AI Explainability 360 Toolkit

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, Yunfeng Zhang
IBM Research AI

ABSTRACT

As machine learning algorithms make inroads into our lives and society, calls are increasing from multiple stakeholders for these algorithms to explain their outputs. Moreover, these stakeholders, whether they be government regulators, affected citizens, domain experts, or developers, present different requirements for explanations. To address these needs, we introduce AI Explainability 360[1], an open-source software toolkit featuring eight diverse state-of-the-art explainability methods, two evaluation metrics, and an extensible software architecture that organizes these methods according to their use in the AI modeling pipeline. Additionally, we have implemented enhancements to bring research innovations closer to consumers of explanations, ranging from simplified, accessible versions of algorithms to guidance material to help users navigate the space of explanations along with tutorials and an interactive web demo to introduce AI explainability to practitioners. Together, our toolkit can help improve transparency of machine learning models and provides a platform to integrate new explainability techniques as they are developed.

CCS CONCEPTS

• **Software and its engineering** → *Software libraries and repositories.*

KEYWORDS

AI, Explainability, Trust, Opensource

ACM Reference Format:
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, and Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, Yunfeng Zhang. 2021. AI Explainability 360 Toolkit. In *8th ACM IKDD CODS and 26th COMAD (CODS COMAD 2021), January 2–4, 2021, Bangalore, India.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3430984.3430987

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 predictions. However, many machine learning techniques are not easily explainable, even by experts in the field. This has led to a growing research community, with a long history, focusing on "interpretable" or "explainable" machine learning techniques. 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 work, we present an open-source toolkit to address this overarching need, taking into account the points of view of many possible consumers of explanations. Our contributions are as follows:

- We have implemented an application programming interface and extensible toolkit that organizes various explainability techniques according to their use in AI modeling pipeline. We have released the toolkit into the open source community under the name AI Explainability 360 (AIX360) [9]. It is the most comprehensive explainability toolkit across different ways of explaining (see Table 1). Additionally, we have algorithmically extended several state-of-the-art interpretability methods to make them consumable in practical data science applications.
- We have developed demonstrations, tutorials, guidance material, and other educational content to make the concepts of interpretability and explainability accessible to non-technical stakeholders. The tutorials cover several problem domains, including finance, healthcare, and human capital management, and provide insights into their respective datasets and prediction tasks in addition to their more general educational value.

[1] The web demo, tutorials, notebooks, guidance material as well as a link to the github repository (https://github.com/Trusted-AI/AIX360) are available at http://aix360.mybluemix.net/. AI Explainability 360 tutorial presented at ACM FAT* 2020 can be accessed at: https://facctconference.org/2020/acceptedtuts.html

CODS COMAD 2021, January 2–4, 2021, Bangalore, India

ACM ISBN 978-1-4503-8817-7/21/01.
https://doi.org/10.1145/3430984.3430987

Toolkit	Data Explanations	Directly Interpretable	Local Post-Hoc	Global Post-Hoc	Persona-Specific Explanations	Metrics
AIX360 [9]	✓	✓	✓	✓	✓	✓
Alibi [17]			✓			
Skater [4]		✓	✓	✓		
H2O [3]		✓	✓	✓		
InterpretML [22]		✓	✓	✓		
EthicalML-XAI [2]				✓		
DALEX [10]			✓	✓		
tf-explain [5]			✓	✓		
iNNvestigate [7]			✓			
modelStudio	✓	✓	✓	✓		
ELI5 [1]		✓	✓	✓		
Iml [21]		✓	✓	✓		
Captum [18]			✓			
WIT [24]	✓		✓	✓		

Table 1: Comparison of AI explainability toolkits.

2 TOOLKIT

The current version of the toolkit contains implementations of eight explainability algorithms which cover different types of explanations relevant to machine learning models and their datasets, as shown in Table 1. In this context, a *local* explanation is for a single prediction of a machine learning model, whereas a *global* explanation describes the behavior of the entire model. A *directly interpretable* model is one that is intrinsically transparent and understandable by most consumers (e.g. small decision trees), whereas a *post-hoc* explanation generally involves training a directly interpretable surrogate model in order to explain a more complex source model or its prediction after it has been trained. A *data* explanation refers to an explanation concerning the dataset used to train a model, for instance a few prototypes which summarize the dataset. A *persona-Specific* explanation refers to local or global explanations tailored for specific types of users. More specifically, the algorithms implemented in the toolkit are as follows:

- *Boolean Decision Rules via Column Generation [11] (Implemented as BRCGExplainer, directly interpretable):* Learns a small, interpretable Boolean rule in disjunctive normal form (DNF) for binary classification.
- *Generalized Linear Rule Models [23] (GLRMExplainer, directly interpretable):* Learns a linear combination of conjunctions for real-valued regression through a generalized linear model (GLM) link function (e.g., identity, logit).
- *ProtoDash [15] (ProtodashExplainer, data explanations):* 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.
- *ProfWeight [13] (ProfWeightExplainer, global explanation):* 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.
- *Teaching Explanations for Decisions [16] (TEDExplainer, Persona-specific explanation):* 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.
- *Contrastive Explanations Method [12] (CEMExplainer, local explanation) :* Generates a local explanation in terms of what is minimally sufficient to maintain the original classification, and also what should be necessarily absent.
- *Contrastive Explanations Method with Monotonic Attribute Functions [20] (CEMMAFImageExplainer, local explanation):* For complex images, creates contrastive explanations like CEM above but based on high-level semantically meaningful attributes.
- *Disentangled Inferred Prior Variational Autoencoder [19] (DIPVAE-Explainer, data explanation):* Learns high-level independent features from images that possibly have semantic interpretation.
- *Explainability Metrics*: The toolkit includes two metrics from the explainability literature: Faithfulness [8] and Monotonicity [20], which help measure the fidelity of the generated explanations.

The AIX360 toolkit provides 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 scientist users, who may not be experts in explainability, as well as algorithm developers. Toward this end, we make use of a 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. Algorithm developers can inherit from a family of base class explainers in order to integrate new explainability algorithms. Listing 1 presents a glimpse of the AIX360 API from the perspective of an end-user who uses the CEMExplainer to obtain local explanations corresponding to predictions made by an MNIST model.

Listing 1: Python code snippet illustrating the use of AIX360

```python
from aix360.algorithms.contrastive import CEMExplainer
from aix360.datasets import MNISTDataset
# load normalized MNIST data
data = MNISTDataset()
# Instantiate a local post-hoc explainer
explainer = CEMExplainer(mnist_model)
# chose an input image
img = np.expand_dims(data.test_data[0], axis=0)
# obtain local explanations for an image
(pn_image, _, _) = explainer.explain_instance(img,
    arg_mode='PN")
```

3 WEB DEMO, TUTORIALS AND RESOURCES

AIX360 was developed with the goal of providing accessible resources on explainability to non-technical stakeholders. Therefore, we include multiple educational materials to both introduce the explainability algorithms provided by AIX360 and to demonstrate how different explainability methods can be applied in real-world scenarios. These educational materials includes a simple yet comprehensive taxonomy[2] that structures algorithms in the explainability space and serves multiple purposes, including guiding practitioners to choose the right explanation methods for their use case, revealing algorithmic gaps to researchers, and informing design choices related to explainability software (e.g., class hierarchy) for data scientists and developers. An interactive web demo[3] has been developed based on the FICO Explainable Machine Learning Challenge dataset [14], a real-world scenario where a machine learning system is used to support decisions on loan applications by predicting the repayment risk of applicants. The demo highlights that three groups of people – data scientists, loan officers, and bank customers – are involved in the scenario and their needs are best served by different explainability methods. For example, although the data scientist may demand a global understanding of model behavior through an interpretable model, which can be provided by the GLRMExplainer, a bank customer would ask for justification for their loan application results, which can be generated by the CEMExplainer. We use storytelling and visual illustrations to guide users of AIX360 through these scenarios of different explainability consumers.

The AIX360 toolkit currently includes five tutorials in the form of Jupyter notebooks that show data scientists and other developers how to use different explanation methods across several application domains. The tutorials thus serve as an educational tool and potential gateway to AI explainability for practitioners in these domains. The tutorials cover:

- Using three different methods to explain a credit approval model to three types of consumers, based on the FICO Explainable Machine Learning Challenge dataset [14].
- Creating directly interpretable healthcare cost prediction models for a care management scenario using Medical Expenditure Panel Survey data [6].
- Exploring dermoscopic image datasets used to train machine learning models that help physicians diagnose skin diseases.
- Understanding National Health and Nutrition Examination Survey datasets to support research in epidemiology and health policy.
- Explaining predictions of a model that recommends employees for retention actions from a synthesized human resources dataset.

Beyond illustrating the application of different methods, the tutorials also provide considerable insight into the datasets that are used and to the extent that these insights generalize into the respective problem domains. These insights are a natural consequence of using explainable machine learning and could be of independent interest.

4 DISCUSSION

In summary, this demo aims to introduce the open-source AI Explainability 360 (AIX360) toolkit, which contains eight state-of-the-art explainability algorithms that can explain an AI model or a complex dataset in different ways to a diverse set of users. Some of the algorithms were enhanced from their published versions for AIX360 to make them more usable and their outputs easier to consume. The toolkit also contains two explainability metrics, a credit approval demonstration, five elaborate tutorials covering different real-world use cases, and 13 Jupyter notebooks, making it accessible to practitioners and non-experts. These resources, together with the breadth of the toolkit and its common extensible programming interface, help distinguish AIX360 from existing explainability toolkits. We believe that the toolkit provides technical, educational, and operational benefits to the community over and above what currently exists. As AI surges forward with trust being one of the critical bottlenecks in its widespread adoption, we hope that the community will leverage and significantly enhance the AIX360 toolkit and enable it to grow.

REFERENCES

[1] [n.d.]. ELI5: Explain Like I'm Five. GitHub repository, https://github.com/TeamHG-Memex/eli5.

[2] [n.d.]. EthicalML-XAI: An eXplainability toolbox for machine learning. GitHub repository, https://github.com/EthicalML/xai.

[3] [n.d.]. H2O.ai: Machine Learning Interpretability Resources. GitHub repository, https://github.com/h2oai/mli-resources.

[4] [n.d.]. Skater: Python Library for Model Interpretation/Explanations. GitHub repository, https://github.com/oracle/Skater.

[5] [n.d.]. tf-explain: Interpretability Methods for tf.keras models with Tensorflow 2.0. GitHub repository, https://github.com/sicara/tf-explain.

[6] Agency for Healthcare Research and Quality. 2019. Medical Expenditure Panel Survey (MEPS). https://meps.ahrq.gov/mepsweb/. Last accessed 2019-08.

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

[8] David Alvarez-Melis and Tommi Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems*. 7775–7784.

[9] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra MojsiloviÄ‡, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John T. Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2020. AI Explainability 360: An Extensible Toolkit for Understanding Data and Machine Learning Models. *Journal of Machine Learning Research* 21, 130 (2020), 1–6. http://jmlr.org/papers/v21/19-1035.html

[10] Przemysław Biecek. 2018. DALEX: Explainers for Complex Predictive Models in R. *Journal of Machine Learning Research* 19, 84 (2018), 1–5.

[11] Sanjeeb Dash, Oktay Günlük, and Dennis Wei. 2018. Boolean Decision Rules via Column Generation. In *Advances in Neural Information Processing Systems*. 4655–4665.

[12] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. 2018. Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. In *Advances in Neural Information Processing Systems*. 592–603.

[13] Amit Dhurandhar, Karthikeyan Shanmugam, Ronny Luss, and Peder Olsen. 2018. Improving Simple Models with Confidence Profiles. In *Advances in Neural Information Processing Systems*. 10296–10306.

[14] FICO. 2018. FICO Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge.

[15] Karthik Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. 2019. Efficient Data Representation by Selecting Prototypes with Importance Weights. In *Proceedings of the IEEE International Conference on Data Mining*.

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

[2] https://github.com/Trusted-AI/AIX360/tree/master/aix360/algorithms

[3] https://aix360.mybluemix.net/

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

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

[19] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2018. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In *Proceedings of the International Conference on Learning Representations*.

[20] Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Karthik Shanmugam, and Chun-Chen Tu. 2019. Generating Contrastive Explanations with Monotonic Attribute Functions. arXiv:1905.12698.

[21] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. 2018. iml: An R package for Interpretable Machine Learning. *Journal of Open Source Software* 3, 26 (2018), 786. https://doi.org/10.21105/joss.00786

[22] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. InterpretML: A Unified Framework for Machine Learning Interpretability. arXiv:1909.09223.

[23] Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Günlük. 2019. Generalized Linear Rule Models. In *Proceedings of the International Conference on Machine Learning*. 6687–6696.

[24] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. 2020. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 56–65.