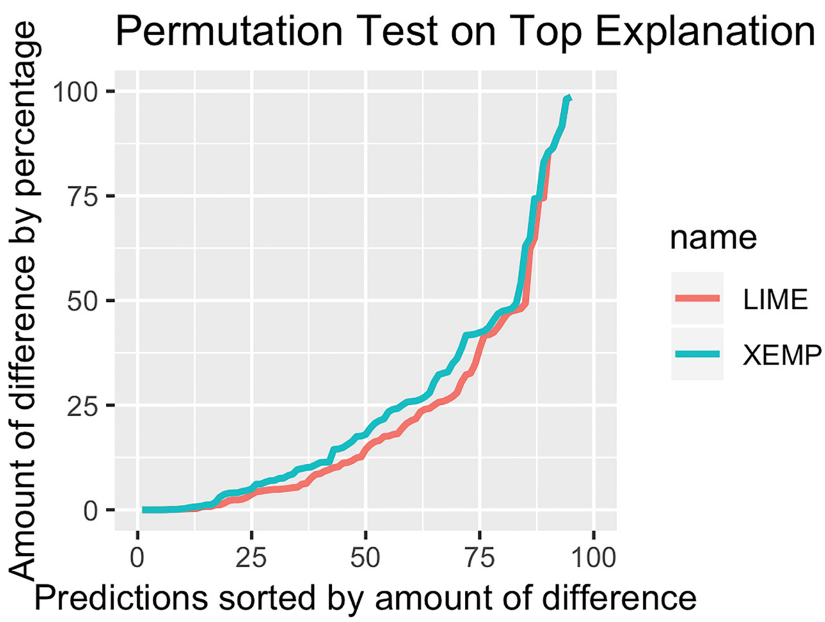


EXPLANATION ACCURACY

The accuracy of the explanations was evaluated using a permutation test with both XEMP and LIME. The permutation test used by Lundberg operates by permuting or changing the value of the top feature in an explanation with a randomly selected value and measuring the difference in predictions. If the explanation is a strong explanatory feature, you would expect a strong effect on the prediction.

The accuracy of the explanations was evaluated using a permutation test with both XEMP and LIME.

The figure below shows the results of the permutation test using the Boston Housing dataset. The graph plots the difference in predictions as a percentage between the original explanation prediction and the permuted explanation prediction for all 101 rows of data, sorted by prediction strength. For most of the predictions, XEMP provides a stronger impact on predictions and, hence, offers a more accurate explanation.



EXPLANATION COMPUTATIONAL SPEED

To evaluate the computational resources required for production explanations, we ran benchmarks on several datasets. Our results consistently found that LIME required substantially more resources than XEMP as shown in the table above. The results shouldn't be surprising if you consider that for every explanation, LIME needs to generate a synthetic dataset, perform 5,000 predictions, and then fit a linear model. In contrast, XEMP for a dataset like Adult with 14 features, XEMP will only need to generate at most a few hundred predictions for every explanation. If you scale this up on the Adult dataset to a one million records, XEMP would take 50 minutes, while LIME would take well over 100 hours.

	Time in Seconds	
	LIME	XEMP
Boston Housing (100 explanations)	42	0.3
Adult (1000 explanations)	423	3

Conclusion

The following chart shows a side-by-side comparison of the LIME versus XEMP prediction explanation methodologies.

Does Your Explainability Method...?	Why it Matters	LIME	XEMP
Work across many types of models	Flexible	✓	✓
Provide consistent explanations	Consistency	✗	✓
Scale in terms of performance	Speed	✗	✓
Does not require subjective hyperparameter tuning	Ease of Use	✗	✓
Explain the deployed model or a surrogate model?	Fidelity	Surrogate	Primary
Answer stakeholder questions?	Communicative	✗	✓
Explanation Diversity Results (# of Top Explanations)	Fidelity		
Adult dataset		1	13
Boston dataset		3	7
Explanation Accuracy Results (higher is better)	Fidelity	646	713
Explanation Computational Speed (sec)	Speed		
Adult dataset (1000 predictions)		423	3
Boston dataset (100 predictions)		43	0.3

As the table above shows, XEMP is generally more suitable and capable than LIME.

1. From a theoretical perspective, XEMP doesn't make assumptions about local linearity.
2. XEMP works directly on the actual model that is deployed, and doesn't rely on a linear approximation model.

3. XEMP is stable and every explanation results in a unique and replicable prediction. You won't get two different explanations for the same prediction.
4. XEMP is easily explainable, answering the questions asked by stakeholders, not using abstract mathematical concepts such as gradients.
5. XEMP provides a greater variety of explanations, keeping fidelity with the individual feature values used within a prediction.
6. XEMP provides stronger explanations. The stronger explanations are very important, because they provide a quantitative basis for evaluating predictions.
7. XEMP is much more scalable, since it is fast, with calculation times being approximately linear with the number of input features. XEMP is an enterprise-ready solution for providing prediction explanations.

DataRobot recommends XEMP for prediction explanations in production environments, due to its scalability, speed, and results in benchmark comparisons.

At DataRobot, we are able to provide human-friendly explanations for all sorts of models and ensembles. We can provide prediction explanations for classification, regression, and even time series models. With DataRobot, you get AI that you can trust.

DataRobot

DataRobot helps enterprises embrace artificial intelligence (AI). Invented by DataRobot, automated machine learning enables organizations to build predictive models that unlock value in data, making machine learning accessible to business analysts and allowing data scientists to accomplish more faster. With DataRobot, organizations become AI-driven and are enabled to automate processes, optimize outcomes, and extract deeper insights.

Learn more at **datarobot.com**