# ICIE 1.0: A Novel Tool for Interactive Contextual Interaction Explanations

Simon B. van der Zon[1(✉)], Wouter Duivesteijn[1(✉)], Werner van Ipenburg[2(✉)], Jan Veldsink[2(✉)], and Mykola Pechenizkiy[1(✉)]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
{s.b.v.d.zon,w.duivesteijn,m.pechenizkiy}@tue.nl
[2] Coöperatieve Rabobank U.A., Utrecht, The Netherlands
{werner.van.ipenburg,jan.veldsink}@rabobank.nl

**Abstract.** With the rise of new laws around privacy and awareness, explanation of automated decision making becomes increasingly important. Nowadays, machine learning models are used to aid experts in domains such as banking and insurance to find suspicious transactions, approve loans and credit card applications. Companies using such systems have to be able to provide the rationale behind their decisions; blindly relying on the trained model is not sufficient. There are currently a number of methods that provide insights in models and their decisions, but often they are either good at showing global or local behavior. Global behavior is often too complex to visualize or comprehend, so approximations are shown, and visualizing local behavior is often misleading as it is difficult to define what local exactly means (i.e. our methods don't "know" how easily a feature-value can be changed; which ones are flexible, and which ones are static). We introduce the *ICIE* framework (Interactive Contextual Interaction Explanations) which enables users to view explanations of individual instances under different *contexts*. We will see that various contexts for the same case lead to different explanations, revealing different feature interactions.

**Keywords:** Explanations · Feature contributions · Feature interactions · Model transparency · Awareness · Trust · Responsible analytics

## 1 Introduction

Within the domain of banking, black box models are used to predict fraud, money laundering and risk in lending, with their main advantage: speed. However, for real world application of such models, the outcome still has to be augmented by human experts for positive cases, as the bank has to be able to explain to its customers why a particular decision was made.

The standard accounting software packages of this World all provide features to perform what is colloquially known as a "What if? analysis": if we change the


C. Alzate et al. (Eds.): MIDAS 2018/PAP 2018, LNAI 11054, pp. 81–94, 2019.
https://doi.org/10.1007/978-3-030-13463-1_6

value of a cell, how does that affect other cells in my spreadsheet? It is a rather basic technique, and therefore by itself not particularly of interest for data miners. As a research field, we are typically more interested in reverse engineering, a "What happened? analysis". A classifier predicts a specific outcome, but which input attributes contribute most? Which input attributes make it happen? How far must we go in changing the input, in order to change the outcome?

In this paper, we propose to stack the two forms of analysis, to provide a more in-depth analysis of feature interaction and how it affects prediction. We do so by building onto the concept of SHAP contribution values [11]. Introduced in 1953 in the context of cooperative game theory, the original Shapley values [10] provide an answer to the question of how much of the total reward should be awarded to each member of a winning coalition, based on the individual contributions. This concept can be exploited [11] to determine to which degree a certain prediction outcome can be attributed to each individual input attribute.

However, not all real-world interactions can naturally be decomposed into individual contributions. For instance, a family history of clubfoot and maternal smoking individually both have a positive influence on the probability that the offspring displays isolated clubfoot. However, if the family history displays clubfoot, this has a multiplicative effect on the influence of smoking on the probability of clubfoot in the offspring [5]. Any decomposition of the final effect into an additive contribution to single input attributes will necessarily misrepresent what is really going on in this dataset. This is a fundamental problem, that this paper will also not solve.

Instead, our main contribution allows an end user to explore more convoluted interactions. A recent paper [6] introduced SHAP interaction values, which allow a user to find pairs of features which interact differently from their expected additive contributions. Although these values provide correct explanations of the case under observation, they are often still hard to interpret, and only give a limited view of what is actually happening (i.e. in the general context, where each feature is equally important). Instead, we propose to explore wider interactions, in the *ICIE* framework (Interactive Contextual Interaction Explanations). Under this framework, users can test various *contexts* to find explanations featuring unusual attribute interactions (i.e. *contexts* under which the "contextual SHAP values" change). Hence, ICIE enables users to find contexts (the "what if?" part) under which attributes interact unusually with the prediction (the "what happened?" part). Ultimately, this enables analysts to perform a more targeted investigation when verifying positive alerts.

## 2 Related Work

Baehrens et al. [1] propose methods which visualize a limited number of dimensions w.r.t. the class label assigned by the model, for individual classification decisions. Goldstein et al. [4], have a similar approach and observe the average global prediction when modifying these features. The LIME method [8] aims at constructing interpretable models in the vicinity of the case under explanation, and is prone to how exactly this vicinity is defined. Most current works

on explanation of individual predictions use some form of sensitivity analysis to determine the impact of a feature. The EXPLAIN method [9] by Robnik-Šikonja and Kononenko, measures the impact of a removing a feature by comparing the original prediction to the average predictions for all of the feature's possible values. This method however cannot cope with interacting features which cannot impact the classifier's outcome alone (e.g. when $f(x) = a_1 \vee a_2$, for $x = \{a_1 = \text{true}, a_2 = \text{true}\}$, both $a_1$ and $a_2$ need to change in order to see that either one of them has an impact on the outcome, consequently, both will be assigned zero contribution). Štrumbelj et al. address this problem with IME [12], by observing each of the $2^n$ feature subsets (in case of binary classification) and hence has an exponential time complexity. While this method is capable of showing contributions for interacting features, it is not feasible for use on datasets with many attributes and/or attribute values. Štrumbelj et al. follow this up with an approximation algorithm [11] using the Shapley values from cooperative game theory [10], which makes computation feasible for larger domains. Lundberg et al. discuss theory and several properties of these SHAP values [7], provide an algorithm for efficient computation for ensembles of trees, and provide a generalization of the SHAP contributions which enables them to measure interaction between two features [6]. Finally, Martens and Foster provide algorithmic approaches to find explanations for high dimensional document data, in the form of minimal sets of words that change the outcome of the classifier when removed from the document [13].

As noted by Lundberg et al., many current methods for interpreting individual machine learning model predictions fall into the class of *additive feature contribution methods* [7]. This class covers methods that explain a model's output as a sum of real values attributed to each input feature. Additive feature contribution methods have an explanation model $g$ that is a linear function of binary variables: $g(z') = \theta_0 + \sum_{i=1}^{n} \theta_i z'_i$, where $z' \in \{0,1\}^n$, $n$ is the number of input features, and $\theta_i \in \mathbb{R}$. The $z'_i$ variables typically represent a feature being observed or unknown, and the $\theta_i$'s are the feature contribution values.

## 3 Preliminaries

Given a dataset $\Omega$, which is a bag of $N$ records $x \in \Omega$ of the form $x = (a_1, \ldots, a_n, \ell)$, where $\{a_1, \ldots, a_n\}$ are the input attributes of the dataset, taken from some collective domain $\mathcal{X}$, and $\ell \in \{\ell_{\text{pos}}, \ell_{\text{neg}}\}$ is the binary class label. A model $f(x)$ can be trained to predict the class label $\ell$.

To explain a model's decision, SHAP values can be computed for each of the attributes. These values reveal the additive contributions to the model's outcome and hence the sum of these values approximate the predicted class label closely. It is important to note that these values reveal the local contributions for a particular instance, consequently, the contribution for some feature $a_1$ can be positive for an instance $x$, while it can be negative for an instance $x'$ (if feature $a_1$ interacts with one of the changed attributes in $x'$).

**Definition 1 (Additive SHAP contribution).** *Let $\theta_i(x)$ be the additive SHAP contribution value for the $i^{th}$ feature of an instance $x$, i.e.*

$$\theta_i(x) = \sum_{S \subseteq \mathcal{X} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [f_x(S \cup \{i\}) - f_x(S)],$$

*where $n$ is the number of attributes in the feature space $\mathcal{X}$, and $f_x(S)$ is the soft classifier output for model $f$, conditioned only on the features present in the feature subset $S$.*

Note that *negative* SHAP values represent feature-values contributing to the *negative* class and *positive* SHAP values to the *positive* class. Exact computation of the SHAP values is not feasible, but they can be approximated by the sampling algorithm described by Štrumbelj et. al. [11, Algorithm 1].

## 4 The ICIE Method

Providing an overview of only the SHAP contributions gives a limited view of the model's logic. After all, changing one feature, could (potentially completely) change the original contributions. We propose a method to manually explore *contexts* under which the SHAP values are different, and allow users to gain more confidence in for example a fraud alert.

We introduce "context" as our key element of interaction with the user. Context can be defined as a set of constraints, to describe a subspace of the feature space. For now, we restrict ourselves to context that take the form of the well-known descriptions from pattern mining (Definition 2). An instance is either covered by a description, or not. And hence can be used as a natural way to restrict the calculation of the SHAP values to a subspace of the feature space.

**Definition 2 (Description).** *A* description *is a set of constraints, mapping an instance from a domain, to a binary value: $D = x \rightarrow \{true, false\}$.*

SHAP values reflect the contribution of a feature $a_i$ for an instance $x$ and its predicted label $\ell$, and are intuitively computed by taking the average change in soft prediction output for perturbed versions of $x$. For each perturbation the difference in output between a version with $a_i$, and one without $a_i$ is summed. We use the context to restrict this perturbation space. Let this be clarified by an illustrative example. Suppose a model is trained on a dataset about car occasions to predict whether the price will be 'low' or 'high', and we are interested in the model's opinion on an instance $x$. Let $x = \{a_{\texttt{mileage}} = 250.000, a_{\texttt{fuel}} =$ 'gasoline'$\}$, where we are interested in the contribution of $a_{\texttt{mileage}}$. Suppose the model predicts $\ell_{\texttt{price}} =$ 'low', and feature $a_{\texttt{mileage}}$ with SHAP value $\theta_{\texttt{mileage}} = 0.5$ is the largest contributor for this decision. When the same case is now observed under the context of only $\texttt{mileage} \geq 200.000$ cars. We likely observe that the contribution of `fuel` increases, as 'diesel' cars can handle a higher mileage, hence making the fuel type a more important selection criterion. Such interactions are not revealed by the SHAP values. In this work we propose a framework to let an analyst explore such scenarios.

### 4.1 Calculation of Contextual SHAP Values

We calculate SHAP values under a given context similarly to [11, Algorithm 1], with a twist. Instead of selecting a sample at random using $\pi(n)$ to perturb features (the set of all possible feature permutations), we select them from $\pi_D(n)$ (the set of all feature permutations satisfying the context $D$). The effect of the features in the context is amplified by disregarding the feature effects captured by the complement of the context. Formally $\pi_D(n)$ is defined as the set of tuples of the form $(d_1, \ldots, d_e, d_1^{\mathsf{c}}, \ldots, d_{n-e}^{\mathsf{c}}) \in \mathcal{X}$, where $A_D = \{d_1, \ldots, d_e\}$ is the set of attribute indices mentioned in $D$, and $(d_1^{\mathsf{c}}, \ldots, d_{n-e}^{\mathsf{c}}) \in d^{\mathsf{c}}$ is the set of all random permutations of the complement of $A_D$. We refer to a SHAP value $\theta$ under context $D$ by the notation $\theta^D$. Note that under the general context, ICIE values reduce to the regular SHAP values.

### 4.2 UI for Context Exploration

Figure 1 shows the user interface of our application. The next sections discuss its various components. The software consists of two parts: (1) the computation of the classification models (from .csv data files) and (contextual) SHAP values are done on the server (Java); and (2) the user interface runs in the client browser (ECMAScript 6), and sends requests to the server via a MySQL database, where asynchronous Java workers are waiting to answer requests (i.e. computing (contextual) SHAP values).

**SHAP Parameters Controller.** In the top of the screen the parameters controller is shown. A dataset (with corresponding model) can be selected here. An instance from this dataset can be retrieved by inserting its **x.id**. The $m$ corresponds to the SHAP value sampling criteria (from [11]). The first $m$ value is the minimum number of samples drawn for each feature to get an initial estimation for the SHAP value of each feature, the second $m$ number is the maximum number of samples that can be draw (multiplied by the number of features), and is divided based on the expect reduction in variance for the SHAP values. The button **fast/quality** can be used to set these two values to preset values.

**Context Controller.** Allows the user to manipulate the context. We restrict ourselves to contexts that describe subgroups of the data by using simple operators on a subset of features. In Fig. 1 a context representing non-Asian people, younger than 46 with "some" capital gain and education are represented. For ease of use, the **0/1** button next to the close button can toggle the context on/off.

**Feature Contributions View.** In this view, we present the contextual SHAP feature contributions. The user can manipulate the case under investigation here, allowing him/her to observe explanations for variations of the case. With each step in either changing the instance or the context, we highlight the aspects that

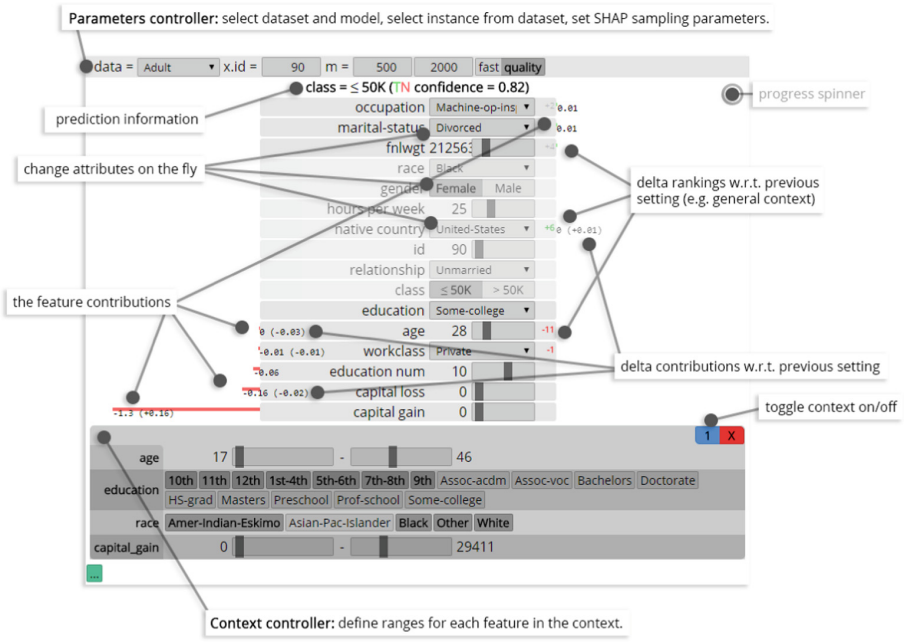



**Fig. 1.** The ICIE user interface. In the top of the screen, in the *parameters controller*, basic domain- and SHAP sampling parameters can be set. In the center of the screen, the feature importances for the current case (based on **x.id**) are shown (red bars on the left side for negative contributors, and green bars on the right for positive contributors). A user can modify the instance on the fly by changing its attributes. In the bottom of the screen the context is shown and can be modified. For each change that the user makes (either in context or the instance), the contribution values are recalculated, and the changes are reflected by the delta contributions (next to each contribution) and the delta rankings (next to the value selector for each attribute). (Color figure online)

change. The difference in contribution value is shown along with the change in ranking of the feature (based on the sorting by contribution).

## 5 Use Cases

We demonstrate our method on six datasets, all taken from the UCI repository [3]. This sample features a mixture of datasets having only binary/nominal or only numeric attributes, as well as datasets mixing attribute types. Characteristics of the used datasets, including statistics on the available attributes per type, can be found in Table 1. For each dataset $\Omega$, a classifier is trained on a random sample of 80% of the data and some common quality measures are reported on the remaining 20% of the data. The models are depth-16 decision trees [2] (characteristics can be found in Table 2). Note that our approach is model agnostic, and we simply use decision trees for practical purposes.

For each dataset we show the additive contributions under the general context, along with a more specific context revealing interacting features (not possible to obtain from the general SHAP values). We comment on the found interactions revealed by the context and explain the process of navigating to the specific context. In the next section, some generalizable remarks on the "navigation process" are discussed.

**Table 1.** Dataset characteristics

| Dataset | $N$ | $n$ | $n_{\text{bin}}$ | $n_{\text{nom}}$ | $n_{\text{num}}$ |
|---|---|---|---|---|---|
| $\Omega_1$ = Adult | 48842 | 14 | 1 | 7 | 6 |
| $\Omega_2$ = Credit-card-default | 3000 | 24 | 0 | 4 | 20 |
| $\Omega_3$ = German-credit | 1000 | 21 | 1 | 13 | 7 |
| $\Omega_4$ = Mushroom | 8124 | 22 | 6 | 16 | 0 |
| $\Omega_5$ = Tic-Tac-Toe | 958 | 9 | 0 | 9 | 0 |
| $\Omega_6$ = Wisconsin | 699 | 9 | 0 | 0 | 9 |

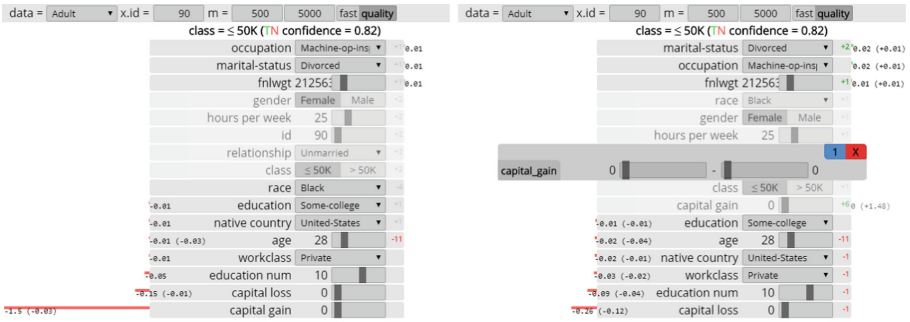
**Table 2.** Model characteristics

| Dataset | Majority | Accuracy | Kappa | Precision | Recall | $F_1$-score |
|---|---|---|---|---|---|---|
| $\Omega_1$ = Adult | 0.92 | 0.83 | 0.45 | 0.91 | 0.91 | 0.91 |
| $\Omega_2$ = Credit-card-default | 0.70 | 0.78 | 0.45 | 0.83 | 0.83 | 0.83 |
| $\Omega_3$ = German-credit | 0.54 | 0.69 | 0.48 | 0.82 | 0.82 | 0.82 |
| $\Omega_4$ = Mushroom | 0.53 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| $\Omega_5$ = Tic-Tac-Toe | 0.63 | 0.85 | 0.69 | 0.82 | 0.82 | 0.82 |
| $\Omega_6$ = Wisconsin | 0.64 | 0.97 | 0.93 | 0.98 | 0.98 | 0.98 |

### 5.1 Adult

This dataset records instances on the annual revenue of individuals and discretizes the individuals in two categories (more than 50k, or less than or equal 50k). For this exploration example we observe the instance with `id` = 90 (the first instance has `id` = 0) from the Adult dataset. Figure 2a shows the "before" situation, and Fig. 2b shows the "after" situation (where the context is limited to `capital_gain=0`). In the before situation, it is clear which attribute is dominating the decision (namely `capital_gain`, with $\theta_{\texttt{capital\_gain=0}} = -1.48$). For this reason, we choose to restrict the context to this particular attribute; the intuition is that now the contributions will be computed only against instances with this same `capital_gain`, hence amplifying the inner effects in that subspace of the data. An unexpected finding reflected by our visualization is that in the general context `age` is the biggest positive contributor (with

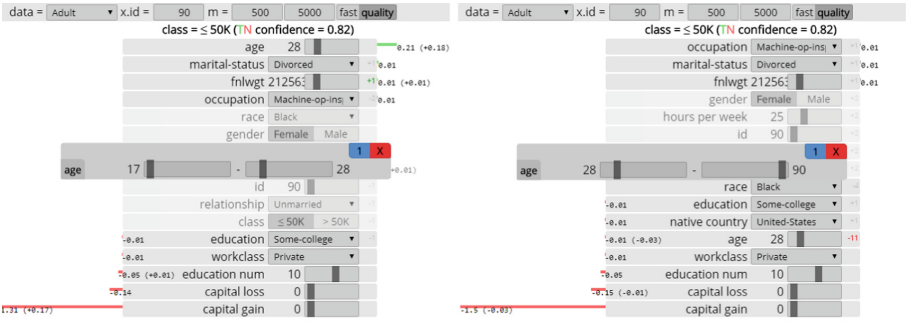
$\theta_{\texttt{age}=28} = 0.02$), whereas in the specific context, `age` becomes a negative contributor (with $\theta^{\texttt{capital\_gain}=0}_{\texttt{age}=28} = -0.02$). This implies an interaction between the two features and can intuitively be interpreted as: generally `age` $= 28$ has a positive impact on the classification, but within the subspace of people with `capital_gain` $= 0$, this particular `age`, contributes negatively.




(a) (b)

**Fig. 2.** Explanations for case 90 from the Adult dataset (we start counting from 0). On the left (Fig. 2a), we see the explanation in the general context, and on the right (Fig. 2b), we see the same explanation, but now under the context $\{\texttt{capital\_gain} = 0\}$.

Another interesting observation for case 90 of the Adult dataset is presented in Fig. 3. The `age` attribute contributes differently depending on the particular `age` ranges we inspect. We observe that the positive contribution of age is amplified by a factor 10 in the context `age` $\leq 28$ (Fig. 3a). Intuitively interpreted as: in the subspace of people with an `age` up to 28, `age` contributes substantially



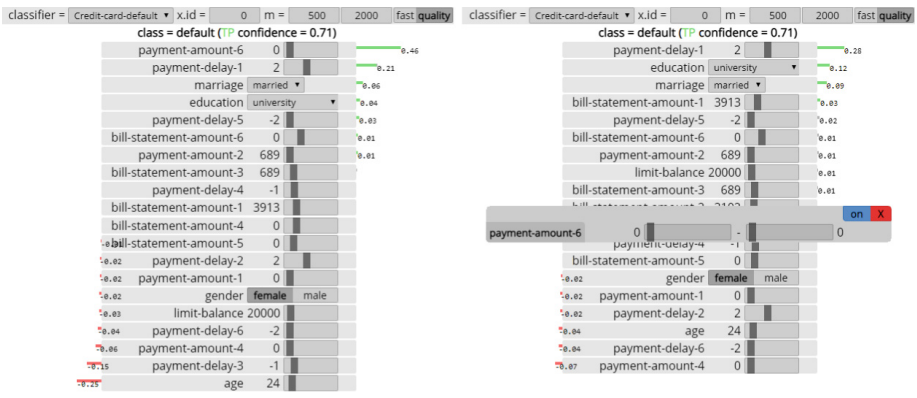

(a) (b)

**Fig. 3.** Explanations for case 90 from the Adult dataset. On the left (Fig. 3a), we see the explanation under the context $\{\texttt{age} \leq 28\}$, and on the right (Fig. 2b), we see the same explanation, but now under the context $\{\texttt{age} \geq 28\}$.

more positively than in the general context (this makes sense as people who are 28 years old in the age group of people of age up to 28, generally make more money than younger ones). When we observe the complement of this context (`age` $\geq 28$), the contribution of `age` is actually inverted (which makes sense for analogous reasons). The general context averages the contributions over the entire age range, where the negative contributions dominate this example, thus sketching a limited view which may be interpreted incorrectly.

### 5.2 Credit-Card-Default

This dataset records credit card clients in Taiwan from April 2005 to September 2005 and are divided in two groups: people who pay duly or people who default on their credit card payments. The model that was trained on this dataset reaches 89% accuracy. Figure 4 presents a client that defaulted on her credit card payment, with biggest contributor `payment-amount-6` $= 0$ (the amount that was payed back six months ago). Note that the contribution of `education` = 'university' has a slight positive impact on defaulting (maybe because this person is actually still in university judging by the age). When inspected under context $\{$`payment-amount-6` $= 0\}$, two interesting observations can be made. Firstly, within the attribute university grows by a factor 3, implying that the within this specific subgroup people in university are apparently more likely to default. Secondly, the importance of the age feature (towards paying duly) gets smaller, meaning that in this subgroup the positive effect of age is less.

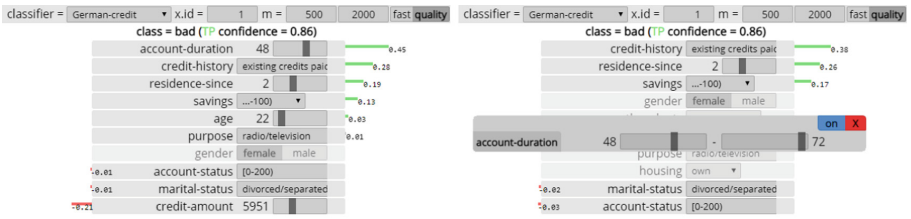



(a) (b)

**Fig. 4.** Explanations for case 0 from the Credit-card-default dataset. On the left (Fig. 4a), we see the explanation in the general context, and on the right (Fig. 4b), we see the same explanation under the context $\{$`payment-amount-6` $= 0\}$.

### 5.3 German-Credit

This dataset records clients from a German bank requesting a loan with a specified amount, purpose and duration for the loan. The clients are divided in two groups: clients who did not manage to pay according to agreement (bad class) and clients who did (good credit class). The model that was trained on this dataset reaches 69% accuracy. Figure 5 presents a client that was assigned the bad credit class, with biggest (bad-class) contributor `account-duration` = 48 (the duration of the loan was 48 months), and as biggest (good-class) contributor `credit-amount` = 5951, which is apparently considered low by the model. If we want to amplify the effects more, we restrict the context to a sub-range of the biggest contributor, namely {`account-duration` $\geq$ 48} (only loans of at least 48 months). We can now observe that under this context the importance for the biggest good-class contributor completely disappears, implying that while in the general setting having this low credit amount is not a risk, in the context of clients with a loan lasting 48 months or longer there actually is an increased risk.




(a) (b)

**Fig. 5.** Explanations for case 1 from the German-credit dataset. On the left (Fig. 5a), we see the explanation in the general context, and on the right (Fig. 5b), we see the same explanation under the context {`account-duration` $\geq$ 48}.

### 5.4 Mushroom

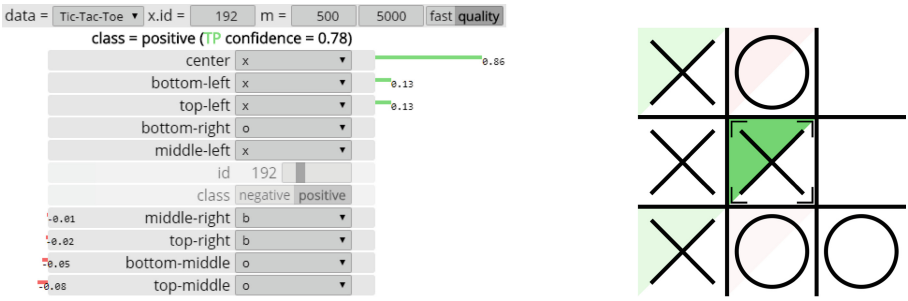
This dataset records mushrooms found in North-America and divides them in two groups: poisonous or edible. The model that was trained on this dataset reaches 99.8% accuracy. Figure 6 presents an edible mushroom with biggest contributor `odor` = 'Almond'. When inspected under context {`odor` = 'Almond'}, two interesting observations can be made. Firstly, there are no negative contributors anymore, implying that this attribute dictates the outcome (when verifying this claim, we indeed find that all 389 mushrooms with `odor` = 'Almond' are edible). Note that this conclusion cannot be drawn from the contributions in the general context. Secondly, `stalk-root` = 'Bulbous' goes from biggest negative contributor (with $\theta_{\texttt{stalk-root}=\text{'Bulbous'}} = -0.14$) to biggest positive contributor (with $\theta^{\texttt{odor}=\text{'Almond'}}_{\texttt{stalk-root}=\text{'Bulbous'}} = 0.13$), telling us "yes" in general `stalk-root` = 'Bulbous' contributes negatively, but when the mushroom smells like almond, the opposite is true.


(a) (b)

**Fig. 6.** Explanations for case 13 from the Mushroom dataset. On the left (Fig. 6a), we see the explanation in the general context, and on the right (Fig. 6b), we see the same explanation under the context {`odor` = 'Almond'}.

### 5.5 Tic-Tac-Toe

This dataset records all possible *end* games for the Tic-Tac-Toe game, with corresponding outcome (either × won, or × did not win; note that both a draw and ◯ wins are counted as negatives). Figure 7 represents a game that was won by ×, with most important move `center` = ×. When observed in this particular context, we find that according to the model × always wins the game when it occupies the `center` (all contributions become 0, meaning that no other attribute influences the game). This is an important finding, as it points out one of the flaws of the model (we can think of many games where × occupies the center, but doesn't win the game).




(a) (b)

**Fig. 7.** Explanations for case 192 from the Tic-Tac-Toe dataset. For the sake of interpretability, we have drawn the board corresponding to this game. On the left (Fig. 7a), we see the explanation in the general context, and on the right (Fig. 7b), we see the same explanation under the context {`center` = ×}, this time visualized for more clarity. The top-left triangles in the cells of the board correspond to the contributions in the general context, and the ones in the bottom-right correspond to the contributions in the context, the (intensity of the) color corresponds to the contribution.

### 5.6 Wisconsin

This dataset records patients diagnosed with breast cancer. The records contain features about images of the (possible) tumor cells. The patients are divided in two groups (malignant and benign). First we report an instance similar to the finding for the Tic-Tac-Toe and Mushroom datasets, namely, where one attribute dictates the outcome. Figure 8 shows the same behavior, but in this case it is even more unclear from the initial visualization that `uniformity-of-cell-size` is actually dictating the prediction, which is reflected by observing it under the context $\{$`uniformity-of-cell-size` $\geq 5\}$.

Next we use our method to inspect a wrong classification and try to find an explanation for the error. Figure 8 shows record 59, which is wrongly classified as 'benign'. In order to find out why the model made this mistake we start limiting the context to the biggest contributor for this case; `uniformity-of-cell-size`. To find out in which "direction" the positive contribution works, we observe the instance under two contexts: $\{$`uniformity-of- cell-size` $\leq 3\}$ and $\{$`uniformity-of-cell-size` $\geq 3\}$. If the former removes the contribution of `uniformity-of-cell-size`, we conclude that this direction can't be used to alter the decision, else we argue the opposite. Figure 8d shows that the former holds, hence we modify `uniformity-of-cell-size` $= 3$ by replacing it with 4. We now see that negative is predicted (Fig. 8e). This may suggest that either

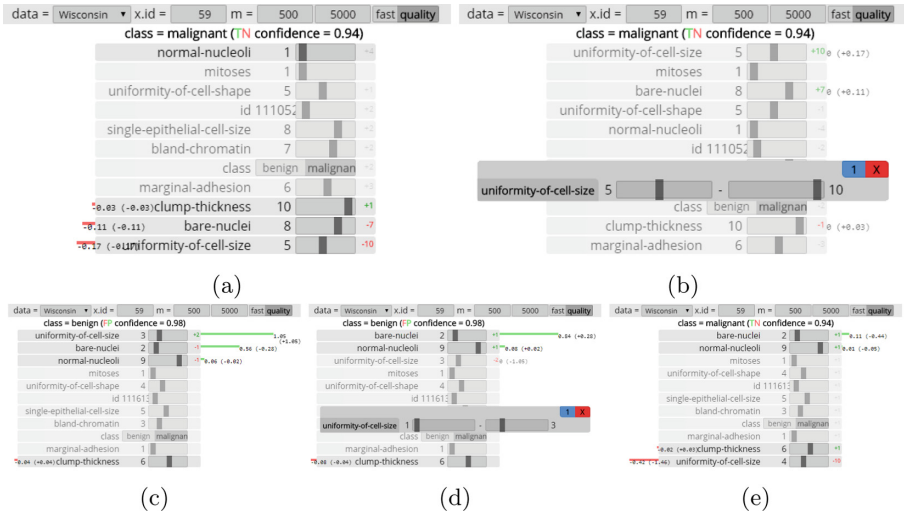



**Fig. 8.** Explanations for cases 50 and 59 from the Wisconsin dataset. On the top-left (Fig. 8a), we see the explanation in the general context, and on the top-right (Fig. 8b), we see the same explanation under the context $\{$`uniformity-of-cell-size` $\geq 5\}$. On the bottom-left (Fig. 8d), we see the explanation in the general context, in the bottom-middle (Fig. 8d), we see the same explanation under the context $\{$`uniformity-of-cell-size` $\leq 3\}$, and on the bottom-right (Fig. 8e), we see the instance where the value for `uniformity-of-cell-size` is replaced by 4.

the model is wrong here or a false measurement was recorded. Note that similar reasoning can be used in an automated search to find the least expensive path to changing the class label.

### 5.7 Guiding the Manual Search for Explanations

When investigating an explanation with our framework it is important to be aware of some basic strategies. It is often a good idea to start investigating the key-players (i.e. top-contributors) first, as it is less likely that contexts consisting of zero-contributors influence the other contributions (however not impossible, e.g. in the case where a zero-contributor is an average of an equal amount of positive contributors as negative contributors). In order to amplify the interactions that are happening *within* the top-contributors, one can set the context equal to its value; restricting the calculation of the SHAP values to this particular feature subspace. For features with large domains (especially numeric feature), the plain SHAP values provide a very limited amount of information, for example, when $\theta_{\texttt{age}=30} = 0.5$, we don't know whether it is increasing or decreasing over the interval of [17–90] (or another interval), or whether its adjacent value $\theta_{\texttt{age}=31}$ would be substantially different or not. By using smartly positioned contexts (usually at these boundaries), we can make deductions from the resulting observations.

## 6 Discussion

Our approach shows to reveal novel information that cannot be obtained by observing the additive SHAP values alone. In particular it can be used to make local feature interactions visible by restricting the context to a dominating feature; to find interacting features (e.g. where one contributes positively in the general context, but negatively in a more specific context), which is of great value when it comes to understanding the classifier and the domain (for accurate classifiers); to help in inspecting wrong predictions of a classifier; and to get insights in the "direction" of the feature contributions when it comes to numeric features.

We are currently exploring strategies to automatically discover interesting contexts, making the manual search less time consuming, and allowing us to find contexts consisting of more features. Ultimately, such tooling helps in answering more involved questions, such as: "given a context, what is the possible/likely adversarial activity leading to a change in this predicted label?" Helping experts in financial domains to target the right sources of information when verifying an alert.

## References


1. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.-R.: How to explain individual classification decisions. J. Mach. Learn. Res. **11**(Jun), 1803–1831 (2010)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. CRC Press, New York (1989)
3. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
4. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. **24**(1), 44–65 (2015)
5. Honein, M.A., Paulozzi, L.J., Moore, C.A.: Family history, maternal smoking, and clubfoot: an indication of a gene-environment interaction. Am. J. Epidemiol. **152**, 658–665 (2000)
6. Lundberg, S.M., Erion, G.G., Lee, S.-I.: Consistent individualized feature attribution for tree ensembles, arXiv preprint arXiv:1802.03888 (2018)
7. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Conference Proceedings on Advances in Neural Information Processing Systems, pp. 4768–4777 (2017)
8. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you? Explaining the predictions of any classifier. In: Proceedings of KDD, pp. 1135–1144 (2016)
9. Robnik-Šikonja, M., Kononenko, I.: Explaining classifications for individual instances. IEEE Trans. Knowl. Data Eng. **20**(5), 589–600 (2008)
10. Shapley, L.S.: A value for n-person games. Contrib. Theory Games **2**(28), 307–317 (1953)
11. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. Knowl. Inf. Syst. **41**(3), 647–665 (2014)
12. Štrumbelj, E., Kononenko, I., Robnik-Šikonja, M.: Explaining instance classifications with interactions of subsets of feature values. Data Knowl. Eng. **68**(10), 886–904 (2009)
13. Martens, D., Foster, P.: Explaining data-driven document classifications. MIS Q. **38**(1) (2014)