XAI Toolsheet: Towards A Documentation Framework for XAI Tools

Surya Karunagaran[1,] Ana Lucic[1], Christine Custis[1]
[1] Partnership on AI
surya@partnershiponai.org, ana@partnershiponai.org, christine@partnershiponai.org

Abstract

The landscape of Explainable AI (XAI) tools, which are practical implementations of XAI algorithms, has substantially grown in recent years. However, information about the XAI tools is spread across several academic and non-academic outlets, making it difficult and time-consuming to adopt an XAI tool from the vast array of options. We introduce the "XAI Toolsheet," a documentation framework with 22 tool features based on a qualitative analysis of 152 XAI tools as an effective means to assess XAI tools. The proposed tool features are classified under three categories: metadata, utility, and usability to assess the functionality and usefulness of various XAI tools. XAI Toolsheet can aid (a) XAI tool developers to evaluate their tools along proposed dimensions critically and (b) XAI tool users to quickly compare the capabilities and limitations of XAI tools and aid in their decision-making process. Our goal is for the XAI Toolsheet to become a standard documentation practice for the ML community. We practically demonstrate the usefulness of our framework by presenting instantiations of the XAI Toolsheet for two different XAI tools.

1 Introduction

There is a growing demand for Machine learning (ML) models to be explainable due to several instances of harm caused by ML systems in real-world applications [Arya et al., 2019]. There is a lack of consensus around the definition of explainability, but a common theme from XAI studies considers explainability to encompass any technical means to make AI understandable by humans [McGregor, 2020]. The most notable contributions to making ML model behavior interpretable and understandable to humans come from the subfield of Explainable AI (XAI) [Hall et al., 2019], where the focus of existing research is around creating new explainability algorithms [Kaur et al., 2020]. In AI, an algorithm refers to *"a set of step-by-step instructions...the algorithm tells the machine how to find answers to a question or solutions to a problem"* [Kingman et al., 2022]. But, in the recent years, there has been a wave of new XAI algorithmic tools that implement the XAI algorithms and methods developed by XAI researchers. An algorithmic tool is defined as "a product, application, or device that supports or solves a specific problem, using complex algorithms" [Kingman et al., 2022]. Similarly, XAI algorithmic tools aim to support the explainability of ML systems. Every XAI algorithmic tool has an underlying XAI algorithm, but every XAI algorithm is not necessarily an XAI tool by it-self. For the remainder of the paper, we will refer to XAI algorithmic tools as "XAI tools".

XAI tools provide capabilities that enable the transition of explainability from theory to practice. However, the information about XAI tools spans across multiple academic and non-academic outlets. A preliminary search yielded 152 XAI tools; choosing from such a wide range of toolsets can be difficult and time-consuming for the intended tool users. Further, the choice of XAI tools will largely depend on the application domain and user preferences [Morley et al., 2021]. Thus, selecting an XAI tool can be seen as a classic case of the psychological phenomenon "choice overload": a situation where many equivalent or similar choices lead to dissatisfaction with or avoidance [Iyengar and Lepper, 2000]. Since no single tool offers a one-size-fits-all solution [Arya et al., 2019; Richard et al., 2018], there is a need for a practical framework that will offer guidance for optimal tool selection. Also, recently several government agencies worldwide are enacting policies to document information about the algorithmic tools they use [Floridi, 2020; Kingman et al., 2022]. Such efforts are directed toward increasing the transparency and accountability of algorithm-assisted decisions. However, there is no consensus about the tool information that needs to be documented. Hence, we address the following research question (RQ):

- ***RQ****: What are the relevant tool features that needs to be documented to assess the functionality and use-fulness of different XAI tools effectively?*

Our aim is to facilitate the discussion regarding the desired tool properties by introducing “XAI Toolsheet” framework consisting of 22 tool features to evaluate different XAI tools. The rest of the paper is organized as follows: we begin with an overview of related work that has inspired our paper, then we explain 22 identified tool features, which we will refer to as XAI Toolsheet dimensions. Afterwards, we show the practical use of XAI Toolsheet by comparing two

XAI tools and conclude with discussing the future scope of research. From here on, we will use terms “tools" and "XAI tools" interchangeably to enhance readability of the paper.

2 Related Work

Most of the existing work is around describing the advantages and disadvantages of XAI algorithms or techniques [Erickson et al., 2017; Morley et al., 202]. Only few studies inform about the XAI tools. For instance, Hohman et al. [2018] surveyed the state of deep learning research visualization techniques and summarized some of the open-source deep learning tools. There is a lack of comprehensive research on the tool properties that would assist a practitioner in choosing one for their desired task. We aim to fill this gap by introducing the XAI Toolsheet, which contains a set of tool criteria to guide interested users in their tool selection process.

The framework most similar to ours is that of Sokol and Flach [2020], they propose a list of characteristics to evaluate XAI algorithms in a self-reporting format called "Explainability Factsheets". It differs from XAI Toolsheet since we focus on detailing the components of implementation tools, not on assessing the quality of explanations provided by the XAI algorithms.

2.1 AI Documentation Efforts

Our inspiration to create XAI Toolsheet as a documentation template stems from the existing documentation efforts that advocates for standardizing and sharing information about datasets and models used in the development of AI solutions. Some prominent data documentation efforts to capture the potential bias and intended uses includes Gebru et al.’s [2021] Datasheets for datasets, which proposes a standardized template for public datasets, Bender and Friedman’s [2018] Data Statement schema for datasets used in natural language processing (NLP), and Holland et al.’s [2018] Data Nutrition label that provides a standardized view of the core components of a dataset. Other documentation tools focus on the model, such as the Model Card proposed by Mitchell et al. [2019] which conveys information about the intended use and context of the ML model. Our proposal is distinguished from prior work in that we focus on tool-specific criteria as opposed to properties specific to datasets or ML models.

3 Methodology

We address our research question (**RQ**) by applying qualitative document analysis method on 152 tools that were made public between 2015 and 2021 to come up with themes and subthemes. The relevant documents include research papers from academic databases [ScienceDirect, IEEE Xplore, SpringerLink, ACM Digital Library, Google Scholar, arXiv, Scopus]. Although most of tools included for the study have a corresponding research paper, some are only accompanied by a GitHub Readme file or a blog post. (See Appendix A for the tools cited in this paper and Appendix B for instructions on how to obtain the complete tool list and survey/review method that are used to create the framework). We include these outlets because of the rapid growth of XAI tools and the lack of a perfect fit for publishing and disseminating work in this area, therefore, the inclusion of these non-traditional sources is important to review, as they are highly influential and impactful to the field. We identified 22 tool features categorized under 3 dimensions: metadata (Section 4), utility (Section 5), and usability (Section 6). We describe each features functionality and the corresponding rationale for its inclusion in the framework and summarize any problems, suggestions, and opportunities for future research.

4 Metadata

This category is based on the concept of metadata from information systems and is most commonly defined as "data about data" [Riley, 2017]. Metadata is a short explanation or summary of basic information about an artifact. In the context of our study, we propose 8 dimensions under metadata to summarize basic information about the tool that will make it easier for the users to find and use the tool.

4.1 Tool Type

This refers to the deployment format of the tool that offers explanation capabilities. Since each tool type requires varying degrees of technical expertise, as well as varying integration and interoperability requirements, this information will allow the tool users to assess the technical capabilities and needs before integrating the tool into their project. Based on our analysis, we classify types of tools into the following categories:

- **Packages or Libraries**: programming language-specific tools (e.g., Python, R, Java, C++) that provide explainability capabilities for an ML project. Some of the packages (e.g., Manifold) offer visualizations as web applications, while others are in the form of visualization libraries (ELI 5)
- **Platforms**: software products that offer explainability as a feature in the product. Platforms are used to complete ML projects from beginning to end and typically support the project through the stages of data analysis, data preparation, modeling and evaluation, and deployment.
- **Visual analytics systems**: graphical user interfaces in the form of dashboards, which often include point and click features that help the user understand the model’s behavior (e.g., Google What-if tool).

We observe XAI tools are developed primarily to help model developers or AI researchers to inspect and debug the model which is in-line with existing research, and it may be challenging for non-technical stakeholders to use these tools directly. It is worth mentioning that the majority of XAI tools focus on ML model explanations and only a few (e.g., H20 Driverless AI, Shapash, AI explainability 360) provide explanations of the data as well as models.

We also observe that tool developers use multiple naming conventions for their tools, the most used terms include: “toolkits”, “platforms”, “library”, “packages”, “framework” and “visual analytics systems”. Tool developers typically do not offer an explanation as to why they use a particular naming convention and some use multiple terms for the same tool type. We suggest that the XAI tool community adopts a consistent naming convention (i.e., the one suggested in this subsection) in order to minimize confusion for both technical and non-technical stakeholders.

4.2 Type of Tool Developers

This dimension contains information about the identity of the tool creators. This information may help tool users to assess the credibility of tools since each creator might have varying levels of AI experience, knowledge, and technical capabilities. We broadly group tool developers into three types:

- **Companies**: large corporations which have a division dedicated to AI (e.g., Microsoft, Google, IBM, Uber, Amazon, Oracle, Facebook), AI-specific companies (e.g., Data robot, H20, Databricks, Alteryx])
- **Individual contributors**: such as academic researchers (mostly computer science PhD students), industry employees (large corporations, startups, research labs).
- **Universities**: academic institutions that focus on research (e.g., University of Washington, Stanford).

4.3 Tool Users

This dimension gives information about the intended users of the tool. The explainability needs of tool users can vary significantly depending on their goals, back-grounds, usage contexts, and other factors [Arya et al., 2019]. For example, technical stakeholders may want to dive deeper into the performance of a system and determine whether it works as intended. In contrast, business-oriented users may require the tools to offer non-technical information along with some summary statistics. We categorize the intended tools users as,

- **Technical Users:** such as data scientists, ML engineers, ML researchers, etc. These stakeholders often use explanations to debug or uncover issues with the model and explain the model to other stakeholders.
- **Non-technical Users**: include (a) business users, (b) impacted groups, and (c) regulatory bodies. Business users typically use explanations to make informed decisions about AI decision support systems (e.g., medical doctors, loan officers, judges, or hiring managers). Usually, these users are not experts regarding the technical details about the functioning of the AI systems they use. Impacted groups are individuals whose lives could be impacted by the AI. They typically use explanations to seek recourse or contest the AI. Possible examples include patients, job or loan applicants. Regulatory bodies are institutions who audit for legal or ethical concerns such as fairness, safety, or privacy.

The majority of tools were developed primarily to help technical stakeholders to inspect the model. We recommend that more research is needed to understand if and how each stakeholder benefits from the explanations offered by the tools.

4.4 Tool Developed Year

This provides information about the tool created year. Recency can play a role in the adoption of new technology, since some users tend to associate recency with state-of-the-art capabilities and adopt new tools right away, while others may wait until the tool gains traction before adopting it. We anticipate tha the tool created year to be a useful filter during tool selection process.

4.5 Type of Tool Access

This dimension provides the cost and support information associated with the tool. The two major types of access are,

- **Open source**: is free to use and offers open collaboration. It is free for anyone to download and use. There are often various types of support offered, such as de-tailed documentation, forums, wikis, newsgroups, email lists and slack channels.
- **Proprietary**: is copyrighted with no open access and limited flexibility. Proprietary tools are typically not free to use.

We find GitHub to be the main destination where open-source tools are published. We find that the majority of XAI tools are developed as open-source and are managed by a distributed community of developers who cooperatively improve and support the source code, often without remuneration. We also observe that some tools are initially built as proprietary solutions but are later released as open source (e.g., Manifold. Similarly, there are distributors of open-source packages who also offer a for-profit, licensed and proprietary version built upon the original opensource platform (e.g., H20 driverless AI).

4.6 Tool License

Tool license is a way for users to gain access to tools while ownership rights remain with the tool developer. Each opensource tool has an associated open-source license, which is a legal and binding contract between the developer and the user of a tool, declaring that the tool can be used in commercial applications under specified conditions. The most common licenses used are as follows:

- **MIT:** is by the Massachusetts Institute of Technology. It allows for free use, modification, and distribution of the tool as long as a copy of the original MIT license and copyright notice is added to it.
- **Apache:** is by the Apache Software Foundation (ASF). It allows for free use, modification, and distribution of the tool as long as the user follows the terms of the Apache License. Most patents are licensed under the Apache license.
- **Berkeley Software Distribution (BSD)**: has a family of permissive free software licenses that lets the user freely modify and distribute the tool as the user retains a copy of the copyright notice, list of conditions, and the disclaimer.

The important difference to note is that the Apache license mandates that changes made to the source code may be documented, which is not the case with the MIT or BSD licenses. Further study is needed to understand the impact of license types in the tool selection process.

4.7 Tool Documentation

This feature refers to documentation detailing the installation and configuration requirements. This includes documentation about the code, APIs, release notes, and design specifications to ensure tool compatibility for the use case at hand. Tool documentation aids users in discovering gaps between their requirements and the tools' capabilities. Most documentation of open-source tools is published using an open-source software documentation hosting platform called "Read the Docs". It generates documentation written with the Sphinx documentation generator. The Sphinx theme is meant to provide a better reader experience for documentation users on both desktop and mobile devices. Whenever there is an update pushed in a GitHub repository, "Read the Docs" will automatically synchronize the code and documentation.

4.8 Tool Compatibility

Compatibility refers to the extent to which an XAI tool can integrate with ML models developed in various environments. This dimension can help users assess their tool integration requirements and capabilities, since incompatible integration could cause significant inconvenience or non-usage of the tool, despite its functionality. The most common forms of tool integration are with models developed in:

- ML platform environments that offer end-to-end functionalities, including both data and model exploration and validation [e.g., Manifold].
- Modular libraries such as PyTorch [Paszke et al., 2019]
- Notebook environments such as Jupyter, Colab [e.g., What-If Tool]
- Cloud-based environments such as AZURE, Google cloud [e.g., Language Interpretability Tool].

We also observe that certain tools are compatible with specific library versions.

5 Utility Dimensions

We adopt the UX (User Experience) design concept of utility, defined as providing functionalities that users need to successfully perform a task in hand [Ann, 2019]. Similarly, the main function of XAI tool is to successfully integrate XAI methods to tool users' projects. We propose the following 8 features as utility dimensions of XAI tools.

5.1 Type of Dataset

At a high level, there are two broad categories of data: structured and unstructured. Structured data typically comes in a tabular format where each row corresponds to a data point and each column corresponds to a feature. In contrast, the format of unstructured data is less explicit and comes in forms such as text, audio, images, or videos. Our survey indicates that tabular, text, and image/video data are the most common types of data types supported by current XAI tools. We find that tool developers do not always explicitly state which types of data are supported by their tools, and we suggest that tool developers provide this information upfront for their users.

5.2 Time of Explanation

Time of explanation refers to the stages where explanation algorithms are applied in an ML lifecycle. We define the time of explanation according to three stages: pre-model, in-model and post-model (i.e., post-hoc) explanations.

- **Pre-model**: Data is one of the most important factors that decides the performance of an AI system, and therefore it is important to have a good understanding of the data before focusing on the model. We observe that many tool developers place emphasis on representing explanation information in the form of visualization. There are tools available specifically for data explanations (e.g., Facets) and tools wherein data explanations algorithms are bundled with model explanations (AIX 360).
- **In-model**: referred as intrinsic, or white-box explanations are obtained by extracting information directly from ML models that are inherently more interpretable (e.g. shallow decision-trees, linear regressions, etc.).
- **Post-model**: are often referred to as black-box or post hoc explanations, which are obtained by using an XAI algorithm on top of the original model, which is treated as a black-box.

5.3 Scope of Explanation

This criterion distinguishes whether the XAI algorithm offers explanations for individual predictions (i.e., local) or the entire model behavior (i.e., global). Global model explanations provide insight into the distribution of the prediction outputs based on the input features in order to form an overall description of the ML model. Local model explanations provide insight into the relationship between the input features and the prediction for a particular instance. This can be done by approximating a small region of interest in a black box model using a simpler interpretable model [Arya et al., 2019]. We observe that the scope of explanations is not explicitly mentioned in many tools, and it is up to users to find this information from reading resources provided such as research papers or tool documentation. We suggest tool developers make this information more explicit upfront.

5.4 Dependance on Model Class

This property distinguishes whether algorithms used to provide explanations are dependent on the specific ML model (model-specific) or independent of the ML model (model-agnostic). Model-specific interpretation algorithms are limited to specific model classes because each algorithm is based on a specific model's internals. On the other hand, model-agnostic algorithms can be applied to any ML model. By definition, model-agnostic algorithms cannot have access to the model inner workings, such as weights or structural information [Kaur et al., 2020], otherwise it would not be possible to decouple them from the black-box model. An advantage of these algorithms is that they allow the user to use whichever model they wish, since they are applied after training. Unlike model-specific algorithms, they do not restrict the user to one model. However, since model-agnostic algorithms do not have access to the inner workings of the model, it is unclear if they are truly explaining the model.

The dependence on model class is related to another utility dimension: the time of explanation (see Section 5.2). By definition, all tools that are model agnostic are also post-model since they are based on treating the model as a black box. However, a model-specific tool can be either in-model or post-model.

5.5 Type of ML Model

ML models use large amounts of data to infer the parameters of a particular problem directly from the data. There are various types of ML models, including deep learning models (i.e., neural networks), tree-based models, and linear models, among others.

5.6 Type of Explanation Algorithm

This feature provides information about the XAI algorithms provided by the tool. All the open-source tools we surveyed offer descriptions of the explanation algorithm along with a corresponding research paper to help justify the design choices of the algorithm. In proprietary tools, information about the explanation algorithms is not available. Some tools offer XAI algorithms developed by others (for e.g., LIME and SHAP algorithms are available in AIX 360. It is unclear as to what improvements that newer XAI algorithms have over older ones. We suggest the tool developers to provide strong reasoning and contextual evidence when they create new algorithms and further research is needed to benchmark the performance of XAI algorithms and tools.

5.7 Type of ML Task

These includes types of tasks performed by ML algorithms such as (semi) supervised learning, unsupervised learning, reinforcement learning.

- **Supervised learning**: requires datasets that have a corresponding label for every single datapoint. For a new datapoint, the model is trained to predict the label based on what it learned from the labelled dataset.
- **Unsupervised learning**: is used for unlabeled datasets, where the task is to infer patterns about the data with-out reference to any labels.
- **Semi-supervised learning**: is used when the dataset has a combination of labelled and unlabeled data.
- **Reinforcement learning**: is conducted in an interactive environment, where an agent learns about its environment by using feedback (i.e., rewards) from its previous action and states.

5.8 Problem Type

This category offers information about the problem types supported by the XAI tools. Below we list some examples of problem types of supervised and unsupervised learning algorithms:

- **Classification (supervised)**: the output of the model is a categorical value.
- **Regression (supervised)**: the output of the model is a continuous value.
- **Clustering (unsupervised)**: grouping data based on their similarities.
- **Dimensionality reduction (unsupervised)**: maps the input into a latent space, which is typically of a lower dimension than the original space, while preserving the original properties of the data.
- **Generative modeling (unsupervised):** generates new data points based on patterns found in the dataset.

6 Usability Dimension

Usability includes capabilities that enables users to understand and operate the tool easily. It includes the following 6 features.

6.1 Explanation Type

It includes the formats in which the explanations are available to the users. We classify the explanation outputs into technical and non-technical explanations.

- **Technical explanations**: includes summary statistics (e.g., coefficients of a linear model) & visualizations that comprises of various plots that can help the user comprehend predictions (e.g., partial dependence plots).
- **Non-technical explanations**: includes non-statistical descriptions, either in natural language (e.g., What-If Tool) or highlighting important regions in an image (Activis). Tools that offer non-technical explanations tend to claim that non-ML experts would be able to understand the explanations offered.

6.2 Explainability Enhancing Features

This refers to any additional tool attributes that make explanations more human interpretable. For example, H20 wireless AI offers "reason codes," which are natural language explanations of Shapley values. The FICO decision management suite provides "reason codes," a textual output that helps business users understand a specific decision instance, such as the reason for loan rejection.

6.3 User Specific Explanations

Checks whether the tool offers ability to customize explanations based on the ML stakeholders' profile. Our survey finds that very few tools (e.g., AI Explainability 360) offer explanations tailored to various user profiles.

6.4 Explanation Documentation

This dimension checks whether tool offers capabilities to automatically provide documentation of the explanations, since documenting and sharing the explanations in an understandable form can improve model governance and stakeholders' trust in the predictive ML model. We find that some tools automate the documentation of prediction explanations to save time for tool users (e.g., Shapash). For e.g., Econ ML automatically generate documents with prediction explanations. Some tools [Fiddler platform, Contextual AI] allow the reports to be shared in multiple formats (e.g., PDF, HTML, e-mail). However, further research is needed to identify what information should be provided and how it should be displayed to various tool users.

6.5 Usecase

The use case feature refers to a concrete example explaining how the user should use the tool to complete the task at hand. Providing use cases can add clarity because they can help explain how the tool behaves in a particular domain. They can also help users understand the scope of the tool. Some open-source tools [LIME, SHAP] provide notebooks to demonstrate the applicability of the tools in various domains (e.g., HR, biomedicine, finance, etc.). We also see instances where tools developed for one specific domain suggest the appropriateness of tools in other fields [Manifold]. However, most tools do not contain information about the application domain; it is often left to the user to assess the relevance and usefulness of the tools for their project. We suggest the tool developers include concrete use cases in their tool documentation, or at least explain why a particular tool can or cannot be used for a given domain.

6.6 Guidance for Use

This includes guidelines that would help users determine which algorithms are appropriate for their use case. For e.g., AIX360 provides a decision tree charts to help users choose an XAI method from the list. We encourage tool developers to provide such additional guidance to ease the decision-making process.

7 XAI Toolsheet Template

Tool users may find it difficult to read all the available tool information. Design fit concepts in HCI suggest presenting the information in tabular format when information acquisition is concentrated on extracting discrete and precise information [Kelley et al., 2009]. As a result, we propose displaying the tool information in a form of one-page summary template. Appendix C shows the instantiation of XAI Toolsheet.

8 Conclusion

In this preliminary work, we ideate the concept of XAI tool documentation framework which is composed of 22 tool features. XAI Toolsheet can be useful to researchers, engineers, product managers, and regulators, who are interested to know what properties to look for in an XAI tool. Although more tool criteria can always be added to the list, we suggest the tool developer community include XAI Toolsheet as a documentation template to promote the trust of ML systems. XAI Toolsheet can further help tool developers to critically evaluate their tools for redundancy and to highlight unique value proposition of their tools.

For future work, we plan to validate the tool features and the template with tool developers and tool users to deter-mine how useful they find XAI Toolsheet in practice. We also want to publish a database of XAI Toolsheet for the 152 XAI tools. Another future direction involves creating Toolsheet for other types of tools focusing on responsible AI practices such as fairness, privacy and security.

A XAI Tools Cited in this Paper

Tools	Retrieved from
ELI5	https://github.com/eli5-org/eli5
What-if tool	https://github.com/PAIR-code/what-iftool
H20-Driverless-AI	https://github.com/Ozgeersoyleyen/H20-Driverless-AI
Shapash	https://github.com/MAIF/shapash
Data Robot	https://www.datarobot.com/platform/
Skater	https://oracle.github.io/Skater/
Databricks	https://databricks.com
Alteryx	https://www.alteryx.com
Lucid	https://github.com/tensorflow/lucid
Facets	https://github.com/PAIR-code/facets
LIME	https://github.com/marcotcr/lime
SHAP	https://github.com/slundberg/shap
AIX360	https://aix360.mybluemix.net
EconML	https://github.com/microsoft/EconML
Manifold	https://github.com/uber/manifold

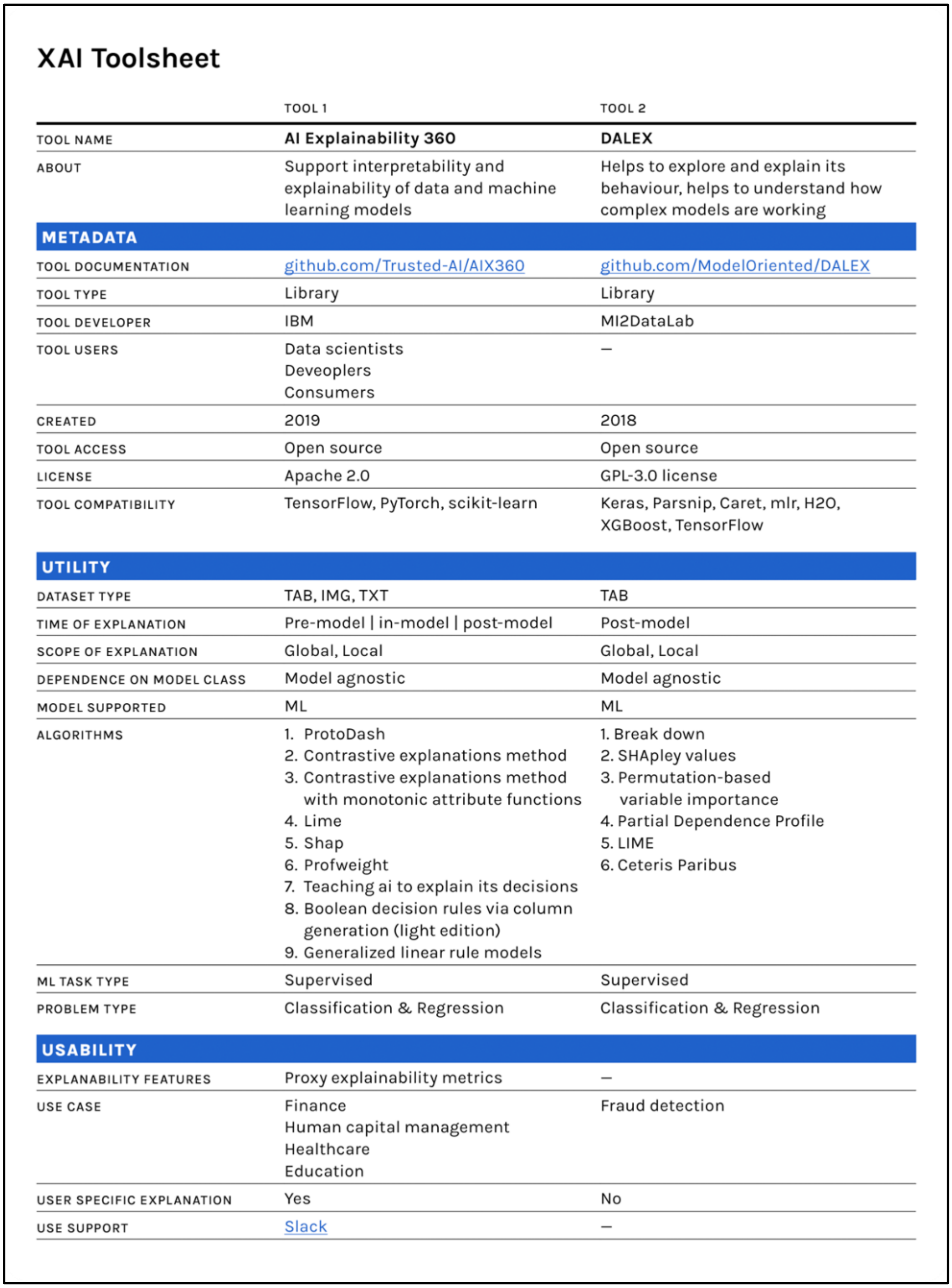
Table 1: XAI Tools Cited in this Paper

B Database search & Analysis Details

Readers who are interested to get the full list of 152 tools, database search results, key words, Document analysis method that was used to create the framework can reach out to surya@partnershiponai.org.

C XAI Toolsheet Instantiation

We demonstrate the practical application of XAI Toolsheet as shown in Figure 1. This is a prototype template, and it is still in its ideation stage, we are currently in the process of evaluating and consolidating both the tool features contents and the template with XAI tool developers and users.



XAI Toolsheet

	TOOL 1	TOOL 2
TOOL NAME	**AI Explainability 360**	**DALEX**
ABOUT	Support interpretability and explainability of data and machine learning models	Helps to explore and explain its behaviour, helps to understand how complex models are working
METADATA		
TOOL DOCUMENTATION	github.com/Trusted-AI/AIX360	github.com/ModelOriented/DALEX
TOOL TYPE	Library	Library
TOOL DEVELOPER	IBM	MI2DataLab
TOOL USERS	Data scientists Deveoplers Consumers	—
CREATED	2019	2018
TOOL ACCESS	Open source	Open source
LICENSE	Apache 2.0	GPL-3.0 license
TOOL COMPATIBILITY	TensorFlow, PyTorch, scikit-learn	Keras, Parsnip, Caret, mlr, H2O, XGBoost, TensorFlow
UTILITY		
DATASET TYPE	TAB, IMG, TXT	TAB
TIME OF EXPLANATION	Pre-model | in-model | post-model	Post-model
SCOPE OF EXPLANATION	Global, Local	Global, Local
DEPENDENCE ON MODEL CLASS	Model agnostic	Model agnostic
MODEL SUPPORTED	ML	ML
ALGORITHMS	1. ProtoDash 2. Contrastive explanations method 3. Contrastive explanations method with monotonic attribute functions 4. Lime 5. Shap 6. Profweight 7. Teaching ai to explain its decisions 8. Boolean decision rules via column generation (light edition) 9. Generalized linear rule models	1. Break down 2. SHApley values 3. Permutation-based variable importance 4. Partial Dependence Profile 5. LIME 6. Ceteris Paribus
ML TASK TYPE	Supervised	Supervised
PROBLEM TYPE	Classification & Regression	Classification & Regression
USABILITY		
EXPLANABILITY FEATURES	Proxy explainability metrics	—
USE CASE	Finance Human capital management Healthcare Education	Fraud detection
USER SPECIFIC EXPLANATION	Yes	No
USE SUPPORT	Slack	—

Figure 1: Practical application of XAI Toolsheet

References

[Adadi et al., 2018] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6, 52138–52160, 2018.

[Agarwal and Das, 2020] Namita Agarwal and Saikat Das. Interpretable machine learning tools: A survey. *IEEE*, 1528–1534, 2020.

[Ann, 2019] Blandford, Ann. HCI for health and wellbeing: Challenges and opportunities. *International journal of human-computer studies* 131: 41-51, 2019

[Arya et al., 2019] 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.

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

[Bender and Friedman, 2018] Emily M Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics* 6, 587–604, 2018.

[Besedes et al., 2015] Tibor Besedeš, Cary Deck, Sudipta Sarangi, and Mikhael Shor. Reducing choice overload without reducing choices. *Review of Economics and Statistics*, 793–802, 2015.

[Chen et al., 2020] Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. CausalML: Python Package for Causal Machine Learning. *arXiv:2002*.11631, 2020.

[Erickson et al., 2017] Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, Timothy Kline, and Kenneth Philbrick. Toolkits and libraries for deep learning. *Journal of digital imaging*, 30(4):400–405, 2017.

[Floridi, 2020] Luciano Floridi. 2020. Artificial Intelligence as a Public Service: Learning from Amsterdam and Helsinki. *Philosophy & Technology*, 33(4):541–546, 2020.

[Gardner et al., 2018] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A Deep Semantic Natural Language Processing Platform. *arXiv:1803*.07640, 2018.

[Gebru et al., 2021] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12); 86–92, 2021.

[Gunning and Aha, 2019] David Gunning and David Aha. DARPA's explainable artificial intelligence (XAI) program. *AI magazine*, 40(2): 44–58, 2019.

[Hall et al., 2019] Mark Hall, Daniel Harborne, Richard Tomsett, Vedran Galetic, Santiago Quintana-Amate, Alistair Nottle, and Alun Preece. A systematic method to understand requirements for explainable AI (XAI) systems.

[Hohman et al., 2018] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2018. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 25(8): 2674– 2693, 2019.

[Holland et al., 2018] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv preprint arXiv:1805.03677*, 2018.

[Iyengar and Lepper, 2000] Sheena S. Iyengar and Mark R. Lepper. 2000. When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, 79(6): 995–1006, 2000.

[Kaur et al., 2020] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning, 1–14, 2020.

[Kelley et al., 2009] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A" nutrition label" for privacy. 1–12, 2009.

[Kingman et al., 2022] Nigel Kingsman, Emre Kazim, Ali Chaudhry, Airlie Hilliard, Adriano Koshiyama, Roseline Polle, Giles Pavey, and Umar Mohammed. Public sector AI transparency standard: UK Government seeks to lead by example. *Discover Artificial Intelligenc*, 2(1): 1–9, 2022

[McGregor, 2020] Sean McGregor. Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database. *arXiv:2011*.08512, 2020.

[Mitchell et al., 2019] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. 220–229, 2019.

[Morley et al., 2021] Jessica Morley, Luciano Floridi, Libby Kinsey, and Anat Elhalal. From what to how: an initial review of publicly available AI ethics tools, methods and research to translate principles into practices. In *Ethics, Governance, and Policies in Artificial Intelligence*. Springer, 153–183, 2021.

[Paszke et al., 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, andLuca Antiga. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Riley, 2017] Jenn Riley and National Information Standards Organization (U.S.). *Understanding metadata: what is metadata, and what is it for?*, 2017.

[Singh et al., 2019] Chandan Singh, W. James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. *arXiv:1806.*05337, 2019.

[Sokol and Flach, 2020] Kacper Sokol and Peter Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. 56–67, 2020.

[Syrgkanis et al., 2019] Vasilis Syrgkanis, Victor Lei, Miruna Oprescu, Maggie Hei, Keith Battocchi, and Greg Lewis. Machine Learning Estimation of Heterogeneous Treatment Effects with Instruments. *arXiv:1905.*10176, 2019.

[Tenney et al., 2020] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. *arXiv:2008.05122*, 2020.

[Richard et al., 2018] Tomsett Richard, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. Interpretable to whom? A role-based model for analyzing interpretable machine learning systems. *arXiv:1806.07552*, 2018.

[Zhang et al., 2019] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S. Ebert. 2019. Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models. *IEEE Trans. Visual. Comput. Graphics*, 25(1): 364–373, 2019.