A Multi-layered Approach for Interactive Black-box Explanations

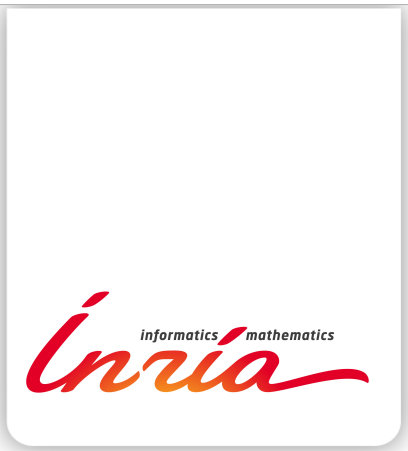
Clement Henin, Daniel Le Métayer

A Multi-layered Approach for Interactive Black-box Explanations

Clément Henin, Daniel Le Métayer

RESEARCH REPORT
N° 9331
March 2020
Project-Team PRIVATICS

ISSN 0249-6399 ISRN INRIA/RR--9331--FR+ENG



A Multi-layered Approach for Interactive Black-box Explanations

Clément Henin, Daniel Le Métayer

Project-Team PRIVATICS

Research Report n° 9331 — March 2020 — 34 pages

Abstract: In order to provide interactive explanations, a system must be generic enough to be able to address a wide range of questions from explainees with different levels of expertise. In this paper, we present a multi-layered approach allowing explainees to express their needs at different levels of abstraction. We describe a proof-of-concept system called IBEX (for "Interactive Black-box Explanations") implementing this approach and show its application to a variety of case studies.

Key-words: Algorithmic decision system, explainability, interactive explanation, transparency, black-box model, machine-learning, artificial intelligence

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Une approche multi-couche pour des explications interactives de boîte noire

Résumé : Pour fournir des explications interactives, le système explicatif doit être suffisemment générique pour répondre à une grande variété de questions des desinataires des explications ayant différents niveaux d'expertise. Dans cet article, nous présentons une approche multi-couche permettant aux destinataires des explications d'exprimer leurs besoins à différents niveaux d'abstraction. Nous décrivons aussi une preuve de concept, appelée IBEX ("Interactive Black-box EXplanations"), qui repose sur cette approche, et nous montrons quelques cas d'usage.

Mots-clés : Système de décision automatique, explicabilité, explications interactives, transparence, modèle boîte noire, apprentissage machine, intelligence artificielle

1 Introduction

Algorithmic Decision Systems (hereafter "ADS") are increasingly used in many areas, sometimes with a major impact on the lives of the people affected by the decisions. Some of these systems make automatic decisions, for example to reduce or to increase the speed of an autonomous car, while others only make suggestions that a human user is free to follow or to dismiss. In some cases, the user is a professional, for example a medical practitioner or a judge, while in other cases he is an individual, for example an internet user or a consumer. Some ADS rely on traditional algorithms, while others are based on machine learning (hereafter "ML") and may involve complex models such as neural networks, support vector machines or random forests. Regardless of these considerations, when an ADS can have a significant impact on people, its design and validation should ensure a high level of confidence that it complies with its requirements.

Explainability has generated increased interest during the last decade because accurate ML techniques often lead to opaque ADS and opacity is a major source of mistrust. Indeed, even if they are not a panacea, well designed explanations can play a key role, not only to enhance trust in a system, but also to allow its users to better understand its outputs and therefore to make a better use of them. In addition, they are necessary to make it possible to challenge the decisions resulting from an ADS. On the legal side, Recital 71 of the European General Data Protection Regulation, which concerns decisions "based solely on automated processing", states that "in any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision."

Explainability methods produce different types of explanations in different ways, based on different assumptions on the system [1]. In this paper, we focus on a category of methods, called "black-box explanation methods", that do not make any assumption on the availability of the code of the ADS or its underlying model. The only assumption is that input data can be provided to the ADS and its outputs can be observed.

In practice, explanations can take different forms, they can target different types of users (hereafter "explainees") with different goals. One of the main challenges in this area is therefore to devise explanation methods that can accommodate this variety of situations. This is especially crucial to avoid the "inmates running the asylum" phenomenon [2] and be able to design a system that can be used by lay persons. The need to conceive an explanation as an interactive process rather than a static product has been argued in a very compelling way by several authors [2, 3, 4, 5, 6]. It must be acknowledged, however, that most contributions in the XAI field still focus on static explanations. A promising development in this direction consists in implementing a toolkit including a variety of methods that can be selected by the users depending on their needs. For example, AIX360 ("AI Explainability 360") [7] contains eight explainability algorithms and allows users to choose among them based on a taxonomy including criteria such as "understand the data or the model" or "self-explaining model or post-hoc explanations". In the same spirit, the What-If tool includes visualization components and facilities to generate counterfactual examples and partial dependence plots. Our objective in this paper is to go one step further in this direction to better fit the needs of lay users through :

- A fine grain decomposition of the context of the explanation process (profile of the user, objective, etc.).
- The possibility for explainees to express their needs at different levels of abstraction depending on their level of expertise.
- A mapping between of the different levels to generate the most suitable explanations.

- A fine grain decomposition of the core components of an explanation system and their parameters.
- The possibility for explainees to react to an explanation (e.g. to request more details, or a simpler explanation, or an explanation in a different form).

In this paper, we present our layered approach in detail, describe a proof-of-concept system called IBEX (for "Interactive Black-box Explanations") implementing it and show its application to a variety of case studies.

The core of our approach is the observation that, beyond their diversity, black-box explanation methods share a number of features. We propose a generic architecture for black-box explanation methods and focus on two of its core components, called respectively *Sampling* and *Generation*, which can be implemented and composed in different ways. This architecture is generic in the sense that many black-box explanation methods (including existing methods [8]) can be characterized by specific instantiations of components and choices of parameters of this architecture. The benefit of working at this fine grain level is that the core components can be composed and parameterized in many other ways than in existing "on the shelves" methods. The wealth of this combinatory is necessary to address the variety of explainees' needs. However, the challenge is to be able to map these needs, which can be expressed at a high level of abstraction, especially by lay users, to technical options (components and parameters) best suited to address them.

To address the above issues, we propose three levels of abstraction, called respectively the *context*, the *requirements* and the *technical options*, to express explainees' needs:

1. The *context* provides high-level information about the profile of the explainee and his objectives.
2. The *requirements* characterize the desired explanations, including their format, degree of simplicity and generality.
3. The *technical options* define precisely the explanation production process; they include in particular the scope and type of sampling to be used and potential constraints on the generation phase.

Broadly speaking, these levels could be summarized as: "What is the question?", "What kind of explanation is needed?" and "How to compute it?". Lay users should be able to express their needs at the highest level of abstraction, without any knowledge of the requirements and technical options. On the other hand, expert users, for example the designers of the ADS, may prefer to express their explanation requests directly as requirements or in technical terms. Regardless of the level of abstraction adopted by the user to express his needs, ultimately these needs have to be translated into technical options. In this paper, we describe a heuristic method to derive (1) requirements from contexts and (2) technical options from requirements. As an illustration of the mapping between contexts and requirements, a lay user whose goal is to challenge a decision will be associated with a low level of generality (focus on his specific case) and preferably simple explanations in the form of realistic counterexamples. On the other hand, an expert user whose goal is to improve the ADS will, by default, be suggested general rule based explanations.

It should be clear that the role of these mappings is to make it easier for users to express their needs and to obtain appropriate explanations. The generation process cannot, by essence, always meet the user's needs at the outset. If the explanation generated by the system does not meet their expectations, users can either impose a different choice (e.g. a lay user may request an explanation in rule based format) or interact with the system after the generation of a first explanation (e.g. to express that they want a simpler or more general explanation).

We first present the two higher levels of abstraction (the context and the requirements) in Section 2 before describing the lower level (the technical options) in Section 3. The technical options rely on a generic explainer architecture and two parameterized explanation components, namely *Sampling* and *Generation*. In Section 4, we define the mappings between the different levels and show the derivation of, respectively, requirements from contexts, and technical options from requirements. In Section 5 we illustrate the approach with the application of our proof-of-concept system IBEX to several case studies. These case studies involve different types of users and show the benefit of the approach in terms of versatility and interactivity. Section 6 discusses related work and Section 7 concludes with prospects for future work. The interested reader can also find in A the presentation of some implementation choices made in the implementation of IBEX and in B a summary of the notations used in this paper.

2 Context and requirements

In this section, we present successively the higher levels of abstraction of our framework: the context (Section 2.1) and the requirements (Section 2.2). The mapping between these levels is described in Section 4.

2.1 Context

The context is the highest level of abstraction, which should be accessible to any explainee, including lay users, to express their needs in a simple, non technical, way. Contexts are made of the *ADS* to be explained and four elements related to the explainee's query: *Profile*, *Objective*, *Focus* and *Point of interest*. Each of them is presented in turn below.

- *ADS* includes the decision algorithm and, if available, the associated learning or historical usage data set.
- *Profile* characterizes the profile of the explainee. It takes a value in the set $\{TE, AU, DE, LU\}$. TE represents technical experts, AU auditors, DE domain experts and LU lay users. Technical experts include designers, developers, testers, i.e. people having some knowledge about the design or the techniques used to implement the ADS. Auditors are also assumed to have a high level of expertise but they are involved in a specific task of auditing or evaluating the ADS. Domain experts are not assumed to have any expertise about the ADS itself or the technology used but they are knowledgeable about the application domain. Examples of domain experts include medical doctors, judges or police officers. The last category, lay users, includes users who are not assumed to possess any specific knowledge. They may be persons affected by decisions relying on the ADS or simple citizens.[1]
- *Objective* characterizes the objective of the explainee. It takes a value in the set $\{I, T, C, A\}$. I represents the improvement of the ADS, T trust enhancement, C challenging a decision and A taking actions based on a decision. The improvement of the ADS includes its testing, assessment of its accuracy and any action to detect potential weaknesses. Trust enhancement includes a variety of objectives related to the use of the ADS (avoiding wrong decisions [1], enhancing the acceptance of the results [1], increasing the predictability of the output [11] and being comfortable with the strengths and limitations of the ADS [12]) or its the purpose (causality, transferability [13, 10]). Challenging a decision and taking

[1] Other explainees' profiles taxonomies were already proposed in previous works, notably in [9] and [10]. Our contribution is consistent with them, but involves some simplifications, justified by pragmatic needs.

an action based on a decision are two alternative reactions for the person affected by a decision [14]. Actions that can be taken based on a decision include actions that can have an impact on the person's record (his input data) and therefore on future decisions.

- *Focus* characterizes the scope of the explanation. It takes a value in the set $\{G, L\}$. G stands for global explanation and L for local explanation. An explanation is global if the explainee is interested in the behaviour of the ADS for the whole input dataset. Otherwise, it is local, which means that the explainee is interested in the behaviour of the ADS for (or around) a specific input value.
- *Point of interest* defines the input value x which is the point of interest of the explainee when the focus of the explanation is local (otherwise, the context does not contain any point of interest).

We should emphasize that some of these elements can be omitted by explainees if they are not sure about them. The only mandatory element is the *ADS*. Explanations can be generated from partially defined contexts. The drawback is that such explanations may not correspond to the expectations of the explainee who may then have to refine his needs through further interaction steps.

2.2 Requirements

The requirements provide an intermediate level of abstraction. They characterize the desired explanations more precisely than the context but still in an abstract way. They can be useful to certain lay users, depending on their level of understanding, and to more expert users. The requirements are made of seven elements[2]: *Format*, *Simplicity*, *Generality*, *Point of interest*, *Realism*, *Actionability* and *Nature*. Apart from *Realism*, which is, to the best of our knowledge, an original contribution, these elements are motivated by previous work and experimental studies, as mentioned below.

- *Format* includes the different forms of explanations that can be generated [1, 13]. The impact of the format on the acceptance of explanations is analyzed in [15, 16]. Examples of formats include rule based explanations (RB), local linear approximation (LA), counterfactual explanations (CF), decision trees (DT), pearson correlation coefficients (PC) and partial dependence plots (PD). The full list of formats implemented in the current version of IBEX is provided in Section 3.4.
- *Simplicity* characterizes the level of simplicity required for an explanation. It is a key requirement as it generally relates to understandability [1, 17, 18]. The current version of IBEX considers three increasing levels of simplicity: Simplicity = $\{1, 2, 3\}$.
- *Generality* characterizes the level of generality required for local explanations, i.e. the size of the class of input values that should be covered by the explanation ([17] p.44). Some authors use the word "cover" to denote the same concept [18, 19]. The current version of IBEX considers three increasing levels of generality: Generality = $\{1, 2, 3\}$. Level 1 covers a single input (the point of interest), level 3 a wide class of inputs and level 2 is intermediate. Note that generality is defined only for local explanations since global explanations cover, by definition, the whole input dataset.
- *Point of interest* has the same definition as above (for contexts). Like generality, the point of interest is defined only for local explanations.

[2] In addition to the ADS, as defined in the context.

- *Realism* characterizes the level of realism required for an explanation. By "realism", we mean the fact that the explanation production process takes into account the actual distribution of the input data. Realistic explanations are preferable for explainees interested in the actual usage of the ADS. On the other hand, explainee interested in the internal logic of the ADS, independently of its actual usage, may proceed without the constraint of realism. Let us consider this notion with the example of a credit scoring system. The ADS systematically outputs the maximum risk when the application file mentions a previous credit fraud. Although this feature has a tremendous impact on the score, it is rarely used in practice, as few credit applicants are in this situation. The realistic approach takes into account the probability of hitting this feature during the computation of the explanation, and thus attributes a rather low importance to this feature, while the non-realistic approach only considers the model itself, and thus attributes a high importance to this feature. The current version of IBEX considers three increasing levels of realism: Realism $= \{1, 2, 3\}$.

- *Actionability* expresses the fact that actionable explanations should be prefered. An actionable explanation is an explanation involving only actionable features of the input dataset ([14] p.42). For example, in the input file of a loan applicant, the age variable is not actionable whereas the number of outstanding loans is actionable. The current version of IBEX considers two options: Actionability $= \{T, F\}$. Value T means that actionability is a requirement. In this case, the explainee has to provide the list of actionable features.

- *Nature* corresponds to the presence or absence of probability in the explanations ([17] p.44). The current version of IBEX considers two options: Nature $= \{T, F\}$. Value F means that probabilistic explanations are not desired and value T that they are acceptable.

Like contexts, requirements can be partially defined. In addition, they may be expressed in terms of preferences rather than fixed choices. For example, a technical expert may characterize simplicity by $3 > 2 > 1$ to express the fact that he prefers simple explanations but can also cope with intermediate or complex explanations. On the other hand, lay users may prefer to characterize simplicity by selecting only value 1. In the following, the former are called soft requirements and the latter hard requirements. In addition, soft requirements may also be prioritized (ranked by order of importance). For example, a technical expert who wants to debug or improve the ADS may consider that generality is more important than simplicity ($general > simple$).

3 Technical options of the generic explainer

The context and requirements introduced in the previous section allow explainees to express their needs in a rather abstract way. In this section, we present the lower, operational level, which defines the actual process to produce explanations meeting these needs. In the following, we call *explainer*, a system producing explanations. In order to define the elements of the technical options, as was done above for contexts and requirements, we first introduce our generic explainer architecture in Section 3.1. Then we describe the two main elements of this architecture, the Sampling and the Generation components, with their parameters, in Section 3.2 and Section 3.3 respectively. These elements allow us to define the technical options in Section 3.4. At this stage, we do not consider the mapping of contexts and requirements on technical options, which is the subject of Section 4.

3.1 Generic explainer architecture

A great variety of needs can be expressed at the context and requirements levels described in the previous section. The first condition to be able to produce a range of explanations meeting all these needs is to be able to express implementation options also at a very fine grain. It should then be possible to combine these options in different ways to address different explainees' needs. To this aim, we introduce in this section a generic parameterized explainer architecture. The architecture is generic in the sense that many black-box explanation methods (including existing methods [8]) correspond to specific choices for its components and parameters. Furthermore, these components can be composed and parameterized in many other ways than in the implementations of existing "on the shelves" methods. The wealth of this combinatory is critical to match the variety of explainees' needs.

To introduce our explainer architecture, let us consider the simple example of a spam classifier. This ADS takes as input the text of an email and outputs the probability of this email being a spam. Since we assume that the code of the ADS is not available, the method can only build emails, submit them to the ADS and analyze the results. For example, to assess the role of the signature part in the classification of a specific email, the explainer can create different versions with and without the signature part, or with different pieces of text in the signature part. The explainer has then to compute the answer based on the results of the ADS and to present it to the explainee.

This simple example highlights the two main components of an explainer architecture: (i) the selection of inputs to submit to the ADS to be explained, which is called the *Sampling* component; and (ii) the analysis of the links between the selected inputs and the corresponding outputs of the ADS to generate the content of the explanation, which is called the *Generation* component. If the input data are not meaningful for humans, as the pixels of an image for example, a preliminary component is required to extract an interpretable representation, as done in LIME [20]. Because the representation step is not essential to the description of the technical options, we postpone its discussion to A and focus now on the two other components. We propose formal characterizations of the sampling and generation components which are generic enough to encompass existing black-box explanation methods[3] and to serve as a basis for the production of explanations meeting user's needs expressed as contexts or requirements, as shown in 3.4. The main notations used in this section and the followings are sketched in Table 1.

Name	Description	Example
F	Black-box model	The spam classifier
X	Input space of F	Set of of all possible emails
Y	Output space of F	$[0, 1]$
E	Scope of the explanation	Email x_e
S	Samples (product of the sampling step)	Emails with modified signature
Θ	Parameters of the sampling	Part of the email
D	Dataset describing the overall population	Training set of F

Table 1: Main notations for the generic explainer

[3]The interested reader can find in [8] an analysis of existing black-box explanation methods and their expression in terms of the components and parameters of our generic architecture.

3.2 Sampling

The role of the *Sampling* component is to select appropriate inputs (or "samples") to answer a question about a model F. The choice of the samples may depend on a number of factors. The first aspect to take into consideration is whether the question concerns the whole model or specific inputs. We call E the scope[4] of the explanation. If the question concerns a single input x_e, then $E = \{x_e\}$; if the question is about the whole model F, then $E = D$ with D a multiset[5] representation of the population (possible inputs to F) available to the explainer. In general, E and D could be any (multi)subset of possible input values. We call X the set of input values, which can be seen as the support set (or type set) of multiset D. In the spam filter example, X is the set of all possible emails (i.e. the set of all texts of a given format) and D represents the actual data set of emails available to the explainer, which can be used, for example, to estimate the distributions of the features. Typical examples of D would be the training or testing sets used during the learning process, or simply historical data accumulated during the use of the model. When the explainer does not have any information about this distribution, D is the empty set ($D = \emptyset$).

The result of the *Sampling* component is a set of samples $S = \{x_1, ..., x_n\} \in X^n$. For example, to address the first question about the impact of the signature on the classification of x_e, a possible option is to select a single sample obtained by removing the signature part of x_e. This strategy does not require any information about the actual distribution of the population and can therefore be applied even if $D = \emptyset$. However, the answer may not be realistic or precise enough. A more elaborate strategy would be to replace the original signature of x_e by real signatures obtained from many other emails. This strategy requires information about the actual distribution of the population ($D \neq \emptyset$) in order to ensure that the sample set reflects the reality. We can now define the sampling procedure as follows [6]:

$$S = \{h_\theta(x_e, x_p) \mid (\theta, x_e, x_p) \in \Theta \times E \times D, Z(\theta, x_e, x_p)\} \tag{1}$$

with

$$h_\theta : E \times D \to X \tag{2}$$

Θ is the set of parameters for the sampling, Z is a filter function and h_θ defines how samples are generated. In a nutshell, the θ parameter makes it possible to generate several samples for each pair (x_e, x_p) while Z restricts the generation of samples to a selection of pairs (x_e, x_p). In our spam filter example, E is limited to a single email to be explained ($E = \{x_e\}$). The email is represented by the content of its different parts (header, body, signature, ...) and $h_\theta(x_e, x_p)$ is a version of x_e that is obtained by replacing a part of x_e by the corresponding part of x_p. The part that is replaced is specified by θ. For instance, taking $\Theta = \{(SIG)\}$ and assuming that D contains 1000 emails, the sampling procedure generates 1000 perturbed versions of x_e with signatures (corresponding to $\theta = (SIG)$) extracted from the emails in D. The role of the θ parameter is therefore to customize the sampling function. For instance, if both the header and the signature of the email are taken into consideration, θ could specify which part of the email is replaced (header, signature or both). With $\Theta = \{(HDR), (SIG), (HDR, SIG)\}$, the sampling procedure would generate 3000 versions of x_e with header, signature or both replaced by the corresponding parts of other emails in D. Another possibility provided by Definition (1) is to use a filter function Z, for example selecting only emails from the same sender as x_e, or relying on

[4] The scope is related to the focus element of the context. The precise correspondence is discussed in Section 4.

[5] D is a multiset because it can contain multiple occurrences of the same value to reflect the distribution of the values in the real population.

[6] If Θ, D or E are empty, they are set to $\{0\}$ in (1), otherwise the product space would also be empty.

a notion of distance to select only emails close to x_e. To make the presentation more concrete, let us present three examples of sampling strategies which are instances of Definition (1). These strategies are available, among others, in the proof-of-concept system IBEX illustrated in Section 5. We focus on local explanations here since global explanations rely on the whole population set. In the first example, called "Select Closest" (SC), Z is used to select from the population set D inputs that are close to x_e by comparing the distance $d(x_e, x_p)$ to a predefined threshold r. In this case, h^{close} simply returns the unmodified input from the population set (cf Figure 1b).

$$S = \{h^{close}_{\theta}(x_e, x_p) = x_p \mid (\theta, x_e, x_p) \in \{r\} \times E \times D, d(x_e, x_p) < \theta\} \tag{3}$$

Since h^{close} returns samples from the population set D, it may be suitable to generate realistic explanations. The number of samples and their closeness to x_e can be tuned using r.

Another strategy, called "Permutation" (P) swaps features among samples to account for the underlying distribution of X. The following sampling function:

$$h^{perm}_{\theta}(x_e, x_p) = (x_e[i] \text{ if } i \in \theta \text{ else } x_p[i]), \text{ with } x_e \in E, x_p \in D \tag{4}$$

combines the features of x_e with the features of x_p ($x[i]$ denotes the i^{th} feature of x) and the parameter θ defines the origin of a feature : the scope or an input of the population set (cf Figure 1c). θ is drawn randomly in such a way that each feature comes from x_p with probability p, which is a parameter of the sampling. The computation of Shapley values in [21] or the generation of local rule based models in Anchors [19] are based on similar sampling strategies. Each feature is independently drawn from the empirical distribution of X and only features included in the same θ are correlated. "Permutation" sampling is an intermediate level of realism.

Finally, "Add random Noise" (AN), generates samples by adding a certain amount of noise to x_e. Samples are then noisy versions of x_e, with the noise drawn from a normal distribution with 0 mean (cf. Figure 1d).

$$h^{noise}_{\theta}(x_e, x_p) = x_e + \theta, \text{ with } \theta \sim \mathcal{N}(0, \sigma^2) \tag{5}$$

The distribution of samples obtained with AN does not use the information of the population set ($D = \emptyset$), and features are independent. Variable σ represents the standard deviation of the added noise: small values of σ generate samples that are close to x_e while bigger values generate samples in a wider space (as depicted with the two circles in Figure 1d). "Add random noise"provides non-realistic samples.

3.3 Generation

The set S of samples and the model F are the inputs of the explanation generation process. Even if explanations can take many different forms, the generation process can be broadly defined as the computation of a *proxy* of the model F followed by the construction of an explanation based on this proxy. In some cases, the proxy model is considered as the explanation itself, and the second phase is therefore just the identity function.

Coming back to the spam classifier example, an option for the *Generation* component is to train a rule-based model on the samples to predict the output of the classifier. An example of rule generated this way could be: "If the signature of the email is less than 60 characters long, then the classifier will consider that it is a spam; otherwise it will consider that it as an acceptable email". Because such rules are easily interpretable, they can be used directly as explanations. In other situations, either because the type of model used is too complex or the model is too big to be understandable (for example if it involves a large number of rules), simpler explanations have to be generated from the proxy model. This phase can return, for example, the most important

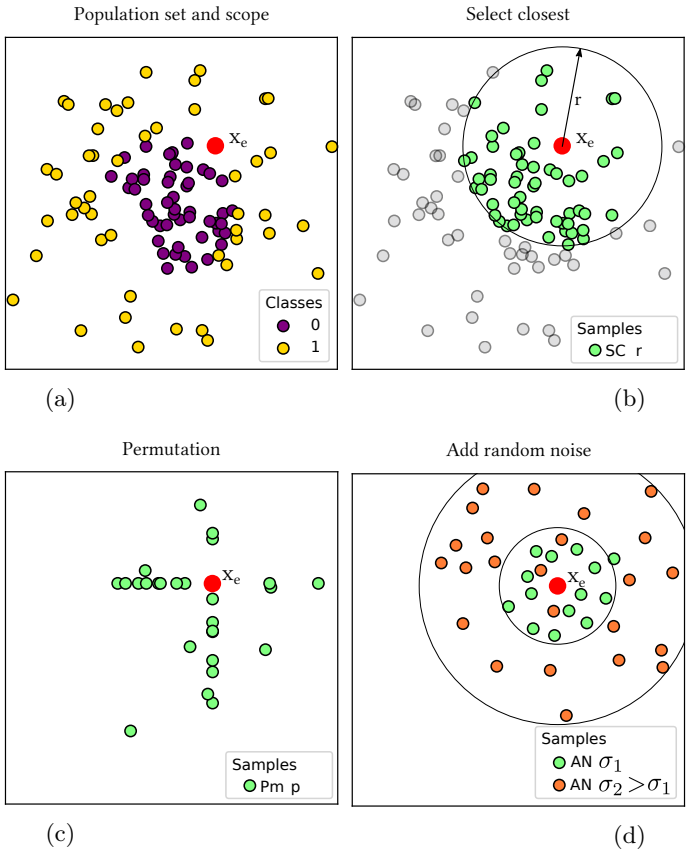


Figure 1: Schematic view of local sampling processes with two dimensional continuous variables. The point of interest of the explanation (or scope) is the red circle x_e. (a) Population set and point of interest for a binary classification problem, classes are depicted with different colors. (b) "Select closest" sampling with threshold r. Samples are inputs of the population set within a circle of size r centered at the point of interest. (c) "Permutation" sampling with probability p. Samples are altered versions of the point of interest with one or two features drawn from the empirical distribution. (d) "Add random Noise" sampling with σ_1 and $\sigma_2 > \sigma_1$. Samples are noisy versions of the point of interest.

feature(s) of the input. For the spam classifier, the generated explanation could then be: “The length of the signature part and the number of typos are the two most important features used by the ADS to decide if an email is a spam”.

Technically speaking, the proxy model is denoted by f_w (the rule-based model in the example), which is a function of the same type as the model F, parameterized by w:

$$f_w : X \to Y \tag{6}$$

The core of the *Generation* component is to find the best proxy f_w to answer the question of the explainee, which amounts to find the optimal values of w. Optimality can be defined formally using constraints $o_i(w, S) \in \mathbb{B}$ and criteria $c_i(w, S) \in \mathbb{R}$ where $\mathbb{R}$ and $\mathbb{B}$ are the sets of real

numbers and booleans respectively. The global objective takes the following form:

$$\begin{aligned} w^* = \underset{w}{\text{argmin}} \quad & \sum_i \lambda_i c_i(w, S) \\ \text{subject to} \quad & o_i(w, S) \end{aligned} \tag{7}$$

where $\lambda_i \in \mathbb{R}$ are used to weight the criteria.

In many methods, the objective is to find the parameters w such that the proxy f_w is as close as possible to F on the elements of S (samples). Indeed, finding a good explanation is often a matter of trade-off. A typical example is finding the right balance between precision and complexity – often used as a rough approximation of understandability. For example, a simple explanation of the spam classifier that would be accurate (i.e. predicting the actual result of the classifier) on only seventy percent of its inputs would not be acceptable; on the other hand, an accurate explanation that would take the form of several pages of rules would provide little insight to the user. Using both criteria and constraints offers flexibility. This distinction is already used in some existing methods. For instance Anchors [19], sets a *constraint* on the precision of the rule-based model and advocates that explanations should be highly precise, while BETA [18] sets the precision of the rule-based model as a criterion and advocates that explanations should first be interpretable.

To make the presentation more concrete, let us consider three examples of generation strategies, which are instances of Definition (7). These strategies are available, among others, in the proof-of-concept system IBEX presented in Section 5. The first example is the "Rule-Based model" (RB) generation f_w with w the set of rules. A possible instance is to use as criterion the number of rules and as constraint the precision of the model, as done in [19], which can be expressed using the following minimization:

$$\begin{aligned} w^* = \underset{w}{\text{argmin}} \quad & ||w|| \\ \text{subject to} \quad & \#\{x \in S, f_w(x) = F(x)\}/\#S > a \end{aligned} \tag{8}$$

with $||w||$ the number of rules, $\#$ denoting the cardinality and a, the minimum accuracy.

Another example is to use a "Local linear Approximation" (LA) of the model, as done in [20]. In this case, f_w can be defined as $f_w(x) = \sum_i w_i x[i]$ with the following minimization:

$$w^* = \underset{w}{\text{argmin}} \quad \lambda||w|| + \sum_{x \in S} (f_w(x) - F(x))^2 \tag{9}$$

which amounts to a classical Lasso regression. The derived coefficients of the Lasso regression provide information about the local behaviour of the ADS. More precisely, by comparing their values, the explainee can estimate what would be the impact of the modification of a variable on the model output. In many cases, it approximates the importance of a feature for a specific output.

Finally, as proposed in [14], the generation step may be used to find a counterfactual example, which can be expressed as follows:

$$\begin{aligned} w^* = \underset{w \in \{x - x_e, x \in S\}}{\text{argmin}} \quad & ||w|| \\ \text{subject to} \quad & f_w(x_e) \neq F(x_e) \end{aligned} \tag{10}$$

with $f_w(x) = F(x + w)$ and $||w||$ denoting the distance between $x + w$ and x. A counterfactual example is the input closest to the point of interest for which the ADS returns an output different

from the output returned for the point of interest. Our formulation of counterfactuals involves the differences between the point of interest and the counterfactual, named w, that should be as small as possible. Equation (10) involves the norm of w, which is the distance between the counterfactual and the point of interest, and a constraint on the output of the ADS for the counterfactual, which should differ from the output of the ADS for the point of interest.

3.4 Set of technical options

In the previous two sections, whe have presented the two main components of the generic explainer architecture, the *Sampling* component and the *Generation* component. These components can be instantiated and parameterized in different ways. These instantiations and parameterization options together make up the technical options available to produce explanations. In this section, we review this set of technical options based on the notions introduced in Section 3.2 and Section 3.3, before showing in Section 4 how they can be derived from explanation needs expressed in terms of contexts and requirements.

The instantiations of the *Sampling* and the *Generation* components currently available in the implementation of IBEX are presented with their parameters in Table 2. Considering that, for local explanations, the two phases (*Sampling* and *Generation*) are independent, there are nine possible combinations of instantiations. For global explanations, three additional options are possible, making a total of twelve options for the instantiation of components. The second part of the technical options, the choice of the parameters, mostly depends on the instantiation of the component, as shown in Table 2. Table 3 provides an overview of the parameters used by the sampling and generation components.

Name	Component	Focus	Parameters	Short description
Add random Noise (AN)	Sampling	Local	σ	Adds Gaussian noise to the point of interest
Permutation (Pm)	Sampling	Local	p	Swaps of values between the scope and population inputs
Select Closest (SC)	Sampling	Local	r	Selects inputs from the population closest to the point of interest
Identity (Id)	Sampling	Global	$\emptyset$	Returns the population set
Replace with Constant (RC)	Sampling	Global	α	Replaces all values of one feature with constant α
Rule-Based model (RB)	Generation	Local	a	Accurate and simple RBM
Local linear Approximation (LA)	Generation	Local	λ	Lasso regression
CounterFactual (CF)	Generation	Local	$\emptyset$	Finds the closest sample to the point of interest leading to a different output
Decision tree (DT)	Generation	Global	a_{DT}	Decision tree (sampling: Id)
Pearson Correlation (PC)	Generation	Global	$\emptyset$	Global linear importance of features (sampling: Id)
Partial Dependence (PD)	Generation	Global	$n^{(i)}$	Computes average output of each features value (sampling: RC)

Table 2: Technical options: components and their parameters. (i) Variable n denotes the number of bins used for continuous variables.

As an illustration of the possibilities of combinations of different instantiations for the sampling and generation components, let us consider the example of counterfactuals (CF). When a counterfactual example is obtained using a realistic sampling strategy, the final explanation looks

like a real email, very similar to the point of interest (with a small number of words modified). In this case, a realistic counterfactual could an altered version of the point of interest with longer words in the signature such that its length exceeds sixty characters. This type of explanation is useful for a domain expert who wants to enhance his trust in the ADS. On the other hand, a counterfactual obtained from a non-realistic sampling does not necessarily look like a real email. For instance, one of the non-realistic sampling strategies that could be used would consist in a random addition of characters to the original email. The additional wording would look like typos for the ADS. This type of counterfactual is more suited for technical experts trying to improve the model as such (for any input data, disregarding their actual "real life" distribution).

Component	Parameter	Short description
Add random noise	σ	Standard deviation of noise
Permutation	p	Probability to change feature value
Select closest	r	Distance to farthest sample
Rule-based model	a	Minimum accuracy
Local linear approximation	λ	Lasso penalization weight
Decision tree	a_{DT}	Minimum accuracy
Partial dependence plot	n	Number of bins

Table 3: List of technical parameters

The choice of the parameters associated with each component further multiplies the number of technical options available. For example, the value of σ (Definition (5)) has an impact on the average distance between samples and the point of interest, which we call the range of the sampling. So explanations obtained with greater values of σ are likely to be more general than explanations with small values of σ. As another example, the value of parameter a (Definition 8) represents the minimum accuracy imposed during the search for the rule-based model. This parameter can be used to control the simplicity of the resulting explanation.

4 From contexts to explanations

In the previous sections, we have presented the three levels of abstraction that can be used by explainees to express their needs, namely context, requirements and technical options. Each level has been presented independently so that different types of users, depending on their level of expertise and types of needs, can use the most suitable level without having to know or to understand the lower levels. However, in order to produce explanations, these needs have in any case to be translated into technical options of the generic explainer. In this section, we present the two phases of this process, the translation of the context into requirements in Section 4.1 and the translation of requirements into technical options in Section 4.2. This process may result in the generation of several solutions (sets of technical options). We describe in Section 4.3 a final step which relies on a post-hoc evaluation of explanations to select the most suitable answer. The whole process is sketched in Figure 2 and we recall that all concepts and notations are summarized in B

4.1 From context to requirements

The context, as defined in Section 2, is made of four elements[7] : *Profile*, *Objective*, *Focus* and *Point of interest*. These elements are used to produce the requirements which are made of seven

[7]In addition to the *ADS* to be explained, which pertains to each level of abstraction.

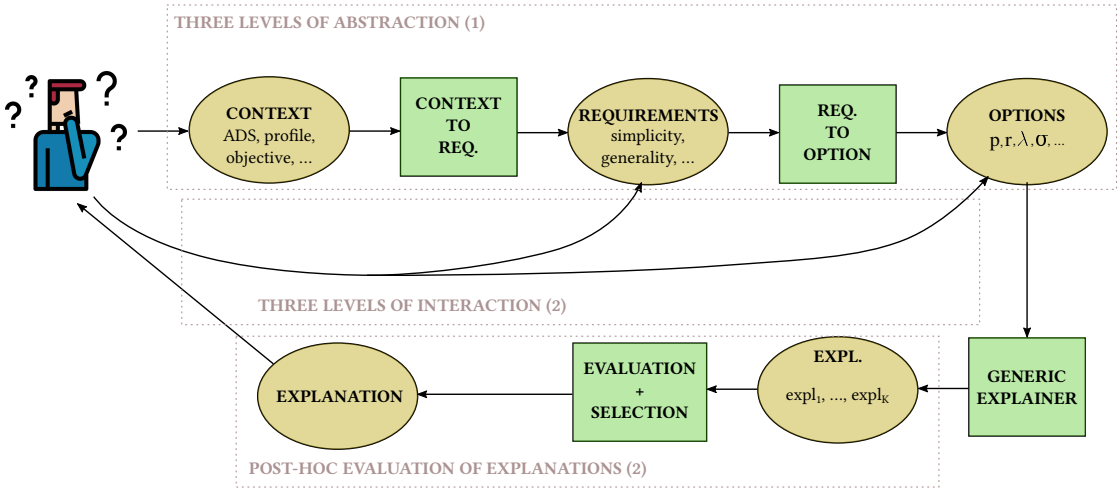


Figure 2: Overview of the approach.

elements: *Format*, *Simplicity*, *Generality*, *Point of interest*, *Realism*, *Actionability* and *Nature*.

As discussed in Section 2, we distinguish *hard requirements*, which impose specific values (*e.g.* general = 1), and *soft requirements*, which express preferences (*e.g.* general: $3 > 2 > 1$). When several preferences are expressed, they should be ordered by importance (*e.g.* general > simple). The operational semantics of requirements consists in filtering out first all explanations that do not meet the hard requirements before applying soft requirements by order of importance until only one explanation is left.

The first step of the translation procedure consists in using the *Focus* element of the context to select the subset of formats that can be used. For example, if *Focus* = G (meaning that the explainee is interested in a global explanation), then counterfactual explanations (CF) are not appropriate. More generally, the *Focus* associated with each format is presented in Table 2. If *Focus* = L (local explanation), then the *Point of interest* element of the requirements is obtained directly from the same element in the context. The other elements of the requirements are derived from the *Profile* and *Objective* elements of the context as presented in Figure 4.

In the following, we provide some intuition about the choices made in Table 4. Usually, simple explanations are preferred over complex explanations ([11] p.44). Simplicity is expressed as a soft requirement with a low priority unless the objective is *Trust*. Lay users generally expect explanations that are as simple as possible, thus a hard requirement is used (*simple* = 3).

The generality of an explanation (which is relevant only for local explanations) enhances the explainee's capabilities to understand the outcomes of the ADS for input values that have similarities with the point of interest. Therefore the values of the generality element should be maximum (*general* = *3*) when the objective is to increase the trust in the model ([11] p.44). On the other hand, when the objective for a lay user is to challenge a specific decision or to take actions to obtain better decisions from the ADS, a lower level of generality is more appropriate.

High levels of realism favour the generation of explanations that are supported by training data [22]. Depending on the context, this choice can be an advantage or a drawback. Explanations that are not supported by training data make it possible to analyze decision boundaries that are part of the model, but are not necessarily reflected in actual field data, as mentioned in the credit scoring example of Section 2.2. When the objective of the explainee is trust enhancement, decision boundaries that are actually used must be the primary concern, which justifies

Technical Expert		**Domain Expert**		
*Improve**	*Trust*	*Trust**	*Challenge*	*Action*
format: RB >DT >LA >PD >PC >CF	format: RB >DT >LA >PD >PC >CF	format: RB >DT >LA >PD >PC >CF	format: RB >DT >LA >PD >PC >CF	format = CF
simplicity: 3 >2 >1	simplicity: 3 >2 >1	simplicity: 3 >2 >1	simplicity: 3 >2 >1	simplicity: 3 >2 >1
generality: 3 >2 >1	generality = 3	generality = 3	generality: 1 >2 >3	generality: 1 >2 >3
realism = 1	realism: 3 >2 >1	realism = 3	realism = 1	realism = 2
actionability = F	actionabililty = F	actionability = F	actionability = F	actionability = T
nature = T	nature = T	nature: T >F	nature : F >T	nature = F
general >form >simple	simple >real >form	simple >nat >form	form >nat >simple	
Auditor		**Lay User**		
Trust	*Challenge**	*Trust**	*Challenge*	*Action*
format: RB >DT >LA PD >PC >CF	format: RB >DT >LA PD >PC >CF	format: RB >DT >LA PD >PC >CF	format: RB >DT >LA PD >PC >CF	format = CF
simplicity: 3 >2 >1	simplicity: 3 >2 >1	simplicity = 3	simplicity = 3	simplicity = 3
generality = 3	generality: 1 >2 >3	generality = 3	generality: 1 >2 >3	generality: 1 >2 >3
realism = 3	realism = 1	realism = 3	realism = 1	realism = 2
actionability = F	actionability = F	actionability = F	actionability = F	actionability = T
nature: T >F	nature: T >F	nature : F >T	nature : F >T	nature = F
simple >nat. >form	form >nat. >simple	nat. >form	form >nat.	

Table 4: Translation of the context into requirements. Hard requirements appear in black type and soft requirements in green type. When the explainee decides to provide only his profile, the starred objective is used as a default setting.

the choice of realistic sampling. On the other hand, technical experts may want to investigate these "theoretical" decision boundaries in order to assess the robustness of the model in all conditions. Similarly, it can be argued that these boundaries, even though they are rarely used in practice, could be helpful to detect potential non-compliance with existing regulations, such as non-discrimination laws. Low realism is thus also appropriate to challenge the ADS. Finally, to help explainees for future actions, likely input changes should be proposed (in the sense that the values are drawn from the actual distribution), but imposing the maximum realism would be too restrictive because this level of realism only considers inputs from the population set. Therefore, the intermediate level is used.

If the goal of the explainee is to take actions to improve his data (and possibly get a different decision from the ADS in the future), explanations should be focused on actionable features in order to suggest only feasible modifications. Therefore, *actionability* is True when the objective is action and False otherwise. As shown by previous studies ([23] p.44), the use of probabilities in explanations is usually not illuminating for explainees ($nature = F$), especially when they are interested in a single point of interest. However some profiles, such as auditors and technical experts, may be interested in a balanced view of the situation, which is provided by the use of probabilities ($nature:\ F > T$).

To conclude this section, we would like to emphasize that the choices presented in Table 4 are not hard-wired in the implementation of IBEX. The architecture of the system can accommodate different choices of translation and this flexibility will be used to improve it based on the feedback of the users and field experience.

4.2 From requirements to technical options

In this section, we show how the six elements[8] of the requirements can be used to choose the technical options of the explanation framework. Let us consider each of these elements in turn :

- **Format:** The format element is used to choose the instantiation of the generation component of the explainer. However, when this element is expressed as a soft requirement, the choice of the generation component may depend on other elements of the requirements, in particular Nature, as discussed below.
- **Simplicity:** simplicity can be translated as a condition on the size (number of items) of an explanation through technical parameters of the generation component. For instance, local linear approximations use the penalization parameter λ to control the number of non-zero coefficients: increasing the weight of the penalization thus increases the simplicity of the resulting explanation. The mapping for other instantiations of the generation component is presented in Table 5.
- **Generality:** generality has an impact on the range of the sampling, *i.e.* the average distance between the point of interest and the samples. An explanation derived from examples that are close to the point of interest (small range) is unlikely to be general. The range of the sampling can be expressed through parameters of the sampling component. For example, "Add random noise" controls the range through σ, the standard deviation of the added noise, as depicted in Figure 1d. The mapping for other sampling components is presented in Table 5.
- **Realism:** the realism of an explanation is fully determined by the sampling component. Sampling strategies are categorized based on their level of usage of the population set. "Add random noise", which makes no use of the population set, is associated with a low level of realism ($realism = 1$), while "select closest" and "identity", which use it heavily, are associated with a high level of realism ($realism = 3$); "permutation" and "replace with constant" are associated with an intermediate level ($realism = 2$).
- **Actionability:** if the user chooses to focus on actionable features, the non-actionable features are removed during the sampling step. Thus they cannot appear in the final explanation.
- **Nature:** This requirement influences the possible instantiations of the generation component based on their use of probabilities. For instance, rule-based explanations include the probability of a sample being correctly predicted, while counterfactual explanations do not involve any probabilities. The components involving probabilities are RB and DT.

In general, the translation procedure presented in this section may yield several possible solutions (sets of technical options), in particular when soft requirements are involved. In the following section, we present a post-hoc evaluation procedure to select, among these options, the solution which "best" meets the requirements.

4.3 Post-hoc explanation evaluation

When the translation procedure presented in the above section yields several solutions, it is necessary to choose among them the set of technical options that is the most likely to address

[8] We do not discuss further the *Point of interest* element which can be used directly as in the previous section.

Requirement	**Component**	**Param.**	**Effect**
Simplicity	LA	λ	High simplicity $\Longrightarrow$ large λ
	RB	a	High simplicity $\Longrightarrow$ small a
	DT	a_{DT}	High simplicity $\Longrightarrow$ small a_{DT}
	PD	n	High simplicity $\Longrightarrow$ small n
Generality	AN	σ	High generality $\Longrightarrow$ large σ
	Pm	p	High generality $\Longrightarrow$ large p
	SC	r	High generality $\Longrightarrow$ large r

Table 5: Mapping requirement to technical parameters

the needs of the explainee. Because we cannot make any assumption on the regularity of the black-box ADS, it is not feasible to predict the exact properties of an explanation based solely on the technical options of the explainer. Let us assume, as an illustration, that the explainee desires a local explanation which has a good level of generality and is expressed as a rule-based model. The shape of the closest decision boundary of the ADS and its distance to the point of interest greatly influence the number of rules needed to achieve an acceptable precision. To overcome this issue and to ensure that the explanation generated by the explainer will meet the requirements, our translation process includes a last *post-hoc evaluation* step: the generation of the explanations corresponding to the different solutions (technical options for the explainer) produced by the previous step followed by an evaluation of their properties.

Generally speaking, the assessment of the qualities of explanations is still an open research question. We consider here their compliance with respect to requirements as defined in Section 2.2. More precisely, we focus on the *Simplicity* and *Generality* elements, which are often expressed as soft requirements. The assessment of simplicity is based on the number of items involved in the explanation (*e.g* the number of rules in a rule-based model, the number of modifications in a counterfactual example, etc.). This use of the size of an explanation as a proxy for simplicity is common [24]. It has some limitations (size does not always reflect simplicity) but it is operational and it can be instantiated to any explanation format.

The assessment of generality relies on a test of the explanation on inputs from the population that are close to the point of interest. If the explanation is not valid for a minimum number of inputs (threshold T_1) then the generality is 1; if it is valid for the T_1 closest inputs but not for T_2 inputs ($T_2 > T_1$), then the generality is 2; if it is valid for the T_2 closest inputs then the generality is 3[9].

To conclude this section, it is important to stress that the definition of the needs of the explainee (at one of the three levels of abstractions defined here) is only the first interaction step of the explainee with IBEX. When an explanation has been generated by IBEX based on the set of technical options resulting from this initial step, the explainee can reply to IBEX with a new request. This request can refer to the initial explanation (e.g. asking for a "richer", or "less simple", explanation, or an explanation in a different format) or can be entirely new and expressed again at any level of abstraction. By allowing explainees to interact at a different abstraction levels, IBEX gives them the opportunity to express their needs in a very precise way and to react to previous explanations.

[9]In the current version of IBEX , threshold T_1 is set to 10 and T_2 is set to 50.

5 IBEX at work: application to case studies

In this section, we illustrate our approach with the application of our proof-of-concept system IBEX to several case studies. The implementation of IBEX follows directly the approach presented in the previous sections and the interested reader can find in A complementary details about some technical choices that are not described in the core of the paper. The code of IBEX is publicly available[10].

Interactions at any level of abstraction are feasible with IBEX. By default, the interaction is done at the context level which is the most appropriate for lay users. The aims of the initial interactions at this level are to elicit the needs of the explainee and to express them in terms of context elements (as defined in Section 2). These interactions take place as follows (questions asked by IBEX):

1. Choose a data set.
2. Are you interested in global (G) or local (L) explanations?
3. What is your point of interest? (optional question: for local explanations only)
4. How do you want to be considered by IBEX: as a technical expert (TE), a lay user (LU), a domain expert (DE) or an auditor (AU) ?
5. What is the objective of the explanation: is it to improve the ADS (I), to enhance your trust in the ADS (T), to challenge the ADS (C), or to take future actions based on results of the ADS (A)?
6. What are your actionable features? (optional question: for objective A only)

The user may skip any of these questions (except the first one) if he is not sure about the answer. In any case, IBEX then generates a first explanation based on this (potentially partial) context and asks the user whether he wants to ask further questions. If so, the user has two options: he can either ask an entirely new question (simple iteration of the protocol) or ask a question based on the previous explanation. In this case, he can express his wishes as a tuning of the requirements, for example "simpler explanation", "more general explanation", or "actionable explanation". Alternatively, he can ask to see the requirements derived from the previous question and modify any of its elements by himself. In both cases, IBEX will then generate new technical options and a new explanation based on the new requirements. The user will then have again the options to stop, ask an entirely new question or a question involving the previous answer.

In order to illustrate the benefits of the approach in terms of versatility and interactivity, we consider successively the use of IBEX in the follwing situations:

1. A lay user requesting explanations about an ADS (based on a 2-layer neural network model) applied to the adult census data set[11] with the objective of enhancing trust (Section 5.1).
2. A domain expert requesting explanations about an ADS (based on a support long short-term memory neural network model) applied to the airline sentiment analysis data set[12] with the objective of enhancing trust (Section 5.2).
3. A lay user requesting explanations about an ADS (based on a random forest model) applied to the German credit data set[13] with the objective of taking actions (Section 5.3).

[10] https://gitlab.inria.fr/chenin/ibex
[11] https://archive.ics.uci.edu/ml/datasets/Adult
[12] https://www.kaggle.com/crowdflower/twitter-airline-sentiment
[13] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

4. An auditor requesting explanations about an ADS (based on a 2-layer neural network model) applied to the adult census data set with the objective of challenging the ADS (Section 5.4).

The three data sets, which are publically available, are used as the population sets (Section 3). Their features are summarized in Table 6.

Data set	Features	Type	Output	Model
Adult census	Personal information about American citizens (age, education, marital status, ...)	Tabular	Yearly income ('<50k', '>=50k')	2-layer NN
German credit	Credit application information (amount, type of job, ...)	Tabular	Risk profile ('bad', 'good')	Random Forest
Airline sentiment analysis	Tweets about airline companies	Textual	Sentiment category	LSTM NN

Table 6: Datasets and black-box models used for the case studies

5.1 Use by a lay user to enhance trust

The first case study involves the adult census data set. This data set, which has been extracted from the 1994 US census, contains personal information about American citizens such as their age, education level or marital status. The goal of the ADS is to predict, based on these features if the individual earns more or less than $50,000$\$ per year. A lay user who wants to enhance his trust in the ADS would choose the following answers: *data set=adult census, focus=G, profile=LU and objective=T*. From this context, IBEX has generated the explanation presented in Figure 3. We can see that the explanation is simple, it is composed of a decision tree with only two nodes and three leaves, which is consistent with the choices presented in Table 4.

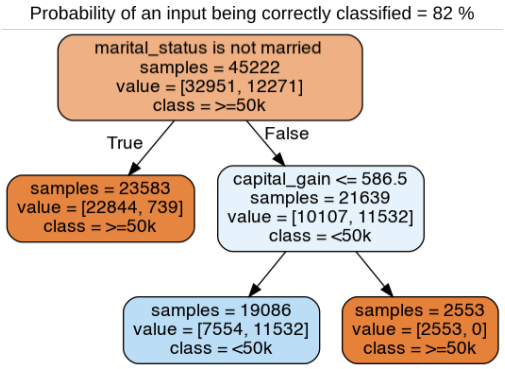


Figure 3: Explanation generated by IBEX for the adult census data set based on the initial context. IBEX has used the following requirements: format=DT, simplicity=3, actionability=F, nature=T, realism=3

.

The requirements generated by IBEX for this context are presented in the left part of Figure 4. We can see that $nature = F > T$, meaning that an explanation that does not involve any probability would have been preferred by the user. Nevertheless, the explanation generated by IBEX involves a probability. The reason is that the first explanation formats that were considered by IBEX (*PC* and *PD*) led to explanations that were considered too complex to satisfy the hard requirement $simple = 3$. For this reason, the post-hoc evaluation step of IBEX made the choice of a decision tree format.

```
HARD REQUIREMENTS:
        actionability : F
        simplicity : 3
        generality :
        realism : 3
SOFT REQUIREMENTS:
        nature : F > T
        format : DT > PD > PC
```

(a)

```
HARD REQUIREMENTS:
        actionability : F
        simplicity : 2
        generality :
        realism : 3
        format : DT
SOFT REQUIREMENTS:
        nature : F > T
```

(b)

Figure 4: (a) Requirements derived by IBEX from the initial context (G, LU, T); (b) Revised requirements based on the user's request "less simple".

Let us assume now that the user is almost satisfied with this first explanation but he suspects that the logic of the ADS is much more complex and this explanation is a bit simplistic. Through the IBEX interface, he can either request a "less simple" explanation or ask IBEX to show the requirements derived from the previous question and modify by himself the simplicity element. In the first case, IBEX would generate the requirements shown in the right part of Figure 4 and the explanation presented in Figure 5. We can see that this explanation is indeed less simple than the previous one and it provides a more accurate description of the logic of the ADS.

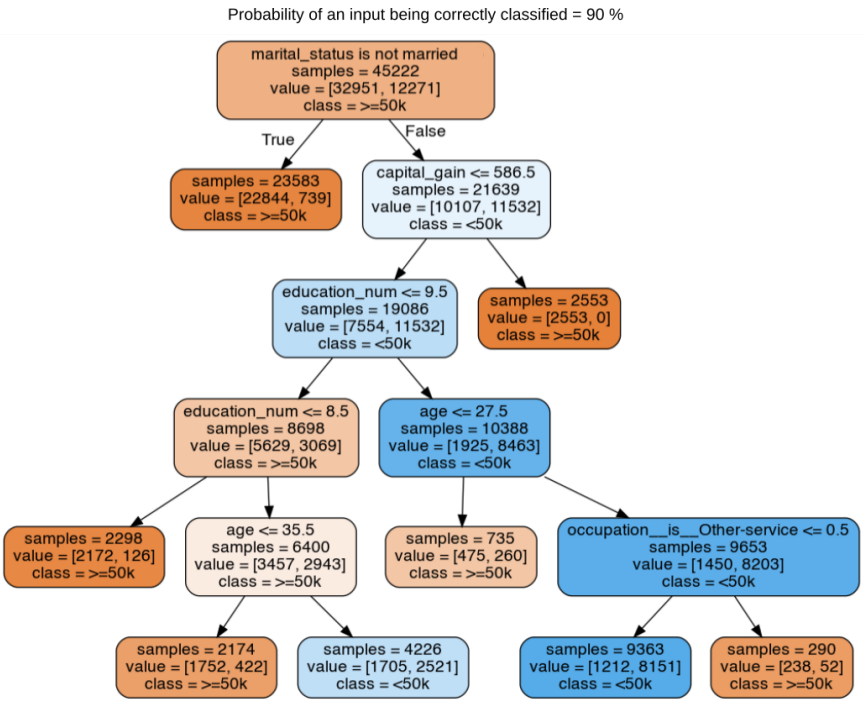


Figure 5

Figure 6: Explanation generated by IBEX for the adult census data set based on the revised requirements: $format = DT, simplicity = 2, actionability = F, nature = T, realism = 3$

5.2 Use by a domain expert to enhance trust

The second case study involves the airline sentiment analysis database. This data set contains tweets about airline companies and the objective of the ADS is to classify them into three categories: negative, neutral or positive. Negative tweets are supposed to express negative emotions (anger, irritation, etc.), positive tweets are supposed to express positive emotions (happyness, gratitude, etc.) and neutral tweets show no or little emotion.

Let us assume that an employee working in the customer relationship service of a company wants to better understand the ADS to make a better use of it. This employee uses IBEX as a domain expert (DE), asking explanations about specific tweets (L) and with the objective of increasing trust (T).

Table 4 shows that IBEX associates this context with general and realistic explanations. Figure 7 presents explanations about specific tweets generated by IBEX based on this request. All these explanations meet the generality requirement: they are valid for large classes of inputs (tweets), which is very helpful in improving the understanding of the model. The fact that the explanations are simple and yet very precise tends to show that the model does not use complex combinations of words and classifies tweets based on the presence or absence of a limited number of keywords (like "cancelled" or "great"). Interestingly, one of the explanations is in the format LA and all others are in the format RB. RB is the preferred format for this context (because it involves probabilities), but, for this specific tweet, the only explanation that matched the generality requirement was of the LA format.

Tweet	Pred	Explanation
@AmericanAir although you have no control of the weather, you came through with a great customer service	Positive	IF the words "great" and "weather" appear THEN the tweet is positive with proba > 90 % among samples
@SouthwestAir Jason (108639) at Gate #3 in SAN made my afternoon!!! #southwestairlines #stellarservice #thanks!	Positive	Estimation of feature contributions (blue=positive) 
@AmericanAir all right, but can you give me an email to write to?	Neutral	IF the words "can" and "right" appear THEN the tweet is neutral with proba > 90 % among samples
@VirginAmerica You have any flights flying into Boston tomorrow? I need to be home and you Cancelled Flightled my flight and didn't do anything	Negative	IF the word "cancelled" appears THEN the tweet is negative with proba > 99 % among samples
@SouthwestAir What can we do to bring you back to Jackson, MS?! We miss you terribly around here. These other airlines are horrible‼	Negative	IF the word "horrible" appears THEN the tweet is negative with proba > 99 % among samples

Figure 7: Example of explanations to enhance trust of a domain expert. All explanations are very general and realistic. All of them, except one, use the RB format ($nature = T$), which is preferred for domain experts.

5.3 Use by a lay user to take actions

The third case study concerns the German credit data set which contains information about the credits (amount, duration, purpose, etc.) and the applicants (type of job, number of ongoing credits, etc.). The ADS classifies applications as risky ("bad") or safe ("good"). Let us consider an individual whose credit application has been declined and who would like to know how to improve it to have it approved in the future. The profile for this query is lay user (LU) and the objective

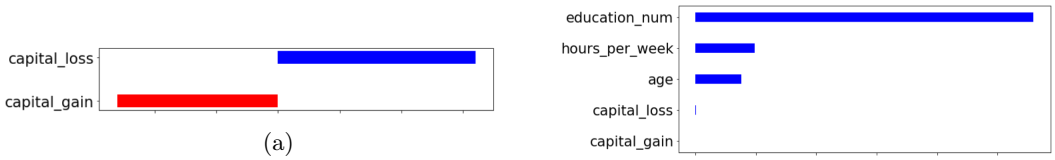
is to prepare future actions (A) for a specific input (L). From Table 4, we can see that IBEX associates this context with the *CF* format, the average level *2* of realism (level *1* would lead to unlikely modifications of the application while level *3* could end up with a counterfactual too far away from the optimal value). Also, possible modifications need to be limited to actionable features (e.g. duration of the credit or number of ongoing credits), which are provided by the explainee. The counterexample generated from this context by IBEX, shown in Table 7, suggests two modifications of the current application: the duration of the credit and telephone ownership.

Actionable features	Credit amount	Duration	Ongoing credits	Job	Telephone ownership	Output
Current	10722	47	1	unskilled resident	yes	**Bad**
CF	10722	36	1	unskilled resident	none	**Good**

Table 7: Realistic counterfactual explanations based on the modification of actionable features.

5.4 Use by an auditor to challenge the ADS

For the last case study, we assume that an auditor (AU) wants to challenge (C) a specific decision (L) concerning the adult census data set. One way to challenge an ADS is to show that it uses features that are not allowed. For this context, IBEX chooses a low value for realism, because it is useful to understand the true shape of the decision boundaries, independently of the use of the ADS in practice. Figure 8 shows an example illustrating the difference between realistic and non-realistic explanations. The explanation on the left side of Figure 8a has been generated by IBEX for this context (with $realism = 1$). The explanation on the right side, which is provided for comparative purposes, has been generated with a high value for realism ($realism = 3$). The latter takes into account the probability of observing a change in the input scope together with the effect of this change on the ADS output. In contrast, the leftmost explanation describes the effect of a change on the output disregarding the actual distribution of the input data.



(a) (b)

Figure 8: Two LA explanations generated by IBEX for the same adult input predicted as '$< 50k$'. The positive values (blue bars) indicate a positive impact of the feature on the output '$< 50k$' and negative values (red bars) indicate a negative impact on the output '$< 50k$'. (a) The explanation on the left (generated with realism=1) reflects the logic of the ADS, disregarding the actual distribution of the input data. (b) The explanation on the right (generated with realism=3) reflects the behaviour of the ADS on real data (based on the distribution of the input data set).

The explanation of the left shows that *capital_gain* and *capital_loss* have the highest impact on the decisions, while the explanation on the the right emphasizes *education_num* and *hours_per_week*. None of these explanations is more "true" than the other, they just explain the behaviour of the ADS from different perspectives. Our analysis of the population of the data set shows that less than 9% of all inputs have capital gains and that they were all classified as '$> 50k$'. This corroborates the fact that, even if *capital_gain* is a very good indicator of the class '$> 50k$', it does not appear in the explanation on the right because it is a rare event. The

choice of a low level for realism made it possible for IBEX to identify this aspect of the model that could be used to challenge a decision, especially if *capital_ gain* is not supposed to play a role in the decision. This type of explanation can be also useful to a technical expert who wants to improve the ADS, as confirmed by Table 4 (*TE* is also associated with *realism=1*).

6 Related works

To the best of our knowledge, no existing explanation system provides the diversity of explanations and the interaction capabilities offered by IBEX. Some authors have already proposed taxonomies of explainee's profiles [9, 25], explanations' objectives [11, 13, 26] or combinations of profiles and objectives [12, 27, 10]. The impact of the type of question on the explanation has been analyzed through a user study in [16]. In the same vein, the impact of the explanation form has been studied in [28]. Some works also aim at identifying appropriate sets of features of explanations [29, 1, 11]. These contributions are related to this paper in the sense that the categorization of explainees' needs is a key element of our interactive approach. However, the goal of these contributions is to identify and categorize these needs, rather than to design a generic interactive explainer. To the best of our understanding, none of them suggests an operational mapping to actual explanations as presented here.

As far as the generation of explanations is concerned, a plethora of methods already exist. In most cases, explanations are seen as static objects that are provided without taking into consideration the explainees' profiles or specific needs [20, 18, 19, 30]. [1, 26, 10] provide exhaustive surveys of the literature. Some surveys and classifications of explanation methods focus on the theoretical underpinnings of explanations while others are general overviews of existing methods. The scope of explanation methods considered in [1] is particularly wide, including black-box, white-box and constructive methods. This very comprehensive survey introduces a glossary and a taxonomy for interpretable and explainable AI. It then identifies a wide range of publications in this field and classifies them according to the taxonomy.

Some contributions involve a form of interaction with explainees. AIX360 [7] contains eight explainability algorithms and allows users to choose among them based on a taxonomy including criteria such as "understand the data or the model" or "self-explaining model or post-hoc explanations". As it takes into consideration the user's needs, AIX360 provides a first level of interaction with explainees. However, the three levels of abstraction available in IBEX allow for richer interactivity, for instance, by allowing to choose the levels of simplicity, generality and realism of the explanation. Moreover, the generic explainer can be customized to fulfill the requirements of the explainee, which is not possible with the portfolio approach of AIX36. Finally, IBEX offers the possibility to react to an explanation, which is also an original feature.

Glass-Box [31] allows explainees to interact with an adaptive explainer through a voice-based (or chat-based) interface. The system provides local explanations, under the form of counterfactuals, and allows explainees to react in order to obtain a new explanation. Although Glass-Box has similarities with IBEX, its interactive capabilities are limited to the choice of actionable features for counterfactuals (which is also included in IBEX). The bLIMEy system (for "build LIME yourself"), is a generic explainer relying on the framework proposed in [8]. As in IBEX, several sampling strategies are available to produce a variety of explanations. However, bLIMEy does not include an analysis of the context of the explainee's query, neither does it include a mapping from this context to technical options, as done in IBEX.

SHAP [32] proposes a unified approach to describe four explanation methods. It is related to our generic explainer, in the sense that it uses a unique theoretical description to describe several explanation methods. However, SHAP is restricted to explanations under the form of feature

importance and it is not interactive. In a related area, many works have been done on interaction with machine learning systems for the sake of improvement or debugging [33]. These proposals involve a form of interaction with the users, but the objective is not to explain a black-box model, as in IBEX.

Some authors consider interactive explanation frameworks from a more theoretical point of view. For instance, [34] defines the specifications of a dialogue system for explanations and [35] proposes an interaction protocol for XAI. These works are related to IBEX, and could be useful sources of inspiration to enhance its interaction facilities. However, their goal is not to propose an operational explanation system.

Finally, on the implementation side, many projects have recently emerged to provide implementations of existing methods [36, 37, 38]. The goal of these projects is to integrate a variety of existing methods, but they do not include a comprehensive interaction module and a fine-grain decomposition of components as done in IBEX.

7 Conclusion

The main goal of the work described in this paper is to address the variety of needs in terms of explanations of ADS and to design an explanation system that can be used by a wide range of explainees, including lay users. To this aim, we have proposed a framework involving different levels of abstraction and a fine-grain decomposition of explanation tasks that can be combined in different ways. As a byproduct of this work, it is possible to use this decomposition into atomic components to compare and classify existing black-box explanation methods more precisely than presented in the various surveys published on this topic. The interested reader can find in [8] a table and discussion showing that existing methods can be seen as particular instantiations of this framework, i.e. particular choices of the technical options presented in this paper. This analysis shows the generality of the framework and the benefits of the fine-grain approach to devise new combinations of options.

In this paper, we have shown, through the IBEX prototype, the feasibility of an interactive explanation system based on our approach. However, as stated above, IBEX is a proof of concept implementation and it can be improved and extended in several directions. It should be noted that the architecture of IBEX is highly modular. For example, new profiles, sampling strategy or data representations can easily be added without major modification of the system. In order to prove its usability as an explanation system in real life, it should be tested through a randomized user study involving different types of explainees, which we plan to do in the near future in collaboration with partners in the health care and the financial sectors. In this perspective, a key aspect of explanations that has not been developed in this paper is their assessment. Different criteria have been proposed to assess the quality of an explanation [39]. Our framework makes it possible to specify quality objectives, either as constraints or as criteria, as presented in Section 3, but it does not provide any help to evaluate the relevance of these objectives (for example through an assessment of the understanding of the explainee). This is a major avenue for further research.

A first improvement of IBEX concerns the user interface, which is very basic in the current version. In particular, it would be interesting to provide a richer and higher-level language to interact with explainees, for instance a restricted version of natural language that could be used by explainees to express questions such as “Why is it the case that my application has been rejected ?” or “Why has this file been accepted and not this one ?” In some cases, requirements or technical options for the generation of explanations could be derived directly from such questions. In other cases, the explanation system would in turn ask a question to the explainee in order to

allow him to refine his initial request. Natural language can also be used to express explanations, for example to return an explanation such as "The most influential factor for explaining the reject decision is the number of outstanding loans". Dialogue specifications could rely on models such as [35].

A second improvement of IBEX could be the use of more elaborate sampling strategies that would provide further advantages in term of flexibility and efficiency of the computation, especially for high-dimensional data. The use of genetic algorithms, as presented in [40], is a promising approach to achieve this goal.

Another important area for further research is the design of new types of explanations and interactions to make it easier for the users of an ADS (or people affected by its decisions) to challenge its decisions. Indeed, in order to support decision challenging, it is necessary to provide interactions about justifications (why a given decision is the good one), and not only about explanations (why the ADS made or suggested this decision). This requires a form of argumentation currently beyond the scope of IBEX.

References

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Computing Surveys (CSUR) 51 (5) (2018) 93.

[2] T. Miller, P. Howe, L. Sonenberg, Explainable AI: beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences, CoRR abs/1712.00547. arXiv:1712.00547.
URL http://arxiv.org/abs/1712.00547

[3] P. Madumal, T. Miller, F. Vetere, L. Sonenberg, Towards a grounded dialog model for explainable artificial intelligence, CoRR abs/1806.08055. arXiv:1806.08055.
URL http://arxiv.org/abs/1806.08055

[4] B. D. Mittelstadt, C. Russell, S. Wachter, Explaining explanations in AI, CoRR abs/1811.01439. arXiv:1811.01439.
URL http://arxiv.org/abs/1811.01439

[5] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, M. Kankanhalli, Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18, ACM Press, Montreal QC, Canada, 2018, pp. 1–18. doi:10.1145/3173574.3174156.
URL http://dl.acm.org/citation.cfm?doid=3173574.3174156

[6] S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, Explanation in human-ai systems: A literature meta-review synopsis of key ideas and publications and bibliography for explainable ai 204.

[7] V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, et al., One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, arXiv:1909.03012 [cs, stat]ArXiv: 1909.03012.
URL http://arxiv.org/abs/1909.03012

[8] C. Henin, D. Le Métayer, Towards a generic framework for black-box explanations of algorithmic decision systems (Extended Version), Inria Research Report 9276, https://hal.inria.fr/hal-02131174.

[9] R. Tomsett, D. Braines, D. Harborne, A. D. Preece, S. Chakraborty, Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, CoRR abs/1806.07552.

[10] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, arXiv:1910.10045 [cs]ArXiv: 1910.10045.
URL http://arxiv.org/abs/1910.10045

The icon of Fig. 2 was made by turkkub from www.flaticon.com

[11] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267. doi:10.1016/j.artint.2018.07.007.

[12] A. Weller, Challenges for transparency, arXiv:1708.01870 [cs]ArXiv: 1708.01870.
URL http://arxiv.org/abs/1708.01870

[13] Z. C. Lipton, The mythos of model interpretability, arXiv:1606.03490 [cs, stat]ArXiv: 1606.03490.
URL http://arxiv.org/abs/1606.03490

[14] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harvard journal of law & technology 31 (2018) 841–887.

[15] S. Stumpf, V. Rajaram, L. Li, M. Burnett, T. Dietterich, E. Sullivan, R. Drummond, J. Herlocker, Toward harnessing user feedback for machine learning 10.

[16] B. Y. Lim, A. K. Dey, D. Avrahami, Why and why not explanations improve the intelligibility of context-aware intelligent systems, in: Proceedings of the 27th international conference on Human factors in computing systems - CHI 09, ACM Press, 2009, p. 2119. doi:10.1145/1518701.1519023.
URL http://dl.acm.org/citation.cfm?doid=1518701.1519023

[17] T. Miller, P. Howe, L. Sonenberg, Explainable AI: Beware of inmates running the asylum, in: IJCAI-17 Workshop on Explainable AI (XAI), Vol. 36, 2017.

[18] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & Explorable Approximations of Black Box Models, arXiv preprint arXiv:1707.01154.

[19] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: AAAI Conference on Artificial Intelligence, 2018.

[20] M. T. Ribeiro, S. Singh, C. 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 - KDD '16, ACM Press, San Francisco, California, USA, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.
URL http://dl.acm.org/citation.cfm?doid=2939672.2939778

[21] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowledge and Information Systems 41 (3) (2014) 647–665. doi:10.1007/s10115-013-0679-x.

[22] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, The dangers of post-hoc interpretability: Unjustified counterfactual explanations, arXiv:1907.09294 [cs, stat]ArXiv: 1907.09294.
URL http://arxiv.org/abs/1907.09294

[23] T. Miller, Explanation in artificial intelligence: insights from the social sciences, arXiv preprint arXiv:1706.07269.

[24] F. Doshi-Velez, B. Kim, Towards A Rigorous Science of Interpretable Machine Learning, arXiv e-prints (2017) arXiv:1702.08608arXiv:1702.08608.

[25] G. Ras, M. van Gerven, P. Haselager, Explanation methods in deep learning: Users, values, concerns and challenges, CoRR abs/1803.07517. arXiv:1803.07517.
URL http://arxiv.org/abs/1803.07517

[26] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160. doi:10.1109/ACCESS.2018.2870052.

[27] C. T. Wolf, Explainability scenarios: towards scenario-based xai design, in: Proceedings of the 24th International Conference on Intelligent User Interfaces - IUI '19, ACM Press, 2019, p. 252–257. doi:10.1145/3301275.3302317.
URL http://dl.acm.org/citation.cfm?doid=3301275.3302317

[28] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, H. Wallach, Manipulating and measuring model interpretability, arXiv:1802.07810 [cs]ArXiv: 1802.07810.
URL http://arxiv.org/abs/1802.07810

[29] M. Hall, D. Harborne, R. Tomsett, V. Galetic, S. Quintana-Amate, A. Nottle, A. Preece, A systematic method to understand requirements for explainable ai (xai) systems 7.

[30] B. Kim, R. Khanna, O. O. Koyejo, Examples are not enough, learn to criticize! criticism for interpretability, in: Advances in Neural Information Processing Systems, 2016, pp. 2280–2288.

[31] K. Sokol, P. Flach, One explanation does not fit all: The promise of interactive explanations for machine learning transparency, KI - Künstliche Intelligenzdoi:10.1007/s13218-020-00637-y.
URL http://link.springer.com/10.1007/s13218-020-00637-y

[32] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, Curran Associates, Inc., 2017, p. 4765–4774.
URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[33] J. A. Fails, D. R. Olsen, Interactive machine learning 7.

[34] D. Walton, A dialogue system specification for explanation, Synthese 182 (3) (2011) 349–374. doi:10.1007/s11229-010-9745-z.
URL http://link.springer.com/10.1007/s11229-010-9745-z

[35] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, A grounded interaction protocol for explainable artificial intelligence, arXiv:1903.02409 [cs]ArXiv: 1903.02409.
URL http://arxiv.org/abs/1903.02409

[36] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, arXiv preprint arXiv:1909.09223.

[37] J. Klaise, A. Van Looveren, G. Vacanti, A. Coca, Alibi: Algorithms for monitoring and explaining machine learning models.
URL https://github.com/SeldonIO/alibi

[38] P. Biecek, Dalex: Explainers for complex predictive models in r, Journal of Machine Learning Research 19 (84) (2018) 1–5.
URL http://jmlr.org/papers/v19/18-416.html

[39] A. Dhurandhar, V. Iyengar, R. Luss, K. Shanmugam, A formal framework to characterize interpretability of procedures, arXiv:1707.03886 [cs]ArXiv: 1707.03886.
URL http://arxiv.org/abs/1707.03886

[40] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv:1805.10820 [cs]ArXiv: 1805.10820.
URL http://arxiv.org/abs/1805.10820

A Implementation Choices

The implementation of IBEX follows directly the approach presented in this paper, in particular Section 2, Section 3 and Section 4. In this Appendix, we provide complementary details about some implementation choices that were not described in the core of the paper.

In Section 3.2, we mentioned the need to build a meaningful representation of the data when its features are not directly interpretable. For textual data, IBEX relies on a positional representation similar to the one used in the implementations of LIME [20]. Texts are represented as fixed-size sequences of words. The tabular representation refers to the words by their positions and, for each position, a boolean indicates if the word is present or not. For instance, an initial text of four words (*e.g* "thanks for the flight") is represented by a list of four booleans $(1, 1, 1, 1)$. A modified version of this input in which the third word is removed (*e.g* "thanks for flight") or modified (*e.g* "thanks for good flight") is represented by the boolean list $(1, 1, 0, 1)$. The explanation is generated based on this positional representation of texts and displayed through a reference to the original value. This type of representation is suitable only for local explanations.

Some readers might be curious of how distances are computed with various types of data. For numeric type, the Manhattan distance is used, while for categorical data (including textual data), we used the Goodall distance.

In Section 4.3, we mentioned the need to do post-hoc evaluations of the simplicity and generality. The definition of simplicity generality depends on the generation component and on the data type. The simplicity of an explanation is approximated by the number of rules (respectively nodes) for RB and DT, by the number of non-zero coefficients for LA and PC and by the number of features to be modified for CF.

Assessing the generality of an explanation is more challenging. Generality refers, for local explanations, to the applicability and validity of an explanation for similar instances and should thus be based on checking the explanation on inputs from the population set that are similar to the point of interest. The idea is therefore to evaluate the explanation on a "neighborhood set". For tabular data, the neighborhood is defined as the 50 inputs that are the closest to the point of interest. For textual data, the neighborhood includes all inputs that contain all the words that are mentioned in the explanation. For RB, the system checks the validity of the rules on the neighborhood set. For CF, the validity of the CF is evaluated by applying the modifications of the CF (if, as in the example Table 7, the CF suggests to change the duration from 47 to 36, duration=36 is applied to all inputs of the neighborhood), applying the black box F and comparing the output to the expected output ("Good" in the example of Table 7). For LA, the systems compute new LA to each input of the neighborhood set and compare the coefficients obtained to the original LA of the point of interest. As the coefficients are float estimations, it is very unlikely that the values match exactly. To cope with this issue, we have to use a tolerance error during the comparison of coefficient values. The reasonable tolerance that we used in our case is the standard deviation of the coefficient, obtain through a bootstrap evaluation.

In rare cases, it is not feasible to find any explanation matching all hard requirements. Then, in order to still provide an explanation, the system has to relax one of the hard requirement. Implementation on specific use case should avoid occurrences of this situation is as it leads to the delivering of an explanation that is not best suited for the explainee. When such a case happens, IBEX notifies the user and specifies which constraint was relaxed.

It should be noted that, although it is not mentioned in the present paper, the implementation of IBEX allows the computation of explanations for images. In particular, a model to classify handwritten digits[14] is available for testing in the implementation, as well as a popular and

[14] http://yann.lecun.com/exdb/mnist/

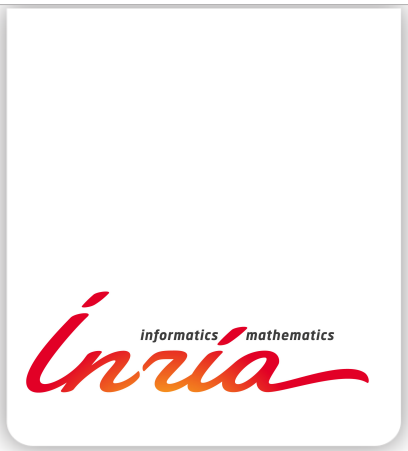
general image classifier, based on ImageNet[15]. We decided not to include these in the paper because no satisfying equivalent for the generality of the explanation could be found for images (the notion of distance for image is ill-defined) and because efficient computation of explanation in very high dimension requires huge effort on optimization, which is far beyond the scope of IBEX.

[15]https://keras.io/applications/#vgg16

B Summary of notations

Name	Abbr.	Type	Description
Context		Level of interaction	Higher level. Describes explainee's query
Requirements		Level of interaction	Intermediate level. Describes explanation features
Technical options		Level of interaction	Lower level. Parameters for the generic explainer
Algorithmic decision system	ADS	Context element	The black-box
Profile		Context element	Defines the expertise of the explainee
Objective		Context element	Objective or goal of explainee's query
Focus		Context element	Local or global: scope of the explanation
Point of interest		Context element	For local explanation, the input that is considered
Format		Requirement element	Form of explanation (cf. instantiations of generation)
Simplicity		Requirement element	Simplicity of the explanation (number of items in IBEX)
Generality		Requirement element	Measures how many inputs are covered
Point of interest		Requirement element	For local explanation, the input that is considered
Realism		Requirement element	Use of the underlying distribution to generate samples
Actionability		Requirement element	Only actionable features are considered
Nature		Requirement element	Presence or absence of probability in the explanations
Hard requirement		Type of requirement	Select one value (e.g. generality=2)
Soft requirements		Type of requirement	As preference (e.g. simplicity: $3 > 2 > 1$)
Sampling		Technical component	Choice of samples used to generate the explanation
Generation		Technical component	Deriving explanation from samples
Criteria		Generation objective	Values being minimized during the generation
Constraint		Generation objective	Constraint imposed during the minimization
Technical expert	TE	Profile	Implements the ADS
Auditor	AU	Profile	External expert auditing / evaluating the ADS
Domain expert	DE	Profile	Expert of the domain of the ADS (e.g. physician)
Lay user	LU	Profile	No specific expertise
Improve	I	Objective	Enhancement of the ADS
Trust	T	Objective	Understand ADS
Challenging	C	Objective	Challenge decision or whole system
Action	A	Objective	Optimize future interaction with the ADS
Global	G	Focus	Explanation about the whole model
Local	L	Focus	Explanation about a single input

Name	**Abbr.**	**Type**	**Description**
Select closest	SC	Sampling	Select from D inputs that are close to the scope
Permutation	Pm	Sampling	Permutations of features among samples
Add random noise	AN	Sampling	Noisy (normal) versions of the scope
Identity	Id	Sampling	Returns the population set
Replace with constant	RC	Sampling	Exchanges values with constants
Rule-based model	RB	Generation	Local: Accurate and simple RBM
Counterfactual	CF	Generation	Local: Close sample with a different output
Decision tree	DT	Generation	Global: small DT with minimum accuracy
Pearson correlation	PC	Generation	Global: linear importance of feature
Partial Dependence	PD	Generation	Global: Average output of each feature value
Black-box	F	Function	The black-box function of the ADS
Input space	X	Type set	Space of input variable for the black-box F
Output space	Y	Type set	Output space of F
Scope	E	Set of inputs	Set of inputs referred to by the explanation
Samples	S	Set of inputs	Set of inputs used to explore the black-box
Population	D	Set of inputs	Set of inputs (historical, training, ...)



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399