University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Michał Kuźba
Student no. 371148

Conversational explanations of Machine Learning models using chatbots

Master's thesis
in COMPUTER SCIENCE

Supervisor:
dr hab. inż. Przemysław Biecek
Institute of Applied Mathematics and Mechanics
University of Warsaw

Warsaw, December 2021

Abstract

Recently we have seen a rising number of methods in eXplainable Artificial Intelligence (XAI). To our surprise, their development is driven by model developers rather than a study of needs for human end-users. Moreover, most of such tools and methods are static and do not reflect the human need for interactivity and communication in the explanation process. In this thesis, we propose a chatbot (XAI-bot) explaining decisions of the predictive model. XAI-bot offers a conversational interface to explanations and allows us to answer the question, “What would a human operator like to ask the ML model?” In this work, we develop the XAI-bot and demonstrate it using a Random Forest model trained on the Titanic dataset. We collect 1000+ human-agent interactions and analyse the patterns among the explanatory queries users have asked the chatbot. To our knowledge, it is the first study that uses a conversational system to collect the needs of human operators from the interactive and iterative dialogue explorations of a predictive model. The proposed methodology enables the study of end-user needs related to XAI and, consequently, the development of XAI methods tailored to their needs.

The results of this work were presented at the ECML PKDD 2020 International Workshop on eXplainable Knowledge Discovery in Data Mining and published in the conference proceedings.

Keywords

Explainable Artificial Intelligence, iterative dialogue explanations, Machine Learning, conversational agents, human-centred Machine Learning

Thesis domain (Socrates-Erasmus subject area codes)

11.4 Sztuczna inteligencja

Subject classification

I. Computing Methodologies
I.2 Artificial Intelligence
I.2.1 Applications and Expert Systems
I.2.7 Natural Language Processing
I.2.m Miscellaneous

Tytuł pracy w języku polskim

Konwersacyjne wyjaśnienia modeli Uczenia Maszynowego z użyciem agentów dialogowych

Contents

Introduction

Motivation

Machine Learning models are widely adopted in all areas of human life. They often become critical parts of the automated systems. For this reason, it becomes increasingly important to interact with the models, understand the decisions they make and explore the rationale behind them. Hence, we currently observe the growing number of tools (explanation systems) in the domain of eXplainable Artificial Intelligence (XAI). We also see the increasing popularity of the Human-centered Artificial Intelligence paradigm. It is a "perspective on AI and ML that algorithms must be designed with awareness that they are part of a larger system consisting of humans" [1].

While ML systems offer progress in numerous areas, the way they affect human lives raises a number of concerns. For instance, Scantamburlo et al. [2] raise an issue of understanding machine decisions and their consequences on the example of computer-made decisions in criminal justice. This single example alone already touches upon a range of features such as fairness, equality, transparency and accountability. These features are desired in every system yet are rarely supported by the plethora of blackbox ML models in the wild.

Ribera and Lapedriza [3] identify five motivations behind designing and utilizing explanations. These are (1) system verification, including bias detection; (2) improvement of the system (debugging); (3) learning from the system's distilled knowledge; (4) compliance with legislation, e.g. "Right to explanation" set by EU [4]; (5) informing people affected by AI decisions. This extensive set of reasons caused the rising number of explanation methods and frameworks. We list and characterize them in Section 1.3 and Section 1.4.

Similarly to other software tools, XAI methods require evaluation [5, 6, 7] and should be implemented in response to human end-user needs. However helpful in many cases, such methods and toolboxes remain focused on the model developer perspective. Most popular methods like Partial Dependence Plots (`PDP`) [8], `LIME` [9] or `SHAP` [10] are designed to be the tools for a post hoc model diagnostic rather than tools linked with the needs of end-users. Therefore, not unlike any other tools or software, an explanation system should be designed with its addressee (explainee) in the central place. Hence, both the form and content of the system should be adjusted to the end-user. And while explainees might not have the AI expertise, explanations are often constructed by engineers and researchers for themselves [11, 12, 13, 14], therefore limiting its usefulness for the other audience [15].

Similarly, both the form and the content of the explanations should differ depending on the explainee's background and role in the model lifecycle. Ribera and Lapedriza [3] name three types of explainees: AI researchers and developers, domain experts and the lay audience. Tomsett et al. [16] introduce six groups: creators, operators, executors, decision-subjects, data-subjects and examiners. These roles are positioned differently in the lifecycle of the automated system. In addition, users differ in the background and the goal of using the explanation system. Finally, they vary in the technical skills and the language they use.

Table 1 lists example questions related to the interpretability of Machine Learning models. We can see the hypothetical behaviour of various consumers of the models and explanation systems. They ask different questions depending on a range of factors. In particular, most end-users are not ML engineers and have little expert knowledge in the field. Crucially, while this is a credible list, it remains hypothetical. To collect an actual set of questions in a particular context requires a study with the end-users of the explanation systems. The task of surveying the ML explanations needs of the diverse audience remains to be solved and tools enabling that to be developed.

Table 1: List of example questions related to the interpretability of Machine Learning models. These questions tend to differ depending on a range of factors. Those factors include the characteristics of the end-users and the context of the decision-making process supported by ML models. Source: own elaboration.

Role	**Field**	**Question**	**Motivation**	**Other factors**
Doctor	Medicine	***What was a diagnosis for similar patients?***	Build trust	High-stakes decision
Bank customer	Finance	***What could I do to improve my credit score?***	Change output	
Facebook user	Advertising	***Why do I see this ad?***	Understand use of data	
Machine Learning engineer	Any	***What are the important variables?***	Improve model	ML expertise

Specifically, the tool we are looking for is an interface between a human and a Machine Learning model along with its explanations. In general, the issue of finding a human-machine interface is a problem in the field of Human-Computer Interaction [17] (see Figure 1.5). In the explainability context, the term Human-centred XAI (HCXAI) was recently coined [18, 19].

Interaction between the explainee and the explainer during the explanation process is well-grounded in social sciences [20]. Assady et al. [21] state that explanation is a cognitive process and social interaction. Moreover, interactive exploration of the model allows personalizing the explanations presented to the explainee [22]. Maadi et al. [23] stress the benefits of a "human-in-the-loop" approach to Machine Learning models lifecycle. Furthermore, Arya et al. [24] identify a space for interactive explanations in a tree-shaped taxonomy of XAI techniques. However, the `AIX360` framework presented in this paper implements only static explanations. Similarly, most other toolkits and methods focus entirely on the static branch of the explanations taxonomy.

While several tools and dashboards [25, 26, 27, 28] allow for a certain degree of interactivity and personalization, they have multiple shortcomings. Firstly, they do not provide a narrative throughout the explanation process. In their work, Baniecki and Biecek [29] introduce the term Interactive Explanatory Model Analysis (IEMA) and claim that an explanation system should offer multi-threaded customizable stories. Secondly, current methods, including the dashboards, often do not have a memory or state. It means they cannot keep track of historical interactions or store information from the user. Usually, what they offer instead are independent single-shot visualizations. More importantly, these tools limit user interaction to a set of buttons, predefined input fields and simple, predictable operations. There is no means for the user to express their own needs, go beyond what was expected by the system developer and pose any question they might have. Finally, it results in a lack of feedback and communication between the end-users and developers.

A prospective human-computer interface in our scenario is the use of natural language [30]. Miller [20] claims that truly explainable agents will use interactivity and language communication. Natural language interface, or specifically, conversational interface, is a good fit in this situation. It is a remedy for a range of problems with the dashboards approach. Natural language allows for open input and asking questions in a way resembling human-human interactions. Additionally, modern chatbot solutions support multi-turn conversations handling the history and state throughout the dialogue. Finally, chatbots allow for an iterative exploration of the model, composing multiple responses into the story about the blackbox model. To sum up, the conversational interface to model and its explanations is a natural and promising approach to exploration and collection of the user needs.

Objective

In this thesis, we address two broad tasks. Firstly, the explanation needs of the end-users are ignored and unstudied. Therefore, the XAI developers fail to recognise the diversity of the end-users and tailor the tools to their needs. Moreover, the few existing attempts to the needs discovery suffer from poor scalability of the manual user studies. Secondly, while most XAI tools are static, the end-users could considerably benefit from interactive and conversational explanations.

Our goal in this thesis is twofold and correspond to the tasks identified above. Firstly, we create a working prototype of a conversational system for XAI. Specifically, we develop a chatbot (XAI-bot), allowing the explainee to interact with a predictive model and its explanations. This particular implementation supports the conversation about the Random Forest model (Section 4.2.2) trained on the Titanic dataset (Section 4.2.1). However, any model trained on this dataset can be plugged into this system. Moreover, this approach can be applied successfully to other datasets with components reused or easily adapted. Secondly, we use this system to discover what questions people ask to understand the model. We collect a corpus of human–chatbot interactions and analyse the user queries. This exploration is enabled by the open input of the chatbot. It means users might type in anything even if the system fails to respond satisfyingly to some of their queries.

As a result of this research, 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 the needs of explainees by addressing their actual questions. It is in contrast to developing new methods blindly or according to the judgement of their developers. The chatbot we build in this work bridges the communication gap between stakeholders of

XAI systems. XAI-bot opens the feedback channel from XAI end-users to XAI developers (Figure 1). Using the proposed methodology in various contexts (audience types, applications and other factors), we could fill in questions as in Table 1 based on the studies of the end-users and their actual questions.

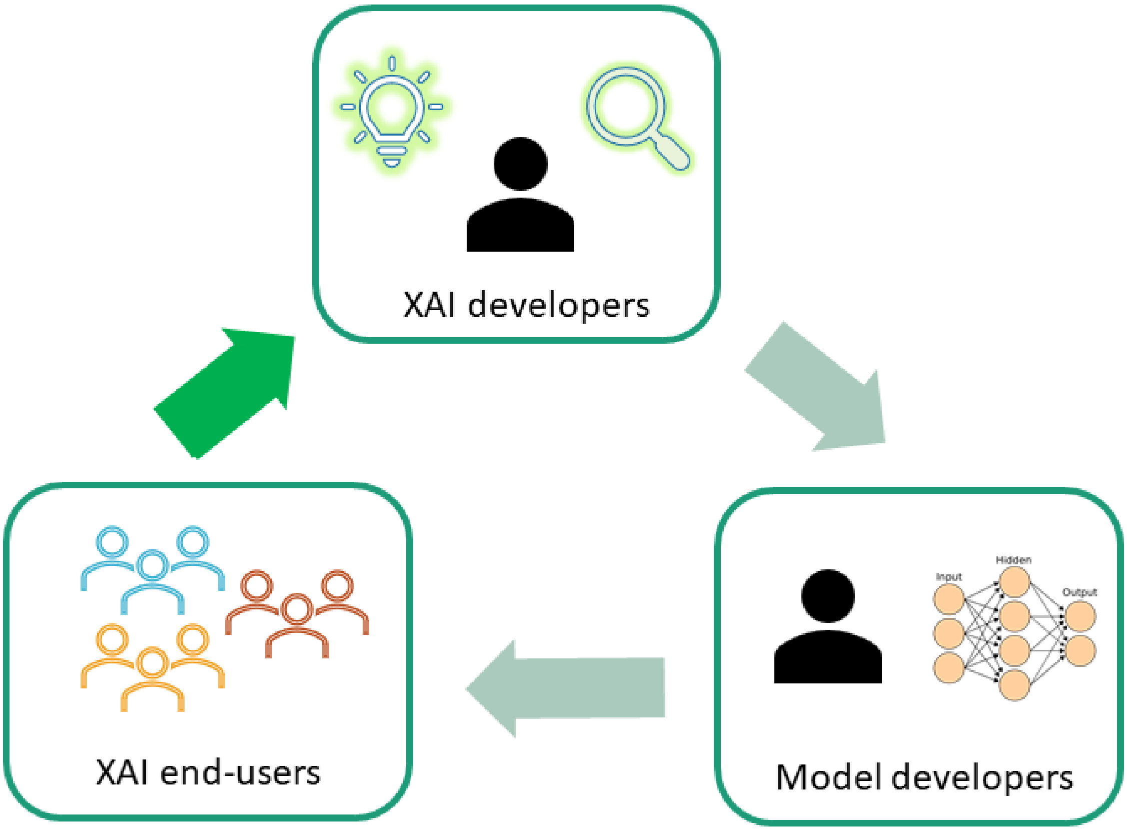


Figure 1: A simplified diagram of stakeholders in the Machine Learning model lifecycle. This work aims to bridge a communication gap between a diverse group of XAI (Explainable Artificial Intelligence) end-users and XAI developers. This new feedback channel is represented with the bold, green arrow on the left.

Structure

This thesis is organized into five chapters.

The first two chapters provide a brief theoretical introduction and describe the domains of Explainable Artificial Intelligence (XAI) and Conversational Artificial Intelligence. Chapter 1 contains a short history of XAI, introduces basic definitions and describes existing tools and methods. Then, we discuss the most pressing challenges of the field and present the explanations from the perspective of the social sciences. Finally, we describe previous attempts at building interactive, textual or conversational explanation systems. Chapter 2 outlines the field of conversational AI and introduces basic definitions and classification of dialogue agents. Finally, it briefly reviews existing frameworks for building chatbots.

Chapter 3 presents the chatbot built in this work (XAI-bot). We start by outlining its capabilities and listing all components and actors of the system. Then, we discuss the versatility and extensibility of the XAI-bot. Finally, we demonstrate XAI-bot, as well as its NLU and NLG components on examples.

Chapter 4 describes the experiment of collecting human-model interactions using XAI-bot. We start by describing previous studies of user needs in the explanatory context. Then, we outline the experiment setup, describe the dataset and model used in the experiment and display statistics of the surveyed sample. Finally, we present the main results of this work — aggregated statistics of the collected conversations, grouped by query categories. We report the statistics and discuss the obtained results.

Finally, Chapter 5 summarizes achieved results and outlines directions for future research.

Acknowledgements

I would like to express my gratitude to my supervisor Professor Przemysław Biecek for his guidance and insightful comments throughout the project and writing of this thesis. This research was financially supported by the NCN Opus grant 2016/21/B/ST6/0217.

Chapter 1

Explainable Artificial Intelligence

In this chapter, we briefly describe the field of Explainable Artificial Intelligence (XAI). This summary concentrates exclusively on explainability in the context of Supervised Learning. Nevertheless, there is an ongoing XAI work in other paradigms, e.g. explainability of Reinforcement Learning models [31].

We start by introducing some basic definitions and terminology. Then, we outline the history of the field. In the next section, we list XAI methods and frameworks along with their classification and taxonomies. We also discuss the perspective of the social sciences to explanations. Finally, we quote some critical voices and outline the challenges of the field.

We dedicate the last three sections to the specific branches of XAI directly related to the topic of this research. These are interactive, textual and conversational explanations.

1.1. Terminology

At the start of this introduction to Explainable AI, we will define some most important terms from the field. The two most common names for the field are: **Explainable Artificial Intelligence** (*XAI*) and **Interpretable Machine Learning** (*IML*). These names stem from the concepts of **interpretability** and **explainability**.

There is not one widely agreed definition of these terms. Nevertheless, we quote two popular definitions of interpretability. Miller [11] says: "*Interpretability is the degree to which a human can understand the cause of a decision.*" Another one by Kim et al. [32] states: "*Interpretability is the degree to which a human can consistently predict the model's result*". Molnar [33] adds to that: "*The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made.*"

However, there exist slightly different understandings of this term. Lipton [34] claims that "*interpretability is not a monolithic concept, but in fact reflects several distinct ideas*". He says ML models usually optimize for accuracy, and it is unclear how to encode a notion of interpretability into the real-valued loss function. For him, Interpretable Machine Learning might be seen as an umbrella term incorporating desired model features such as fairness, accountability, trust, reliability, transparency, safety and privacy. Lastly, yet another perspective from Rudin [35]: "*Interpretability is a domain-specific notion, so there cannot be an all-purpose definition*".

Another question is the relation between interpretability and explainability. Sometimes, these two are assigned different meanings. In such case, interpretability is understood as an intrinsic property of the model. Consequently, we can call such models naturally or inherently interpretable. Some examples might be Linear Regression or a simple decision tree. We also call these models **white boxes**, **glass boxes** or **transparent boxes**. It is in contrast to so-called **blackboxes**. Blackbox models are not interpretable themselves and require an additional explanation. Such a post hoc approach is understood as **explainability**.

However, in this work, we follow the popular approach of treating interpretability and explainability as synonyms. For consistency, we primarily use the terms **explainability** and **XAI**. If there is a need to differentiate between the two aforementioned types, we refer to them descriptively, e.g. post hoc explanations or naturally interpretable models. Consequently, we use the terms **explanation system** or **explainer** for a method, tool or system offering an explanation of the model.

1.2. History of Explainable AI

This section describes the history of explainable AI. It is primarily based on the book "*Interpretable Machine Learning — a Brief History, State-of-the-Art and Challenges*" by Molnar et al. [36].

Machine learning dates back to the middle of the 20th century. Recently, after an AI winter, the Deep Learning revolution of the 2010s [36] caused an explosion of interest and research in the Machine Learning area. The field of explainable AI, however, was not very active until recently. Having said that, naturally interpretable models such as Linear Regression date back to the early 19th century. One of the first explainability methods was a feature importance measure for Random Forests (2001) [37]. Additionally, there was some earlier explanation work in the field of expert systems (1993) [38].

A growing interest in the field of interpretability or explainability and the usage of these terms is observed from around 2015 (see Figure 1.1 and Figure 1.2). Explainable AI has now reached its readiness and is maturing [36]. There are books [33, 41, 42, 43] and surveys of methods [44, 45, 46, 47, 48]. The number of academic papers is rapidly growing (see Figure 1.1). The topic of interpretability is present at top Machine Learning conferences (e.g. ICML, NeurIPS, ICLR, FAccT) and the industry [49, 27]. Importantly, there is an increasing number of tools and frameworks (see Section 1.3) with `LIME` (2016) [9] and `SHAP` (2017) [10] being among the top-cited and most popular tools. There have been interpretability-related competitions (FICO Explainable Machine Learning Challenge [50]) and datasets (e.g. `COMPAS` [51]). Finally, academic work on explainability affects the legal efforts on creating regulations and policy documents [52].

1.3. Methods and frameworks

In this section, we describe a classification of XAI methods along with some example tools and frameworks. There is a number of XAI taxonomies [24, 53, 54, 55, 56, 57]. In this brief overview, we will cover three standard division criteria for methods in Explainable AI.

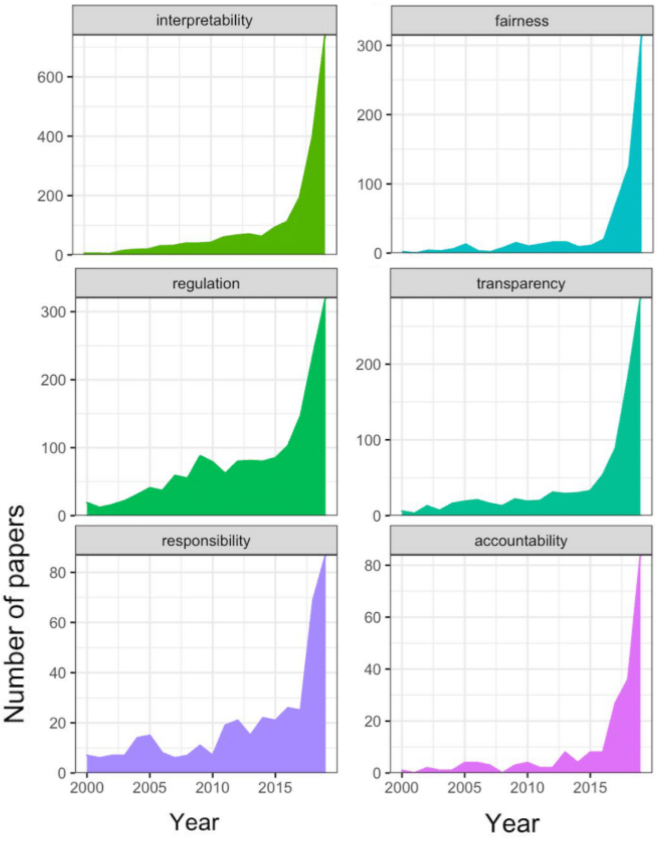
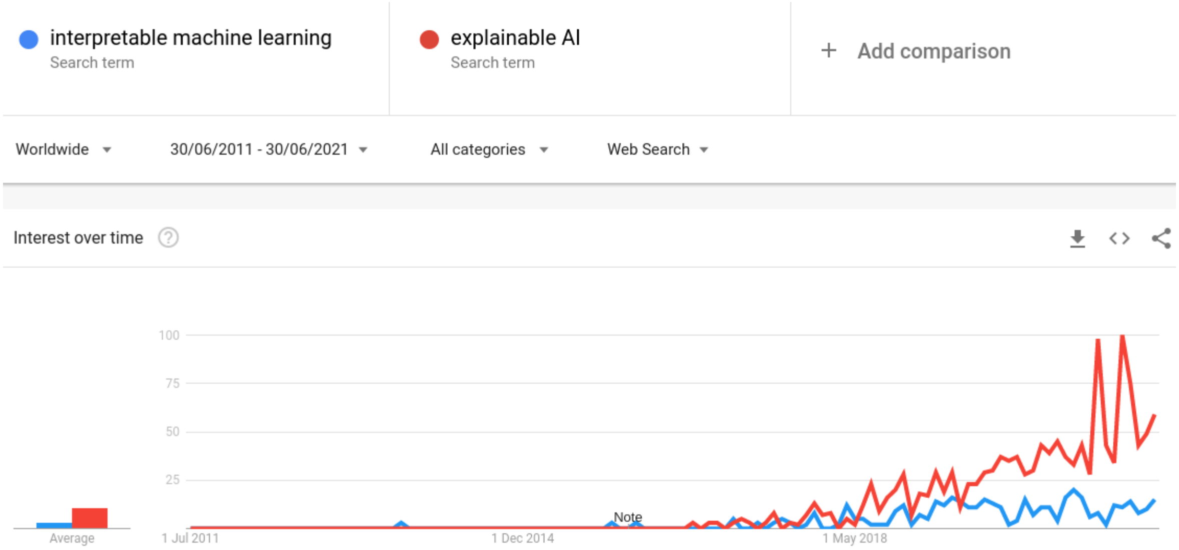


Figure 1.1: Interpretability-related trends among the academic papers. We see the rising numbers of papers for given topics. This analysis is based on 557,681 AI-related papers from the *S2ORC* corpus [39]. We have seen an explosion of new papers since 2015. Adapted from other work [40].



● interpretable machine learning
Search term

● explainable AI
Search term

+ Add comparison

Worldwide ▾ 30/06/2011 - 30/06/2021 ▾ All categories ▾ Web Search ▾

Interest over time

Figure 1.2: Google Search trends for phrases "*interpretable machine learning*" and "*explainable AI*" over the last ten years. This screenshot is based on [36] (accessed November 03, 2021).

1.3.1. Naturally interpretable vs post hoc

Naturally (or inherently) interpretable models are models which themselves can be assigned a particular interpretation. Linear Regression models, decision trees, nearest neighbour methods and decision rule models are generally considered interpretable [58, 59]. However, Linear Regression models with hundreds of features or deep decision trees might be already too complex and not interpretable anymore [36]. Another problem comes with heavy feature engineering. Frequently, simple and interpretable models perform comparably with the blackboxes because of the advanced feature engineering. Therefore, while the model itself might be interpretable, the overall system, including the feature transformation, becomes a blackbox [34]. Some of the latest, top-performing methods for naturally interpretable models include Explainable Boosting Machines (EBMs) [60] or Neural Additive Models (NAMs) [61].

In the second group, we have post hoc explainability methods, which produce an explanation to the blackbox model. All explainers we discuss in this work fall into that type.

1.3.2. Local vs global

Post hoc explanation methods might be further divided into groups depending on the scope of the explanation. Local methods explain the model's decision on the instance level, while global methods explain the global behaviour of the model on the dataset level.

Example local methods include Shapley values techniques [10], counterfactual explanations, local surrogate models or saliency maps for Convolutional Neural Networks (CNNs). **Shapley values** [62] provide a way to distribute the payout of a collaborative game between players. In this setting, features are players, and their payout corresponds to their contribution to the model decision. An example of such a method implemented in the `iBreakDown` package [63] can be seen in Figure 1.3. We use this explainer in our conversational XAI system Section 3.2.5.

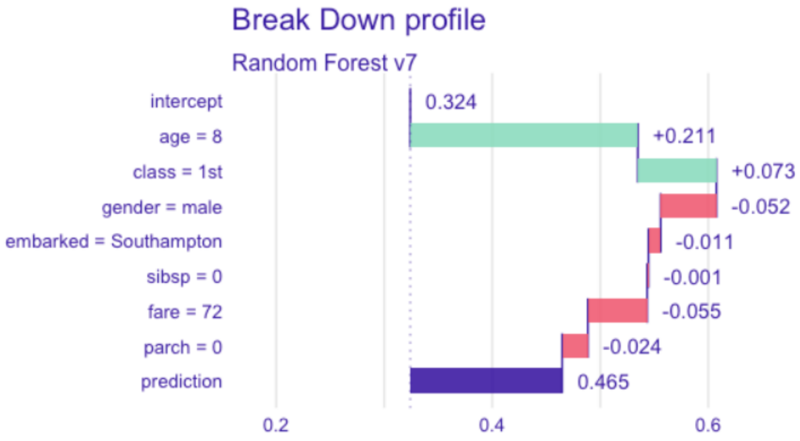


Figure 1.3: An example Break Down profile from the `iBreakDown` package. It presents the contribution of each variable to the final prediction. For this passenger, we see that being a small child contributes the most to the above-average survival chance (compare prediction bar with the intercept level). Predicted survival chance on X-axis.

Another method leverages **surrogate models**. They rely on an approximation of the blackbox model with one that is naturally interpretable. A popular local explanation method based on this technique is `LIME` [9].

Counterfactual explanations are based on the "*what-if*" scenarios. They are well-grounded in philosophy and social sciences [20]. An example tool using this approach is the `CeterisParibus` package [64] demonstrated in Figure 1.4. We utilise this tool in our conversational XAI system Section 3.2.5.

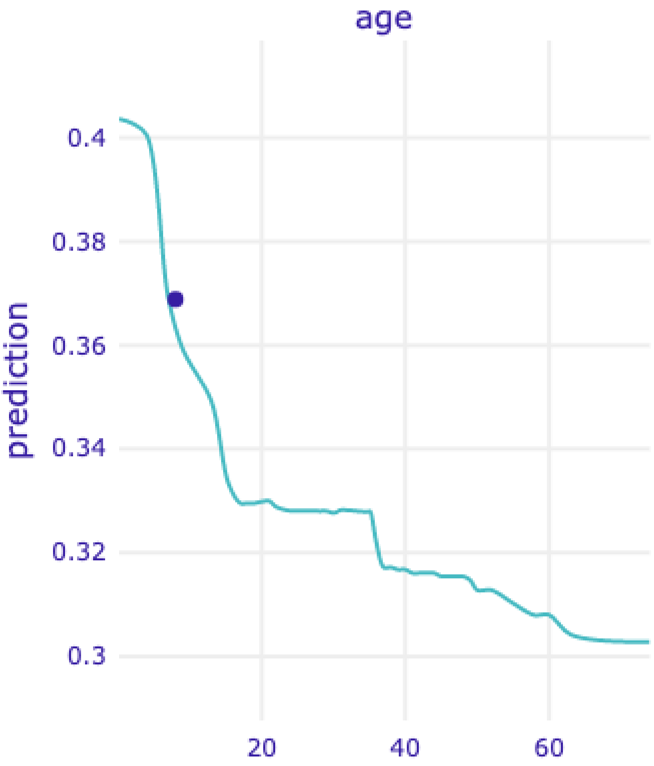


Figure 1.4: An example Ceteris Paribus plot. It presents how the prediction changes with one of the variables changing its value. In this case, a survival prediction (Y-axis) of the passenger (marked with a dot) on Titanic lowers with increasing age (X-axis).

Finally, **saliency maps** [65, 66] use network gradients to explain individual predictions of CNNs. They show how a change in the value of a pixel can affect the classification output.

A second class of the global methods explains the model behaviour on an entire dataset. These methods include **feature importance** measures [67, 68] and **feature effect** techniques such as PD plots. Yet another possibility is to detect and analyse **single observations** influential to the model behaviour [69]. Some examples include low confidence, misclassified observations, high leverage points or outliers. Lastly, global **surrogate models** that copy the original model's behaviour using a naturally interpretable model are in use [70, 71]. Such a form of model distillation allows drawing conclusions about the blackbox model by interpreting its surrogate.

1.3.3. Model-agnostic vs model-specific

The third aspect of explainability is model-agnosticism. Most of the methods we have discussed are **model-agnostic**. It means they treat the model as a blackbox and assume nothing about its internals. Thus, these explainers operate exclusively on the input and output of the model.

However, there is a range of **model-specific** methods. For instance, saliency maps mentioned before fall into this class. Other examples include explanations of Random Forest models [72] or transformers [73].

1.4. Explainable AI frameworks

There is a number of open source XAI frameworks. Some example are: `AIX360` [24], `InterpretML` [60], `Alibi` [74], `iNNvestigate` [75], `Skater` [76], `PyTorch Grad-CAM` [77] (Python), `DALEX` [78] (R and Python), `iml` [79] (R) or `H2O Driverless` interpretability module [27].

Additionally, there are several fairness-related toolkits such as `AIF360` [80], `Fairlearn` [81], `fairmodels` [82] or `DALEX`.

Concluding, we observe a wide range of XAI methods and their corresponding software implementations. Moreover, there have emerged several specialised frameworks offering multiple tools to address XAI tasks. The next step should be to tailor future and existing tools to end-users needs that we study in this work.

1.5. Perspective from social sciences

In his work, *"Explanation in Artificial Intelligence: Insights from the Social Sciences"*, Miller [20] claims that the field of XAI should build on research from areas such as philosophy, cognitive science or social psychology. This section is primarily based on his summary of the general topic of explanation.

Figure 1.5 shows that XAI lies at the intersection of **AI**, **HCI** and **social science**. Miller analyses how humans explain to each other. Based on that, he carries insights from social sciences to the field of Explainable AI. Among the major finding of this work is that explanations are contrastive, selected and social.

Contrastive explanations refer to explanatory questions of a form: *"Why **P** rather than **Q**?"*. P and Q are called **fact** and **foil**, respectively. It is claimed that all *"why-questions"* are contrastive, but sometimes the foil is not explicitly stated. Additionally, a frequent trigger for the human need for explanation is abnormality. It is often expressed as a contrastive *"why"* question with an implicit foil — *"Why P (and not the expected default case Q)"*? The second conclusion from Miller's work is that explanations are selected in a biased manner. It means people rarely want the explanation to consist of a complete cause of an event. They rather expect to see one or two major causes. Additionally, people have cognitive biases and incorporating these biases and adapting the explanation to the user knowledge and beliefs yield a better effect. Lastly, explanations are social. They are part of conversation or interaction. For the XAI context, Miller states that *"truly explainable agents will have to be interactive and adhere to maxims of communication"*.

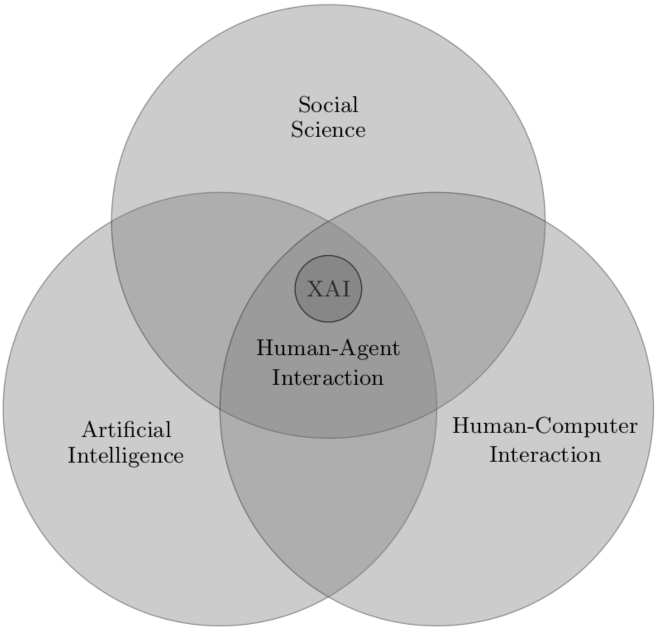


Figure 1.5: Diagram presenting the location of Explainable AI (XAI). **XAI** is a subfield of **Human-Agent interaction** and lies at the intersection of **AI**, **HCI** and **social science**. Reprinted from [20].

In this thesis, we address this call and implement a conversational explanatory agent (XAI-bot). We describe previous work regarding interactive, text and conversational explanations in sections 1.8, 1.9 and 1.10, respectively.

1.6. Critical voices

The field of XAI has reached some initial maturity and proved useful in various applications. Nevertheless, its current state faces some criticism and challenges.

One critical voice arises in the work of Cynthia Rudin [35] "*Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*". The author claims that post hoc explainability is wrong and urges the use of naturally interpretable models instead. She distinguishes two alternatives. The first option is that an explanation is completely faithful to the original model. In this case, the explanation could as well replace the original model. Alternatively, the explainer is not entirely faithful to the original and therefore does not explain its actual decisions. She concludes that post hoc explanations are always wrong. Furthermore, she claims that naming the post hoc methods as explanations is misleading. Instead, she proposes other names such as: "*summaries of predictions*," "*summary statistics*," or "*trends*" as more suitable. Consequently, the author opposes the right to explanation which allows such post hoc methods. Instead, she proposes that regulations should require naturally interpretable models whenever adequate and possible. Finally, she claims that a trade-off between accuracy and interpretability is a myth.

Contrarily, Molnar anticipates that the focus of interpretability will stay on model-agnostic tools [33]. He claims that their advantage lies in modularity and scalability. Additionally, Miller proposes an explanatory model of self — a second model that should specifically produce explanations. He claims such construct would be of use even for naturally interpretable models. For that, such a model should have the information useful for the process of explanation. It could model the learning process rather than the learned task itself.

There is also some criticism related to specific methods such as saliency maps. Arun et al. [83] and Adebayo et al. [65] evaluate existing saliency methods finding that several of them are unreliable and fail on a range of evaluation criteria, such as model parameters randomisation test. They conclude that the use of these methods requires additional scrutiny and careful choice of the applied technique.

1.7. Challenges

Research on explainability faces many challenges and requires further work in multiple directions. We will list some most critical or most related to this thesis.

One significant challenge is a lack of a widely agreed **definition** of interpretability. This is a considerable flaw and is a common critique of the field [34, 84]. Definition issues affect the **evaluation** of explanations which is very challenging. Arguably, unreliable explanations are useless and systematic quality evaluation is necessary [5, 6, 7]. However, we are lacking objective metrics. Only certain aspects are measurable such as fidelity, sparsity or uncertainty. For instance, Zhang et al. [85] describe the uncertainty of explanations and Molnar et al. [86] provide a way to quantify the interpretability of the model. Another approach to evaluation relies on user studies. Unfortunately, such an evaluation procedure is very costly unless performed automatically. Finally, like ML models themselves, explanations turn out to be vulnerable to adversarial attacks [87, 88, 89].

A challenge we consider very significant and therefore address in this thesis is **discovering user needs** and addressing them. There is a number of explanation systems designed by ML specialists. However, mostly without studying **lay audience needs** [12, 13, 14]. A related challenge is to build tools that will address the needs of regulatory documents, e.g. ensuring the right to explanation and allowing complex models to be used in high-stakes or sensitive domains.

In this work, we address these challenges by building a **conversational system (XAI-bot)**, allowing XAI developers to collect and study these needs. Consequently, this methodology enables the creation of better tools, methods and interfaces **tailored to the needs of the end-users** of explanations.

1.8. Interactive explanations

Miller calls: "truly explainable agents will have to be interactive". Furthermore, Arya et al. [24] identify a space for **interactive explanations**. It is the first split (static vs interactive) in their tree-shaped taxonomy of XAI techniques . However, the `AIX360` framework presented in this paper implements only static explanations. Similarly, most other toolkits and methods focus entirely on the static branch of the explanations taxonomy.

Baniecki and Biecek [29] introduce a concept of Interactive Explanatory Model Analysis proposing "multi-threaded customizable story about the blackbox model" rather than single aspect model explanations. They also provide a `modelStudio` framework for interactive model explanations [25]. There is a number of dashboard-like frameworks for interactive explanations. Some of them are H2O `Driverless AI` [27], `InterpretML` [60], aforementioned `modelStudio` [25], `explainerDashboard` [26] or `Shapash` [90]. See [29] for a full comparison.

1.9. Natural language explanations

Natural language has certain advantages over formal language or visual communication. Firstly, humans explain their decisions verbally [20, 34], so this approach is the closest to human explanations. Consequently, it is relatively easy to collect a dataset of golden standard explanations from humans or mine them from existing free-form text datasets. Secondly, natural language is readily comprehensible to end-users of the explanation systems [91].

Some attempts towards **textual explanations** also named as **natural language explanations** use a second model generating text explanations on top of the predictions of the original model [91, 92, 93]. However, this approach usually requires additional manual labelling with human-created explanations for the training set.

Another approach is to use rule-based or template-based systems. For instance, Hendricks et al. [94] generate counterfactual explanations with natural language in the image classification task. An example explanation for a single image of a bird:

> *This is a* Red Bellied Woodpecker *because this is a black and white spotted bird with a red crown.*
>
> *This is not a* Yellow Billed Cukoo *because it does not have a grey crown.*

One implementation of natural language explanations comes with an R package `ingredients` [95]. It offers simple template-based, textual explanations on top of the visual explanations available in the `DALEX` package [78]. Below we provide an example natural language explanations for a single prediction on Titanic dataset (see Section 4.2.1). These predictions correspond to the visual output of Figure 1.3.

> *Random forest predicts that the prediction for the selected instance is 0.465.*
>
> *The most important variables that increase the prediction are age, class.*
>
> *The most important variables that decrease the prediction are class, fare.*

1.10. Conversational explanations

Conversational explanations might be seen as both interactive and textual explanations described in the previous sections. The idea of structuring explanation in the conversational form is well-grounded in social science research. Hilton [96] proposes a *conversational model of causal explanation* claiming that explanation is a conversation. Miller [20] describes the concept of "explanation as a conversation" . Walton [97] proposes a formal dialogue model called CE. Walton [98] also describes a formal dialogue system for explanation. Madumal et al. [15] propose an interaction protocol and identify components of an explanation dialogue.

Nonetheless, the topic of conversational explanations of Machine Learning models is largely unexplored. Miller [20] includes a short example dialogue between a person and a hypothetical explanation agent in his work. Due to engineering challenges, some user studies employ the "Wizard of Oz" proxy approach as an alternative [22, 99] to building an actual dialogue agent. We elaborate on both this approach and chatbots in Chapter 2. Finally, some other articles propose the use of dialogue agents for explanation [100, 101] or declare an initial effort towards the construction of such system [102, 103].

Here we list the few existing implementations of conversational XAI agents. Sokol and Flach [104] propose conversation using class-contrastive counterfactual statements. This idea is implemented as a conversational system for the credit score systems lay audience [22, 105]. Pecune et al. [106] describe conversational movie recommendation agent explaining its recommendations. Gao et al. [107] propose a chatbot explanation framework with proactive explanations of AI model including the confidence level. They implement a simple Slack chatbot for the anomaly detection scenario. Finally, they collect the results from four experienced users of the model.

Chapter 2

Conversational Artificial Intelligence

Conversational Artificial Intelligence is a field of Natural Language Processing solving the task of human-computer dialogue. Systems solving this task are called conversational agents, dialogue agents or chatbots. We will use these names interchangeably following [108].

In this chapter, we provide an introduction to chatbots primarily based on [109]. In the second part, we describe software frameworks for their development.

2.1. Chatbots

Chatbots can be divided into two classes based on their application. The first class is **task-oriented** agents. They chat with the user to complete certain tasks. Commercial chatbots usually fall into this type. Their example objectives might be to guide users through the reservation process or answer users questions in the automated customer support. Moreover, digital assistants such as Siri, Alexa or Cortana fall into this class. Their tasks include making calls or finding restaurants [110]. The second class is **open-ended** chatbots which can chat about anything. These agents are not restricted to a predefined set of topics. Their goal is usually to mimic interaction with a human. The ultimate objective of many such agents is to pass a Turing test and convince a human they talk to a another person rather than a bot.

A dialogue is a sequence of **turns** — a sequence of alternating contributions from both speakers. A related task of question answering might be seen as a special case of a chatbot with a single turn from both speakers. For a chatbot to be more than a sequence of unrelated question answering tasks, it is necessary to condition the output on the previous turns. It means the agent should keep track of the overall history of the conversation. A **dialogue state** represents the current situation of the conversation as well as a model of the previous turns and the knowledge acquired from the interaction with the human.

Depending on the architecture, the dialogue state might be modelled as an automaton with a finite number of states or a continuous representation, e.g. using a neural network. The first is used in task-oriented agents with naturally defined conversation states, such as "choosing a hotel" or "specifying dates".

Chatbots differ by who carries the **initiative** of the conversation. The first type is a user-initiative system. One such example is the digital assistant, such as Siri or Alexa, with the user asking for guidance. Conversely, the system with the user responding to the agent-initiated queries is called a system-initiative architecture. Finally, a mixed-initiative architecture is one when both speakers can take the initiative. This framework corresponds to how humans usually communicate in natural conversations.

There are three main approaches to chatbot development: (1) rule-based, (2) corpus-based and (3) frame-based [109]. The history of the field starts in the 1960s with the rule-based chatbots. This class includes one of the most remarkable chatbots in the field — ELIZA [111] and PARRY [112].

ELIZA was a bot simulating a psychologist. Due to the specific psychotherapy setup, ELIZA can assume the pose of not knowing anything and concentrating on the interlocutor. Therefore, ELIZA relies on a set of transformation patterns reflecting patients statements at them. For instance, the sentence *"You hate me"* may be answered with *"What makes you think I hate you"*. This tactic relies on pattern detection and applying a corresponding transformation to the patient's utterance. ELIZA prioritises the available transformation rules based on the list of predefined keywords. For instance, the keyword *"everybody"* is ranked higher than *"I"*. Thus, if both are present in the utterance, ELIZA will apply the rule corresponding to the former. Finally, ELIZA uses a memory that allows it to refer to one of the previous patient questions. Surprisingly, ELIZA was able to make a believable impression of having a conversation with a human psychologist.

A following rule-based bot called PARRY implements an additional component of a mental state. The state modelled two variables — levels of fear and anger. Certain conversational topics increased these levels, conditioning PARRY's responses according to its emotional state. PARRY was the first chatbot to pass the Turing test [113] — psychiatrists could not distinguish between interview transcripts of PARRY and actual paranoids.

The second class of corpus-based agents learns from the corpus of historical conversations. It might generate the response in the Information Retrieval manner — retrieving the best answer from the corpus. Another option is to use the corpus to train a response generation system using Machine Learning algorithms, usually Deep Learning architectures.

Thirdly, frame-based agents [114] are used in task-oriented systems. The frame includes a set of user **intents** and a collection of **slots**. An intent is a goal or an intention of the user utterance. Slots are parameters to be filled with one of the possible values of a specific **entity** type. Example outputs of intent classification and slot filling for the chatbot implemented in this work are presented in Section 3.5. One special type of intent is the **fallback intent**. It means the agent failed to understand the user's intention or, alternatively, none of the intents scored above the confidence threshold.

Frame-based chatbots have two important Natural Language Processing (NLP) components: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU module is responsible for the intent classification and slot filling or entity extraction. The first can be seen as a text classification problem, yet the output is conditioned on the dialogue state. The second is a semantic parsing problem. Frame-based agents usually rely on **template-based** generation, which might be implemented as a set of hardcoded sentences (templates) with the variable slots filled depending on the conversation state. Examples of NLU outcome and NLG templates for the XAI-bot implemented in this work can be seen in Section 3.5 and Section 3.6, respectively.

Finally, there are chatbots using the hybrid approach and combining the aforementioned methods [115]. One such example is the conversational system (XAI-bot) developed in this work (see Chapter 3). XAI-bot is primarily frame-based. Nevertheless, it uses Machine Learning solutions to classify user intents based on the training corpus.

Some chatbots, such as mobile assistants, have additional voice capabilities (voicebots). It means they have Automatic Speech Recognition (ASR) and Text to Speech (TTS) components. However, in this work, we implement a chatbot whose primary interface is textual, and the voice interface is only an addition.

The conversational agents are systems with the direct involvement of the user. Therefore, their design is related to principles of the Human-Computer Interaction field. Some of these user-centred design principles are:

- **Iterative design**. This principle relies on the cyclic process of alternating test and modification phases [116]. In the chatbot context, previous interactions are used to identify new intents, validate the design or retrain the data-driven components of the chatbot.
- **Wizard-of-Oz system**. It is the technique when a human acts as a conversational agent and speaks with the user under disguise [117, 118]. It helps to combat the cold start problem and test ideas in the early stages before developing the actual automated dialogue agent.

In addition to the open text input, some chatbots use prompt buttons. These buttons serve as hints to users. A click on a button with a given text is equivalent to typing this query manually. This technique is used to design more satisfying user experiences and guide users through the conversation [119].

2.2. Existing frameworks for chatbot development

The popularity of the commercial applications of chatbots resulted in a plethora of frameworks facilitating the construction of the dialogue agents. There exist both open-source frameworks with free plans such as Rasa [120] or Botpress [121] as well as commercial solutions including Google's Dialogflow [122], Microsoft Bot Framework [123], Amazon Lex [124] or IBM's Watson Assistant [125]. Additionally, one can implement a chatbot in-house or leverage existing tools for specific subtasks such as Natural Language Understanding or Named Entity Recognition. The final group are no-code solutions that allow for simple, FAQ-like agents with few clicks or drag and drop method. In this section, we briefly describe the criteria of choice for our application and justify the final selection of Google Dialogflow.

On the one hand, our application requires the flexibility of writing code to call APIs of model and explainers and to handle more complex dialogue logic. On the other hand, we want to leverage existing solutions and not reinvent the wheel. Therefore we consider only full-fledged commercial and open-source frameworks. Furthermore, since this is a standalone one-off project, we do not have any preference for a specific cloud ecosystem. Thus, we decide to choose between two prominent representatives of their categories — Google's Dialogflow and Rasa Stack.

When choosing the chatbot framework, we follow several criteria we consider significant: pricing, customisation, quality, user-friendliness (including GUI options), deployment and data collection options. Note that this list is subjective, and for different applications, other factors such as multi-language support or an option for in-house data storage might be relevant.

Table 2.1: Comparison of Dialogflow and Rasa — two popular chatbot frameworks. State for 16.11.2021.

Criterion	**Dialogflow**	**Rasa**
Pricing	Paid per traffic, free quota and free credit for a. trial period.	Free, open-source library. Paid enterprise version with deployment, analytics and other extra features.
Customization	Limited to few options, such as NLU algorithm (ML-based or rule-based), confidence threshold for fallback or use of spellcheck. Developers write fulfillment code.	Control over components — open-source code. In particular, the developer can plug in their own NLU module.
Quality	High — pre-trained NLP models for NLU and entity extraction are available. Built-in entity types.	High — Spacy or BERT are among solutions for NLP modules.
User-friendliness	Easy to use. Dialogflow offers GUI for training or analytics.	More difficult to start. GUI is available via external project — Rasa UI [126].
Deployment	Offers out-of-the-box deployment to Slack or Messenger (on approval of the third party). Web integration is available via an external tool [127].	Own deployment — no support.
Data collection	Data stored in the Cloud ecosystem and the Dialogflow environment.	Own deployment — no support.

Table 2.1 compares Rasa and Dialogflow in their free plans according to the criteria listed above.

Ultimately, we decided to use Dialogflow in our work. The main reasons are GUI, data collection and deployment options combined with satisfactory functionalities and customisation available within the free plan.

Chapter 3

XAI-bot — conversational system for model explanations

In this work, we propose a conversational interface to explore user needs in the context of XAI. For this purpose, we build the XAI-bot. It is a chatbot with a task of answering people's questions regarding the decisions of the Machine Learning model and explanations of these decisions. For the experiment conducted in this work (Chapter 4), we choose a well-known Titanic dataset (Section 4.2.1) and the Random Forest blackbox model (Section 4.2.2) trained to predict the survival of the passengers during the infamous maritime disaster in 1912.

We start this chapter by describing the capabilities of XAI-bot. Then, we list all actors and components within the system. Later we discuss the versatility of the design and suggest how to reuse it for experiments with conversations around other datasets and models. Finally, we demonstrate the chatbot on example dialogue presented from various perspectives. We also illustrate NLU and NLG components on examples.

3.1. Capabilities

This chatbot is a *user-initiative*, *multi-turn* agent. It uses the hybrid design approach. On the one hand, it is a *task-oriented*, *frame-based* agent with a *dialogue state*. On the other hand, it utilises Machine Learning NLU algorithms and the template-based NLG. The agent uses a textual interface with visual elements. It was implemented in two iterations (see Figure 4.1) using the Dialogflow framework and the Google Cloud Platform (GCP) [128] technological stack. The bot takes advantage of prompt buttons (also named as suggestion buttons) in some limited use cases. For instance, if a user asks to list variables of the dataset or requests help, the bot replies with suggestion buttons (an example usage in Figure 3.2). However, we want the prompt buttons to serve only as a support of the conversation and use them sparingly since the primary goal is to learn the questions posed by users and explore their needs rather than impose any specific utterances.

We have built the chatbot iteratively based on the collected human-agent interactions. It started with an initial collection of intents, each defined with a set of training sentences. The main results of the retraining process were: (1) adding new intents for unaddressed queries, (2) extending the training set with the actual user utterances. For further details on this process, see Section 4.2. We deployed the final version of the agent for the experiment described in

Chapter 4. The chatbot understands 40 user intents backed by 874 training sentences (see a complete list in Appendix B). As a result, the dialogue agent is capable of understanding and responding to several groups of queries:

- **Supplying data** about the passenger. For instance, specifying their age or gender. Users can omit this step by impersonating one of two predefined passengers with different model predictions.
- **Inference** — telling users what are their chances of survival. The inference is made based on the information specified by the users. The model imputes any missing values.
- **Visual explanations** from the Explanatory Model Analysis toolbox [41]: Ceteris Paribus profiles [64] (addressing "what-if" questions) and Break Down plots [63] (presenting feature contributions). Importantly, we use these explanations to offer a warm start to the system by answering some of the anticipated queries. However, the principal purpose of this work is to learn what are the actual user queries and to perform a quantitative study of these needs.
- **Dialogue support** queries, such as listing and describing available variables or restarting the conversation.

Admittedly, the scope of the chatbot is limited, and it will likely fail to understand or answer some of the relevant user queries. However, restricting the agent scope to chatting about the model and its explanation is a deliberate choice — there is no need for the chatbot to conduct an off-topic conversation. Consequently, the agent resorts to the *fallback intent* for any user query it does not understand and informs the user about this fact. Nevertheless, the primary goal of this work is to explore user queries, including the discovery of questions that we had not anticipated initially. Lastly, the agent design allows the smooth addition of the new intent in the following development iteration.

3.2. System overview — actors and components

The conversational system proposed and implemented in this work consists of two actors and several components. Figure 3.1 provides an overview of the building blocks of this conversational framework. Additionally, the diagram presents the role and position of the actors within the system. Finally, we can track the conversation flow through the system components.

All actors and components of the system are described in detail below, following the logical order.

3.2.1. Explainee

Explainee is a human operator — an addressee of the system. They chat about the blackbox model, its decisions and explanations of these predictions. Explainee is a central actor to this framework. The goals behind this chatbot are (1) offering a conversational interface to explainees and (2) collecting their interactions with the chatbot.

3.2.2. Interface

This layer serves as an interface between an **explainee** and the **dialogue agent**. It communicates with the agent's engine passing messages between a user and the agent.

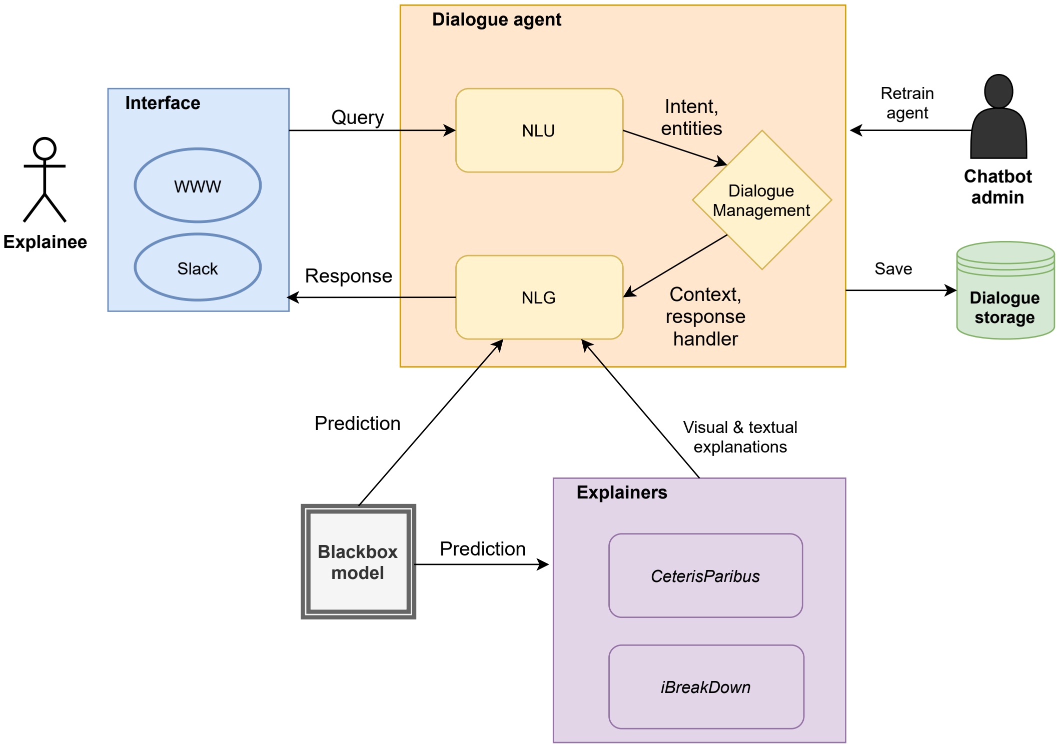
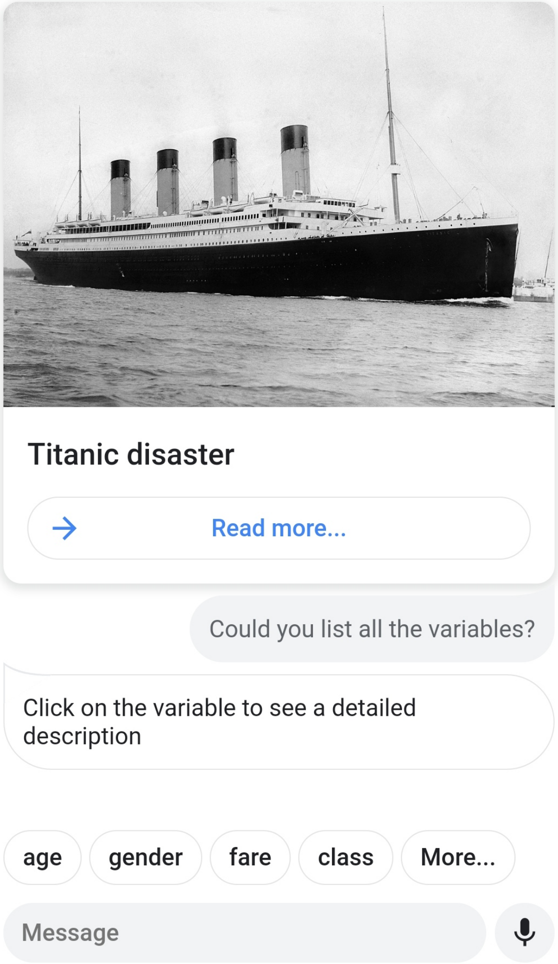


Figure 3.1: Overview of the XAI-bot architecture. **Explainee** uses the system to talk about the **blackbox model**. They interact with the system using one of the **interfaces**. The **dialogue agent** holds the conversation with an **explainee** and orchestrates the overall system. In order to respond to a user query, the **dialogue agent** asks the **blackbox model** for its predictions and **explainers** for visual explanations. All human-chatbot interactions are saved to the **dialogue storage**. The **chatbot admin** creates the **dialogue agent** and iteratively retrains it by leveraging the historical conversations from the **dialogue storage**. Adapted from [129].

The chatbot might be deployed to various conversational platforms independently of the backend and each other. The only minor exception to that is rendering some of the graphical, rich messages. These require slight adaptation between the interfaces. In this work, we experimented with Slack, Messenger and a custom web integration. Finally, we decided to proceed with the last one — a dedicated website. Such standalone web integration has a number of advantages. Firstly, it does not require a login (compared to Messenger, Slack or similar). It is open for everyone, while Slack limits a bot within a workspace and Messenger to the platform users. Furthermore, it allows a good balance between privacy and user data collection. Importantly, all visitors of the chatbot's website remain anonymous. Yet, we use Google Analytics to keep track of the aggregated statistics on the surveyed population (Figure 4.2). Finally, a custom web integration offers complete control over the rendering of the visual elements.

The frontend of the chatbot uses `Vue.js` and is based on a Dialogflow for Web project [127]. It provides a chat interface and renders rich messages, such as plots and suggestion buttons (see Figure 3.2). This integration allows having a voice conversation leveraging the browser's speech recognition and speech synthesis capabilities.



Titanic disaster

Could you list all the variables?

Click on the variable to see a detailed description

Figure 3.2: Demonstration of the XAI-bot rich responses including the graphical elements. This is a screenshot of an extract from an example conversation. It is the view on the web integration (a mobile device view). We can see an image-based response of the chatbot on the top. Below, we see clickable suggestion buttons representing variable names in this case. A click on such a button is equivalent to typing its query manually. At the very bottom, we can see text and speech input options. Photography "*RMS Titanic departing Southampton on April 10, 1912*." comes from the public domain.

3.2.3. Dialogue agent

A **dialogue agent** is an engine of the XAI-bot. It is responsible for holding a conversation with an **explainee** and orchestrating the overall conversational system. This dialogue agent is implemented using the Dialogflow framework [122]. The agent itself consists of three subcomponents (see Figure 3.1):

- **Natural Language Understanding (NLU)**
  The Natural Language component detects the intent of the user query. It also extracts and classifies the entities from this query (see Section 3.5 for a demonstration). NLU module recognises 40 intents. Some examples include: posing a what-if question, asking about a variable or specifying its value (see Appendix B for a complete list). Additionally, the NLU module comes with four entities. One allows capturing the variable name. The three other entities correspond to values of the categorical variables — gender, class and the place of embarkment. For numerical features, the entity extractor leverages the Dialogflow built-in numerical entity.

  The training set consists of 874 sentences distributed across all intents. Some of these sentences come from the initial subset of the collected conversations, so they are bits of the actual human-chatbot interactions. This intent classifier uses the Dialogflow built-in rule-based and Machine Learning algorithms. Finally, we take advantage of a built-in system for spellcheck correction.

- **Dialogue Management**
  The dialogue manager is responsible for orchestrating the conversation. Firstly, it calls an NLU module sending the raw user query. It receives a rich response, including the detected intent and a set of extracted entities. Then, it calls the NLG component to produce a response to the user. Additionally, the dialogue manager keeps track of the previous turns and maintains a representation of history with the notion of **dialogue state**.

  The dialogue state is a structure maintaining the current state of the conversation, including its previous turns. Dialogflow offers two flavours of the dialogue state — **state** and **context**. We follow the framework's terminology and introduce this distinction in our work. Dialogflow state is a data structure storing information obtained throughout the conversation. For this particular model, the state keeps track of the user (Titanic's passenger) information. Next, context is a representation of the current situation of the dialogue. It allows to condition a response on more than the last query. One example is when a user query is a single number. In general, this is an ambiguous utterance. However, when preceded by a query about age or fare, it is likely an attempt to specify this information. Dialogflow further introduces two views on contexts — input and output variants. Output contexts are activated on certain situations during the conversation. In our example, a question related to the "age" variable activates the corresponding output context. Contexts have configurable timespans counted in the number of dialogue turns. Therefore, if an intent is defined with the same input context, such intent has a higher probability of detection while the context is active. To summarize, Dialogflow's state and context provide powerful options for dialogue state management, allowing to condition responses on the entire conversation rather than the last query.

A technical realization of the dialogue management uses a fulfillment webhook. A corresponding function serves each intent with appropriate code logic. These functions are invoked by the dialogue manager when a corresponding intent is detected in the user query. We use `Node.js` language and run the fulfillment code with Google's Cloud Functions [130]. The fulfillment code along with the model and its explanations might be found in the project repository [131].

- **Natural Language Generation (NLG)**
  Response generation system. This is primarily a **template-based** generation system. It means the response is built using one of the hardcoded sentences (templates). The variable slots of the template are filled with available information according to the current conversation state (see Section 3.6 for examples).

  Moreover, an output of the NLG module might be an image or a visual template, such as an explanation plot with parameters set according to the query or state. Therefore, the dialogue agent might need to use explanations or model predictions to produce a chatbot's utterance. For this, the NLG component will query explainers or the model correspondingly. Plots, images and suggestion buttons that are part of the chatbot response are rendered as rich messages on the front end.

3.2.4. Blackbox model

A blackbox Machine Learning model. It is accessible by the dialogue agent via API. In this particular case, we use a Random Forest model trained to predict the chance of survival on Titanic. However, this design is versatile (see Section 3.3) and works for any model or dataset. This model was trained in R [132] and converted into REST API with the `plumber` package [133]. Details about the Titanic dataset and the model can be found in Section 4.2.1 and Section 4.2.2, respectively.

3.2.5. Explainers

The main task of the XAI-bot is to explain the decisions of the ML model to an explainee. To do that, it makes use of various explanation tools (explainers). We use visual and textual explainers from `iBreakDown` [63] and `CeterisParibus` [64] in this implementation. It is straightforward to plug in other explainers of the blackbox model. The communication schema relies on REST APIs. One API exposes the output of the explainers to the agent. Explainers, in turn, call the API of the blackbox model to retrieve model predictions. See the `xai2cloud` package [134] for more details on exposing explainers with API.

3.2.6. Dialogue storage

The primary goal of this work is to collect and explore human-chatbot interactions. For this reason, all conversations are saved and stored for later analysis. There are two destinations in which the logs are stored. There are also multiple uses of this data.

The first destination is Dialogflow's internal storage. The data from this destination is used in three different ways. Firstly, the Dialogflow Analytics module keeps track of the intent classification flow and presents the historical data using aggregated statistics. An example screenshot from Dialogflow Analytics can be seen in Figure 3.4. The second use is the Dialogflow Training interface (see Figure 3.5). It displays the historical conversations with an extended GUI. It allows the chatbot admin to add a user query to the training set, possibly coupled with the label reassignment in case of an intent misclassification. The third section — Dialogflow History — presents historical conversations with a simple GUI for the chatbot admin inspection. In this case, the chatbot admin might also view raw interaction logs (see an example in Appendix A). Raw interaction logs include additional metadata such as detected intent and confidence score for the decision on top of the plain utterances.

The second log destination is Google Stackdriver [135]. It is a cloud logging storage. The data saved here is stored in a raw format with metadata.

3.2.7. Chatbot admin

Human operator — developer of the system. Chatbot admin manually retrains the dialogue agent based on misclassified intents and misextracted entities. This particular chatbot was iteratively retrained on the initial subset of the collected dialogues. Chatbot admin uses historical conversations from the dialogue storage (see Section 3.2.6) and the GUI exposed by the Dialogflow Training tool.

3.3. Versatility and extensibility

In this work, we develop XAI-bot — a chatbot allowing the user to interact with a Random Forest model trained on the Titanic dataset and the explanations of this model. Notably, we chose this particular model and dataset purely for demonstration purposes. The same experiment could and we believe should be repeated for a wide range of datasets and models. The architecture from the Figure 3.1 is versatile. In this section we discuss its reusability potential and limitations.

First of all, we treat the model as a **blackbox** and only query it for its predictions. Therefore, models trained on the same dataset are fully interchangeable. In this particular case, we work with a Random Forest model for Titanic survival prediction. However, another Random Forest model, SVM or a neural network would work as good. This system is **model-agnostic**.

If another dataset is to be used, still much of the components can be reused straightaway. These include the XAI-bot's interface, storage and agent communication with the model and explainers. The main required adaptation lies within the dialogue agent. Change of the dataset needs to be reflected at least in updating the data-specific entities and intents. For instance, a new set of variables needs to be covered. It must also be followed by modifications of the training sentences for the new intents in the NLU module. Usually, NLG templates will require some adaptation as well. However, the general patterns for the dialogue support queries, explanation questions, and user intents design are largely transferable. Finally, the process of training, deployment and data collection stays the same regardless of these changes.

Extending the chatbot to address a new question comes down to the addition of a new intent. To plug in a new post hoc explanation method is a straightforward process.

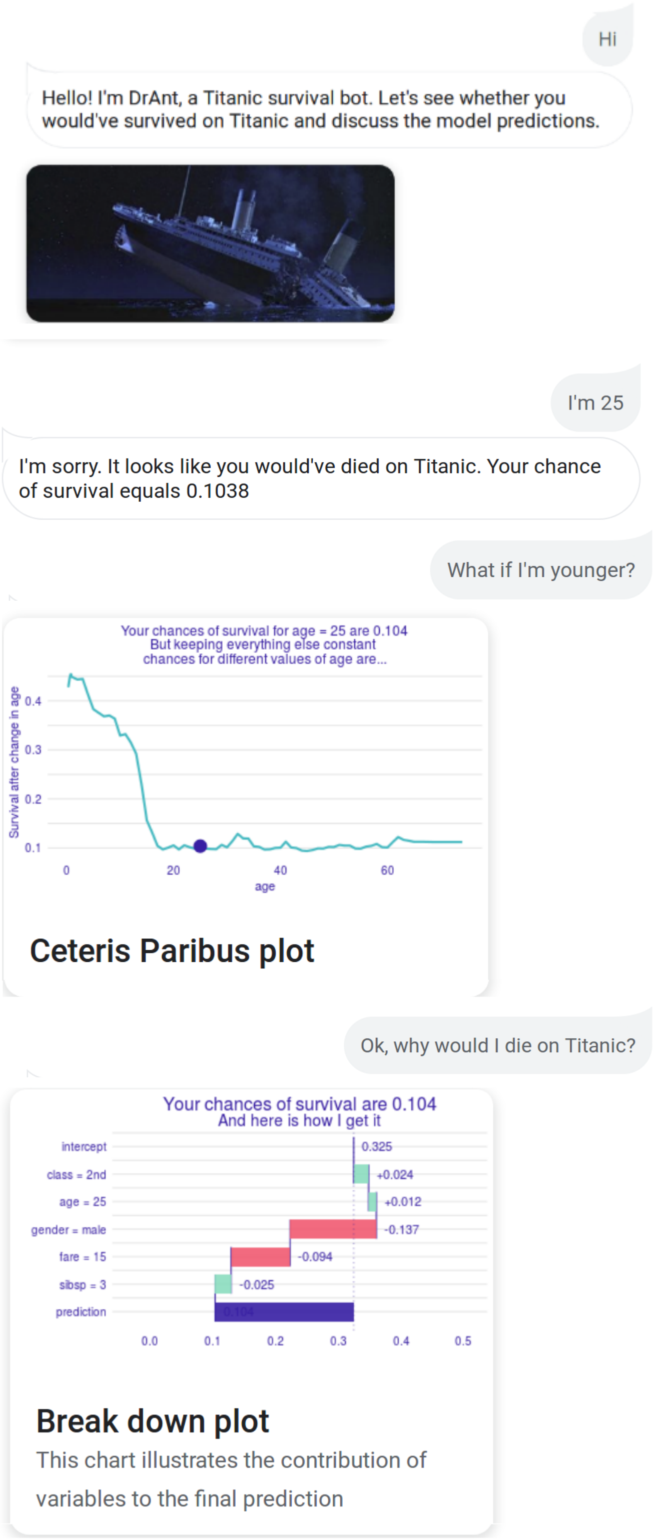
Admittedly, this framework comes with several limitations. Firstly, it is primarily designed to work with tabular datasets. It operates best when the variables are explicit, well-defined, and their number remains relatively small. However, it does not disqualify the XAI-bot from working with datasets from text, vision or speech fields. For instance, one could imagine, the user uploads a picture and asks queries about the predictions of the Convolutional Neural

Network. Finally, this framework is designed to work with Supervised Learning models. They can be both classifiers or regression models.

Extension of this framework to other data modalities as well as Machine Learning paradigms such as Reinforcement Learning is a direction for future research. However, the scope of applications for this framework remains broad. XAI-bot is a powerful way to explore user needs and offer conversational explanations. Furthermore, one can reuse XAI-bot easily for experiments with a range of datasets, models and audiences.

3.4. Example dialogue

In this section, we present a multi-turn excerpt from an example human-chatbot conversation from three different perspectives. Figure 3.3 presents the conversation from the front end perspective (as seen by the user). The figure illustrates that agent responses are conditioned by the state. In this particular case the state contains the age information specified by the user. Finally, we can see a bimodal form of the conversation — agent responses contain textual and visual elements (usually visual explanation plots). Secondly, Figure 3.4 presents the same conversation from an intent classification perspective. It is a flow chart representing the decision path of the NLU module. The green, highlighted path corresponds to the intent classification flow for the conversation from Figure 3.3. Finally, Figure 3.5 presents a view of this conversation from the Dialogflow Training tool.



Hi

Hello! I'm DrAnt, a Titanic survival bot. Let's see whether you would've survived on Titanic and discuss the model predictions.

I'm 25

I'm sorry. It looks like you would've died on Titanic. Your chance of survival equals 0.1038

What if I'm younger?

Your chances of survival for age = 25 are 0.104
But keeping everything else constant
chances for different values of age are...
Survival after change in age
0.1
0.2
0.3
0.4
0
20
40
60
age

Ceteris Paribus plot

Ok, why would I die on Titanic?

Your chances of survival are 0.104
And here is how I get it
intercept
class = 2nd
age = 25
gender = male
fare = 15
sibsp = 3
prediction
0.325
+0.024
+0.012
-0.137
-0.094
-0.025
0.104
0.0
0.1
0.2
0.3
0.4
0.5

Break down plot

This chart illustrates the contribution of variables to the final prediction

Figure 3.3: An excerpt from an example conversation. Queries of the explainee are on the right side and coloured grey. XAI-bot responses are on the left in white boxes. We can see the agent responds with textual and visual content. See Figure 3.5 for a more detailed technical view of this conversation, including detected user intents and extracted entities. The image on the top is a screenshot of the RMS Titanic breaking in half from the film Titanic (1997). Source: `https://en.wikipedia.org/wiki/Titanic_(1997_film)`. This figure is reprinted from [129].

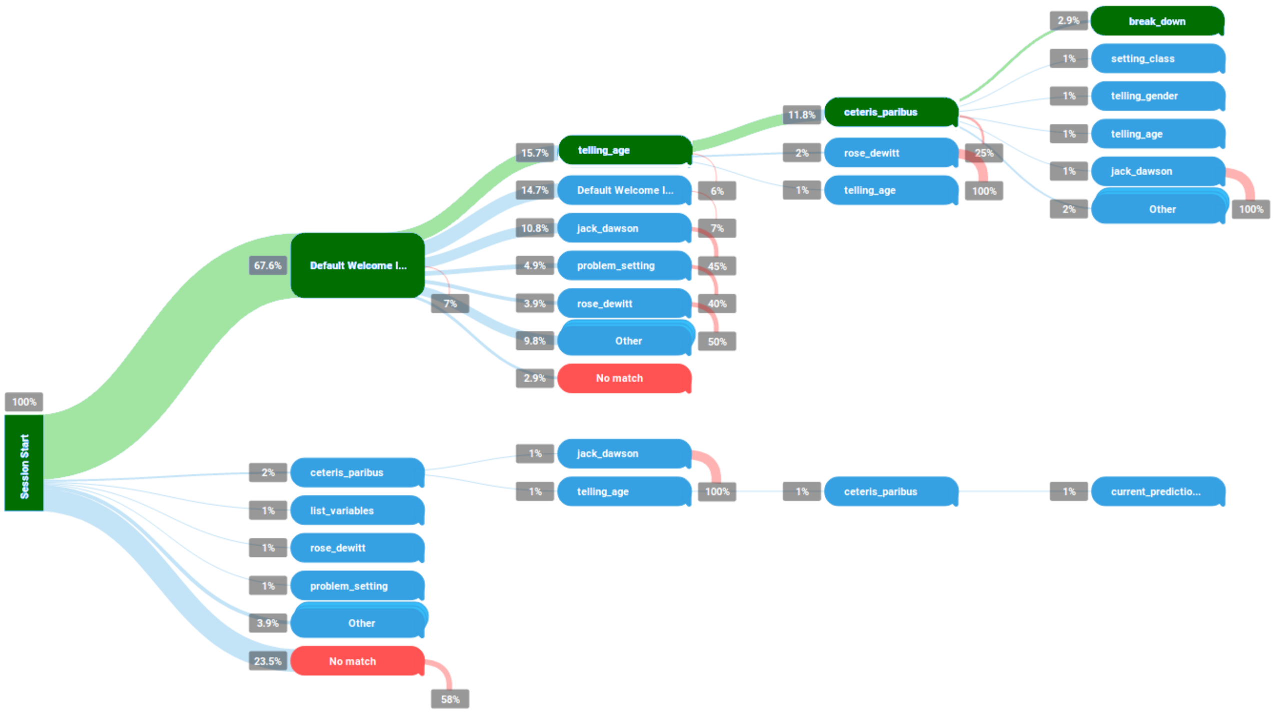
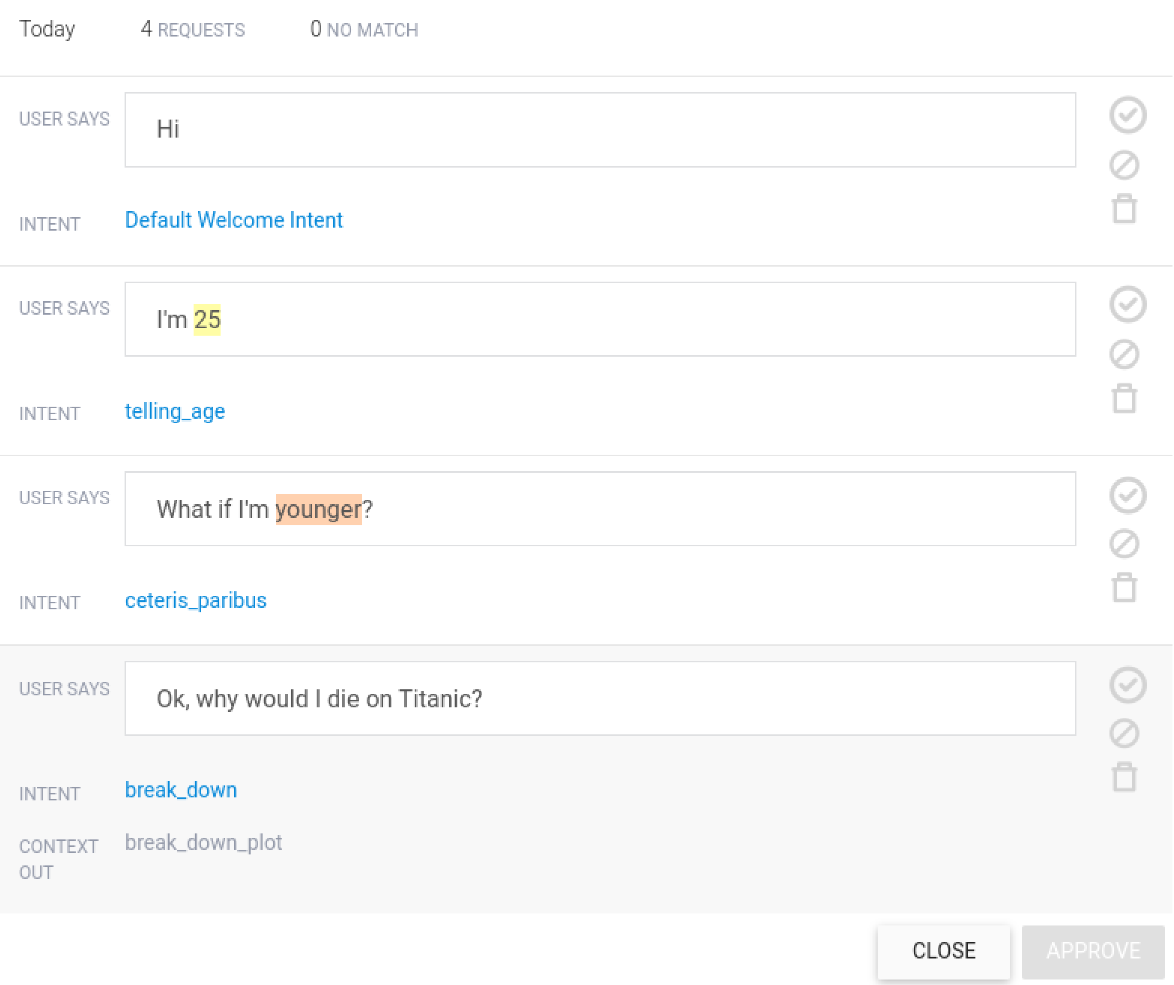


Figure 3.4: Screenshot from the Dialogflow Analytics. This flow chart demonstrates the work of the intent classifier of the NLU module. Each box corresponds to a classified intention of the query, e.g. *telling_ age* or *ceteris_ paribus*. This flow chart shows an intent tree based on a sample of dialogues. With an entry box marking a session start on the left, every next box along a path is a detected intent of the following query. For instance, the conversation from Figure 3.3 contributes to the topmost (green) path. Percentages and line widths indicate the share of the flow in the overall sample. Reprinted from [129].



Today 4 REQUESTS 0 NO MATCH

USER SAYS Hi

INTENT Default Welcome Intent

USER SAYS I'm 25

INTENT telling_age

USER SAYS What if I'm younger?

INTENT ceteris_paribus

USER SAYS Ok, why would I die on Titanic?

INTENT break_down

CONTEXT OUT break_down_plot

Figure 3.5: Screenshot from Dialogflow Training for the example conversation from Figure 3.3. This tool offers a detailed view of historical conversations with various options for agent retraining. It allows the chatbot admin to review the past dialogues and add the actual user utterances to the training data. The chatbot admin can add these sentences as positive (for an intent correctly detected by the system) or negative examples (for the fallback intent). In addition, the tool allows the correction of detected intent (under the user utterances, marked with blue) or extracted entities (highlighted with colours). Finally, using this tool, we can see information about the chatbot contexts. In this example, we can see that the last query activates the *break_down_plot* output context. Therefore, the following few user questions would be interpreted with information about a recent conversation turn related to the Break Down plot.

3.5. Natural Language Understanding examples

The Natural Language Understanding (NLU) module of this chatbot is responsible for two tasks: (1) **intent classification** and (2) **slot filling** or **entity extraction**.
In the first one, the user utterance and the conversation state are used to predict the user intent (see Appendix B for the complete list of intents with examples). Each of the possible values is assigned a confidence score. Then, the output of the intent classifier is the intent with the highest score. However, if none of the intents has a confidence score above the threshold, the classifier resorts to the fallback intent.
The role of the entity extractor is to detect and extract entities from the user sentences. Definitions of the entity types are part of the system design. Some of them leverage the Dialogflow built-in types, such as a numerical type for age. Other entity types are defined with lists of their possible values. For instance, an entity type denoting a dataset variable is defined with a list of possible variable names (e.g. age, gender, class). Example outputs of intent classifier and entity extractor are presented below.

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: {}

Query: *What is the meaning of life?*
Intent: *fallback_intent*
Entities: {}

3.6. Natural Language Generation templates

The NLG module of this chatbot relies on **template-based** responses. For each user intent, the bot response is generated using one of the available templates. To enrich the conversation, we provide multiple equivalent templates for many of the intents. Thus, at any turn, one of the templates is chosen randomly. Each such template might have a set of parameters (possibly empty), filled according to the state of the conversation. Note that these parameters do not have to equal the entities extracted from the very last query. See below for examples:

User query: *What is my chance of survival?*
User intent: *current_prediction*
Response template: *Your chance of survival equals* ***{probability}****.*
Response parameters: *{probability: 0.45}*
Chatbot response: *Your chance of survival equals 0.45. It's close to a toss of a coin!*

User query: *Reset age.*
User intent: *clear_variable*
Response template: *Variable* ***{variable}*** *was cleared.*
Response parameters: *{variable: age}*
Chatbot response: *Variable age was cleared.*

User query: *What model are you using?*
User intent: *model_behind*
Response template: *The predictions are made by a Random Forest model.*
Response parameters: *{}*
Chatbot response: *The predictions are made by a Random Forest model.*

Chapter 4

Experiments

In Chapter 3, we describe the XAI-bot — conversational explanatory system using a chatbot. The development of this system is the first goal of our work. The second goal is to use XAI-bot for end-user queries collection. In this chapter, we propose and demonstrate this procedure, addressing the second goal. For that, we perform a wide-scale experiment with a human audience. As a result, we collect and analyse explanatory needs based on 1000+ human-agent interactions. We start this chapter with a brief overview of previous approaches and studies related to end-user needs discovery. Then, we describe the setup of the experiment — we present the project lifecycle, as well as the dataset and model used for this particular experiment. Next, we display the basic statistics of the surveyed sample. Finally, we present the aggregated results of the experiment. These results are user queries grouped into certain categories, along with the occurrence count of each class. We conclude with a discussion of the collected results.

4.1. Previous user studies

XAI methods are usually designed for ML practitioners as end-users [12, 13, 14]. Thus, the **explanatory needs of the lay audience remain largely unstudied and consequentially ignored**. Recently, the term of Human-Centred Explainable AI (**HCXAI**) was coined [18, 19]. Consequently, we can see initial attempts towards discovering user needs related to XAI. In this section, we list the few existing efforts in this area. Most of them rely on manual user studies in the form of interviews. Unfortunately, such studies are costly, not reusable and do not scale well across tasks and audience types. At the end of this section, we review the previous attempts at using conversational agents for user needs collection — similar to our work here.

In one user study, Liao et al. [136] diagnose the gap between algorithmic XAI work and user needs. First, the authors interview 20 UX and design practitioners from IBM. Then, they create the XAI Question Bank based on the taxonomy of the existing tools and supplement it with the collected interview questions. Similarly, Dhanorkar et al. [137] interview 30 ML practitioners within a single company to discover their perspective on the XAI needs throughout the ML project lifecycle.

Gao et al. [107] use a dialogue agent to explain the anomaly detector for IT Operations events. They use the chatbot for interviewing four engineers who are the end-users of the model and not ML practitioners, while they do have some knowledge of the field. In another work, Pecune et al. [106] use their conversational agent to explain movie recommendations. They conduct an experiment with users from Amazon Mechanical Turk and have collected 60 interactions. They perform a survey after each conversation to evaluate the explanation quality based on the feedback from the explainees.

Concluding, there are few existing surveys of user needs. They are limited in scale and usually confined to ML practitioners. The XAI-bot we propose in this work is promising to survey various audiences in a scalable and reusable manner.

4.2. Experiment setup

The research from this thesis was a **three-step work** (see Figure 4.1). We first trained the Random Forest model (see Section 4.2.2) on Titanic dataset (see Section 4.2.1) and created the first version of the chatbot (see Chapter 3). In the second stage, we collected the initial set of conversations from actual human-agent interactions and retrained the agent based on this data (see Section 3.2.7). Finally, we froze the agent development to conduct the full-fledged experiment with users. In this section, we describe the setup of this experiment conducted in the third stage of the project.

The audience of this experiment is primarily Data Science and R communities. We have collected the data throughout two weeks. This period proved sufficient to yield a large enough dataset. We do not fully control the surveyed sample other than by sharing the link to the chatbot's website on the groups and forums of choice. However, by using Google Analytics, we track basic information about the audience (see Section 4.2.3).

At the beginning of the conversation, users are instructed about the goal of the experiment. Then, they are generally encouraged to explore the model and its explanations and think of any questions they have. In Section 4.3, we describe the collected queries and declare any hints from the system or sources of bias users might have. However, the general idea is to support the user through the conversational process without suggesting any specific explanatory queries.

4.2.1. Dataset

The Titanic's sinking in 1912 was a deadly disaster resulting in over 1,500 deaths. Survival prediction from this catastrophe became a classic Machine Learning problem. The Titanic dataset is widely used in tutorials to demonstrate data processing techniques and various Machine Learning algorithms. This task is a **classification problem**. The predicted target is a **binary variable** representing the **survival** from the ship disaster.

In this work, we demonstrate the XAI-bot on the example of an ML model trained to predict survival on Titanic. We chose this problem for a range of reasons. Firstly, an object of the prediction is a human, namely the ship passenger. Therefore, it allows users to engage smoothly in the conversation about the model by impersonating the Titanic passenger. Secondly, the task is relatively easy to understand by a wider audience. It has a relatively small number of variables. However, while simple, this dataset has several interesting characteristics. It comes with both categorical and numerical features as well as interactions between variables.

There are multiple sources of information about the disaster and the Titanic's passengers [138, 139]. In this work, we use a dataset included in the `DALEX` package [78] and described in [41]. It contains 2,207 rows, each corresponding to one of the passengers or crew members.

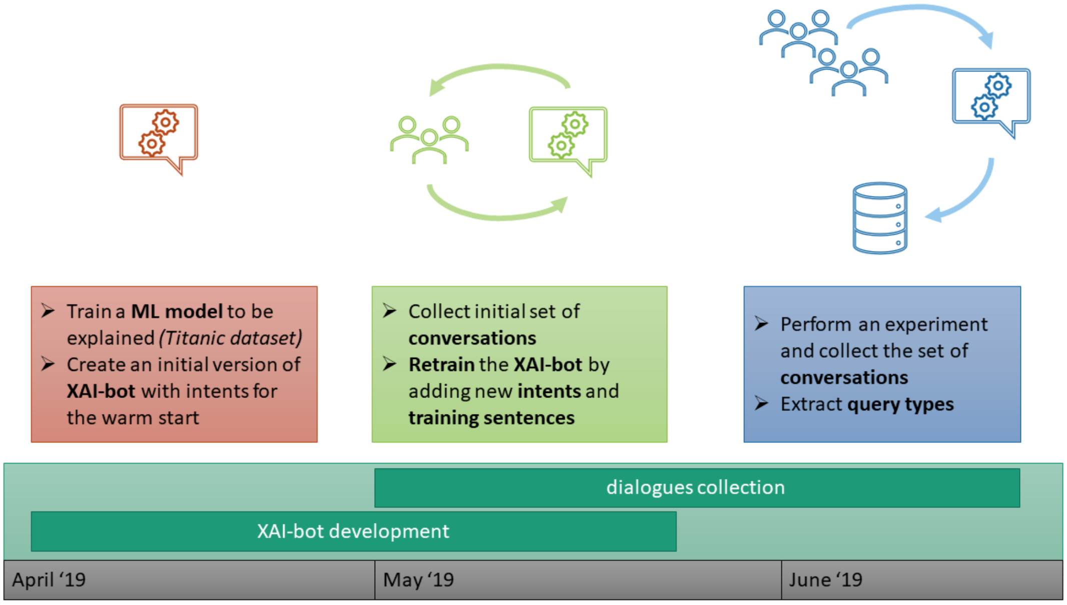

Train a ML model to be explained (Titanic dataset)
Create an initial version of XAI-bot with intents for the warm start
Collect initial set of conversations
Retrain the XAI-bot by adding new intents and training sentences
Perform an experiment and collect the set of conversations
Extract query types
dialogues collection
XAI-bot development
April '19
May '19
June '19

Figure 4.1: Three phases of the project lifecycle. In the **initialization** step, we train the model and create the explanatory agent. Then, we update the agent in the **retraining** phase based on the initial set of collected interactions. The final third stage is the **experiment** with a broader audience when we collect the user queries from the human-agent conversations.

Table 4.1 lists all variables along with their short description. Additionally, Table 4.2 presents a small data sample.

4.2.2. Model

In this work, we use a Random Forest classification model. Random Forest models display good predictive performance and the ability to capture low-order variable interactions [37]. Significantly to our application, the Random Forest models can output the probability scores along with the predicted class. Crucially, as we are concerned with the blackbox models, Random Forests are viewed as such. While small decision trees are deemed interpretable, it is no longer the case with Random Forests [140].

This model was trained with the default set of hyperparameters using the `randomForest` package [141]. Data preprocessing includes imputation of missing values. The performance of the model on the test dataset was: AUC 0.84 and F1 score 0.73. This model can be downloaded from the `archivist` [142] database with the following hook: `archivist::aread("pbiecek/models/42d51")`.

4.2.3. Statistics of surveyed sample

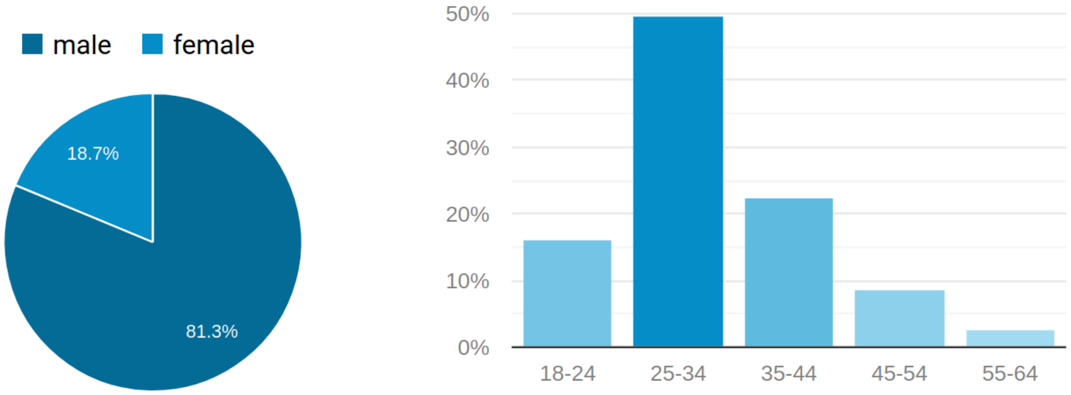
We use Google Analytics to gain insights into the audience of the experiment. Users are distributed across 59 countries, with the top five (Poland, United States, United Kingdom, Germany and India, in this order) accounting for 63% of the users. Figure 4.2 presents demographic data on the subset of the audience (53%) for which this information is available.

Table 4.1: List of variables for the Titanic dataset used in this work.

Variable	**Description**
Gender	Passenger's gender (male/female).
Age	Age in years (integer), range 0–74.
Class	Class of the passenger ticket (1–3) or a duty class of the crew — one of deck, engineering, victualling crew or the restaurant staff.
Embarked	Harbour of embarkment — one of Belfast, Cherbourg, Queenstown or Southampton.
Fare	Ticket price (numerical) or 0 for the crew. Range of 0–512.
Sibsp	Number of siblings/spouses on the board. Numerical in range of 0–8.
Parch	Number of parents/children on the board. Numerical in range of 0–9.
Survived	Target variable, categorical (0/1).

Table 4.2: Sample of rows from the Titanic dataset used in this work.

gender	age	class	embarked	fare	sibsp	parch	survived
male	13	3rd	Southampton	20.05	0	2	0
female	39	3rd	Southampton	20.05	1	1	1
male	25	3rd	Southampton	7.13	0	0	1
female	28	2nd	Cherbourg	24.00	1	0	1
male	30	2nd	Cherbourg	24.00	1	0	0



(a) Gender distribution.

(b) Age distribution.

Figure 4.2: Demographic statistics of the surveyed sample from Google Analytics. Note, this demographic information is only available to a subset of users (53%).

4.3. Results

As a result of this experiment, we collected a number of human-agent interactions (see 3.2.6 for details on the dialogue storage). Next, we filter out conversations with totally irrelevant content and those with less than 3 user queries. Finally, we have obtained 621 dialogues consisting of 5,765 user queries in total. The average interaction length is 9.14, maximum 83 and median 7 queries. The histogram in Figure 4.3 presents the summary of interaction length. Note that the number of turns is two times bigger (each user query receives a chatbot response).

4.3.1. Statistics of extracted query types

In this section, we analyse the collected conversations. We extract explanatory queries, manually classify and group them into certain categories. For each such category, we calculate the number of conversations with at least one query of this type. These occurrence counts are presented in Table 4.3.

Notably, the taxonomy defined below is independent of user intents recognised by the NLU module. While generally, the chatbot does not prompt users to ask any specific explanatory question, we explicitly flag one such exception. Finally, there might be a certain bias resulting from the potential audience familiarity with the tools and methods of explainable AI. Below we list the query types in the decreasing order of frequency defined as in Table 4.3. We describe each category and list some example queries.

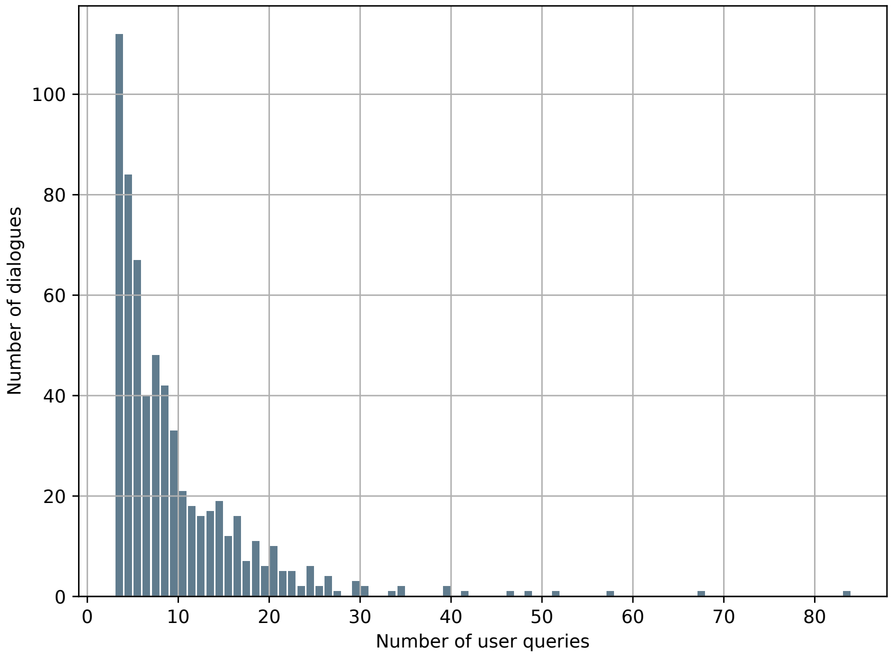


Figure 4.3: Histogram of conversations length (number of user queries) after filtering out interactions shorter than 3 queries. Reprinted from [129].

Table 4.3: Results of the analysis for 621 conversations obtained in the experiment. The last column presents the number of dialogues with at least one query of a given type. A single conversation might contain multiple or none of these queries. Adapted from [129].
* Query prompted to the user by the chatbot

	Query type	**Dialogues count**
Q.1	why/how	73
Q.2	what-if	72
Q.3	EDA	54
Q.4	feature importance	31
Q.5	counterfactuals (how to improve)	24
Q.6	feature comparison/global feature effect	22
Q.7	who has the best score	20
Q.8	model-related	14
Q.9	explicitly contrastive	1
Q.10	plot interaction	0
Q.11	similar observations	0
Q.12	what do you know about me*	57
	Number of all analysed dialogues	**621**

Q.1 **why/how** — general explanation queries, typical examples of such are:

- *"why?"*
- *"explain it to me"*
- *"how was this calculated?"*
- *"why is my chance so low?"*

Q.2 **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 variants such as: *What if I'm older and travel in a different class?*.

Q.3 **EDA** — a general category on Exploratory Data Analysis. All questions related to the exploration of data rather than the model 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* and similar.

Q.4 **feature importance** — here, we group all questions about the feature's relevance, influence, importance or the contribution to 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?*

Q.5 **counterfactuals (how to improve)** — actionable queries for maximizing the prediction, e.g. *what should I do to survive*, *how can I increase my chances.* These queries might be seen as a call for counterfactual explanations.

Q.6 **feature comparison/global feature effect** — comparison of the predictions across different values of the categorical variable. It might be seen as a global variant of the *what-if* question. Examples: *which class has the highest survival chance*, *are men more likely to die than women.*

Q.7 **who has the best score** — here, users 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.

Q.8 **model-related** — these are the queries related directly to the model rather than its predictions. Most of the time, we see questions about the algorithm and the code. Additionally, we see users asking about metrics (accuracy, AUC), confusion matrix and confidence. However, these are observed about a dozen times.

Q.9 **explicitly 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* (Jack is a name for one of the two predefined characters users can impersonate).

Q.10 **plot interaction** — follow-up queries to interact with the displayed visual content. Not observed.

Q.11 **similar observations** — queries regarding "neighbouring" observations. For instance, *what about people similar to me.* Not observed.

Q.12 **what do you know about me** — this is the only query suggested to the explainee using the prompt button. When the user inputs their data manually, it usually serves to understand what is yet missing. However, especially in the scenario when the explainee impersonates a movie character, it can aid understanding which information about the user is possessed by the system. As such, it might be thought of as an explanation query related to data control and privacy.

4.3.2. Discussion

We identify **ten categories** with at least one corresponding query. While most of them relate to the decisions of the model, there are two exceptions. Firstly, "EDA" (Q.3) is a group of questions focused on understanding the dataset. Secondly, "model-related" (Q.8) is a set of queries about the model and its statistics rather than predictions.

Arguably, the types of most frequent XAI queries (Q.1, Q.2, Q.4, Q.5, Q.6) are not surprising and are consistent with theoretical work [20] and other user studies such as [136]. The less expected is the negative outcome — the queries with little or no occurrences. We record no questions related to similar observations (Q.11) — a form of case-based explanations. However, we can see several questions related to the single observations with extremum values (Q.7). The lack of explicitly contrastive queries (Q.9), while surprising, might be explained by the theory of implicit foils (see Section 1.5). Accordingly, such implicit contrastive queries might be found in other categories. One example of such query (from Q.1) is *"why is my chance so low"*.

Our results are consistent with the XAI Question Bank from [136]. In fact, we offer more than that. We present a **study-driven audience-specific** and **application-specific XAI question bank** with a **quantitative dimension**. We hypothesise that such a question bank would be qualitatively and quantitatively different for other audiences and domains. For instance, we do not expect a lay audience without ML expertise to operate on terms such as model feature or variable. Similarly, following Table 1, we anticipate different questions in different application fields.

The advantages of this study and the proposed procedure are the **quantitative outcome** of the survey, **scalability** and **repeatability** of the process. Following this procedure, we obtain concrete figures based on large-scale studies with users. If we were to create a XAI system for the given context (audience and dataset), we could address the very questions from the experiment and prioritise them based on their frequency. Finally, one additional insight from this study is the considerable interest in EDA, which might also benefit from human-agent interactions.

Chapter 5

Summary

Explainable AI is an important and growing field. It requires methods and tools addressing the actual needs of the diverse set of end-users. We diagnose four significant issues that we address in this thesis:

I.1 Explainability could considerably benefit from **interactivity** and **conversation** [20].

I.2 Depending on the area of application or the audience type, **different needs** are linked with the concept of explainability [16, 24, 34].

I.3 Explanation systems are usually designed by ML engineers for themselves. Consequently, the needs of the **diverse audience** of ML-related explanations often remain **unstudied** and therefore **ignored** (see [12, 13, 14] and Figure 1).

I.4 Collection of end-user needs relies on manual user studies (see Section 4.1). Such a solution is costly and **scales poorly**.

In this chapter, we recapitulate achieved results that address the issues listed above. Additionally, we envision directions for future research and propose potential continuations to this work.

5.1. Achieved results

In this thesis, we have presented a novel application of the dialogue system for conversational explanations of a predictive model.

The three detailed contributions are listed below:

1. **We build a XAI-bot and demonstrate it on the example of a binary classification model for the Titanic dataset.** It is a promising, novel approach to the human-agent XAI interface addressing the call for interactivity and use of conversation (ad Issue **I.1**). Users receive a tool for the interactive explanation of the model's predictions. In the future, such systems might be beneficial in bridging the gap between automated systems and their end-users.
2. **We propose a process based on a dialogue system allowing for an effective collection of user expectations related to model explanation.** This is a repeatable and scalable process (ad Issue **I.4**) allowing for the deployment of an agent to

various end-user groups (ad Issue **I.2**). We recommend this methodology to support the user-centred design of any XAI system deployed to the users (ad Issue **I.3**).

3. **We perform an example experiment with 1000+ collected conversations, analyse and discuss the achieved results.** This analysis identifies several frequent patterns among user queries and validates our initial hypotheses about such questions (ad Issue **I.3**). To sum up, in this experiment, we gather requirements for the XAI system in one particular scenario and prove the potential of the proposed procedure.

To the best of our knowledge, this is one of the first implementations of the conversational agent for XAI and the first use of such an agent to collect user needs at scale.

The results of this work were presented at the ECML PKDD 2020 International Workshop on eXplainable Knowledge Discovery in Data Mining. Additionally, these results were published in the ECML PKDD 2020 conference proceedings [129]. Finally, the author presented the preliminary version of this work at the MLinPL 2019 conference during the poster session [143].

5.2. Future Work

We believe this work is an initial effort towards conversational explanations and opens up several avenues for further research. We envision that this process of user needs collection should be repeated for other datasets and domains, e.g. legal, medical and financial. Likewise, we would like to deploy such agents to various audience types, repeat a similar experiment and compare the collected queries. In particular, it would be interesting to analyse the results from explainees without Machine Learning expertise. Additionally, we propose to extend the conversational agent to encourage a feedback loop to gather detailed insights into user needs. Furthermore, based on the collected results, we foresee two related domains benefiting from human-agent interactions — Exploratory Data Analysis (EDA) and algorithmic fairness. We justify the latter with most questions referring to sensitive and bias-prone features such as gender or age. Therefore, the XAI-bot could offer insights into model fairness. However, this should be further examined with a more suitable dataset. Finally, we believe that this work could be extended to adapt non-tabular datasets.

Appendix A

Chatbot raw interaction logs

From a high-level perspective, the communication process of the dialogue agent looks simple: receive a query from the user and send back a response. However, if we look at the system architecture (Figure 3.1), there is a number of components exchanging information with each other. Additionally, these messages have a structured JSON format including some metadata.

We present two excerpts from the raw interaction log of the dialogue agent related to an example user query. Listing A.1 demonstrates an output of the NLU component. We can see extracted entities, detected intent along with the confidence score and some additional metadata. In Listing A.2, we abstract out the NLG response and the diagnostics of the fulfillment webhook. It is a rich response format, ready to be rendered on the frontend side.

Listing A.1: An extract of the NLU component raw API output. It is a structured response to the user query: *"What would be my chance of survival had I been younger?"*. It is an extract — we present only a subset of fields from the entire JSON response file.

```
{
    "queryText": "What would be my chance of survival had I been younger?",
    "intent": {
      "displayName": "ceteris_paribus"
    },
    "intentDetectionConfidence": 0.7057004,
    "parameters": {
      "variable": [
        "age"
      ]
    },
    "allRequiredParamsPresent": true,
    "languageCode": "en",
}
```

Listing A.2: An extract of the NLG component raw API output. It is a continuation of Listing A.1. We see an end result — a list of messages in a structured format to be displayed on the fronted part. Additionally, we can see some diagnostics of the fulfillment webhook. It is an extract — we present only a subset of fields from the entire JSON response file.

```
{
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "Creating a plot. It may take a few seconds..."
          ]
        }
      },
      {
        "card": {
          "title": "Ceteris Paribus plot",
          "imageUri": "http://52.31.27.158:8787/ceteris_paribus?age=25&gender=X&fare=X&class=X&parch=X&sibsp=X&embarked=X&variable=age",
          "buttons": [
            {
              "text": "See a larger plot",
            }
          ]
        }
      }
    ],
    "diagnosticInfo": {
      "webhook_latency_ms": 71
    },

    "webhookStatus": {
        "message": "Webhook execution successful"
    },
}
```

Appendix B

List of intents of the dialogue agent

NLU module of the chatbot recognises 40 user intents. We list them in Table B.1. For each intent we provide one example of the training sentence.

Table B.1: List of all intents of the NLU module of the chatbot. Each user query is assigned one of these intents. The example sentence is an instance from the training data for each intent. Note that intent classification also depends on the current context. For illustration, the utterance "two" can be understood as the specification of age or fare if the context indicates so.

Category	Intent name	Example sentence
Supplying data	*set_age*	*I'm 25 years old*
	specify_age [1]	*25*
	set_gender	*I'm female*
	set_fare	*I paid 210*
	specify_fare	*210*
	set_class	*I traveled in 1st class*
	set_parch	*I travelled with two children*
	specify_parch	*2*
	set_sibsp	*I travelled with my brother and wife*
	specify_sibsp	*two*
	set_embarked	*I boarded in Belfast*
	multi_slot_filling	*I'm 25 yo man in 3rd class*
	travelling_alone	*I traveled on my own*
	clear_variable	*reset age*
	jack_dawson [2]	*I'm Jack*
	reset_jack	*I'm not Jack*
	rose_dewitt	*I'm Rose from the movie*
	reset_rose	*Stop being Rose*
Inference	*current_prediction*	*What is my chance of survival?*

[1] Intents of a form *set_variable* assume a self-contained query, whereas intents *specify_variable* require particular context based on the previous turns to be interpreted adequately.

[2] Jack and Rose are the names for predefined characters users can impersonate. These names originate from the Titanic movie (1997).

(Visual) explanations	*available_plots*	*What plots are available?*
	break_down	*What is the most important feature?*
	break_down_description	*What is break down plot?*
	ceteris_paribus	*What if I am older?*
	specify_variable_ceteris	*place of embarkment*
	how_to_survive	*What should I do to survive?*
	intercept	*What does intercept in the break down plot mean?*
Dialogue support	*Default Fallback Intent* [3]	*What is the capital of Germany?*
	Default Welcome Intent	*Hello*
	end_conversation	*I want to finish this*
	list_variables	*List me all variables*
	current_knowledge	*What do you know about me?*
	explain_feature	*What do you mean by sibsp?*
	expressing_dissatisfaction	*Ok, you've told me that already...*
	help_needed	*What can I do here?*
	restart	*I want to start again*
	praise	*Nice!*
	problem_setting	*Decribe the problem*
Misc	*authorship*	*Who authored this?*
	how_many_survived	*How many survived on Titanic?*
	model_behind	*What model are you using?*

[3] Fallback intent is a special case. It might have training sentences as "negative" examples. However, it primarily serves the purpose of answering the queries not classified to any other defined intent.

Bibliography

[1] Mark O. Riedl. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33–36, 2019.

[2] Teresa Scantamburlo, Andrew Charlesworth, and Nello Cristianini. Machine decisions and human consequences. *arXiv preprint arXiv:1811.06747*, 2018.

[3] Mireia Ribera and Àgata Lapedriza. Can we do better explanations? A proposal of user-centered explainable AI. In *Intelligent User Interfaces (IUI) Workshops*, 2019.

[4] European Commission. Data protection in the EU. Recital 71. `https://gdpr-info.eu/recitals/no-71/`. Accessed 2021-10-19.

[5] LH Gilpin, D Bau, BZ Yuan, A Bajwa, M Specter, and L Kagal. Explaining explanations: An approach to evaluating interpretability of ML. *arXiv preprint arXiv:1806.00069*, 2018.

[6] Shane T Mueller, Robert R Hoffman, William Clancey, Abigail Emrey, and Gary Klein. Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. *arXiv preprint arXiv:1902.01876*, 2019.

[7] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019.

[8] Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[10] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[11] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. *arXiv preprint arXiv:1712.00547*, 2017.

[12] Juliana Jansen Ferreira and Mateus Monteiro. Designer-User Communication for XAI: An epistemological approach to discuss XAI design. *arXiv preprint arXiv:2105.07804*, 2021.

[13] Oyindamola Williams. Towards Human-Centred Explainable AI: A Systematic Literature Review. `https://dx.doi.org/10.13140/RG.2.2.27885.92645`, 2021.

[14] Aaron Springer. Enabling Effective Transparency: Towards User-Centric Intelligent Systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 543–544. Association for Computing Machinery, 2019.

[15] Prashan Madumal, Tim Miller, Frank Vetere, and Liz Sonenberg. Towards a grounded dialog model for explainable artificial intelligence. *arXiv preprint arXiv:1806.08055*, 2018.

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

[17] Stuart K Card, Thomas P Moran, and Allen Newell. *The psychology of human-computer interaction.* Crc Press, 2018.

[18] Upol Ehsan and Mark O Riedl. Human-centered explainable AI: Towards a reflective sociotechnical approach. In *International Conference on Human-Computer Interaction*, pages 449–466. Springer, 2020.

[19] Upol Ehsan, Philipp Wintersberger, Q Vera Liao, Martina Mara, Marc Streit, Sandra Wachter, Andreas Riener, and Mark O Riedl. Operationalizing Human-Centered Perspectives in Explainable AI. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2021.

[20] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[21] Mennatallah El-Assady, Wolfgang Jentner, Rebecca Kehlbeck, Udo Schlegel, Rita Sevastjanova, Fabian Sperrle, Thilo Spinner, and Daniel Keim. Towards XAI: Structuring the Processes of Explanations. In *ACM Workshop on Human-Centered Machine Learning*, 2019.

[22] Kacper Sokol and Peter Flach. One Explanation Does Not Fit All. *KI - Künstliche Intelligenz*, 2020.

[23] Mansoureh Maadi, Hadi Akbarzadeh Khorshidi, and Uwe Aickelin. A Review on Human–AI Interaction in Machine Learning and Insights for Medical Applications. *International Journal of Environmental Research and Public Health*, 18(4), 2021.

[24] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019.

[25] Hubert Baniecki and Przemyslaw Biecek. modelStudio: Interactive Studio with Explanations for ML Predictive Models. *The Journal of Open Source Software*, 2019.

[26] Oege Dijk. ExplainerDashboard documentation. `https://explainerdashboard.readthedocs.io/en/latest/`. Accessed: 2021-10-15.

[27] Patrick Hall, Navdeep Gill, Megan Kurka, and Wen Phan. Machine learning interpretability with H2O driverless AI. *H2O.ai*, 2017.

[28] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.

[29] Hubert Baniecki and Przemyslaw Biecek. The grammar of interactive explanatory model analysis. *arXiv preprint arXiv:2005.00497*, 2020.

[30] Martin G Helander. *Handbook of human-computer interaction*. Elsevier, 2014.

[31] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 77–95. Springer, 2020.

[32] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[33] Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2019.

[34] Zachary C Lipton. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[35] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[36] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer, 2020.

[37] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[38] William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[39] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020. Association for Computational Linguistics.

[40] Stanisław Giziński, Michał Kuźba, and Przemyslaw Biecek. Meta-analysis of academic discourse about interpretability, transparency, and fairness. 10.13140/RG.2.2.29514.80322, 2021.

[41] Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, New York, 2021.

[42] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[43] Leonida Gianfagna and Antonio Di Cecco. *Explainable AI with Python*. Springer, 2021.

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

[45] Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*, 2020.

[46] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[47] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[48] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[49] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. Explainable AI in Industry. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 3203–3204, New York, NY, USA, 2019. Association for Computing Machinery.

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

[51] ProPublica. Compas recidivism dataset. `https://github.com/propublica/compas-analysis`, 2017.

[52] Stanisław Gizinski, Michał Kuzba, Bartosz Pielinski, Julian Sienkiewicz, Stanisław Łaniewski, and Przemysław Biecek. MAIR: Framework for mining relationships between research articles, strategies, and regulations in the field of explainable artificial intelligence. *arXiv preprint arXiv:2108.06216*, 2021.

[53] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

[54] Szymon Maksymiuk, Alicja Gosiewska, and Przemyslaw Biecek. Landscape of R packages for eXplainable Artificial Intelligence. *arXiv preprint arXiv:2009.13248*, 2020.

[55] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[56] Saikat Das, Namita Agarwal, Deepak Venugopal, Frederick T Sheldon, and Sajjan Shiva. Taxonomy and Survey of Interpretable Machine Learning Method. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 670–677. IEEE, 2020.

[57] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2021.

[58] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

[59] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[60] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[61] Rishabh Agarwal, Nicholas Frosst, Xuezhou Zhang, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*, 2020.

[62] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, 2016.

[63] Alicja Gosiewska and Przemyslaw Biecek. Do not trust additive explanations. *arXiv preprint arXiv:1903.11420*, 2019.

[64] Michal Kuzba, Ewa Baranowska, and Przemyslaw Biecek. pyCeterisParibus: explaining Machine Learning models with Ceteris Paribus Profiles in Python. *Journal of Open Source Software*, 4(37):1389, 2019.

[65] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[66] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[67] Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

[68] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):1–11, 2008.

[69] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[70] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

[71] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017.

[72] Aleksandra Paluszynska, Przemyslaw Biecek, and Yue Jiang. 'randomForestExplainer': Explaining and visualizing random forests in terms of variable importance. *CRAN*, 2017.

[73] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. exBERT: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*, 2019.

[74] Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. Alibi explain: Algorithms for explaining machine learning models. *Journal of Machine Learning Research*, 22(181):1–7, 2021.

[75] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. iNNvestigate Neural Networks! *Journal of Machine Learning Research*, 20(93):1–8, 2019.

[76] Oracle. Skater. `https://github.com/oracle/Skater`, 2018.

[77] Jacob Gildenblat and contributors. PyTorch library for CAM methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

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

[79] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. iml: An R package for interpretable machine learning. *Journal of Open Source Software*, 3(26):786, 2018.

[80] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[81] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, 2020.

[82] Jakub Wiśniewski and Przemysław Biecek. fairmodels: A Flexible Tool For Bias Detection, Visualization, And Mitigation. *arXiv preprint arXiv:2104.00507*, 2021.

[83] Nishanth Arun, Nathan Gaw, Praveer Singh, Ken Chang, Mehak Aggarwal, Bryan Chen, Katharina Hoebel, Sharut Gupta, Jay Patel, Mishka Gidwani, et al. Assessing the (un) trustworthiness of saliency maps for localizing abnormalities in medical imaging. *Radiology: Artificial Intelligence*, page e200267, 2021.

[84] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[85] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. *arXiv preprint arXiv:1904.12991*, 2019.

[86] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Quantifying Interpretability of Arbitrary Machine Learning Models Through Functional Decomposition. *Ulmer Informatik-Berichte*, page 41, 2019.

[87] Laura Rieger and Lars Kai Hansen. A simple defense against adversarial attacks on heatmap explanations. *arXiv preprint arXiv:2007.06381*, 2020.

[88] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 180–186, New York, NY, USA, 2020. Association for Computing Machinery.

[89] Hubert Baniecki and Wojciech Marek Kretowicz. *Adversarial attacks on Explainable AI methods.* Bachelor's thesis, Zakład Projektowania Systemów CAD/CAM i Komputerowego Wspomagania Medycyny, 2021.

[90] MAIF. Shapash. `https://github.com/MAIF/shapash`, 2021.

[91] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *arXiv preprint arXiv:1812.01193*, 2018.

[92] Samantha Krening, Brent Harrison, Karen M Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. Learning from explanations using sentiment and advice in RL. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, 2016.

[93] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018.

[94] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. *arXiv preprint arXiv:1806.09809*, 2018.

[95] Pzemyslaw Biecek, Hubert Baniecki, Adam Izdebski, and K Pekala. ingredients: Effects and Importances of Model Ingredients. *CRAN*, 2019. `https://cran.r-project.org/package=ingredients`.

[96] Denis J Hilton. A conversational model of causal explanation. *European review of social psychology*, 2(1):51–81, 1991.

[97] Douglas Walton. Dialogical Models of Explanation. *Explanation-aware Computing (ExaCt)*, 2007:1–9, 2007.

[98] Douglas Walton. A dialogue system specification for explanation. *Synthese*, 182(3):349–374, 2011.

[99] Sophie F Jentzsch, Sviatlana Höhn, and Nico Hochgeschwender. Conversational interfaces for explainable AI: a human-centred approach. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 77–92. Springer, 2019.

[100] Navid Nobani, Fabio Mercorio, and Mario Mezzanzanica. Towards an Explainer-agnostic Conversational XAI. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4909–4910. International Joint Conferences on Artificial Intelligence Organization, 2021.

[101] Richard Dazeley, Peter Vamplew, Cameron Foale, Charlotte Young, Sunil Aryal, and Francisco Cruz. Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*, 299:103525, 2021.

[102] Mareike Hartmann, Ivana Kruijff-Korbayová, and Daniel Sonntag. Interaction with Explanations in the XAINES Project. `https://dataninja.nrw/wp-content/uploads/2021/09/7_Hartmann_XAINES_Abstract.pdf`. Accessed: 2021-11-20, 2021.

[103] Christian Werner. Explainable AI through Rule-based Interactive Conversation. In *EDBT/ICDT Workshops*, 2020.

[104] Kacper Sokol and Peter Flach. 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*, pages 5785–5786. International Joint Conferences on Artificial Intelligence Organization, 2018.

[105] Kacper Sokol and Peter Flach. 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*, pages 5868–5870. International Joint Conferences on Artificial Intelligence Organization, 2018.

[106] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. A model of social explanations for a conversational movie recommendation system. In *Proceedings of the 7th International Conference on Human-Agent Interaction*, pages 135–143, 2019.

[107] Mingkun Gao, Xiaotong Liu, Anbang Xu, and Rama Akkiraju. Chat-XAI: A New Chatbot to Explain Artificial Intelligence. In *Proceedings of SAI Intelligent Systems Conference*, pages 125–134. Springer, 2021.

[108] Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. A Survey on Conversational Agents/Chatbots Classification and Design Techniques. In Leonard Barolli, Makoto Takizawa, Fatos Xhafa, and Tomoya Enokido, editors, *Web, Artificial Intelligence and Network Applications*, pages 946–956, Cham, 2019. Springer International Publishing.

[109] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, USA, 1st edition, 2000.

[110] Matthew B. Hoy. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1):81–88, 2018. PMID: 29327988.

[111] Joseph Weizenbaum. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM*, 9(1):36–45, 1966.

[112] Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. Artificial Paranoia. *Artificial Intelligence*, 2(1):1–25, 1971.

[113] Kenneth Mark Colby, Franklin Dennis Hilf, Sylvia Weber, and Helena C Kraemer. Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. *Artificial Intelligence*, 3:199–221, 1972.

[114] Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. GUS, a frame-driven dialog system. *Artificial Intelligence*, 8(2):155–173, 1977.

[115] Satoshi Akasaki and Nobuhiro Kaji. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. *arXiv preprint arXiv:1705.00746*, 2017.

[116] John D. Gould and Clayton Lewis. Designing for Usability: Key Principles and What Designers Think. *Commun. ACM*, 28(3):300–311, 1985.

[117] Norman M. Fraser and G.Nigel Gilbert. Simulating speech systems. *Computer Speech & Language*, 5(1):81–99, 1991.

[118] N. Dahlbäck, A. Jönsson, and L. Ahrenberg. Wizard of Oz studies — why and how. *Knowledge-Based Systems*, 6(4):258–266, 1993. Special Issue: Intelligent User Interfaces.

[119] C. Limaheluw. The role of buttons in the conversational interface of buttons: An experiment about the influence of buttons on the customer experience, brand attitude and brand trust by using chatbots. `http://essay.utwente.nl/80511/`, 2020. Accessed: 2021-10-20.

[120] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*, 2017.

[121] Botpress. Botpress. `https://github.com/botpress/botpress`, 2021.

[122] Google. Dialogflow documentation. `https://cloud.google.com/dialogflow/docs`. Accessed: 2021-10-07.

[123] Microsoft. Microsoft Bot Framework documentation. `https://aka.ms/bot-framework-www-portal-docs`. Accessed: 2021-10-07.

[124] Amazon. Amazon Lex. `https://aws.amazon.com/lex/`. Accessed: 2021-11-17.

[125] IBM. Watson Assistant Service. `https://www.ibm.com/products/watson-assistant`. Accessed: 2021-11-17.

[126] Paul Aschmann. Rasa UI. `https://github.com/paschmann/rasa-ui`, 2019.

[127] Mikhail Ushakov. Dialogflow for Web v2. `https://github.com/mishushakov/dialogflow-web-v2`, 2021.

[128] Google. Google Cloud documentation. `https://cloud.google.com/docs`. Accessed: 2021-10-20.

[129] Michał Kuźba and Przemysław Biecek. What Would You Ask the Machine Learning Model? Identification of User Needs for Model Explanations Based on Human-Model Conversations. In Irena Koprinska, Michael Kamp, Annalisa Appice, Corrado Loglisci, Luiza Antonie, Albrecht Zimmermann, Riccardo Guidotti, Özlem Özgöbek, Rita P. Ribeiro, Ricard Gavaldà, João Gama, Linara Adilova, Yamuna Krishnamurthy, Pedro M. Ferreira, Donato Malerba, Ibéria Medeiros, Michelangelo Ceci, Giuseppe Manco, Elio Masciari, Zbigniew W. Ras, Peter Christen, Eirini Ntoutsi, Erich Schubert, Arthur Zimek, Anna Monreale, Przemyslaw Biecek, Salvatore Rinzivillo, Benjamin Kille, Andreas Lommatzsch, and Jon Atle Gulla, editors, *ECML PKDD 2020 Workshops*, pages 447–459, Cham, 2020. Springer International Publishing.

[130] Google. Cloud Functions documentation. https://cloud.google.com/functions/docs. Accessed: 2021-10-20.

[131] Michał Kuźba and Przemysław Biecek. Xai-bot Titanic. https://github.com/ModelOriented/xaibot, 2020.

[132] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, 2019.

[133] Trestle Technology, LLC. plumber: An API Generator for R. *CRAN*, 2018. https://cran.r-project.org/package=plumber.

[134] Adam Rydelek. *xai2cloud: Deploys An Explainer To The Cloud*, 2020. https://modeloriented.github.io/xai2cloud.

[135] Google. Google Stackdriver documentation. https://cloud.google.com/stackdriver/docs. Accessed: 2021-10-20.

[136] Q Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the AI: informing design practices for explainable AI user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2020.

[137] Shipi Dhanorkar, Christine T Wolf, Kun Qian, Anbang Xu, Lucian Popa, and Yunyao Li. Who needs to know what, when?: Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle. In *Designing Interactive Systems Conference 2021*, pages 1591–1602, 2021.

[138] Kaggle. Titanic dataset. https://www.kaggle.com/c/titanic/data. Accessed: 2021-10-15.

[139] Encyclopedia Titanica. https://www.encyclopedia-titanica.org/. Accessed: 2021-10-15.

[140] Heping Zhang and Minghui Wang. Search for the smallest random forest. *Statistics and its Interface*, 2(3):381, 2009.

[141] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[142] Przemyslaw Biecek and Marcin Kosinski. archivist: An R package for managing, recording and restoring data analysis results. *Journal of Statistical Software*, 82(11):1–28, 2017.

[143] Michał Kuźba and Przemyslaw Biecek. What would you ask your ML model? Explainable AI chatbot. https://dx.doi.org/10.13140/RG.2.2.29719.32166, 2019.