ML Works - XAI Documentation

What is Explainability

Explainable AI (XAI) refers to methods and techniques in the application of artificial intelligence (AI) such that the results of the solution can be understood by humans. It contrasts with the concept of the "black box" in machine learning where even its designers cannot explain why an AI arrived at a specific decision. XAI may be an implementation of the social right to explanation. XAI is relevant even if there is no legal right or regulatory requirement—for example, XAI can improve the user experience of a product or service by helping end-users trust that the AI is making good decisions. This way XAI aims to explain what has been done, what is done right now, what will be done next and unveil the information the actions are based on. These characteristics make it possible to

(i) confirm existing knowledge
(ii) challenge existing knowledge
(iii) generate new assumptions.

Need for Explainability

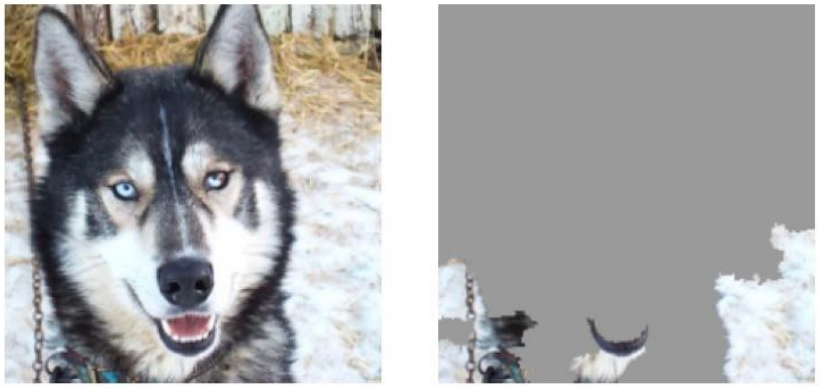
If a machine learning model performs well, why do not we just trust the model and ignore why it made a certain decision? "The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks."

When it comes to predictive modelling, one must make a trade-off:

Is it enough to know only what is predicted? For example, the probability that a customer will churn or how effective some drug will be for a patient. **OR**

Is it worthwhile to possibly bear a loss in performance for better explanation of the prediction?

In some cases, it is enough to know that the predictive performance on a test dataset was good. But in other cases, knowing the 'why' can help learn more about the problem, the data, and the reason why a model might fail. Some models may not require explanations because they are used in a low-risk environment, meaning a mistake will not have serious consequences, (e.g. a movie recommender system) or the method has already been extensively studied and evaluated (e.g. optical character recognition). The need for explainability arises from an incompleteness in problem formalization, which means that for certain problems or tasks it is not enough to get the prediction (the what). The model must also explain how it came to the prediction (the why) because a correct prediction only partially solves the original problem.



(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

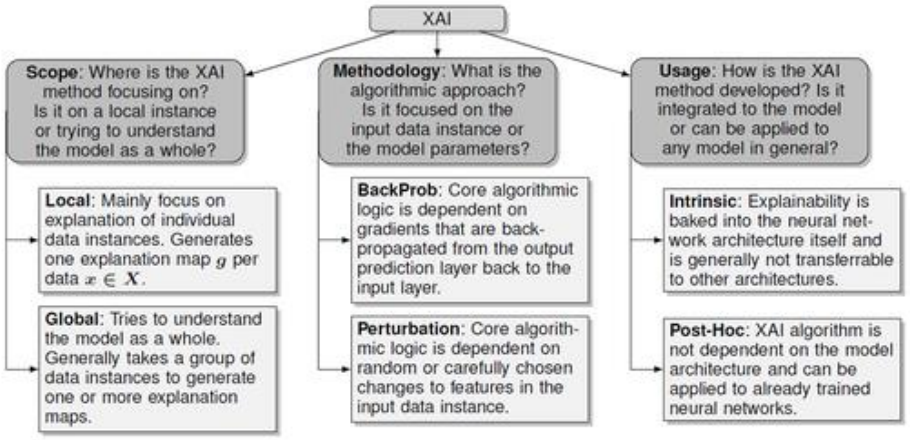
Machine learning models can only be debugged and audited when they can be explained. Even in low-risk environments, such as movie recommendations, the ability to explain is valuable in the research and development phase as well as after deployment. Later, when a model is used in a product, things can go wrong. An explanation for an erroneous prediction helps to understand the cause of the error. It shows a direction to the data scientist & engineer to fix the system.

Consider an example of a husky versus wolf classifier that misclassifies some huskies as wolves (as shown above). Using explainable machine learning methods, you would find that the misclassification was due to the snow on the image. The classifier learned to use snow as a feature for classifying images as "wolf", which might make sense in terms of separating wolves from huskies in the training dataset, but not in real-world use.

At the very least, explainability can facilitate the understanding of various aspects of a model, leading to insights that can be utilized by various stakeholders, such as
• **Data scientists** can be benefited when debugging a model or when looking for ways to improve performance.
• **Business owners** caring about the fit of a model with business strategy and purpose.
• **Model Risk analysts** challenging the model, to check for robustness and approval for deployment.
• **Regulators** inspecting the reliability of a model, as well as the impact of its decisions on the customers.
• **Consumers** requiring transparency about how decisions are taken, and how they could potentially affect them.

Types of Explainability

Methods of ML Explanations can be classified according to various criteria such as **Intrinsic** vs **Post-hoc**, **model-specific** vs **model agnostic** and **global** vs **local**. Let us try to define it based on whether the method explains a model at a global level or does it explain individual prediction instances.



Global level explainability helps to understand the overall distribution of your target outcome based on the features. For example, the value of a house, in general, depends linearly on its size, the higher the size of the house, the higher the price of the house also considering other factors. So, a global explanation would suggest that size of the house has a high impact on predicting the value of the house.

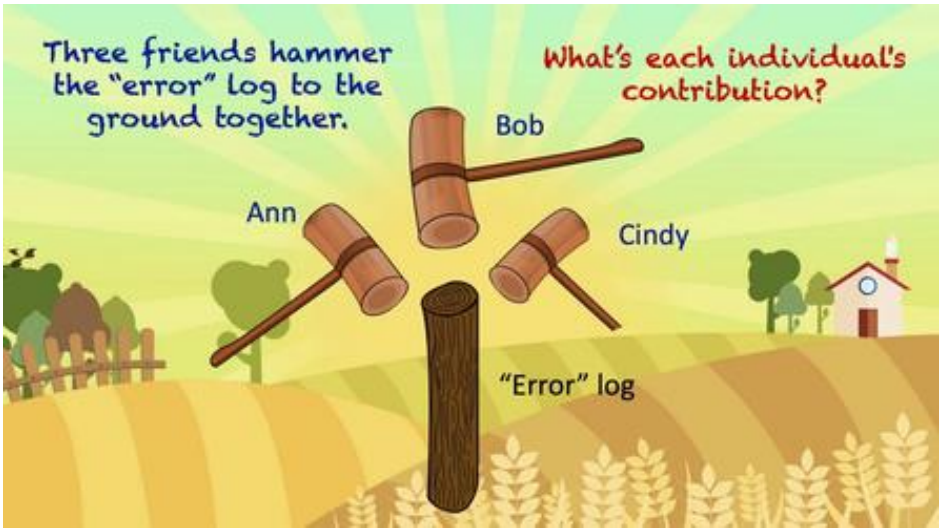
Local level explainability, the prediction might only depend linearly or monotonically on some features, rather than having a complex dependence on all of them. For example, the value of a house although at a global level suggest that high size is equivalent to high value, a house in a locality can have a high size but lower than the average value because of other factors like distance from the city center.

Introducing Shapely Values

In game theory, the Shapley value is a solution concept of fairly distributing both gains and costs to several actors working in a coalition. The Shapley value applies primarily in situations when the contributions of each actor are unequal, but they work in cooperation with each other to obtain the payoff.
"work" here refers to the prediction task, "gains" refers to the difference in actual prediction and average predictions, and "actors" refer to the feature values of the instance.

An intuitive way to understand the Shapley value is the following illustration: The feature values enter a room in random order. All feature values in the room participate in the game (contribute

to the prediction). The Shapley value of a feature value is the average change in the prediction that the coalition already in the room receives when the feature value joins them.



Let's use a story to explain the Shapley value: Assume Ann, Bob, and Cindy together were hammering an "error" wood log, 38 inches, to the ground. After work, they went to a local bar for a drink and Ian, a mathematician, came to join them. Ian asked a very bizarre question: "What is everyone's contribution (in inches)?"

	Marginal contribution			inches
Combination	Ann	Bob	Cindy	Total
A, B, C	2	32	4	38
A, C, B	4	34	0	38
B, A, C	2	32	4	38
B, C, A	0	28	10	38
C, A, B	2	36	0	38
C, B, A	0	28	10	38
Average	2	32	4	38

How to answer this question? Ian listed all the permutations and came up with the data in Table A. When the ordering is A, B, C, the marginal contributions of the three are 2, 32, and 4 inches respectively.

The table shows the coalition of (A, B) or (B, A) is 34 inches, so the marginal contribution of C to this coalition is 4 inches. Ian took the average of all the permutations for each person to get each individual's contribution: Ann is 2 inches, Bob is 32 inches and Cindy is 4 inches. **That's**

the way to calculate the Shapley value: It is the average of the marginal contributions across all permutations. Let's see how it is applied in machine learning.

Let us call the wood log the "error" log for a special reason: It is the loss function in the context of machine learning. The "error" is the difference between the actual value and prediction. The hammers are the predictors to attack the error log. How do we measure the contributions of the hammers (predictors)? The Shapley values!

Shapely values to SHAP

The SHAP (SHapley Additive exPlanations) deserves its own space rather than an extension of the Shapley value. Inspired by several methods on model interpretability, Lundberg and Lee (2016) proposed the SHAP value as a united approach to explaining the output of any machine learning model. Three benefits worth mentioning here.

1. The first one is *global interpretability* — the collective SHAP values can show how much each predictor contributes, either positively or negatively, to the target variable. This is like the variable importance plot but it can show the positive or negative relationship for each variable with the target.
2. The second benefit is *local interpretability* — each observation gets its own set of SHAP values. This greatly increases its transparency. We can explain why a case receives its prediction and the contributions of the predictors. Traditional variable importance algorithms only show the results across the entire population but not in each case. The local interpretability enables us to pinpoint and contrast the impacts of the factors.
3. Third, the SHAP values can be calculated for any tree-based model, while other methods use linear regression or logistic regression models as surrogate models.

Tredence custom SHAP

At this point, it is clear that once we have a model and train/test dataset, we can explain almost all models. But, in industrial setting datasets can range from a few hundred thousand to millions of records and can have various inconsistencies existing in the data like missing values, outliers, etc., hence calculating Shapley values is not always scalable to production datasets.

To evaluate model explanation on real-world datasets at scale, ML Works XAI library has several feature and performance enhancements over the open-source SHAP Explainers to reduce the computation time and improve accuracy of the explanation simultaneously.

The library identifies data issues and notifies the end user, plus is able to handle all model types except Deep Learning and Time Series. Performance has been benchmarked with statistical methods (SSOT) and opensource library with ML Works XAI resulting in more accurate results for every test case.

As speed is a critical aspect in product setting, performance enhancements in ML Works XAI library allow for 10X faster execution, cases where evaluating explanation might take hours is reduced to minutes and minutes to seconds.

The library intelligently chooses the best method to explain the provided model without compromising on speed and without the user worrying about the choice.

Example use case

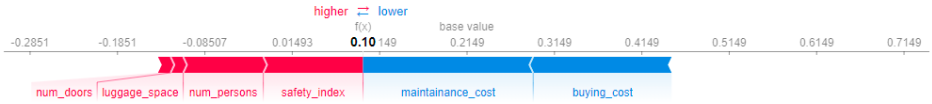
Here a “Car Evaluation data” is used to make a model and explain the predictions. The target value of this dataset is the evaluation result of the car with values as [“unacceptable “, “acceptable”].

The input variables are the qualitative measures of the car such as *buying cost*, *maintenance cost*, *number of doors*, *passenger capacity*, *luggage boot space* and *safety index*. There are 1,728 such samples/datapoints.

Using the above data, a binary classification model is created, and ML Works XAI library is used to compute shapely values to find the most important features at the global and local level.

For example: The following plot for local explanation of an instance (index=42) in the dataset,

Local explanation for class “unacceptable”:



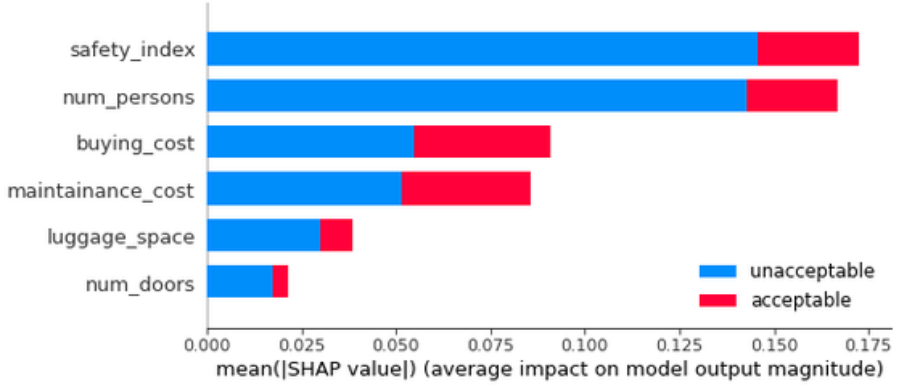
Local explanation for class “acceptable”:

To describe this plot in greater detail:

- The f(x) or *output value* is the prediction for that observation, more specifically predicted probability for that class.
- The *base value*: The original paper [1] explains that the base value E(y_hat) is “the value that would be predicted if all features for the current output were unknown.” In other words, it is the mean prediction, or mean(yhat). One may wonder why it is 0.7053 or 0.2149. This is because the mean prediction probability of Y_test for class “unacceptable“ and “acceptable” is 0.2149 and 0.7053 respectively.
- *Red/blue*: Features that push the prediction probability higher (to the right) are shown in red, and those pushing the prediction lower are in blue.

- *buying_cost*: has a positive impact on the quality rating. The cost of the car is very high which is higher than the average. So it pushes the prediction to the right.
- *safety_index*: has a negative impact on the quality rating. A lower than the average value drives the prediction to the left.
- How to calculate the average values of the predictors. As the ML Works XAI library is built on the training data set. The means of the variables are: `X_train.mean()`

Incase of Global explanation, the absolute shapely values of each feature is averaged over the dataset (as suggested in the X-axis label) to get global feature importance as shown below.



The variable importance here is intuitive, on average feature value of "*safety_index*" contributes (positively or negatively) a value of 0.15 or 15% points to the probability of car belonging to "unacceptable" class. Similarly "*num_doors*" contributes only 0.02 to the model output on average. Interesting point here is once "*safety_index*" is taken into account, the features like "*buying_cost*" and "*maintainance_cost*" have high impact on the quality rating of the car. In case of "acceptable" class, both "*buying_cost*" and "*maintainance_cost*" have highest feature importance within features compared to "unacceptable" class.

The variable importance plot lists the most significant variables in descending order. The top variables contribute more to the model than the bottom ones and thus have high predictive power.

Future Enhancements

ML Works XAI library (community version) alone cannot answer all questions that a stakeholder might be interested in, so to provide a holistic view we have created a suite of libraries (like LIME, Deeplift, etc.) that cater to specific needs. Like SHAP implementation, these libraries have been reverse engineered and customized to handle scale and performance required for an enterprise production deployment. The upgraded ML Works XAI library is available as part of the enterprise version of the platform.

Due to the inherent nature of some models with non-linear decision boundary like SVM, even with all enhancements to speed up the calculation, it might still take some time for larger datasets although still being faster than the native implementation.

Another limitation in terms of models is explaining Time series models, although deep learning-based Time series models are explainable, mainstream models like ARIMA, SARIMAX are difficult to explain.

Thing to always remember

It is important to point out the Shapley values do not provide causality.

This question concerns correlation and causality. **The Shapley values do not identify causality, which is better identified by experimental design or similar approaches**

The Shapley value **can be misinterpreted**. The Shapley value of a feature value is not the difference of the predicted value after removing the feature from the model training. The interpretation of the Shapley value is: Given the current set of feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction is the estimated Shapley value.

Reference

1. "A Unified Approach to Interpreting Model Predictions", Scott M. Lundberg, Su-In Lee, https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf
2. https://github.com/slundberg/shap
3. “Towards A Rigorous Science of Interpretable Machine Learning”, Finale Doshi-Velez, Been Kim, https://arxiv.org/abs/1702.08608
4. https://christophm.github.io/interpretable-ml-book/shap.html