Causality-based Explanation of Classification Outcomes

Leopoldo Bertossi*
Universidad Adolfo Ibáñez
Santiago, Chile
& RelationalAI Inc.

Jordan Li†
Carleton University
Ottawa, Canada

Maximilian Schleich
University of Washington
Seattle, USA
& RelationalAI Inc.

Dan Suciu
University of Washington
Seattle, USA
& RelationalAI Inc.

Zografoula Vagena
RelationalAI Inc.
New York, USA

ABSTRACT

We propose a simple definition of an explanation for the outcome of a classifier based on concepts from causality. We compare it with previously proposed notions of explanation, and study their complexity. We conduct an experimental evaluation with two real datasets from the financial domain.

CCS CONCEPTS

• **Explanations**; • **Classification**;

KEYWORDS

explainable AI, classification models, machine learning

ACM Reference Format:
Leopoldo Bertossi, Jordan Li, Maximilian Schleich, Dan Suciu, and Zografoula Vagena. 2020. Causality-based Explanation of Classification Outcomes. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine-learning (ML) models are increasingly used today in making decisions that affect real people's lives, and, because of that, there is a huge need to ensure that the models and their decisions are interpretable by their human users. Motivated by this need, there has bee a lot of interest recently in the ML community in studying *Interpretable models* [18]. There is currently no consensus on what interpretability means, and no benchmarks for evaluating interpretability [5, 10]. The only consensus is that simpler models such as linear regression or decision trees are considered more interpretable than complex models like, say, deep neural nets. However, two general principles for approaching interpretability have emerged in the literature that are relevant to our paper. The first is the idea of simplifying the model. Rudin [18] defines *explanation* as approximating a model so that it becomes interpretable; thus, at a very high level, we will use the term *explanation* to mean a simple piece of information that helps interpreting a model. Citing Doshi-Velez and Kim [5][1]: *explanations are . . . the currency in which we exchanged beliefs*. The second is the idea that a good notion of explanation should be grounded in *causality* [15]. This idea has frequently been mentioned in the literature, but no consensus exists on how to convert it into a formal definition.

There are two levels at which one can provide explanations [12, 16]. A *global explanation* aims at explaining the model as a whole, while a *local explanation* concerns a particular outcome, i.e. a single decision. For example, consider a bank that uses a machine learning model to decide whether to grant loans to individual customers. Then the global explanation concerns the entire model, for example it explains it to a developer or an auditor, while, in contrast, a local explanation concerns a single decision, for an individual customer. For example the customer applies for a loan, the bank runs the model, and outcome is to deny the loan, and the customer asks *why?* The bank needs to provide an explanation. In this paper we are concerned only with local explanations. We argue that they are of particular interest to the data management community, because they need to be provided interactively, and they often require processing a large amount of data, for example to compare the current customer with the entire population of the bank's customers.

The golden standard for explanations are *black-box explanations* [12, 16], which are independent of the inner workings of the classifier. LIME [16] explains an outcome by learning an interpretable model locally around the outcome; SHAP [12, 13] explains an outcome by modeling it as a Shapley cooperative game and assigning a score to each feature. However, lacking a common benchmark for evaluating the *quality* of the explanation, researchers are also considering *white-box* explanations, based on the intuition that knowledge of the inner workings of the model can help better explain a particular outcome. For example, a successful explanation framework has been developed by Chen et al. [3] for the Credit Risk assessment problem, by co-designing the model and the explanation.

The goal of this paper is three-fold. First, we introduce a black-box notion of explanation based on causality. Our definition extends the simplified notion of *actual cause* and *responsibility* described

*Member of the "Millenium Institute for Foundational Research on Data (IMFD, Chile)
†Work done as an intern at RelationalAI.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1]The attribute this quote to Lombrozo.

in [14] to a new black-box explanation score, called RESP. Unlike SHAP, which is grounded in cooperative games, RESP is grounded in causality. Second, we examine the data-management challenge of computing black box explanations. Both RESP and SHAP explanations require access to a probability distribution over the population. We consider two probability spaces, the *product space* and the *empirical space*, and show that computing SHAP is #P-hard in the former, while RESP is meaningless in the latter. Our finding suggests that future research needs to focus on designing realistic, yet tractable probability spaces, in order to support the efficient computation of black-box explanations. Finally, we conduct an empirical evaluation of the quality of black-box explanations. We compare both RESP and SHAP to the white-box explanation for the Credit Risk problem [3] and find that RESP is very close to the white-box explanation. On closer inspection of the cases where the two explanations differ reveals that the reason is that the white-box explanation is based only on the current entity, and ignores the population as a whole; in contrast, RESP is based on causality and, thus, takes into account counterfactual outcomes, which can be computed by examining the entire population. We also compare RESP and SHAP on a second dataset (where no white-box explanation is available) and measure their sensitivity to bucketisation.

In this paper we only consider *feature-based explanations*, which rank the features by some score, representing their importance for a particular outcome. This is similar to previous work on SHAP [12, 13] and to [3]. Feature-based explanations are attractive because they very simple. Other approaches have been proposed in the literature: LIME [16] provides as explanation a simple model learned locally around the outcome, Goyal et al. al. [7] and Ghorbani et al. [6] introduced *concept-based explanations*, while Khanna et al. [9] provide explanations consisting of items in the training data.

In summary, this paper makes the following contributions:

- We define the explanation problem for a black-box classifier; Sec. 2.
- We introduce RESP, a black-box explanation score based on causality. Sec. 3
- We describe the formal connection between RESP and SHAP, showing that, while they are somewhat correlated, they are quite different. Sec. 4.
- We establish the computational complexity of RESP and SHAP over two simple probabilistic models of the population. Sec. 5.
- We conduct an extensive experimental evaluation. Sec. 6

2 PROBLEM DEFINITION

Fix a set of features $\{F_1, \ldots, F_n\}$, with finite domains $Dom(F_i)$. We assume to have a classification model denote by L; given values $x_1, \ldots, x_n$ in the corresponding domains, the classifier returns a value $L(x_1, \ldots, x_n) \in \{0, 1\}$. The inner workings of the classifier are irrelevant to us. It may be a decision tree, or a boosted model, or a deep neural network, or any other program that given input features, returns a 0 or a 1. We will only assume to have access to a black-box computing L, given values of the features. Our discussion also applies to the case when L returns a continuous value in $\mathbb{R}$, but for simplicity we will assume its value is in $\{0, 1\}$.

We consider the scenario in which L is applied to an entity $\mathbf{e}$, for example an individual customer petitioning for a loan. We identify the entity by its features, $\mathbf{e} = \langle x_1, \ldots, x_n \rangle$, and denote the classifier's output by $L(\mathbf{e})$. Our goal is to provide an explanation for this output $L(\mathbf{e})$. Throughout the paper we will assume that the output $L = 0$ represents the "good" outcome while $L = 1$ represents the "bad" outcome, for example, $L(\mathbf{e}) = 0$ means that the loan is granted, while $L(\mathbf{e}) = 1$ means the loan is denied. We are mainly interested in explaining the outcome $L(\mathbf{e}) = 1$, but our discussion also applies to entities with the good outcome 0. We consider only *feature-based* explanations, which assign a score to each feature F_i, and return as explanation the feature with the highest score, or a small set of features with highest scores. For example, if the explanation is the feature `NumberOfInquiresLast7Days`, then we would tell the customer "your loan was denied because of the number of inquires to your credit history in the last 7 days". Banks often deny loans when there are a large number of recent inquires to the customers' credit history, in order to prevent customers from quickly opening multiple credit accounts at independent banks.

In order to produce meaningful explanations, we will assume that, in addition to the current entity $\mathbf{e}$ and the black box classifier L we also have access to a probability space Ω on entities. For example this can be the known distribution of the population of customers applying for loans. A meaningful explanation should be informed by typical customers over that population, as we explain below. We will write $\mathbf{e} \sim \Omega$ to denote the fact that $\mathbf{e}$ is chosen at random from the space Ω.

In this paper we study two black-box explanation scores. The first is our new proposal, called the RESP-score, and is grounded in the principle of *actual cause*. The second score, SHAP-score, has been proposed recently [13], and is based on the Shapley value of a cooperative game. Both are black-box explanations, in that they are oblivious to the inner workings of the classifier L. We evaluate empirically these two explanations, and compare them to a white-box explanation defined by Chen et al. [3], for a specific classification task.

Throughout the paper we denote the set of n features by $\mathcal{F}$. For any subset $S \subseteq \mathcal{F}$ we denote by $\mathbf{e}_S$ the restriction of an entity $\mathbf{e}$ to the features in S.

3 THE COUNTER-SCORE AND RESP-SCORE

We introduce here our novel black-box explanation score, by applying the principle of causal reasoning. Halpern and Pearl [8] give a formal definition of an actual cause, while Chockler and Halpern extend it to degree of causality, called responsibility [4]. Both notions were simplified and adapted to database queries [14]. We review them here, then adapt them to feature-based explanations.

Fix an entity $\mathbf{e}^\star$, and assume its outcome is $L(\mathbf{e}^\star) = 1$. A *counterfactual cause* [8, 14] is a feature F_i and a value v such that $L(\mathbf{e}) = 0$, where $\mathbf{e} \stackrel{\text{def}}{=} \mathbf{e}^\star[F_i := v]$ is the entity obtained from $\mathbf{e}^\star$ by setting $F_i = v$ and keeping all other values unchanged. Thus, (F_i, v) is a counterfactual cause, if, by changing only F_i to v, the outcome changes from 1 to 0, i.e. from "bad" to "good". Notice that the notion of a counterfactual applies to the *pair* (F_i, v), i.e. the value is important.

We adapt this notion to define the COUNTER-score of a feature F_i (without any value). The COUNTER-score is defined as the expected counterfactual causality over the random choices of the values v:

Definition 3.1. Fix an entity $\mathbf{e}^\star$. The COUNTER-score of a feature F_i is:

$$\text{COUNTER}(\mathbf{e}^\star, F_i) \stackrel{\text{def}}{=} L(\mathbf{e}^\star) - \mathbf{E}\left[L(\mathbf{e}) | \mathbf{e}_{\mathcal{F}-\{F_i\}} = \mathbf{e}^\star_{\mathcal{F}-\{F_i\}}\right]$$

Here, the conditional expectation is taken over the random entity $\mathbf{e} \sim \Omega$ conditioned on having the same features as $\mathbf{e}^\star$ except for F_i.

To see the intuition behind our definition, fix some value v, and let $\mathbf{e} = \mathbf{e}^\star[F_i := v]$. If (F_i, v) is a counterfactual cause, then the difference $L(\mathbf{e}^\star) - L(\mathbf{e})$ is 1; otherwise the difference is 0. The COUNTER-score is simply the expected value of this difference.

We explain now the importance of the probability space Ω, over which we take the expectation $\mathbf{E}[\ldots]$. Suppose a bank is deciding loan applications for customers in the US. A customer, Alice, was denied the loan, and she asks for an explanation. After examining millions of prior customers, the clerk notices that one customer had exactly the same features as Alice, but was born in Luxembourg, and was granted the loan. Luxembourg is a tiny state in Europe whose population is financially very savvy, which explains why that loan was granted while Alice was denied. Thus, place-of-origin=Luxembourg is a counterfactual cause. However, it is a poor explanation, because it is not representative of the population for Alice. The definition of COUNTER-score takes this into account: COUNTER(`Alice`, `place-of-origin`) is very small, because the vast majority of customers identical to Alice except for place-of-origin were also denied the loan application. Thus, our bank will provide Alice with a different explanation, one having a larger COUNTER-score.

It is possible for an entity $\mathbf{e}^\star$ to have no counterfactual cause. This happens when, changing any single feature to any other value, the modified entity $\mathbf{e}$ has the same outcome $L(\mathbf{e}^\star) = L(\mathbf{e}) = 1$. A pair (F_i, v) is called an *actual cause* with contingency $(\Gamma, \mathbf{w})$, where Γ is a set of features and $\mathbf{w}$ is a set of values, if (F_i, v) is a counterfactual cause for $\mathbf{e}^\star[\Gamma := \mathbf{w}]$ [8, 14]. In other words, denoting $\mathbf{e}' \stackrel{\text{def}}{=} \mathbf{e}^\star[\Gamma := \mathbf{w}]$ and $\mathbf{e}'' \stackrel{\text{def}}{=} \mathbf{e}'[F_i := v]$, the pair (F_i, v) is an actual cause for $\mathbf{e}^\star$ with contingency $(\Gamma, \mathbf{w})$ iff $L(\mathbf{e}^\star) = L(\mathbf{e}') = 1 \neq L(\mathbf{e}'') = 0$. For an illustration, suppose Alice is 30 years old and denied the loan. Setting `NumberOfInquiresLast7Days = 0` will not change the outcome, however, if Alice where 5 years older, then setting `NumberOfInquiresLast7Days = 0` would grant Alice the loan. Thus, `NumberOfInquiresLast7Days = 0` is an actual cause with contingency set (`Age`, 35). Chockler and Halpern [8] defined the *responsibility* of an actual cause (F_i, v) with contingency $(\Gamma, \mathbf{w})$ as $1/(1 + |\Gamma|)$; intuitively, smaller Γ results in a larger responsibility, in particular a counterfactual cause has responsibility 1, because then $\Gamma = \emptyset$.

We introduce now the RESP-explanation score.[2]

Definition 3.2. Fix an entity $\mathbf{e}^\star$, and a contingency $(\Gamma, \mathbf{w})$ such that $L(\mathbf{e}^\star) = L(\mathbf{e}')$, where $\mathbf{e}' \stackrel{\text{def}}{=} \mathbf{e}^\star[\Gamma := \mathbf{w}]$. The RESP-score of a feature F_i w.r.t. to the contingency $(\Gamma, \mathbf{w})$ is:

$$\text{RESP}(\mathbf{e}^\star, F_i, \Gamma, \mathbf{w}) \stackrel{\text{def}}{=} \frac{L(\mathbf{e}') - \mathbf{E}\left[L(\mathbf{e}'') | \mathbf{e}''_{\mathcal{F}-\{F_i\}} = \mathbf{e}'_{\mathcal{F}-\{F_i\}}\right]}{1 + |\Gamma|}$$

We define $\text{RESP}(\mathbf{e}^\star, F_i)$ (without $\Gamma, \mathbf{w}$) as $\max_{\mathbf{w}} \text{RESP}(\mathbf{e}^\star, F_i, \Gamma, \mathbf{w})$, where Γ is the smallest set for which this score is $\neq 0$.

In other words, the RESP-explanation score is defined as follows. We first try setting $\Gamma = \emptyset$, in which case $\text{RESP}(\mathbf{e}^\star, F_i, \Gamma, \emptyset) = \text{COUNTER}(\mathbf{e}^\star, F_i)$. If this is non-zero, then its value defines the RESP-score. If it is zero, then we increase Γ, until we find a contingency such that $\text{RESP}(\mathbf{e}^\star, F_i, \Gamma, \mathbf{w}) \neq 0$; this value represents the RESP-score.

4 THE SHAPLEY-SCORE

Motivated by machine learning applications in the medical domain, Lundberg and Lee [13] have proposed the Shapley explanation score of a feature F_i, in short SHAP-score. This score is not grounded in causality, but instead it is based on the *Shapley value* of cooperative games [17, 19].[3] We review it here briefly.

Fix an entity $\mathbf{e}^\star$ and a feature F_i. Let π be a permutation on the set of features $\mathcal{F}$; in other words, π fixes a total order on the set of features. Denote by $\pi^{<F_i}$ the set of features F_j that come before F_i in the order π; similarly, $\pi^{\leq F_i}$ denotes $\pi^{<F_i} \cup \{F_i\}$. The *contribution* of the feature F_i is defined as:

$$c(\mathbf{e}^\star, F_i, \pi) \stackrel{\text{def}}{=} \mathbf{E}\left[L(\mathbf{e}) | \mathbf{e}_{\pi^{\leq F_i}} = \mathbf{e}^\star_{\pi^{\leq F_i}}\right] - \mathbf{E}\left[L(\mathbf{e}) | \mathbf{e}_{\pi^{<F_i}} = \mathbf{e}^\star_{\pi^{<F_i}}\right]$$

Definition 4.1. Fix an entity $\mathbf{e}^\star$. The SHAP-score of a feature F_i is the average contribution of F_i over all permutations π, in other words:

$$\text{SHAP}(\mathbf{e}^\star, F_i) \stackrel{\text{def}}{=} \frac{1}{n!} \sum_\pi c(\mathbf{e}^\star, F_i, \pi)$$

where $n = |\mathcal{F}|$ is the number of features.

The intuition is as follows. Extend the classifier L to entities with missing features, as follows. If $\mathbf{e}^\star_S$ is an entity with features $S \subseteq \mathcal{F}$, then define L' to be the expected value over the missing features: $L'(\mathbf{e}^\star_S) \stackrel{\text{def}}{=} \mathbf{E}[L(\mathbf{e}) | \mathbf{e}_S = \mathbf{e}^\star_S]$. In particular, when *all* features are missing, then $L'(\emptyset) = \mathbf{E}[L(\mathbf{e})]$, and when all features are present then $L'(\mathbf{e}^\star) = L(\mathbf{e}^\star)$. Consider the following process: we present the features of $\mathbf{e}^\star$ to the classifier L' one by one, in some order π. The output of L' changes, step by step, from $\mathbf{E}[L(\mathbf{e})]$ to $L(\mathbf{e}^\star)$. The contribution $c(\mathbf{e}^\star, F_i, \pi)$ of the feature F_i represents the amount of change observed when we introduce F_i. The SHAP-score simply averages this contribution over all permutations π.

The appeal of the SHAP-score is that it splits the difference $L(\mathbf{e}^\star) - \mathbf{E}[L(\mathbf{e})]$ among the n features $F_1, \ldots, F_n$, in other words:

$$\sum_i \text{SHAP}(\mathbf{e}^\star, F_i) = L(\mathbf{e}^\star) - \mathbf{E}[L(\mathbf{e})] \tag{1}$$

This follows immediately from the fact that, for any fixed permutation π, $\sum_i c(\mathbf{e}^\star, F_i, \pi) = L(\mathbf{e}^\star) - \mathbf{E}[L(\mathbf{e})]$. Several good properties

[2] In [2] contingency sets for causes at the attribute- value level for query answers from databases were defined along similar lines.

[3] In [11] Shapley scores have been assigned to database tuples to quantify their contribution to query results.

of the SHAP-score are discussed in [12]. However, unlike the RESP-score, there is no causal semantics associated to the SHAP-score. In particular, it is possible for the SHAP-score of some feature to be < 0, a fact that we observed in our experiments.

We now explain the connection between the SHAP-score and the causality-based scores introduced in Sec. 3. Fix a set $S \subseteq \mathcal{F} - \{F_i\}$ and define the contribution of F_i w.r.t. S as:

$$c'(\mathbf{e}^\star, F_i, S) \stackrel{\text{def}}{=} \mathbf{E}\left[L(\mathbf{e})|\mathbf{e}_{S\cup\{F_i\}} = \mathbf{e}^\star_{S\cup\{F_i\}}\right] - \mathbf{E}\left[L(\mathbf{e})|\mathbf{e}_S = \mathbf{e}^\star_S\right]$$

For a number $0 \le \ell < n$, denote by $\binom{\mathcal{F}-\{F_i\}}{\ell}$ the subsets of size ℓ of $\mathcal{F} - \{F_i\}$. We define the SHAP-score at level ℓ as

$$\text{SHAP}(\mathbf{e}^\star, F_i, \ell) \stackrel{\text{def}}{=} \frac{\ell!(n-\ell-1)!}{n!} \sum_{S\in\binom{\mathcal{F}-\{F_i\}}{\ell}} c'(\mathbf{e}^\star, F_i, S) \quad (2)$$

It is immediate to check that the SHAP-score is the sum of all n levels, $\text{SHAP}(\mathbf{e}^\star, F_i) = \sum_{\ell=0,n-1} \text{SHAP}(\mathbf{e}^\star, F_i, \ell)$. Using this property, we prove the following connection between the SHAP-score and the causality-based scores:

Lemma 4.2. $\text{SHAP}(\mathbf{e}^\star, F_i, n-1) = \frac{1}{n}\text{COUNTER}(\mathbf{e}^\star, F_i)$.

The proof follows immediately from the definitions. This "connection" reveals more about how different the RESP- and SHAP-scores are. Level $n-1$ is only one of the many levels of the SHAP-score, while the RESP-score agrees with the COUNTER-score only when the contingency is empty. Thus, the RESP- and SHAP-scores are derived from different principles (causality and Shapley value respectively) and, while somewhat correlated, have different mathematical definitions.

5 PROBABILITY SPACES AND ALGORITHMS

A key difficulty in computing the RESP- and SHAP-scores consists in defining the probability space Ω. In practice we do not have access to the population defining Ω, but only to a sample, for example the training data, or the test data T. The RESP- and SHAP-scores require the computation of many conditional expectations, and it is not possible to estimate them from a sample T. In this paper we propose two simple approaches to defining Ω, and study how the RESP- and SHAP-scores can be computed in each case.

In this section we assume to have a dataset T with N tuples and $n+1$ attributes, $F_1, \ldots, F_n, C$, where each row represents an entity $\mathbf{e}$ and C represents a count, i.e. the number of times $\mathbf{e}$ occurs in the sample. We denote by $D_i \stackrel{\text{def}}{=} \Pi_{F_i}(T)$, the domain of the feature F_i. Our goal is to define a probability space Ω over $D_1 \times \cdots \times D_n$ that is a generative model for T, over which we compute the RESP- and SHAP-scores.

5.1 The Product Space

Let $p(F_i = x)$ be the observed marginal probability of the value $F_i = x$ in the data T. The *product space*, Ω_P, is defined by choosing all feature values independently:

$$p(\langle x_1, \ldots, x_n\rangle) \stackrel{\text{def}}{=} \prod_i p(F_i = x_i)$$

The advantage of the product space is that it covers the entire domain $D_1 \times \cdots \times D_n$, and it preserves the marginal probabilities of each feature. On the negative side, Ω_P does not capture any correlations.

```
-- create views from all domains:
create view D1 as select F1, sum(C) from T group by F1;
create view D2 as select F2, sum(C) from T group by F2;
...

-- Compute the RESP-score of F3:
-- starting with Γ = ∅, compute the query below
-- then increase Γ until the answer is ≠ 0

-- We show here only Γ = {F1, F4}
with temp as
  (select y.F1, z.F4,
     L(y.F1, F2*, x.F3, z.F4, F5*, ...)*sum(x.C)/M as S
  from D1 y, D4 z, D3 x
  group by y.F1, z.F4
  having L(y.F1, F2*, F3*, z.F4, F5*, F6*, ...)=1)
select max(1-S)/3 from temp;
```

Figure 1: Illustration of how to compute the RESP-score over the product domain Ω_P using SQL queries. T is the input data, $M \stackrel{\text{def}}{=} \sum_{\mathbf{e}\in T} sum(\mathbf{e}.C)$ is a constant, and $L(\cdots)$ is a User Defined Function (the black-box classifier). The input entity $\mathbf{e}^\star$ given by the constants $F1^\star, F2^\star, \ldots$, and we assume $L(F1^\star, F2^\star, \ldots) = 1$. The figure shows only how to compute the RESP-score for F_3, with contingency F_1, F_4; the general case is similar. The division by 3 represents the division by $1 + |\Gamma|$.

Score Computation Algorithm We start by showing how to compute the RESP-explanation. For that need to compute the RESP-score of each feature, then return the feature with highest score. For a fixed feature F_i, we start by computing the COUNTER-score, by applying Def. 3.1 directly; we need to perform single scan over the data T and at most $|T|$ calls to the oracle L, because:

$$\mathbf{E}\left[L(\mathbf{e})|\mathbf{e}_{\mathcal{F}-\{F_i\}} = \mathbf{e}^\star_{\mathcal{F}-\{F_i\}}\right] = \sum_{x\in D_i} L(\mathbf{e}^\star[F_i := x])p(F_i = x)$$

In practice, most often COUNTER $\neq 0$ and then this is the RESP-score of the feature. Otherwise, we consider contingency sets of size 1, 2, ... until we find a non-zero score. Fig. 1 sketches this computation for the case when the feature F_i is F_3, and the contingency set is F_1, F_4.

We now turn to the task of computing the SHAP-score. Unfortunately, the SHAP-score is intractable in this model: we show that it is #P-hard, even if the classifier L is given as a white-box:

Theorem 5.1. *Suppose all features are binary, $F_i \in \{0,1\}$. Then the following problem is #P-hard:* "Given a classifier L specified as a monotone 2CNF function with variables $F_1, \ldots, F_n$, and an entity $\mathbf{e}^\star \in \{0,1\}^n$, compute the SHAP-scores of all its features".

Proof. The following problem is known to be #P-complete [20]: *"Given a monotone 2CNF Boolean formula $L = \bigwedge_{(i,j)\in E}(F_i \vee F_j)$ (where $E \subseteq [n]\times[n]$), compute the number of truth assignments #L".* It follows immediately that the following problem is also #P-hard:

```
VALUES := ∅
forall non-empty S ⊆ {F1, ..., Fn}, in increasing size:
if S ∉VALUES then
   compute V:= E[L(e)|e_S = e*_S] (one pass over T)
   insert (S,V) in VALUES
   if V= 0 or V= 1 then
      forall supersets S' ⊇ S do:
      insert (S',V) in VALUE
```

Figure 2: Computation of the SHAP-score. We show only how to compute all conditional expectations $E[L(\mathbf{e})|\mathbf{e}_S = \mathbf{e}^\star_S]$; these values are then used in Eq.(2). The algorithm computes the conditional for sets S of increasing cardinality, and inserts the results in VALUES. If some value is either 0 or 1, then all supersets of S will have the same conditional expectation, and we insert all of them in VALUES.

compute the probability $p(L) \stackrel{\text{def}}{=} \#L/2^n$, when each variable F_i is set independently to 1 with probability 1/2. We describe polynomial-time Turing reduction from the SHAP-scores computation problem to the probability computation problem. Let L be any monotone 2CNF L for which we want to compute $p(L)$. Consider the following dataset T with two entities: $\langle 0, 0, \ldots, 0\rangle$ and $\langle 1, 1, \ldots, 1\rangle$, where the count is $C = 1$ for both entities. Then each marginal probability is $p(F_i = 0) = p(F_i = 1) = 1/2$, and the product space is precisely that in which we want to compute $p(L)$, i.e. all Boolean variables F_i are set independently to true with probability 1/2. Consider the input entity $\mathbf{e}^\star = \langle 1, 1, \ldots, 1\rangle$ (i.e. all variables are set to 1), thus $L(\mathbf{e}^\star) = 1$ (because the formula is monotone). With n calls to the oracle for SHAP, we can obtain $p(L)$ as follows: $p(L) = \mathbf{E}[L(\mathbf{e})] = L(\mathbf{e}^\star) - \sum_i \text{SHAP}(\mathbf{e}^\star, F_i) = 1 - \sum_i \text{SHAP}(\mathbf{e}^\star, F_i)$, by (1). □

Thus, we cannot hope to compute the SHAP-score over the product space Ω_P. Instead, we consider a second probability space.

5.2 The Empirical Distribution

The *empirical distribution* defined by T is simply T itself. In other words, the outcomes with non-zero probability are precisely the tuples in T, and the probabilities are given by their frequencies in T. We denote Ω_E the empirical distribution defined by the set T. One advantage of the empirical distribution is that it captures not only the marginal probabilities, but also the correlations between features. One disadvantage of the empirical distribution is that it associates a zero probability to every unseen entity.

Score Computation Algorithm We start by describing how to compute the SHAP-score over the probability space Ω_E, and for that we use Eq.(2). To apply this formula, we compute the conditional expectation $E[L(\mathbf{e})|\mathbf{e}_S = \mathbf{e}^\star_S]$ for each set of features $S \subseteq \{F_1, \ldots, F_n\}$. Each conditional expectation requires a complete pass over the data T, and this becomes impractical when n is larger than 20 or so. We propose to use an optimization borrowed from the apriori algorithm [1]. As the set S increases, the set of entities $\mathbf{e} \in T$ that satisfy $\mathbf{e}_S = \mathbf{e}^\star_S$ decreases, until it becomes a singleton $\{\mathbf{e}\}$. At that point, for every superset S' the conditional expected value is the same $L(\mathbf{e})$, and hence we can stop increasing the set S. In fact, we can stop even earlier: when all entities in the set $\{\mathbf{e} \mid \mathbf{e}_S = \mathbf{e}^\star_S\}$ have the same outcome $L(\mathbf{e})$ (either 0 or 1). While the worst case runtime of the algorithm is still exponential in n, in practice this optimization is quite effective and we were able to compute the SHAP-score for up to $n = 30$ attributes. We sketch the algorithm in Fig. 2.

While it is possible to compute the RESP-score in this model too, unfortunately this score is meaningless: for most entities $\mathbf{e}^\star$ the RESP-score will require a very large contingency set, leading to meaningless explanations. To see the intuition behind this, fix the set T and consider a new entity $\mathbf{e}^\star$, not necessarily in T. For example, T is our training set, while $\mathbf{e}^\star$ is a new, random customer, not present in the training set. In order to have a non-zero RESP-score with empty contingency set, we need to find some entity $\mathbf{e} \in T$ that agrees in all features, except one, with $\mathbf{e}^\star$: this is very unlikely given a random choice for $\mathbf{e}^\star$. Extending this simple observation we prove:

Theorem 5.2. *Let L be any classifier, and let $T \subseteq D_1 \times \cdots \times D_n$ be a set of size N that defines the empirical probability space Ω_E. Fix an integer $c \geq 0$. Then, for a randomly chosen entity $\mathbf{e}^\star$, the probability (over the random choices of $\mathbf{e}^\star$) that $\text{RESP}(\mathbf{e}^\star, F_j, \Gamma) \neq 0$, for some feature F_j and some contingency set Γ of size $\leq c$, is $\leq N(1 + \sum_j |D_j|)^{c+1}/\prod_j |D_j|$.*

Proof. If $\text{RESP}(\mathbf{e}^\star, F_j, \Gamma) \neq 0$ for some feature F_j and some contingency Γ, then there must exists two entities $\mathbf{e}', \mathbf{e}'' \in T$ such that (a) $\mathbf{e}^\star$ and $\mathbf{e}'$ agree on all features except Γ, (b) $\mathbf{e}'$ and $\mathbf{e}''$ agree on all features except F_j, and (c) $L(\mathbf{e}^\star) = L(\mathbf{e}') \neq L(\mathbf{e}'')$. In particular, (a) and (b) imply that $\mathbf{e}^\star$ satisfies the following property:

$$\exists \mathbf{e}'' \in T: \ \mathbf{e}^\star \text{ and } \mathbf{e}'' \text{ agree on all but } c+1 \text{ features} \qquad (3)$$

We claim that its probability is $\leq N(1 + \sum_j |D_j|)^{c+1}/\prod_j |D_j|$, which proves the theorem. To prove the claim, start by fixing an entity $\mathbf{e}'' \in T$. Consider a set of features $(F_j)_{j\in J}$, for some set $J \subseteq [n]$: the probability that a randomly chosen entity $\mathbf{e}^\star$ agrees with $\mathbf{e}''$ on all features except those in J is $1/\prod_{k\in[n]-J} |D_k| = \prod_{j\in J} |D_j|/\prod_{k\in[n]} |D_k|$. By the union bound, the probability that a randomly chosen $\mathbf{e}^\star$ agrees with $\mathbf{e}''$ on *any* set of features J, with size $|J| \leq c$, is: $\leq \sum_{J:|J|\leq c} \prod_{j\in J} |D_j|/\prod_{k\in[n]} |D_k| \leq (1 + \sum_j |D_j|)^{c+1}/\prod_j |D_j|$. Finally, the claim follows from the union bound applied to $\mathbf{e}'' \in T$. □

In essence, the theorem says we are very unlikely to find good RESP-explanations using the empirical space. For a concrete example, assume $n = 30$ features, each with a domain of size $|D_i| = 10$, and a test data T with $N = 10000$ entities. We are interested in the probability that a randomly chosen $\mathbf{e}^\star$ has non-zero RESP-score with a contingency set of size $\leq c$. By the theorem, this probability is $\leq 10000 \cdot (1 + 300)^{c+1}/10^{30} \approx 300^{c+1} \cdot 10^{-26}$. When $c = 8$, this quantity is $2 \cdot 10^{-5}$. It is *very* unlikely that a randomly chosen entity will have a explanation with a contingency of size 8 or smaller. On the other hand, explanations with contingency sets 8 or larger become meaningless. In other words, the empirical distribution is not practical for computing the RESP-score.

5.3 Discussion

In this paper we will use the marginal probability space Ω_P when computing the RESP-score, and will use the empirical probability

space Ω_E when computing the SHAP-score. We are forced with this choice by our two results above. One of the main take-aways of this paper is that future work on explanation needs to explore more sophisticated choices for the probability space Ω.

6 EXPERIMENTAL EVALUATION

In this section, we empirically evaluate the RESP and SHAP-scores on two datasets. First, we consider a FICO dataset and evaluate how the black-box RESP and SHAP-explanations compare to a well-known white-box explanation for FICO data. Then, we consider a Kaggle dataset used to detect credit card fraud, for which the explanations require commonly used data transformations. We evaluate how robust the scores are to these transformations.

Evaluation Setup. All experiments are implemented in Python 3.7, and performed on an Intel i7-4770 3.40GHz/64bit/32GB with Linux 3.13.0. We use the Pandas library for the data-management optimizations.

For the RESP-score, we restrict the size of the contingency sets to at most 1. If an entity does not have an non-zero RESP-score for contingency sets of size 1, we return no explanation for this entity.

Datasets. We next overview of the two datasets used for the evaluation.

FICO Dataset. We first consider the dataset from the public FICO challenge. The objective of the challenge is to provide explanations for credit risk assessments.[4]

The dataset consists of 23 continuous features and 10,459 entities. The features are shown in the left column of Table 1. The dependent variable *RiskPerformance* encodes whether the applicant will make all payments within 90 days of being due (*good*, 0), or will make a payment over 90 days after due (*bad*, 1). We remove 588 entities from the dataset, for which all values are missing (indicated by the value -9). This is because we cannot provide explanations for entities for which we have no information.

We separate the input dataset into training and test data. The test dataset is a random sample of 1,975 entities (20% of the original dataset). We use the training dataset to learn the prediction model, and the test dataset to evaluate the model and to explain the predictions of the model.

As noted by the FICO community, several input features are *monotonically increasing*, i.e., the probability of the outcome being bad increases with the feature value.

Credit-Card Fraud Dataset. The second dataset we consider is the Kaggle Credit Card Fraud dataset[5], which is used to detect fraudulent credit card transactions. The dataset consists of 284,807 credit card transactions. Each transaction is described by 30 numerical input variables, out of which 28 are normalized and anonymized for privacy reasons. The other two features are *Time* (seconds elapsed between the transaction and the first transaction in the dataset), and *Amount* (the amount of the transaction). The dependent variable takes value 1 in case of fraud, and 0 otherwise. The dependent variable in the dataset is highly unbalanced. Out of all transactions, only 492 transactions are labeled as fraud.

We normalize *Amount* and *Time* with the scikit-learn RobustScaler, and then separate the dataset into training and test data. The test dataset is a random sample of 28,481 entities (10% of the dataset).

6.1 Experiments with FICO Dataset

Our first experiment compares the black-box RESP- and SHAP-explanation scores with a white-box explanation score described by Chen et al. [3] specifically for the FICO dataset. We denote this white-box explanation score as FICO-explanation. In order to explain the FICO-explanation, we need to do a rather detailed review of the model used [3].

Classification Model. Our goal was to use the exact model and explanation score in [3], but, unfortunately, the paper does not provide sufficient information to replicate the model, and the online demonstration[6] of the model does not seem to follow the description in the paper. Hence, we re-implemented the model using our best understanding of the description in the paper. Our implementation makes one design choice that differs from the original model: the input features are bucketized features into disjoint ranges, as opposed to overlapping ranges. This ensures that each entity can have at most one explanation per feature, whereas the original model can provide several explanations for a single feature in case the feature value falls into several buckets (c.f., Table 1 in [3]). We next describe the model as we implemented it.

The classifier is a two-layer neural network, where each layer is defined by logistic regression models. A logistic regression model $LR_{\boldsymbol{\theta}}(\boldsymbol{x})$ with features $\boldsymbol{x} = (x_1, \ldots, x_n)$ is defined by $n+1$ weights $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_n)$, where θ_0 is the bias term of the model. For a given entity with features $\boldsymbol{x}$, the model computes $p = \texttt{sigmoid}(\theta_0 + \sum_{i \in n} \theta_i \cdot x_i)$, which returns a value between 0 and 1. We refer to the outcome of *LR* as the probability of risk. The model classifies the entity as a "bad" outcome if the probability of risk is above 0.5.

The model in [3] requires that the continuous input features are bucketized and one-hot encoded. We use exactly the same buckets as [3], and describe them in Appendix A. One-hot encoding turns the bucketized feature values into indicator vectors, with one entry for each bucket, which is 1 if the value is in this bucket, and 0 otherwise. After bucketization an one-hot encoding, the 23 *input features* become 165 *binary features*, used by LR.

Next, the binary features are categorized into 10 disjoint groups, each group consisting of all binary features derived from 1-4 input features, called *subscales*, as shown in Table 1. Features within a group describe a similar or related aspect of an applicant. For example, *MScienceOldestTradeOpen*, *MScinceMostRecentTradeOpen* and *AverageMInFile* are grouped together into a single subscale, *TradeOpenTime*, because they are all related to the number of months that a trade is open.

The first layer of the classifier consists of one logistic regression model for each subscale. Each of these models returns the probability of risk associated with the features in the corresponding subscale. The second layer is defined by a single logistic regression model, whose inputs are the subscale risk predictions of the models in the first layer, and whose output is the prediction of the risk for this entity. The models are trained in the R library *glmnet*, and

[4] For details see: https://community.fico.com/s/explainable-machine-learning-challenge
[5] https://www.kaggle.com/mlg-ulb/creditcardfraud
[6] The online demonstration can be found here: http://dukedatasciencefico.cs.duke.edu

Feature Name	Subscale Name	Entity	Feature Score	Subscale Risk	Subscale Weight	Subscale Score	Global Risk
ExternalRiskEstimate	**ExternalRiskEstimate**	61	**2.9896**	0.8262	1.566	**1.2934**	
MSinceOldestTradeOpen		198	0.2453				
MSinceMostRecentTradeOpen	TradeOpenTime	14	0.0311	0.4690	2.527	1.1842	
AverageMInFile		96	0.2960				
NumSatisfactoryTrades	NumSatisfactoryTrades	25	0.0001	0.4513	2.156	0.9729	
NumTrades60Ever2DerogPubRec		0	0.0003				
NumTrades90Ever2DerogPubRec	TradeFrequency	0	0.1515	0.4425	0.359	0.1588	
NumTotalTrades		27	0.2653				
NumTradesOpeninLast12M		0	0.0000				
PercentTradesNeverDelq		89	**0.5686**				
MSinceMostRecentDelq	**Delinquency**	1	0.4015	0.6847	2.545	**1.7425**	0.6146
MaxDelq2PublicRecLast12M		4	**1.0046**				
MaxDelqEver		6	0.0000				
PercentInstallTrades		11	0.0009				
NetFractionInstallBurden	Installment	75	0.3706	0.5273	0.913	0.4817	
NumInstallTradesWBalance		2	1.4898				
MSinceMostRecentInqexcl7days		11	0.8318				
NumInqLast6M	Inquiry	0	0.0002	0.3172	3.004	0.9529	
NumInqLast6Mexcl7days		0	0.0000				
NetFractionRevolvingBurden	RevolvingBalance	67	1.3938	0.6500	1.924	1.2505	
NumRevolvingTradesWBalance		7	0.1176				
NumBank2NatlTradesWHighUtilization	Utilization	2	0.8562	0.6490	0.987	0.6406	
PercentTradesWBalance	TradeWBalance	75	0.4528	0.6113	0.296	0.1808	

Table 1: (left) Features and subscales of the classifier for the FICO dataset. (right) Entity e, and the scores, weights, and risk for both the features and subscales that are computed by the classifier for e.

using monotonicity constraints to model the monotonicity of the input features

The classification model has an ROC-AUC score of 0.812 and classifies 1020 entities as a high risk applications (label 1). In the following experiments, we focus on explaining these 'bad' outcomes.

Table 1 depicts, for illustration of the methodology, the nested structure of the model for one particular entity $\boldsymbol{e}$ in the dataset. For each subscale S, the LS model is applied to the binary features of that subscale. For example, the first feature In Table 1, *ExternalRiskEstimate*, has value 61, and falls in the first of the 8 buckets (Appendix A), hence the *Subscale risk* of this subscale is $\texttt{sigmoid}(\theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 0 + \cdots \theta_8 \cdot 0)$, which is 0.8262 in our example. For another example, the subscale risk of the *Delinquency* subscale is 0.6847. Next, all subscale risks become the input to the logistic regression model at the second layer: they are multiplied by the weight of the 2nd layer LS, shown in the column *Subscale Weight* in the Table, added up and passed through the sigmoid function. This results in the final risk probability of the entity $\boldsymbol{e}$, which in our example is 0.6146; this means high risk, hence $L(\mathbf{e}) = 1$ (because the risk is > 0.5). This completes our review of [3]; the detailed description was necessary in our to describe their white-box explanation.

FICO-**explanation.** We next describe the FICO-explanation from [3]. The main idea is the following. For a single regression model, the explanation score of feature F_i is the product $\theta_i x_i$, called *feature score*. The FICO-explanation computes the top two highest feature scores in the second layer, and for each of them returns the top two feature scores in that group, for a total of maximum four features. For the example in Table 1, the top features are shown in green. For the second layer they are *ExternalRiskEstimate* = 1.2934 and *Delinquency* = 1.7425. *ExternalRiskEstimate* has a single feature, which is returned as part of the explanation, while for the *Delinquency* subscale the top two feature scores are *PercentTradesNeverDelq* = 0.5686 and *MaxDelq2PublicRecLast12M* = 1.0046.

For completeness, we describe now the FICO-explanation in detail. Let $M < 10$ and $N < 4$ be two positive integers (in the experiments $M = N = 2$). For a given entity $\boldsymbol{e}$, the FICO-explanation is computed in four steps:

(1) Run the model on $\boldsymbol{e}$ to compute the subscale and final risks, and store the scores for each feature and subscale.
(2) Rank the subscales in decreasing order of the subscale scores obtained in Step 1, and keep only the top-M subscales.
(3) For each of subscale in the top-M, rank its input features in decreasing order of their feature scores, and keep only the top-N (keep all features if there are $< N$).
(4) Concatenate the top-N features for each of the top-M subscales, sorting first by the subscale score, then by the feature score. This is the final FICO-explanation ranking.

Using $M = N = 2$ in Table 1, the top-2 subscales are *Delinquency* and *ExternalRiskEstimate*, and the top-2 features in *Delinquency* are *PercentTradesNeverDelq* and *MaxDelq2PublicRecLast12M*. Thus, the final FICO-explanation order is: *PercentTradesNeverDelq*, *MaxDelq2PublicRecLast12M*, *ExternalRiskEstimate*.

Experimental Result. To compare the FICO, RESP, and SHAP-explanations, we compare (1) the distribution of the top explanations for each score, and (2) the average similarity of the top-4 explanations for each entity between all pairwise combinations of the scores. Recall that both RESP- and SHAP-scores are black-box models: although we used the same classification model as for FICO-explanation, they only use the outputs of the model.

We report on the explanations for the 1020 entities for which the classifier predicts 1, i.e., the 'bad' outcome. We use $M = N = 2$ in Steps 2 and 4 of the FICO-explanations. The RESP-score fails to provide explanations for 101 entities, due to the restriction on the size of the contingency set.

Fig. 3 presents the distribution of the top explanations returned by the FICO, RESP, and SHAP-explanations.

The first observation is a reasonable correlation between the FICO-explanation (white-box) and the RESP-explanation (black-box). Their most frequent top is the same, *MSinceMostRecentInqexcl7days*, and there are a few other features that are popular top explanations for both scores.

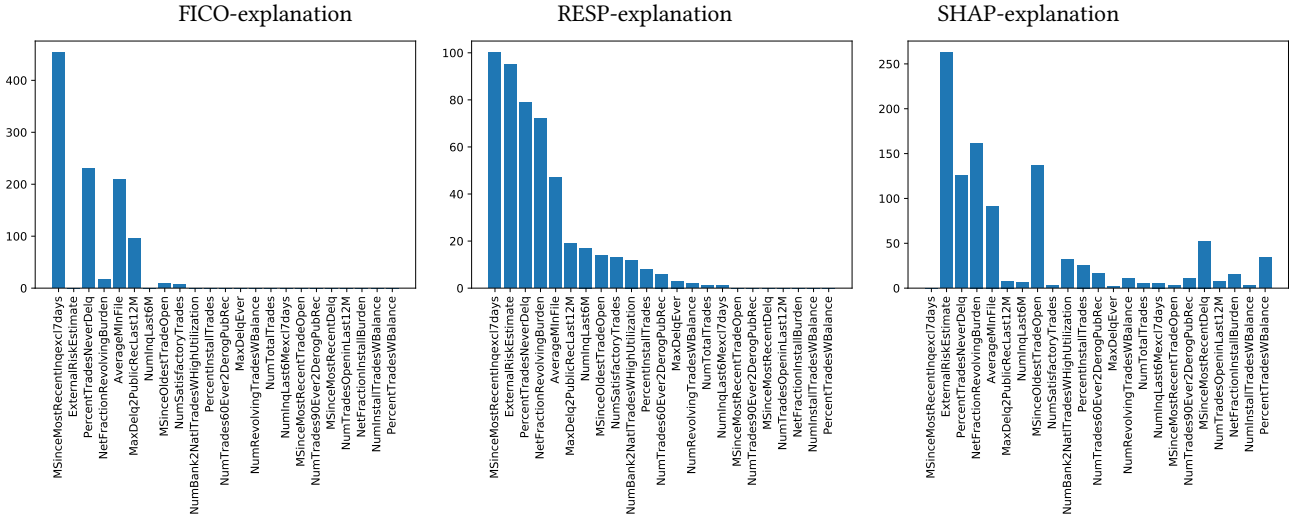

Figure 3: Distribution of the top ranked features for the (left) fico-score, (middle) resp-score, and (right) shap-score. Each bar represents for how many of the 1020 entities was that feature the top explanation. To facilitate comparison, all three bar charts list the features in the same order, which is the decreasing order for the RESP**-explanation (middle chart).**

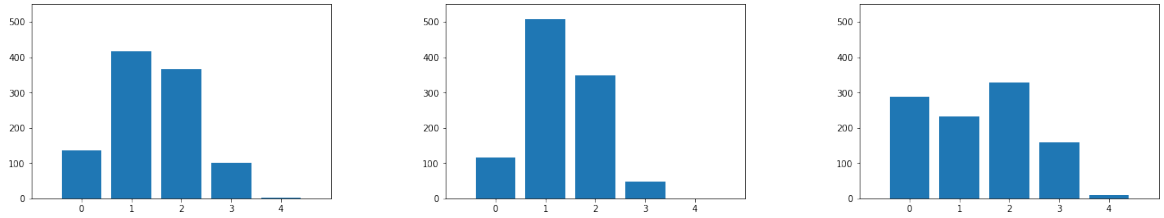

Figure 4: Distribution of the intersection size for top-4 explanations for the (left) FICO **and** RESP**-explanations, (middle)** FICO **and** SHAP**-explanations, and (right)** RESP **and** SHAP**-explanations.**

However, there is an obvious difference with *ExternalRiskEstimate*, which is the second most popular explanation for RESP, yet is never the top FICO-explanation. To understand it, we looked deeper at the data, and illustrate our findings with the example that we chose to show in Table 1. As we saw, the top-2 subscales are *Delinquency* and *ExternalRiskEstimate* in this order(!), hence *ExternalRiskEstimate* will not be the top explanation, instead it will be preceded by the two features in the *Delinquency* subscale. Thus, although the feature-score of *Delinquency* is the highest, it ranks lower only because of the specific way the FICO-explanation is ranked, namely it is ranked first by the sub-scale score, and then by the feature score. This could be adjusted by tweaking the way one computes the ranking. However, there is a deeper reason why the FICO-explanation fails to report *ExternalRiskEstimate* as top explanation: it is because the FICO-explanation is based only on the *current* entity $\mathbf{e}$, and ignores other entities in the population. In contrast, the RESP-score is based on causality and considers the *entire population* of entities. *ExternalRiskEstimate* is a counterfactual cause of our entity, because by changing its value from 61 to 81 the outcome changes from $L(\mathbf{e}) = 1$ to $L(\mathbf{e}) = 0$, while neither *PercentTradesNeverDelq*, *MaxDelq2PublicRecLast12M* nor are causal. The reason why the first is counterfactual while the others are not lies in the weights θ_i associated to the buckets of these features. The values 61 to 81 lie in buckets 1 and 5 of *ExternalRiskEstimate* respectively (see Appendix A), and their weights vary significantly: $\theta_1 = 2.9896$ and $\theta_5 = 0$, In contrast, the buckets of *PercentTradesNeverDelq* and *MaxDelq2PublicRecLast12M* have weights in a small range $[1.5, 1.9]$, and changing their values is insufficient to change the outcome. The FICO-explanation looks only at the current bucket, and fails to notice that other buckets have significantly different values. In contrast, the RESP-score considers the entire model because it examines the entire population, checking for a counterfactual feature.

In general, we observe that the top FICO-explanations are correlated to the weights in the second layer of the classifier. In fact, the four most common top explanations come from the three subscales with the highest weight (their weights are at least 2.5, c.f. Table 1). Thus, they are more likely to have a high subscale score, and consequently, to be among the top subscales in Step 2 of the FICO-explanation. This also explains why the FICO-explanation is less diverse that the RESP-explanation: it tends to choose features from the same three subscales. We argue that a good diversity is a desired quality of an explanation score: we want to be able to give individualized explanations to the customers, and, assuming

all features are relevant to the outcome, we expect a diverse distribution of the top explanation. Of the three graphs in Fig. 3, the RESP-explanation has clearly the most diversity.

Next, we compared the RESP-explanation to the SHAP-explanation, and notice that they are rather distinct. To understand the source of the difference, we focused on the fact that SHAP-score never returns *MSinceMostRecentInqexcl7days* as the top-explanation, which is the most frequent top-explanation for the other two scores. Recall that the SHAP-score is the sum of $n-1$ levels, see Eq. (2), and the weights of the levels is an inverse binomial term, $\frac{\ell!(n-\ell-1)!}{n!}$; thus, most of the mass of the score consists of the first levels $\ell = 0, 1, \ldots$ and the last levels $\ell = n-1, n-2, \ldots$ in other words the weights of the middle levels decrease very fast (exponentially). After examining the data, we found that, for each value of *MSinceMostRecentInqexcl7days*, the distribution of positive and negative outcomes in the test dataset are fairly even. As a result, the first layer of the SHAP-score ($\ell = 0$) is always close to zero. On the other hand, recall from Sec. 5 that for the SHAP-score we are forced to use the empirical probability distribution and, as we argued there, the contribution of the higher levels is zero. This explains why *MSinceMostRecentInqexcl7days* has a very low SHAP-score. We believe that this is an artifact of the empirical probability space that underlies the computation of the SHAP-score, and we conjecture that this phenomenon would not occurs if the SHAP-score was computed over more sophisticated probability scores.

Finally, we compare the top-4 explanations for each pairwise combination of the three scores. We assume the top-4 explanations are sets, and do not consider their ranking. We compute two statistics on each entity: (1) the size of the intersection, and (2) the Jaccard coefficient for set similarities.

Fig. 4 depicts the distribution of the intersection size of the top-4 explanations. For the FICO- and RESP-explanations, we observe that 86.8% of the entities share a common explanation. The remaining cases are mostly entities for which the RESP-score does not provide any explanation, due to the restriction on the contingency set. For the FICO- and SHAP-explanations, 88.5% of the entities share a common explanation. For both comparisons, the number of common explanations is usually less than three. This is not surprising, because the top-4 FICO-explanations must come from the top-2 subscales, whereas the explanations for the RESP and SHAP-scores tend to be more diverse by allowing for explanations from different subscales.

The average Jaccard similarity coefficient for each of the pairwise score comparisons are:

- 0.276 for the FICO and RESP-explanations
- 0.213 for the FICO and SHAP-explanations
- 0.263 for the RESP and SHAP-explanations

The Jaccard coefficients underline our observation from Fig. 3: On average the FICO-explanation is more similar to the RESP- than the SHAP-explanation. Overall, there is only a limited overlap of the top-4 explanations for the three scores. We believe that this is due to the different explanation methodologies, and also the differences in the underlying probability space for the RESP- and SHAP-scores.

6.2 Experiments with the Kaggle Credit Card Fraud Dataset

We next present our results for the evaluation of the black-box RESP and SHAP-scores on the Kaggle Credit Card Fraud dataset.

Classification Model. We use as classifier a logistic regression model. Since the dependent variable in the dataset is highly unbalanced, we trained the classifier over an undersample of the training data, which has an equal distribution of positive and negative outcomes.[7] The classifier labels 846 entities in the original test dataset as fraudulent, which means it misclassifies less than 3% of the entities. This results in an ROC-AUC score of 0.95 on the test dataset.

Score Computation. All features in this dataset are high-precision continuous variables, which leads to complications in the computation of the RESP- and SHAP-scores. For the SHAP-score, the issue is that conditioning on one feature returns a single entity, and, thus, no representative distribution to compute the expected value over the outcome. For the RESP-score, the feature domains are so large that the score becomes very expensive to compute, especially for non-empty contingency sets.

To mitigate this issue, we bucketize the features into varying numbers of equi-depth buckets. Each bucket is represented by the mean over the values that fall into this bucket. Note that, there is a tradeoff between the number of buckets and the time it takes to compute the score. This tradeoff is reversed for the two scores. For the RESP-score, as the number of buckets increases, the computation time increases. For the SHAP-score, as the number of buckets decreases, the computation time increases. We evaluate how sensitive the two scores are to the number of buckets.

Evaluation Results. Fig. 5 presents the distribution of the top explanations for the two scores and varying number of buckets. For the case of 500 buckets, both scores frequently report the variables V_4 and V_{14} as top explanations. Beside these two features, however, the distributions are not very comparable. For instance, the third most common top explanation with the SHAP-score is V_{17}, which is rarely a top explanations for the RESP-score.

We observe that the distribution of the RESP-score is very consistent when the number of buckets ranges between 10 and 500. The number of buckets do however effect the time it takes to compute the score. On our commodity machine, we computed the RESP-score for 10 buckets in less than 4 minutes (roughly 0.3 seconds per entity), while for 500 buckets it took around 11 hours. This is mostly due to the computation for contingency sets of size 1, which is quadratic in the number of features and domain sizes. For the SHAP-score, however, as the number of buckets increases, the distribution over the features becomes more uniform. This is because as we increase the number of buckets there are, number of entities after conditioning on the feature any feature decreases. The time to compute the SHAP-score ranged from 17 minutes for 8000 buckets to over 16 hours for 500 buckets. We attempted to compute the SHAP-score for 100 buckets, but the experiment took longer than the pre-defined timeout of 24 hours. We conclude that the RESP-score is robust to the bucketization of the features, whereas for SHAP-scores there is a tradeoff between the number of buckets and the ability to attain good explanations.

[7] See https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

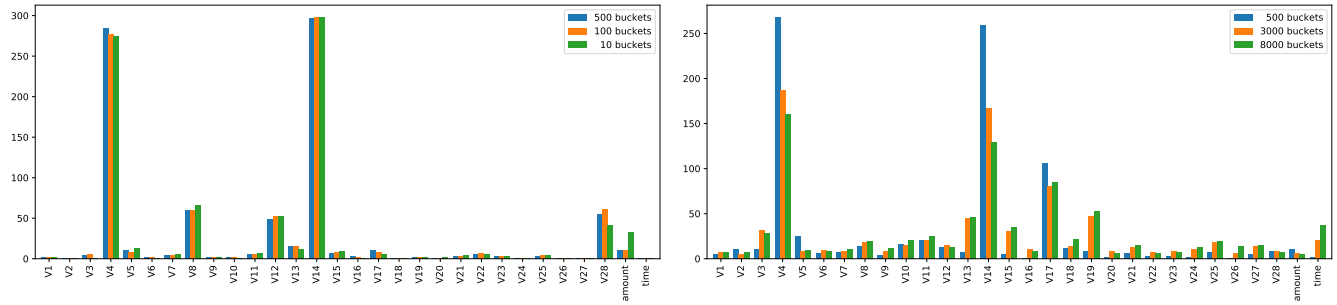

Figure 5: Distribution of the top ranked features for the (left) resp-score, and (right) shap-score where the features are bucketized into varying number of equi-depth buckets.

We further compare the top-4 explanation sets for each entity for both scores and 500 buckets. The majority of the entites (84%) share one or two explanations in the top-4 explanation sets of the two scores, while 10.5% share no explanation. Remarkably, for 75.4% of the entities, the top-explanation for the RESP-score is also one of the top-4 explanations for the SHAP-score. On the other hand, the top explanation in the SHAP-score is one of the top-4 RESP-score explanations for only 67.4% of the entities. Therefore, we conclude that there is some overlap in the scores, but there is only a limited overlap across the top-4 explanations.

7 DISCUSSION

In this paper we introduced a simple notion of explanation, RESP, for a classification outcome, which is grounded in the notion of causality. We have compared RESP to the SHAP-explanation and to a white-box explanation for a specific classification problem. While no benchmarks exists for explanations [5, 10], our empirical evaluation suggests that RESP can provide similar or better quality. Our initial goal of this project was to evaluate the SHAP-score, but we soon ran into difficulties due to its high computational complexity. A polynomial-time implementation is descrbed in [13] but is restricted to decision trees. In contrast, we have shown that its complexity is #P-hard. Because of these difficulties, we proposed the RESP-explanation, based on the simple concept of *counterfactual cause*, and found it quite natural to interpret its results on real data. On the other hand, our experiments have limited the SHAP-explanation to the empirical distribution. Future work is needed to evaluate SHAP on richer probability spaces.

Acknowledgments We thank Guy van den Broeck for insightful discussion on the complexity of SHAP.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499, 1994.

[2] Leopoldo E. Bertossi and Babak Salimi. From causes for database queries to repairs and model-based diagnosis and back. *Theory Comput. Syst.*, 61(1):191–232, 2017.

[3] Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. *CoRR*, abs/1811.12615, 2018.

[4] Hana Chockler and Joseph Y. Halpern. Responsibility and blame: A structural-model approach. *J. Artif. Intell. Res.*, 22:93–115, 2004.

[5] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017.

[6] Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 9273–9282, 2019.

[7] Yash Goyal, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *CoRR*, abs/1907.07165, 2019.

[8] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach — part 1: Causes. *CoRR*, abs/1301.2275, 2013.

[9] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting black box predictions using fisher kernels. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 3382–3390, 2019.

[10] Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018.

[11] Ester Livshits, Leopoldo E. Bertossi, Benny Kimelfeld, and Moshe Sebag. The shapley value of tuples in query answering. *CoRR*, abs/1904.08679, 2019. To appear in Proc. ICDT 2020.

[12] Scott M. Lundberg, Gabriel G. Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. Explainable AI for trees: From local explanations to global understanding. *CoRR*, abs/1905.04610, 2019.

[13] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.

[14] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F. Moore, and Dan Suciu. The complexity of causality and responsibility for query answers and non-answers. *PVLDB*, 4(1):34–45, 2010.

[15] Judeea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge Univ. Press, 2009. 2nd ed.

[16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[17] A. (ed.) Roth. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge Univ. Press, 1988.

[18] Cynthia Rudin. Please stop explaining black box models for high stakes decisions. *CoRR*, abs/1811.10154, 2018.

[19] L. S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

[20] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

A BUCKETIZATION FOR FICO DATASET

We illustrate the bucketization of *ExternalRiskEstimate*. Its buckets are $[0, 63]$, $[64, 70]$, $[71, 75]$, $[76, 80]$, $[81, \infty]$, $\{-7\}$, $\{-8\}$, $\{-9\}$, where the last three buckets are correspond to special generic values that indicate missing, outdated, or inapplicable records. For the entity in Table 1, *ExternalRiskEstimate* = 61 and, after one-hot encoding, this value is represented as the vector $[1, 0, 0, 0, 0, 0, 0, 0]$, because 61 falls into the first bucket.