Journal of Machine Learning Research 21 (2020) 1-6 Submitted 12/19; Revised 5/20; Published 6/20

AI Explainability 360: An Extensible Toolkit for Understanding Data and Machine Learning Models

Vijay Arya VIJAY.ARYA@IN.IBM.COM
Rachel K. E. Bellamy RACHEL@US.IBM.COM
Pin-Yu Chen PIN-YU.CHEN@IBM.COM
Amit Dhurandhar ADHURAN@US.IBM.COM
Michael Hind HINDM@US.IBM.COM
Samuel C. Hoffman SHOFFMAN@IBM.COM
Stephanie Houde STEPHANIE.HOUDE@IBM.COM
Q. Vera Liao VERA.LIAO@IBM.COM
Ronny Luss RLUSS@US.IBM.COM
Aleksandra Mojsilović ALEKSAND@US.IBM.COM
Sami Mourad SAMI.MOURAD@IBM.COM
Pablo Pedemonte PPEDEMON@AR.IBM.COM
Ramya Raghavendra RRAGHAV@US.IBM.COM
John T. Richards AJTR@US.IBM.COM
Prasanna Sattigeri PSATTIG@US.IBM.COM
Karthikeyan Shanmugam KARTHIKEYAN.SHANMUGAM2@IBM.COM
Moninder Singh MONINDER@US.IBM.COM
Kush R. Varshney KRVARSHN@US.IBM.COM
Dennis Wei DWEI@US.IBM.COM
Yunfeng Zhang ZHANGYUN@US.IBM.COM
IBM Research

Editor: Alexandre Gramfort

Abstract

As artificial intelligence algorithms make further inroads in high-stakes societal applications, there are increasing calls from multiple stakeholders for these algorithms to explain their outputs. To make matters more challenging, different personas of consumers of explanations have different requirements for explanations. Toward addressing these needs, we introduce AI Explainability 360, an open-source Python toolkit featuring ten diverse and state-of-the-art explainability methods and two evaluation metrics (`http://aix360.mybluemix.net`). Equally important, we provide a taxonomy to help entities requiring explanations to navigate the space of interpretation and explanation methods, not only those in the toolkit but also in the broader literature on explainability. For data scientists and other users of the toolkit, we have implemented an extensible software architecture that organizes methods according to their place in the AI modeling pipeline. The toolkit is not only the software, but also guidance material, tutorials, and an interactive web demo to introduce AI explainability to different audiences. Together, our toolkit and taxonomy can help identify gaps where more explainability methods are needed and provide a platform to incorporate them as they are developed.

Keywords: explainability, interpretability, transparency, taxonomy, open source

1. Introduction

The increasing deployment of artificial intelligence (AI) systems in high stakes domains has been coupled with an increase in societal demands for these systems to provide explanations for their

Toolkit	Data Explanations	Directly Interpretable	Local Post-Hoc	Global Post-Hoc	Self Explaining	Metrics
AIX360	✓	✓	✓	✓	✓	✓
Alibi			✓			
Skater		✓	✓	✓		
H2O		✓	✓	✓		
InterpretML		✓	✓	✓		
EthicalML-XAI				✓		
DALEX			✓	✓		
tf-explain			✓	✓		
iNNvestigate			✓			
modelStudio	✓	✓	✓	✓		
ELI5		✓	✓	✓		
Iml		✓	✓	✓		
Captum			✓			
WIT	✓		✓	✓		

Table 1: Comparison of AI explainability toolkits: Alibi (Klaise et al., 2020), Skater (ska), H2O (h2o), InterpretML (Nori et al., 2019), EthicalML-XAI(eth), DALEX (Biecek, 2018), tf-explain (tfe), iNNvestigate (Alber et al., 2018), modelStudio (Baniecki and Biecek, 2019), ELI5 (eli), Iml (Molnar et al., 2018), Captum (Kokhliyan et al., 2019) and WIT (Wexler et al., 2020). By Self-explaining, we refer to methods which may not be directly interpretable at a global level, but can provide local explanations. By Metrics, we mean methods to quantitatively evaluate explanations.

predictions. However, despite the growing volume of publications, there remains a gap between what society needs and what the research community is producing. One reason for this gap is a lack of a precise definition of an explanation as different people in different settings may require different kinds of explanations. For example, a doctor trying to understand an AI diagnosis of a patient may benefit from seeing known similar cases with the same diagnosis; a denied loan applicant will want to understand the main reasons for their rejection and what can be done to reverse the decision; a regulator, on the other hand, will want to understand the behavior of the system as a whole to ensure that it complies with the law; and a developer may want to understand where the model is more or less confident as a means of improving its performance.

Since there is no single approach to explainable AI that always works best, we require organizing principles for the space of possibilities and tools that bridge the gap from research to practice. In this paper, we provide a taxonomy and describe an open-source toolkit to address the overarching need, taking into account the points of view of many possible consumers of explanations. Our contributions are as follows: 1) *Taxonomy Conception:* We propose a simple yet comprehensive taxonomy of AI explainability that considers varied perspectives. This taxonomy is actionable in that it aids users in choosing an approach for a given application and may also reveal gaps in available explainability techniques. 2) *Taxonomy Implementation:* We architect an application programming interface and extensible toolkit that realizes the taxonomy in software. This effort is non-trivial given the diversity of methods (see Table 1). We have released the toolkit into the open source community under the name AI Explainability 360 (AIX360). 3) *Algorithmic Enhancements:* We take several state-of-the-art interpretability methods from the literature and further develop them algorithmically to make them more appropriate and consumable in practical data science applications (for more details see Arya et al. (2019)). 4) *Educational Material:* We develop demonstrations, tutorials, and other educational material to make the concepts of interpretability and explainability accessible to non-technical stakeholders. The web demo is based on the FICO Explainable Machine Learning Challenge dataset (FICO, 2018) that illustrates the usage of different explainability methods corresponding to the needs of three different stakeholders (data scientist, loan officer and customer). The tutorials cover several problem domains, including lending, health care, and human capital management, and

provide insights into their respective datasets and prediction tasks in addition to their more general educational value.

The current version of the toolkit contains ten explainability algorithms: 1) *Locally Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016):* Learns local explanations by fitting a local sparse linear model. 2) *Shapley Values (SHAP) (Lundberg and Lee, 2017):* Identifies feature importances based on Shapley value estimation. 3) *Disentangled Inferred Prior Variational Autoencoder (DIP-VAE) (Kumar et al., 2018):* Learns high-level independent features from images that may have semantic interpretation. 4) *Boolean Decision Rules via Column Generation (BRCG) (Dash et al., 2018):* Learns a small, interpretable Boolean rule in disjunctive normal form for binary classification. 5) *Contrastive Explanations Method (CEM) (Dhurandhar et al., 2018a):* Generates a local explanation in terms of what is minimally sufficient to maintain the original classification, and also what should be necessarily absent. 6) *ProfWeight (Dhurandhar et al., 2018b):* Learns a reweighting of the training set based on a given interpretable model and a high-performing complex neural network. Retraining of the interpretable model on this reweighted training set is likely to improve the performance of the interpretable model. 7) *Teaching Explanations for Decisions (TED) (Hind et al., 2019):* Learns a predictive model based not only on input-output labels but also on user-provided explanations. For an unseen test instance both a label and explanation are returned. 8) *Generalized Linear Rule Models (GLRM) (Wei et al., 2019):* Learns a linear combination of conjunctions for real-valued regression through a generalized linear model link function (e.g., identity, logit). 9) *ProtoDash (Gurumoorthy et al., 2019):* Selects diverse and representative samples that summarize a dataset or explain a test instance. Non-negative importance weights are also learned for each of the selected samples. 10) *CEM with Monotonic Attribute Functions (CEM-MAF) (Luss et al., 2019):* For complex images, creates contrastive explanations like CEM, but based on high-level semantically meaningful attributes. The toolkit also includes two metrics: Faithfulness (Alvarez-Melis and Jaakkola, 2018) and Monotonicity (Luss et al., 2019), which measure the accuracy and consistency of local feature based explanations.

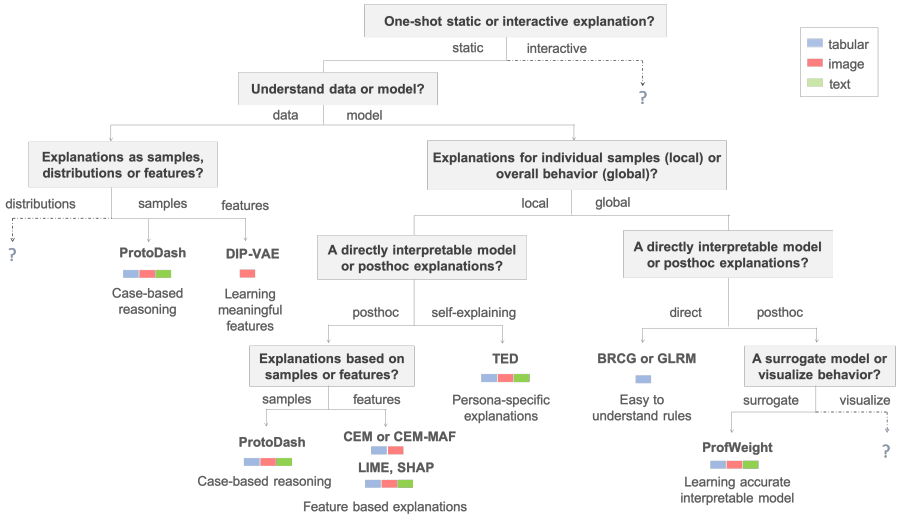

Figure 1: The proposed taxonomy based on questions about what is explained (e.g., data or model), how it is explained (e.g., direct/post-hoc, static/interactive) and at what level (i.e. local/global). The decision tree leaves indicate the methods currently available through the toolkit along with the modalities they can work with. '?' indicates absence of one in the particular category.

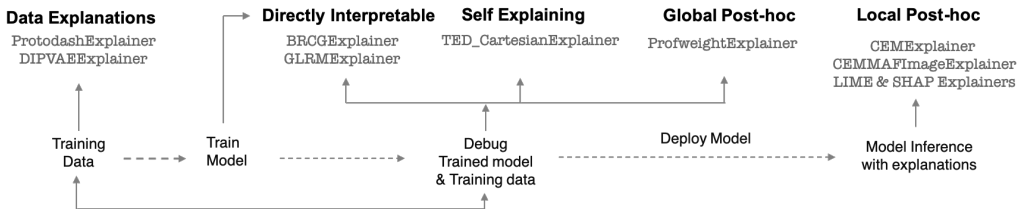

Figure 2: AIX360 class structure organized according to use in the AI pipeline.

2. Taxonomy for AI Explainability and its Implementation

It is important to understand the diverse forms of explanations that are available and the questions that each can address. A proper structuring of the explanation space can have large bearing not just on researchers who design and build new algorithms, but perhaps more importantly on practitioners who want to understand which algorithms might be most suitable for their application. In Figure 1, we provide one such structuring of the explainability space. The structure is in the form of a small decision tree (which is itself considered to be interpretable (Freitas, 2014)). Each node in the tree poses a question to the consumer about the type of explanation required. For users who are not experts in explainable AI, the AIX360 website provides a glossary of terms used in the taxonomy and a guidance document for further context. *Our intention is to propose a simple yet comprehensive taxonomy useful for different types of users at the risk of it being incomplete.*

The AIX360 toolkit aims to provide a unified, flexible, and easy to use programming interface and an associated software architecture to accommodate the diversity of explainability techniques required by various stakeholders. The goal is to be amenable both to data scientists, who may not be experts in explainability, as well as algorithm developers. Toward this end, we make use of a intuitive programming interface that is similar to popular Python model development tools (e.g., scikit-learn) and construct a hierarchy of Python classes corresponding to explainers for data, models, and predictions which expose common methods to users. Algorithm developers can inherit from a family of base classes to integrate new explainability algorithms. We have organized the classes based on their use in different stages of the AI pipeline shown in Figure 2. For instance, the `CEMExplainer` inherits from the base class `LocalWBExplainer` (i.e. local post-hoc white-box explainer), whereas `BRCGExplainer` inherits from `DISExplainer` (directly interpretable supervised explainer). In fact, in the 2^{nd} release of our toolkit LIME and multiple variants of SHAP were seamlessly integrated leveraging our class hierarchy. AIX360 also includes dataset classes to facilitate loading and processing of commonly used datasets so that users can easily experiment with the implemented algorithms.

References

ELI5: Explain like I'm five. GitHub repository, `https://github.com/TeamHG-Memex/eli5`.

EthicalML-XAI: An explainability toolbox for machine learning. GitHub repository, `https://github.com/EthicalML/xai`.

H2O.ai: Machine learning interpretability resources. GitHub repository, `https://github.com/h2oai/mli-resources`.

Skater: Python library for model interpretation/explanations. GitHub repository, `https://github.com/oracle/Skater`.

tf-explain: Interpretability methods for tf.keras models with Tensorflow 2.0. GitHub repository, `https://github.com/sicara/tf-explain`.

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Grégoire Montavon Kristof T. Schütt, Wojciech Samek, Sven Dähne Klaus-Robert Müller, and Pieter-Jan Kindermans. iNNvestigate neural networks! arXiv:1808.04260, 2018.

David Alvarez-Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pages 7775–7784. 2018.

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. arXiv:1909.03012, 2019.

Hubert Baniecki and Przemyslaw Biecek. modelStudio: Interactive studio with explanations for ML predictive models. *Journal of Open Source Software*, 4(43):1798, 2019.

Przemysław Biecek. DALEX: Explainers for complex predictive models in R. *Journal of Machine Learning Research*, 19(84):1–5, 2018.

Sanjeeb Dash, Oktay Günlük, and Dennis Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, pages 4655–4665, 2018.

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018a.

Amit Dhurandhar, Karthikeyan Shanmugam, Ronny Luss, and Peder Olsen. Improving simple models with confidence profiles. In *Advances in Neural Information Processing Systems*, pages 10296–10306, 2018b.

FICO. FICO Explainable Machine Learning Challenge. `https://community.fico.com/s/explainable-machine-learning-challenge`, 2018.

Alex A. Freitas. Comprehensible classification models — a position paper. *ACM SIGKDD Explorations*, 15(1):1–10, 2014.

Karthik Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. Efficient data representation by selecting prototypes with importance weights. In *Proceedings of the IEEE International Conference on Data Mining*, 2019.

Michael Hind, Dennis Wei, Murray Campbell, Noel C. F. Codella, Amit Dhurandhar, Aleksandra Mojsilovic, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. TED: Teaching AI to explain its decisions. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2019.

Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. Alibi: Algorithms for monitoring and explaining machine learning models, February 2020. URL `https://github.com/SeldonIO/alibi`.

Narine Kokhliyan, Edward Wang, Vivek Miglani, and Orion Richardson. Captum. In *https://github.com/pytorch/captum*, 2019.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *Proceedings of the International Conference on Learning Representations*, 2018.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Karthik Shanmugam, and Chun-Chen Tu. Generating contrastive explanations with monotonic attribute functions. arXiv:1905.12698, 2019.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. iml: An r package for interpretable machine learning. *Journal of Open Source Software*, 3(26):786, 2018. doi: 10.21105/joss.00786. URL `https://doi.org/10.21105/joss.00786`.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. arXiv:1909.09223, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Günlük. Generalized linear rule models. In *Proceedings of the International Conference on Machine Learning*, pages 6687–6696, 2019.

J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.