What Would You Ask the Machine Learning Model? Identification of User Needs for Model Explanations Based on Human-Model Conversations.

Michał Kuźba [1 2] **Przemysław Biecek** [1 2]

Abstract

Recently we see a rising number of methods in the field of eXplainable Artificial Intelligence. To our surprise, their development is driven by model developers rather than a study of needs for human end users. To answer the question "What would a human operator like to ask the ML model?" we propose a conversational system explaining decisions of the predictive model. In this experiment, we implement a chatbot called `dr_ant` and train a model predicting survival odds on Titanic. People can talk to `dr_ant` about the model to understand the rationale behind its predictions. Having collected a corpus of 1000+ dialogues, we analyse the most common types of questions that users would like to ask. To our knowledge, it is the first study of needs for human operators in the context of conversations with an ML model. It is also a first study which uses a conversational system for interactive exploration of a predictive model trained on tabular data.

1. Introduction

Machine Learning models are widely adopted in all areas of human life. As they often become critical parts of the automated systems, there is an increasing need for understanding their decisions and ability to interact with such systems. Hence, we are currently seeing the growth of the area of eXplainable Artificial Intelligence (XAI). For instance, Scantamburlo et al. (2018) raise an issue of understanding machine decisions and their consequences on the example of computer-made decisions in criminal justice. This example touches upon such features as fairness, equality, transparency and accountability.

[1]Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland [2]Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland. Correspondence to: Michał Kuźba <kuzba.michal@gmail.com>, Przemysław Biecek <przemyslaw.biecek@gmail.com>.

Ribera & Lapedriza (2019) identify the following motivations for why to design and use explanations:

- system verification, including bias detection,
- improvement of the system (debugging),
- learning from the system's distilled knowledge,
- compliance with legislation, e.g. "Right to explanation" set by European Union,
- inform people affected by AI decisions.

We see the rising number of explanation methods, such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) and XAI frameworks such as AIX360 (Arya et al., 2019), InterpretML (Nori et al., 2019), DALEX (Biecek, 2018), modelStudio (Baniecki & Biecek, 2019), exBERT (Hoover et al., 2019) and many others. As stated by (Gilpin et al., 2018), (Mueller et al., 2019) and (Lage et al., 2019) such systems require a systematic quality evaluation. For instance, (Tan et al., 2019) describe the uncertainty of explanations and (Molnar et al., 2019) describe a way to quantify the interpretability of the model.

These methods and toolboxes are focused on the model developer perspective. Most popular methods like Partial Dependency Profiles, LIME or SHAP are tools for a post-hoc model diagnostic rather than tools linked with the needs of end users. But it is also important to remember that human is the addressee (explainee) of any explanation system. Thus, the implementation of such a system should put the human in the centre while deciding about the form and content.

Also, both the form and the content of the explanations should differ depending on the explainee's background and role in the model lifecycle. Ribera & Lapedriza (2019) describe three types of explainees: AI researchers and developers, domain experts and the lay audience. Tomsett et al. (2018) introduce six groups: creators, operators, executors, decision-subjects, data-subjects and examiners. These roles are positioned differently in the pipeline. Users differ in the background and the goal of using the explanation system. They vary in the technical skills and the language they

use. Finally, explanations should have a comprehensible form – usually visual or textual. Explainees benefit from the interactivity of the explanation system. Sokol & Flach (2018a) propose conversation using class-contrastive counterfactual statements. This idea is implemented by Sokol & Flach (2018b) as a conversational system for the credit score systems lay audience. Also, Miller (2017) claims that truly explainable agents will use interactivity and communication.

To address this problem we create an open-ended explanation system using dialogue. We develop a chatbot allowing the explainee to interact with a Machine Learning model and its explanations. We implement this particular system for the random forest model trained on Titanic dataset but this approach might be transferred successfully to other models and datasets. Our goal is twofold. Firstly, we create a working prototype of a conversational system for XAI. Secondly, we want to discover what questions people ask to understand the model. This exploration is enabled by the open-ended character of the chatbot. It means that the user might ask any question even if the system is unable to give a satisfying answer for each of them.

As a result, we gain a better understanding of how to answer the explanatory needs of a human operator. With this knowledge, we will be able to create explanation systems tailored to explainee's needs by addressing their questions. It is in contrast to developing new methods blindly or according to the judgement of their developers.

2. Dialogue system

This dialogue system is a chatbot with the user initiative. It offers a conversation about the underlying random forest model trained on the well-known Titanic dataset. We deliberately select a black box model with no direct interpretation together with a dataset and a problem that can be easily imagined for a wider audience. The dialogue system was built to understand and respond to several groups of queries:

- **Supplying data** about the passenger, e.g. specifying age or gender. This step might be omitted by impersonating one of two predefined passengers with different model predictions.
- **Inference** – telling users what are their chances of survival. Model imputes missing variables.
- **Visual explanations** from the Explanatory Model Analysis toolbox (Biecek & Burzykowski, 2020): Ceteris Paribus profiles (Kuzba et al., 2019) (addressing "what-if" questions) and Break Down plots (Gosiewska & Biecek, 2019) (presenting feature contributions). Note this is to offer a warm start into the system by answering some of the anticipated queries. However, the principal purpose is to explore what other types of questions might be asked.
- **Dialogue support** queries, such as listing and describing available variables or restarting the conversation.

3. Implementation

A top-level chatbot architecture is depicted in Figure 1. The system consists of several components:

1. **Explainee**
   Human operator – addressee of the system. They chat about the blackbox model and its predictions.
2. **Interface**
   This dialogue agent might be deployed to various conversational platforms independently from the backend and each other. The only exception to that is rendering some of the graphical, rich messages. We used a custom web integration as a major surface. It communicates with the dialogue agent's engine sending requests with user queries and receiving text and graphical content. The frontend of the chatbot uses `Vue.js` and is based on `dialogflow`[1] repository. It provides a chat interface and renders rich messages, such as plots and suggestion buttons. This integration allows to have a voice conversation using the browser's speech recognition and speech synthesis capabilities.
3. **Dialogue agent**
   Chatbot's engine implemented using `Dialogflow` framework and `Node.js` fulfilment code run on `Google Cloud Functions`.
   - **NLU**
     The Natural Language Understanding (NLU) component classifies query intent and extracts entities. This classifier uses the framework's builtin rule-based and Machine Learning algorithms. NLU module recognizes 40 intents such as posing a what-if question, asking about a variable or specifying its value. It was trained on 874 training sentences. Some of these sentences come from the initial subset of the collected conversations. Additionally, NLU module comes with 4 entities – one for capturing the name of the variable and 3 to extract values of the categorical variables – gender, class and the place of embarkment. For numerical features, a builtin numerical entity is utilized. See examples in Section 3.1.
   - **Dialogue management**
     It implements the state and context. Former is

[1] https://github.com/mishushakov/dialogflow-web-v2

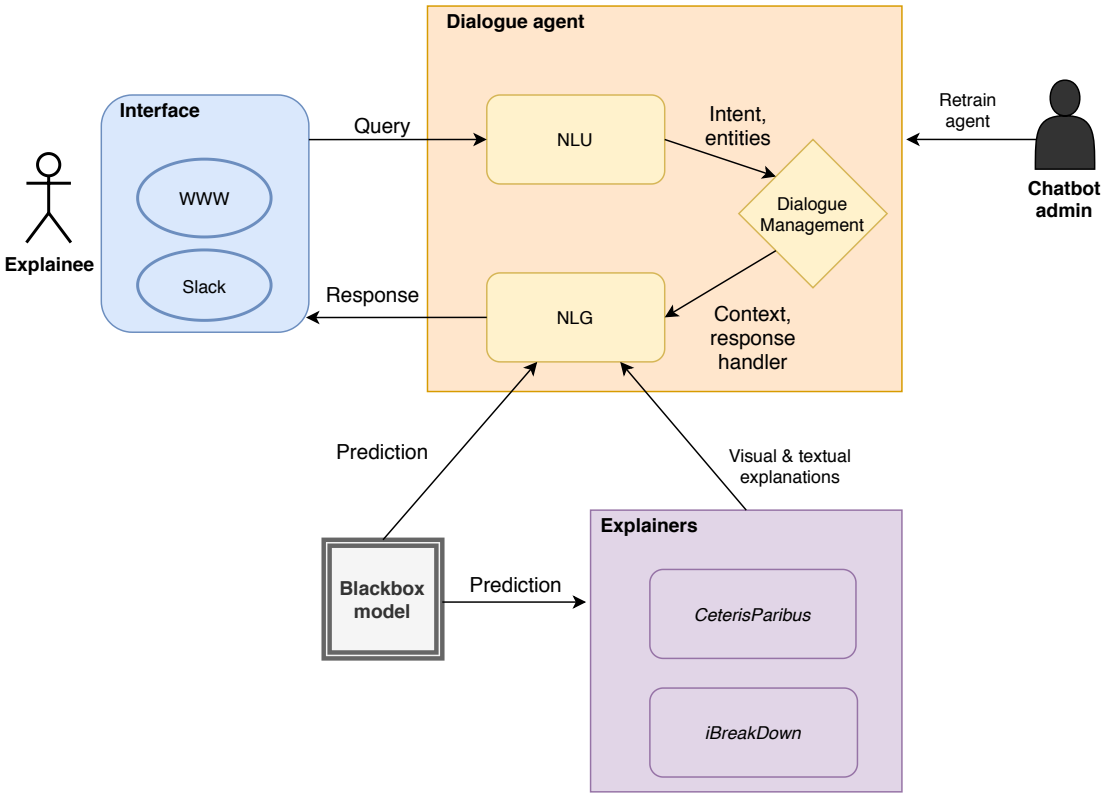

Figure 1. Overview of the system architecture. **Explainee** uses the system to talk about the **blackbox model**. They interact with the system using one of the **interfaces**. The conversation is managed by the **dialogue agent** which is created and trained by the **chatbot admin**. To create a response system queries the **blackbox model** for its predictions and **explainers** for visual explanations.

used to store the passenger's data and the latter to condition response on more than the last query. For example, when the user sends a query with a number it might be classified as age or fare specification depending on the current context.

- **NLG**
  Response generation system. It might query the model or its explainers to produce a final response. Plots, images and suggestion buttons are included as rich messages.

4. **Blackbox model**
   A random forest model predicting the chance of survival on Titanic. The model was trained in R package (R Core Team, 2019) and converted into REST api with the `plumber` package (Trestle Technology, LLC, 2018). The model is stored in the `archivist` database (Biecek & Kosinski, 2017) and can be downloaded with a following R command: `archivist::aread("pbiecek/models/42d51")`.

5. **Explainers**
   REST API exposing visual and textual model explanations from `iBreakDown` (Gosiewska & Biecek, 2019) and `CeterisParibus` (Kuzba et al., 2019) libraries. They query the blackbox model to create an explanation.

6. **Chatbot admin**
   Human operator – developer of the system. They manually retrain the system based on misclassified intents and misextracted entities. For instance, this dialogue agent was iteratively retrained based on the subset of the collected dialogues.

The source code for the prototype is available at `https://github.com/ModelOriented/xaibot`.

3.1. NLU examples

These are the examples of the NLU output.

***Query**: What If I had been older?*
***Intent:** ceteris_paribus*
***Entities:** [variable: age]*

***Query** I'm 20 year old woman*
***Intent:** multi_slot_filling*
***Entities:** [age: 20, gender: female]*

Query: *Which feature is the most important?*
Intent: *break_down*
Entities: *[]*

3.2. Example dialogue

An excerpt from an example conversation might be seen in Figure 2. The corresponding intent classification flow is highlighted in Figure 3.

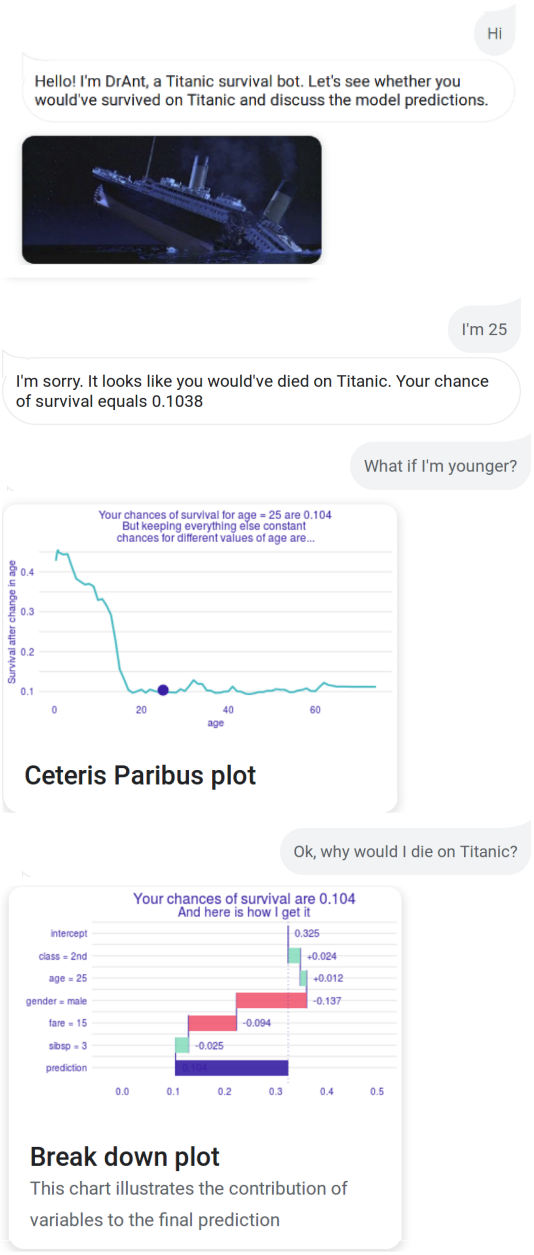

Figure 2. An example conversation. Explainee's queries in the grey boxes.

4. Results

The initial subset of the collected dialogues is used to improve the NLU module of the dialogue agent. As a next step, we conduct an experiment by sharing the chatbot in the Data Science community and analyzing the collected dialogues.

4.1. Experiment setup

For this experiment, we work on data collected throughout 2 weeks. This is a subset of all collected dialogues, separate from the data used to train the NLU module. Narrowing the time scope of the experiment allows to define the audience and ensure the coherence of the data. As a next step, we filter out conversations with totally irrelevant content and those with less than 3 user queries. Finally, we obtain 621 dialogues consisting of 5675 user queries in total. The average length equals 9.14, maximum 83 and median 7 queries. We see the histogram of conversations length in Figure 4. Note that by conversation length we mean the number of user queries which is equal to the number of pairs (user query, chatbot response).

4.2. Query types

We analyze the content of the dialogues. Similar user queries are grouped together. For each such category, we calculate the number of conversations with at least one query of this type. Results are presented in Table 1. Note that users were not prompted or hinted to ask any of these with an exception of the *"what do you know about me"* question. Moreover, the taxonomy defined here is independent of the intents recognized by the NLU module.

List of the query types ordered decreasingly by the number of conversation they occur in:

- **why** – general explanation queries, such as "why", "explain it to me", "how was that derived/calculated".
- **what-if** – alternative scenario queries. Frequent examples: *what if I'm older*, *what if I travelled in the 1st class*. Rarely, we see multi-variable questions such as: *What if I'm older and travel in a different class*.
- **what do you know about me** – this is the only query hinted to the user using the suggestion button. When the user inputs their data manually it usually serves to understand what is yet missing. However, in the scenario when the explainee impersonates a movie character it also aids understanding which information about the user is possessed by the system.
- **EDA** – a general category on Exploratory Data Analysis. All questions related to data rather than the model

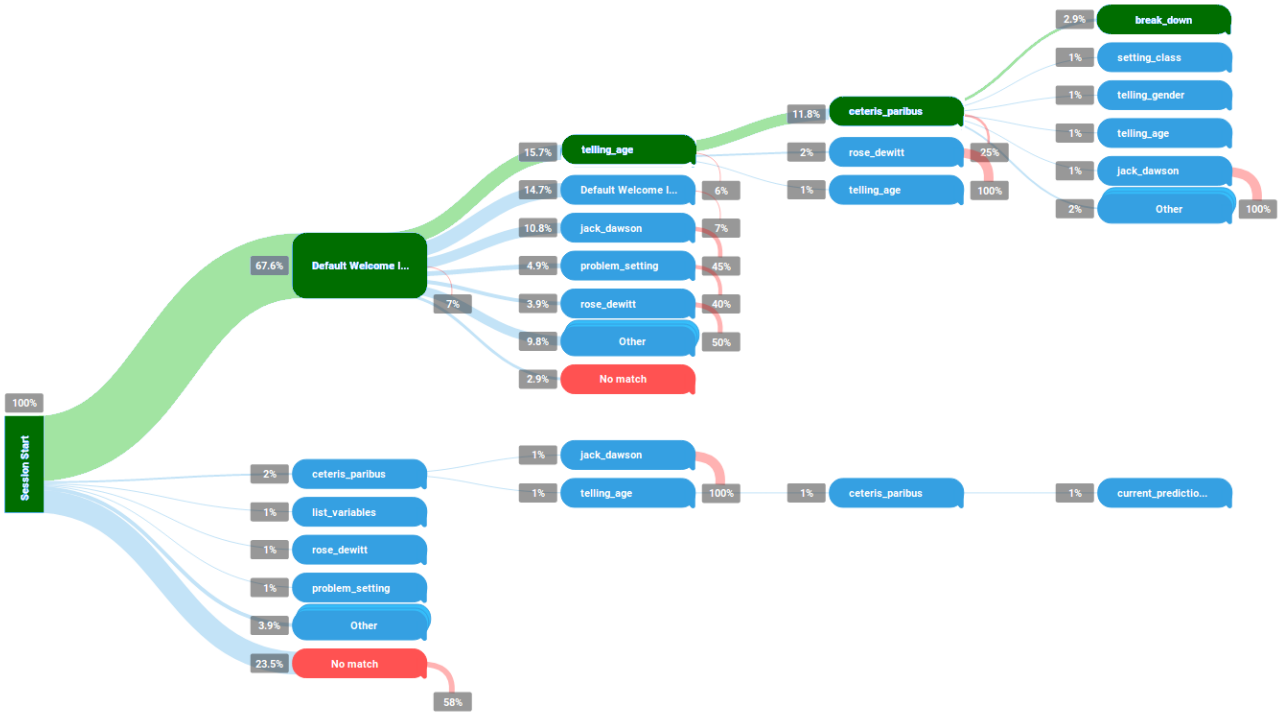

Figure 3. Screenshot from the `Dialogflow Analytics`. This flow chart demonstrates the results of the NLU module on a sample of collected dialogues. Example conversation from Figure 2 contributes to the topmost (green) path. Each box corresponds to a classified intention of the query, e.g. *telling_age* or *ceteris_paribus*.

fall into this category. For instance, *feature distribution*, *maximum values*, *plot histogram for the variable v*, *describe/summarize the data*, *is dataset imbalanced*, *how many women survived*, *dataset size* etc.

- **feature importance** – here we group all questions about the relevance, influence, importance or effect of the feature on the prediction. We see several subtypes of that query:
  - *Which are the most important variable(s)*
  - *Does gender influence the survival chance*
  - **local importance** – *How does age influence my survival*, *What makes me more likely to survive*
  - **global importance** – *How does age influence survival across all passengers*
- **how to improve** – actionable queries for maximizing the prediction, e.g. *what should I do to survive*, *how can I increase my chances*.
- **class comparison** – comparison of the predictions across different values of the categorical variable. It might be seen as a variant of the *what-if* question. Examples: *which class has the highest survival chance*, *are men more likely to die than women*.
- **who has the best score** – here, we ask about the observations that maximize/minimize the prediction. Examples: *who survived/died*, *who is most likely to survive*. It is similar to *how to improve* question, but rather on a per example basis.
- **model-related** – these are the queries related directly to the model, rather than its predictions. We see questions about the algorithm and the code. We also see users asking about metrics (accuracy, AUC), confusion matrix and confidence. However, these are observed just a few times.
- **contrastive** – question about why predictions for two observations are different. We see it very rarely. However, more often we observe the implicit comparison as a follow-up question – for instance, *what about other passengers*, *what about Jack*.
- **plot interaction** – follow-up queries to interact with the displayed visual content. Not observed.
- **similar observations** – queries regarding ”neighbouring” observations. For instance, *what about people similar to me*. Not observed.

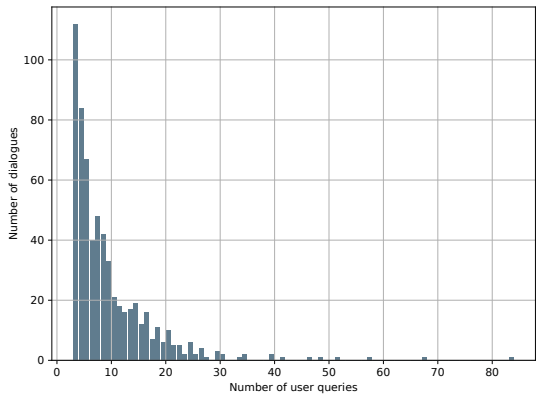

Figure 4. Histogram of conversations length (number of user queries), after filtering out conversations shorter than 3 queries.

We also see users creating alternative scenarios and comparing predictions for different observations manually, i.e. asking for prediction multiple times with different passenger information. Additionally, we observe explainees asking about other sensitive features, that are not included in the model, e.g. nationality, race or income. However, some of these, e.g. income, are strongly correlated with class and fare.

5. Conclusions and Future Work

Depending on the area of application, different needs are linked with the concept of interpretability (Lipton, 2016; Tomsett et al., 2018). And even for a single area of application, different actors may have different needs related to model interpretability (Arya et al., 2019).

In this paper, we presented a novel application of the dialogue system for conversational explanations of a predictive model. Detailed contributions are following (1) we presented a process based on a dialogue system allowing for effective collection of user expectations related to model interpretation, (2) we presented a xai-bot implementation for a binary classification model for Titanic data, (3) we conducted an analysis of the collected dialogues.

Conversational agent proved to work as a tool to explore and extract user needs related to the use of the Machine Learning models. This method allowed us to validate hypotheses and gather requirements for the XAI system on the example from the experiment.

In this analysis, we identified several frequent patterns among user queries. When given a tool for the model exploration, we saw users engaging in the conversation about model's decisions rather than focusing on its metrics.

Table 1. Results of the analysis for 621 conversations in the experiment. The second column presents the number of conversations with at least one query of a given type. A single dialogue might contain multiple or none of these queries.

Query type	**Dialogues count**
why	73
what-if	72
what do you know about me	57
EDA	54
feature importance	31
how to improve	24
class comparison	22
who has the best score	20
model-related	14
contrastive	1
plot interaction	0
similar observations	0
Number of all analyzed dialogues	**621**

Conversational agent is also a promising, novel approach to XAI as a model-human interface. In the future, such systems might be useful in bridging the gap between automated systems and their end users. An interesting and natural extension of this work would be to compare user queries for different explainee's groups in the system, e.g. model creators, operators, examiners and decision-subjects. In particular, it would be interesting to collect needs from explainees with no domain knowledge in Machine Learning. Similarly, it is interesting to take advantage of the process introduced in this work to compare user needs across various areas of applications, e.g. legal, medical and financial. Additionally, based on the analysis of the collected dialogues we see two related areas that would benefit from the conversational human-model interaction – *Exploratory Data Analysis* and *model fairness* (based on the queries about the sensitive and bias-prone features).

Acknowledgments

Micha Kuba was financially supported by the NCN Opus grant 2016/21/B/ST6/0217.

References

Arya, V., Bellamy, R. K. E., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilovi, A., Mourad, S., Pedemonte, P., Raghaven-

dra, R., Richards, J., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K. R., Wei, D., and Zhang, Y. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, 2019.

Baniecki, H. and Biecek, P. modelStudio: Interactive Studio with Explanations for ML Predictive Models. *The Journal of Open Source Software*, Nov 2019. doi: 10.21105/joss.01798. URL https://doi.org/10.21105/joss.01798.

Biecek, P. DALEX: Explainers for Complex Predictive Models in R. *Journal of Machine Learning Research*, 19: 1–5, 2018.

Biecek, P. and Burzykowski, T. *Explanatory Model Analysis. Explore, Explain and Examine Predictive Models*. 2020. URL https://pbiecek.github.io/ema/.

Biecek, P. and Kosinski, M. archivist: An R package for managing, recording and restoring data analysis results. *Journal of Statistical Software*, 82(11):1–28, 2017. doi: 10.18637/jss.v082.i11.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. Explaining explanations: An approach to evaluating interpretability of machine learning. *CoRR*, abs/1806.00069, 2018. URL http://arxiv.org/abs/1806.00069.

Gosiewska, A. and Biecek, P. Do Not Trust Additive Explanations. *arXiv e-prints*, 2019.

Hoover, B., Strobelt, H., and Gehrmann, S. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformers Models, 2019.

Kuzba, M., Baranowska, E., and Biecek, P. pyCeterisParibus: explaining Machine Learning models with Ceteris Paribus Profiles in Python. *JOSS*, 4 (37):1389, 2019. doi: 10.21105/joss.01389. URL http://joss.theoj.org/papers/10.21105/joss.01389.

Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S., and Doshi-Velez, F. An evaluation of the human-interpretability of explanation. *CoRR*, abs/1902.00006, 2019. URL http://arxiv.org/abs/1902.00006.

Lipton, Z. C. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016. URL http://arxiv.org/abs/1606.03490.

Lundberg, S. M. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Miller, T. Explanation in artificial intelligence: Insights from the social sciences, 2017. URL http://arxiv.org/abs/1706.07269.

Molnar, C., Casalicchio, G., and Bischl, B. Quantifying Interpretability of Arbitrary Machine Learning Models Through Functional Decomposition. *arXiv e-prints*, art. arXiv:1904.03867, Apr 2019.

Mueller, S. T., Hoffman, R. R., Clancey, W. J., Emrey, A., and Klein, G. Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI. *CoRR*, abs/1902.01876, 2019. URL http://arxiv.org/abs/1902.01876.

Nori, H., Jenkins, S., Koch, P., and Caruana, R. InterpretML: A Unified Framework for Machine Learning Interpretability, 2019. URL https://arxiv.org/abs/1909.09223.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Ribeiro, M. T., Singh, S., and Guestrin, C. *Why Should I Trust You?: Explaining the Predictions of Any Classifier*, pp. 11351144. ACM Press, 2016. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL http://dl.acm.org/citation.cfm?doid=2939672.2939778.

Ribera, M. and Lapedriza, À. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*, 2019.

Scantamburlo, T., Charlesworth, A., and Cristianini, N. Machine decisions and human consequences. *CoRR*, abs/1811.06747, 2018. URL http://arxiv.org/abs/1811.06747.

Sokol, K. and Flach, P. Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 5785–5786. International Joint Conferences on Artificial Intelligence Organization, 7 2018a. doi: 10.24963/ijcai.2018/836. URL https://doi.org/10.24963/ijcai.2018/836.

Sokol, K. and Flach, P. Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 5868–5870. International Joint Conferences on Artificial Intelligence Organization, 7 2018b. doi: 10.24963/ijcai.2018/865. URL https://doi.org/10.24963/ijcai.2018/865.

Tan, H. F., Song, K., Udell, M., Sun, Y., and Zhang, Y. Why should you trust my interpretation? Understanding uncertainty in LIME predictions. *CoRR*, abs/1904.12991, 2019. URL http://arxiv.org/abs/1904.12991.

Tomsett, R., Braines, D., Harborne, D., Preece, A., and Chakraborty, S. Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, 2018.

Trestle Technology, LLC. *plumber: An API Generator for R*, 2018. URL https://CRAN.R-project.org/package=plumber. R package version 0.4.6.