Explaining Data-Driven Decisions made by AI Systems: The Counterfactual Approach

Carlos Fernandez cfernand@stern.nyu.edu
New York University

Foster Provost fprovost@stern.nyu.edu
New York University

Xintian Han xintian.han@nyu.edu
New York University

Abstract

Lack of understanding of the decisions made by model-based AI systems is one of the main barriers for their adoption. We examine counterfactual explanations, which are becoming an increasingly accepted alternative for explaining AI decisions. The counterfactual approach defines an explanation as a set of the system's data inputs that causally drives the decision (meaning that removing them changes the decision) and is irreducible (meaning that removing any subset of the inputs in the explanation does not change the decision). We generalize previous work on counterfactual explanations, resulting in a framework that (a) is model-agnostic, (b) can address features with arbitrary data types, (c) may explain decisions made by complex AI systems that incorporate multiple models, and (d) is scalable to very large numbers of features. We also propose a heuristic procedure to find the most useful explanations depending on the context. We contrast counterfactual explanations with another alternative that has become popular—methods that explain *model predictions* by weighting features according to their importance (e.g., SHAP, LIME). This paper presents two fundamental reasons why explaining model predictions is not the same as explaining the decisions made using those predictions, suggesting that we should carefully consider whether importance-weight explanations are well-suited to explain decisions made by AI systems. Specifically, we show through several examples that (1) features that have a large importance weight for a model prediction may not actually affect the corresponding decision, and (2) importance weights are insufficient to communicate whether and how the features actually influence system decisions. We demonstrate this first using three simple examples. Then we present three detailed studies using real-world data to compare and contrast the counterfactual approach with SHAP, a popular importance weighting method. The examples and case studies illustrate various conditions under which counterfactual explanations explain data-driven decisions better than feature importance weights.

Keywords: Explanations, System Decisions, Predictive Modeling

1. Introduction

Data and predictive models are used by artificial intelligence (AI) systems to make decisions across many applications and industries. Yet, many data-rich organizations struggle when adopting AI decision-making systems because of managerial and cultural challenges, rather than issues related to data and technology (LaValle et al., 2011). In fact, as predictive models become more complex and difficult to understand, stakeholders often become more skeptical and reluctant to adopt or use them, even if the models have been shown to improve decision-making performance (Arnold et al., 2006; Kayande et al., 2009).

Explanations are also useful for other reasons beyond increasing adoption (Martens and Provost, 2014). For example, explanations may help customers understand the reasoning behind automated decisions that affect them. Users of the model, such as managers or analysts, may use explanations to obtain insights about the domain in which the system is being used. Data scientists and machine learning engineers may also use the explanations to identify, debug, and address potential flaws in the system. Many researchers have tried to reduce the gap in stakeholders understanding of AI systems in recent years, most notably by proposing methods for explaining predictive models and their predictions.

Methods for explaining AI models and their predictions include extracting rules that represent the inner workings (e.g., Craven and Shavlik, 1996; Jacobsson, 2005; Martens et al., 2007) and associating weights to each feature according to their importance for model predictions (e.g., Lundberg and Lee, 2017; Ribeiro et al., 2016). Importance weights, in particular, have become increasingly popular because "model-agnostic" methods that produce importance weights have been introduced: the weights explain predictions in terms of features, so users can understand any specific prediction without any knowledge of the underlying model or the modeling method(s) used to produce the model. For example, two of the most popular methods for explaining model predictions, LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), are model-agnostic and produce importance-weight explanations.

This paper points at two fundamental reasons why importance-weight explanations may not be well-suited to explain data-driven decisions made by AI systems. First, importance weights are designed to explain model predictions, but explaining model predictions is not the same as explaining the *decisions* made using those predictions. Notably, and perhaps counter-intuitively, features that have a large impact on a prediction may not necessarily have an impact on the decision that was made using that prediction. The examples in this paper illustrate this in detail. Therefore, importance weights that are obtained with respect to model predictions may portray an inaccurate picture of how features influence system decisions.

Second, identifying (and quantifying) important features is not sufficient to explain system decisions, even when importance is assessed with respect to the decisions being explained. As an example, suppose that a credit scoring system denies credit to a loan applicant, and that feature importance weights reveal that the two most important features in the credit denial decision were annual income and loan amount. While informative, this "explanation" does not in fact explain what it was that made the system decide to deny credit. Would changing either the annual income or the loan amount be enough for the system to approve credit? Would it be necessary to change both? Or perhaps even changing both would not be enough. From the weights alone, it is not clear how the important features may influence the decision. To be fair, this is not an indictment of methods that calculate feature importance; they were not designed to explain system decisions. However, we are not aware of papers or posts that clarify this in research or in practice.

An alternative to importance-weight explanations are counterfactual explanations—explanations explicitly designed to explain system decisions proposed by Martens and Provost (2014); Provost (2014). For the question why did the model-based system make a specific decision?, the counterfactual approach asks specifically, which data inputs caused the system to make its decision?. This approach is advantageous because (i) it explains decisions rather than the outputs of the model(s) on which the decisions are based; (ii) it standardizes the form that an explanation can take; (iii) it does not require all features to be part of the explanation, and (iv) the explanations can be separated from the specifics of the model.

Martens and Provost (2014) originally applied this framework to explain document classifications, and although it has been applied to other contexts beyond document classification (Moeyersoms et al., 2016; Chen et al., 2017; Ramon et al., 2019), researchers don't all see how the framework can be generalized to settings beyond text (see, e.g., Molnar, 2019; Wachter et al., 2017; Biran and Cotton, 2017). To our knowledge, this approach has not been extended beyond classification models using sparse features in high-dimensional settings. Therefore, we introduce a multi-faceted generalization that focuses on providing explanations for general data-driven system decisions, resulting in a framework that (a) may explain decisions made by systems that incorporate multiple models, (b) is model-agnostic, (c) can address features with arbitrary data types, and (d) is scalable to very large numbers of features. We also propose and showcase a heuristic procedure that may be used to search and sort counterfactual explanations according to their context-specific relevance.

Finally, we illustrate the advantages of our proposed counterfactual approach by comparing it to SHAP (Lundberg and Lee, 2017), an increasingly popular method to explain model predictions that unites several feature importance weighting methods. Via three business case studies that use real-world data, we detail the ways in which counterfactual explanations explain data-driven decisions better than the popular alternative of feature importance weights.

2. AI Systems and Explanations

In this paper, we focus specifically on explaining decisions made by systems that use predictive statistical models to support or automate decision-making (Shmueli and Koppius, 2011), and in particular on systems that make or recommend discrete decisions. We refer to these as artificial intelligence (AI) systems.

2.1 Explaining system decisions

Discrete decision making is closely related to classification, and indeed the subtle distinction often can be overlooked safely—but for explaining system decisions it is important to be clear. First there is a definitional difference: a classification model might classify someone as defaulting on credit or not; a corresponding decision-making system would use this model to make a decision on whether or not to grant credit. Deciding not to grant credit is not the same (at all) as saying that the individual will default—which brings us to the technical difference.

Classification tasks usually are modeled as scoring problems, where we want our predictive models to score the observations such that those more likely to have the "correct" class will have higher scores. These scores may then be used by a system to make a decision that is related to (but usually not the same as) the classification. For example, for binary decisions (and corresponding classifications) typically the scores rank observations, and decisions are made using a chosen threshold appropriate for the problem at hand (Provost and Fawcett, 2013). In many cases, estimated probabilities of class membership are computed from the models, which allows the use of decision theory to combine them with application-specific information on costs and benefits (Provost and Fawcett, 2013) to produce a next stage of more nuanced scores. Thus, decision-making problems are often modeled as "classification tasks" by associating a class with each decision.

However, it is important to emphasize that the final output of the system (i.e., the decision) may not correspond to the labels in the training data. As another example, for a system deciding whether to target a customer with a promotion, scores could consist of expected profits. In this case, we could estimate a classification model to predict the probability that the customer will make a purchase and a regression model to estimate the size of the purchase (conditioned on the customer making a purchase); the expected profits would be the multiplication of these two predictions (Provost and Fawcett, 2013)—and the ranking of the customers by expected profit could be different from the ranking based simply on the classification model score. The final output of the decision-making system would be whether the customer should be targeted with a promotion (and because of selection bias and other complications, we often patently would not want to learn models based on training data about who was targeted with a promotion).

Explaining the decisions made by intelligent systems has received both practical and research attention for decades (Gregor and Benbasat, 1999). Prior work has shown that the ability for intelligent systems to explain their decisions is necessary for their effective use: when users do not understand the workings of an intelligent system, they become skeptical and reluctant to use it, even if the system is known to improve decision-making performance (Arnold et al., 2006; Kayande et al., 2009). More recently, for example, a field study in a Department of Radiology showed that the use of AI systems slowed down, rather than sped up, the radiologists decision-making process because the AI systems often provided recommendations that conflicted with the doctors judgement (Lebovitz et al., 2019). Lacking critical understanding of the opaque AI systems, the doctors often relied on their own diagnoses, which did not concur with the system's. Our paper provides a methodological framework to make the decisions of such AI systems more transparent.

2.2 Explaining predictive models

Over the past several decades, many researchers have worked on explaining predictive models—in contrast to explaining their predictions or decisions made using them. Because symbolic models, such as decision trees, are often considered straightforward to explain when they are small,[1] most research has focused on explaining non-symbolic (black box) models or large models.

Rule-based explanations have been a popular approach to explain black-box models. For example, in many credit scoring applications, banking regulatory entities require banks to implement globally comprehensible predictive models (Martens et al., 2007). Typical techniques to provide rule-based explanations consist of approximating the black box model with a symbolic model (Craven and Shavlik, 1996), or extracting explicit if-then rules (Andrews et al., 1995). Proposed methods are often tailored to the specifics of the models being explained, and researchers have invested significant effort attempting to make state-of-the-art black box models more transparent. For example, Jacobsson (2005) offers a review of explanation techniques for deep learning models, and Martens et al. (2007) propose a rule extraction method for SVMs. Importantly, these "global" explanations (Martens and Provost, 2014) attempt to explain the model as a whole, rather than explaining particular decisions made. As Martens and Provost point out, this can be viewed as explaining every possible decision the model might make—but the methods are not tailored to explain individual decisions.

2.3 Explaining model predictions

A different approach, that has become quite popular recently, is to explain the predictions of complex models, framing the explanations in terms of feature importance by associating a weight to each feature in the model. Each weight can be interpreted as the proportion of the information contributed by the corresponding feature to the model prediction. The main strength of this approach is that the explanations are defined in terms of the domain (i.e., the features), separating them from the specifics of the model being explained. As a result, models can be replaced without replacing the explanation method; end users (such as customers or managers) do not need any knowledge of the underlying modeling methods to understand the explanations, and different models may be compared in terms of their explanations in settings where transparency is critical.

A common way of assessing feature importance is based on simulating lack of knowledge about features (Robnik-Šikonja and Kononenko, 2008; Lemaire et al., 2008). For example, one could compare the original models output with the output obtained when removing a specific feature from the data and the model (e.g., by imputing a default value for the feature). If the output changes, it means that the feature was important for the model prediction. Methods that use this approach often decompose each prediction into the individual contributions of each feature and use the decompositions as explanations, allowing one to visualize explanations at the instance level.

Continuing with the earlier credit scoring example, Figure 1 shows an importance-weight explanation for an individual who has an above-average probability of default. These importance weights were generated using SHAP (Lundberg and Lee, 2017), which we will discuss in more detail in the following sections. Each weight in the explanation represents the impact that its respective feature had on the prediction. Thus, the weight of (roughly)

1. Recent work has been revisiting this assumption, working to produce models explicitly designed to be both accurate and comprehensible; see Wang and Rudin (2015) for an illustrative example.

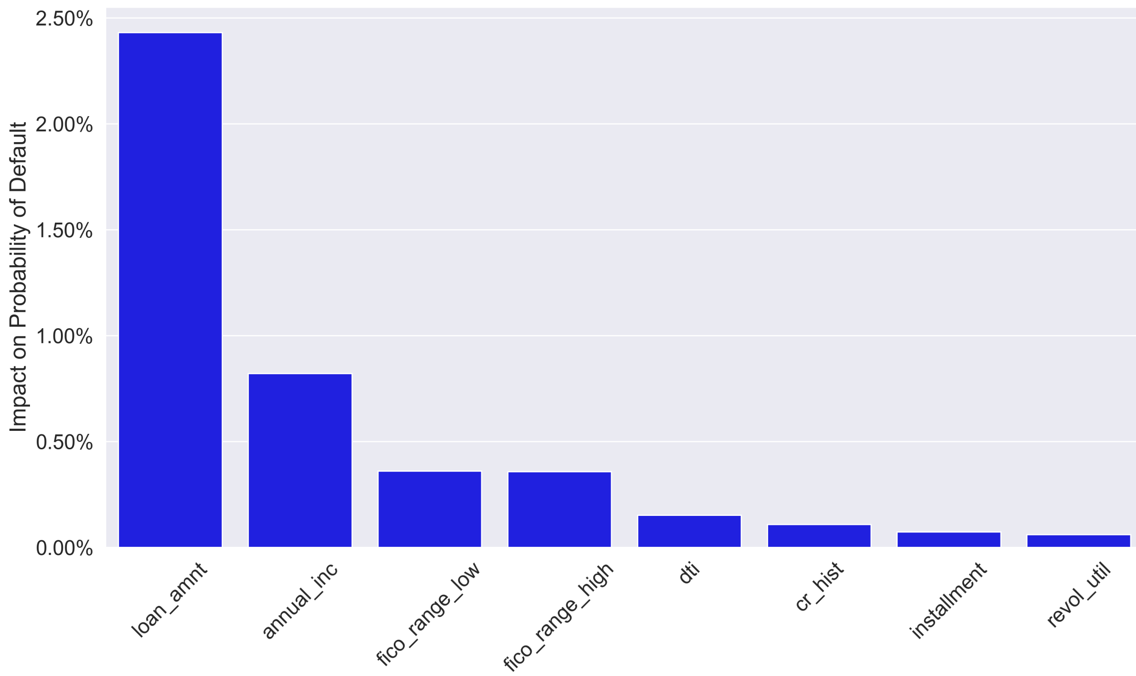


Figure 1: Example of an importance-weight explanation for a model prediction

2.5% that is attributed to the loan amount feature ('loan_amnt') implies that the feature increased the probability of default of that particular individual by 2.5%.

A notable challenge, however, is that interactions between features may lead to ambiguous explanations, because the order in which features are removed may affect the importance attributed to each feature. As a result, subsequent work proposed assessing feature importance by removing all possible subsets of features (rather than only one feature at a time), retraining models without the removed features, and comparing how predictions change (Štrumbelj et al., 2009). However, such approaches may take hours of computation time even for a single prediction and have been reported to handle only up to about 200 features. Alternative formulations (such as SHAP) have attempted to reduce computation time by sampling the space of feature combinations and by using imputation to deal with removed features, resulting in sampling-based approximations of the influence of each feature on the prediction (Štrumbelj and Kononenko, 2010; Ribeiro et al., 2016; Lundberg and Lee, 2017; Datta et al., 2016).

Nevertheless, importance weights are tailored to explain model predictions and may not be adequate to explain system decisions, namely because they dont communicate how the features actually influence decisions. We will illustrate this with several examples below. Moreover, complex systems may incorporate many features in their decision making. In these settings, hundreds of features may have non-zero importance weights for any given instance, yet only a handful of the features may be critical for understanding the system's decisions (Martens and Provost, 2014; Chen et al., 2017).

3. Counterfactual explanations

The idea of using a causal perspective to explain model predictions with counterfactuals was first proposed (to our knowledge) by Martens and Provost (2014) (see also Provost (2014)). Other researchers followed with similar causal, counterfactual explanation approaches (see Molnar, 2019, for examples). In this paper, we generalize the counterfactual explanations originally proposed for document classification (Martens and Provost, 2014) and used subsequently to explain ad-targeting decisions (Moeyersoms et al., 2016), targeting decisions based on Facebook Likes (Chen et al., 2017), and classifications based on other high-dimensional, sparse data (Ramon et al., 2019). We provide a more precise definition of counterfactual explanations below, but as with the prior work, we define explanations in terms of input data—or evidence—that would change the decision if it were not present.

3.1 Example: explaining the decision to flag a transaction

For illustration, suppose a credit card transaction was flagged for action by a data-driven AI system after it was registered as occurring outside the country where the cardholder lives, and suppose the system would have not flagged the transaction absent this location.[2] In this case, it is intuitive to consider the location of the transaction as an explanation for the system decision. Of course, there could be other explanations. Perhaps the transaction also involved a consumption category outside the profile of the cardholder (e.g., a purchase at a casino), and excluding this information from the system would also change the decision to "do not flag". Both are counterfactual explanations—they comprise evidence without which the system would have made a different decision.

A subtle implication of this perspective is that counterfactual explanations are generally applied to "non-default" decisions, because data-driven systems usually make default decisions in the absence of evidence suggesting that a different decision should be made. In our example, a transaction would be considered legitimate unless there is enough evidence suggesting fraud. As a result, explaining default decisions often corresponds to saying, "because there was not enough evidence of a non-default class".[3] Thus, as with prior work, in this paper we focus primarily on explaining non-default decisions.

3.2 Defining counterfactual explanations

Following Martens and Provost (2014) and Provost (2014), we define a counterfactual explanation for a system decision as a set of features that is **causal** and **irreducible**. Being causal means that removing the set of features from the instance causes the system decision

2. We should keep in mind the decision-rather-than-classification perspective. The decision is to flag the transaction for one or more actions, such as sending a message to the account holder to verify. Flagging may be based on a threshold on the estimated likelihood of fraud, but may also consider the existence of evidence from other transactions and the potential loss if the transaction were indeed fraudulent.

3. However, this is not always the case. For example, if a credit card transaction was made in a foreign country, but the cardholder recently reported a trip abroad, the trip report could be a reasonable explanation for the transaction being classified as legitimate. So, the evidence in favor of a non-default classification may be cancelled out by other evidence in favor of a default classification.

Explanation 1	Credit approved if {'loan_amnt'} is removed.
Explanation 2	Credit approved if {'annual_inc'} is removed.
Explanation 3	Credit approved if {'fico_range_high', 'fico_range_low'} are removed.

Table 1: Examples of counterfactual explanations for a system decision

to change.[4] Irreducible means that removing any proper subset of the explanation would not change the system decision. The importance of an explanation being causal is straightforward: the decision would have been different if not for the presence of this set of features. The irreducibility condition serves to avoid including features that are superfluous, which relates to the fact that some of the features in a causal set may not be necessary for the decision to change.

More formally, consider an instance I consisting of a set of m features, $I = \{1, 2, ..., m\}$, for which the decision-making system $C : I \rightarrow \{1, 2, ..., k\}$ gives decision c. A feature i is an attribute taking on a particular value, like `income=$50,000` or `country=FRANCE`. Then, a set of features E is a counterfactual explanation for $C(I) = c$ if and only if:

$$E \subseteq I \text{ (the features are present in the instance)} \tag{1}$$

$$C(I - E) \neq c \text{ (the explanation is causal)} \tag{2}$$

$$\forall E' \subset E : C(I - E') = c \text{ (the explanation is irreducible)} \tag{3}$$

As mentioned, our approach builds on the explanations proposed by Martens and Provost (2014), who developed and applied counterfactual explanations for document classifications, defining an explanation as an irreducible set of words such that removing them from a document changes its classification. Our definition generalizes their counterfactual explanations in three important ways. First, it makes explicit how the explanations may be used for broader system decisions, which may incorporate predictions from multiple predictive models. Second, their practical implementation of explanations (and subsequent work) consists of removing features by setting them to zero, whereas we generalize to arbitrary methods for removing features (and note the important relationship to methods for dealing with missing data). Third, while their approach has been applied in other contexts beyond document classification (Chen et al., 2017; Moeyersoms et al., 2016; Ramon et al., 2019), these applications all have the same data structure: high-dimensional, sparse features. Our generalization applies to features with arbitrary data types.

Going back to our credit scoring example, suppose a decision-making system using the model prediction explained in Figure 1 decides not to grant credit to that individual. Table 1 shows some possible counterfactual explanations for the credit denial decision. Each explanation represents a counterfactual world in which specific evidence is not considered when making the decision, resulting in a default decision (approving credit in this case).

4. It is critical to differentiate what is causing the data-driven system to make its decisions from causal influences in the actual data-generating processes in the "real" world. Our definition of counterfactual explanations relates to the former.

3.3 Removing "evidence" from the input to a data-driven decision procedure

A vital practical question that is raised by the counterfactual approach discussed here is what does it mean to "remove" evidence (i.e., features) from a data instance that will be input to a model-based decision-making procedure? Prior methods for counterfactual explanations and model sensitivity analyses have replaced input feature values with some other specified value. For example, Martens and Provost (2014) replace the presence (binary indicator, count, TFIDF value, etc.) of a word in a document with a zero. This makes sense in the context of their application, because if we consider the presence of a word as evidence for a document classification, removing that evidence—that word—would be represented by a zero for that feature.[5]

More generally, we should consider carefully the notion of removing features from the input to a data-driven model. If we step away for a moment from explaining AI systems, we can think of explaining other sorts of evidence-driven decisions within the same framework. For instance, in a murder case, we might explain our decision to bring in the suspect based on the fact that the murder weapon was found in her apartment; if there were no murder weapon, we would not have brought her in. If we would have brought her in anyway, then the presence of the weapon does not suffice as an explanation for our decision. So, in this case, we are imagining our collection of evidence with the focal piece of evidence missing. We can do the same *in principle* with data-driven decisions: we can make the feature in question be missing and ask if we would still make the same decision. Thus, we can generalize to data inputs of any kind: removing the feature means "making it missing" in the data instance.

We emphasize that we can do this "in principle" because in practice it may or may not be practicable to simply make a feature be missing. Some AI models and systems deal with missing features naturally and some do not. Importantly, note that here we are talking about dealing with missing values at the time of use of the model, not dealing with missing values during machine learning. There are different ways for dealing with missing features when applying (as opposed to learning) a predictive model (Saar-Tsechansky and Provost, 2007), such as imputing default values for the missing features, using an alternative model trained with only the available features, etc.

Therefore, the generalized explanation framework we present is agnostic to which method is used to deal with the removed features—taking the position that this decision is domain and problem dependent. Within a particular domain and explanation context, the user should choose the method for dealing with missing values. For example, in settings where features are often missing at prediction time, replacing the value of a feature with a "missing" categorical value might make the most sense to simulate missingness, whereas in cases where all attributes must have values specified in order to make the decision, replacing the value with the mean or the mode might make more sense. What matters is that the decision may change when some of the features are not present at the time of decision making, and that the method for dealing with missing values allows the change in the decision to be attributed to the absence of these features.

This framework naturally incorporates other techniques used in prior counterfactual approaches: the common case of replacing a feature in a sparse setting with a zero corresponds

5. They discuss the case where absence of a word would be evidence as well; see the original paper.

to mode imputation; replacing a numeric feature with the mean value for that attribute corresponds to mean imputation. In the empirical examples presented below, we use mean imputation for continuous variables and mode imputation for sparse numeric, binary, and categorical variables. Saar-Tsechansky and Provost (2007) discuss other alternatives for dealing with missing values when applying predictive models; any of them could be used in conjunction with this counterfactual explanation framework.

3.4 A procedure for finding useful counterfactual explanations

This definition of counterfactual explanations for system decisions allows any procedure for finding such explanations. For example, fast solvers for combinatorial problems may be used to find counterfactual explanations (Schreiber et al., 2018). For this paper, and for the examples that follow, we adopt a heuristic procedure to find the most useful explanations depending on the context.

The algorithm proposed by Martens and Provost (2014) finds counterfactual explanations by using a heuristic search that requires the decision to be based on a scoring function, such as a probability estimate from a predictive model. We also will presume that the decision making is based on comparing some score to a threshold. This scoring function is used by the search algorithm to first consider features that, when removed, reduce the score of the predicted class the most. This heuristic may be desirable when the goal is to find the smallest explanations, such as when explaining the decisions of models that use thousands of features. Another possible heuristic is to remove features according to their overall importance for the prediction, where the importance may be computed by a feature importance explanation technique (Ramon et al., 2019).

However, the shortest explanations are not necessarily the best explanations. For instance, users may want to use the explanations as guidelines for what to change in order to affect the system decision. As an example, suppose that a system decides to warn a man that he is at high risk of having a heart attack. An explanation that the system would have not made the warning if the patient were not male is of very little use as a guide for what to do about it. In practice, some features are easier to change than others, and some may be practically impossible to change.

Therefore, we allow the incorporation of a cost function as part of the heuristic procedure in order to search first for the most relevant explanations. The underlying idea is that the cost function may be used to associate costs to the removal (or adjustment) of features, so that sets of features that satisfy desirable characteristics are searched first. Importantly, the cost function is meant to be used as a mechanism to capture the relevance of explanations, so the cost of changing or removing the features might not represent an actual cost (we will show an example of this in one of the case studies below). For example, the cost may be fixed (e.g., when removing a word from a document), may be contingent on the value of the variable (e.g., when adjusting a continuous variable), contingent on the value of other features, or may even be practically infinite.

Subsequently, instead of searching for the feature combinations that change the score of the predicted class the most, the heuristic could search for the feature combinations for which the output score changes the most per unit of cost. The motivation behind this new heuristic is to find first the explanations with the lowest costs. Returning to the heart attack example, if we assign an infinite cost to changing the gender feature, the heuristic would not select feature combinations that include it, regardless of its high impact on the output score. Instead, the heuristic would prefer explanations with many modest but cheap changes, such as changing several daily habits. To the extent that the system also has a scoring function (which could be the result of combining several predictive models), the procedure proposed by Martens and Provost (2014) could be easily adjusted to find the most useful explanations for the problem at hand. A similar approach has been suggested for classifiers that have a known and differentiable scoring function (Lash et al., 2017).

3.5 Other advantages of counterfactual explanations

Counterfactual explanations have other benefits as well. First, as with importance weights, they are defined in terms of domain knowledge (features) rather than in terms of modeling techniques. As mentioned above, this is of critical importance to explain individual decisions made by such models to users. More importantly, these explanations can be used to understand how features affect decisions, which (as we will show in next sections) is not captured well by feature importance methods. Also, because only a fraction of the features will be present in any single explanation, the present approach may be used to explain decisions from models with thousands of features (or many more). Studies show cases where such explanations can be obtained in seconds for models with tens or hundreds of thousands of features and that the explanations typically consisted of a handful to a few dozen of features at the most (Martens and Provost, 2014; Moeyersoms et al., 2016; Chen et al., 2017).

4. Limitations of importance weights

In this section, we use three simple, synthetic (but illustrative) examples to highlight two fundamental reasons why importance-weight explanations may not be well-suited to explain data-driven decisions made by AI systems. The first example (Example 1) is meant to illustrate that features that have a large impact on a prediction (and thus large importance weights) may not have any impact on the decision made using that prediction. The next two examples show that importance weights are insufficient to communicate how features actually affect decisions (even when importance is determined with respect to system decisions rather than model predictions). More specifically, we show cases in which importance weights remain the same despite substantial changes to decision making (Examples 1, 2, and 3) and in which features deemed unimportant by the weights actually affect the decision (Example 3). Similar examples to the ones discussed in this section will come up again in the case studies in Section 5, when comparing importance weights with counterfactual explanations using real-world data.

Throughout this section, the examples assume that we want to explain the binary decision made for three-feature instance I and decision procedure C_i as defined here:

$$I = \{F_1 = 1, F_2 = 1, F_3 = 1\}, \tag{4}$$

$$C_i(I) = \begin{cases} 1, & \text{if } \hat{Y}_i(I) \geq 1 \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $\{F_1, F_2, F_3\}$ are binary features, and C_i is the decision-making procedure (an AI system) that uses the scoring (or prediction) function $\hat{Y}_i$ to make decisions. The examples that follow will employ different $\hat{Y}_i$. We assume that domain knowledge has guided us to replace the values of missing features with a default value of zero.

We compute importance weights using SHAP (Lundberg and Lee, 2017), a popular approach to explain the output of machine learning models. Before we focus on the disadvantages of importance weights for explaining system decisions, let us point out that SHAP has several advantages for explaining data-driven model predictions: (i) it produces numeric "importance weights" for each feature at an instance-level, (ii) it is model-agnostic, (iii) its importance weights tie instance-level explanations to cooperative game theory, providing a solid theoretical foundation, (iv) and SHAP unites several feature importance weighting methods, including the relatively well-known LIME (Ribeiro, Singh and Guestrin, 2016).

In the case of SHAP, importance weights consist of the (approximated) Shapley values of the features for a model prediction. Shapley values correspond to the impact each feature has on the prediction, averaged over all possible joining orders of the features. A major limitation of Shapley values is that computing them becomes intractable as the number of features grows. SHAP circumvents this limitation by sampling the space of feature combinations, resulting in a sampling-based approximation of the Shapley values. There are only 3 features in the examples that follow, so the approximations are not necessary here, but they will be for the case studies discussed in Section 5, where the number of features is much larger. We illustrate the computation of Shapley values in more detail in the examples below.

4.1 Example 1: Distinguishing between predictions and decisions

All importance weighting methods (that we are aware of) are designed to explain the output of scoring functions, not system decisions. This is problematic because a large impact on the scoring function does not necessarily translate to an impact on the decision. This example illustrates this by defining $\hat{Y}_1$ as follows:

$$\hat{Y}_1(I) = F_1 + F_2 + 10F_1F_3 + 10F_2F_3, \tag{6}$$

so the prediction and the decision for instance I are $\hat{Y}_1(I) = 22$ and $C_1(I) = 1$ respectively.

Table 2 shows how to compute the Shapley values of the features with respect to $\hat{Y}_1$. Each row represents one of the six possible joining orders of the features, and each column corresponds to the impact of one of the three features across those joining orders. The last row shows the average impact of the features, which corresponds to the Shapley values.

According to Table 2, SHAP gives F_3 a larger weight than F_1 or F_2 due to its large impact on $\hat{Y}_1$. However, if we take a closer look at C_1 and $\hat{Y}_1$ simultaneosuly, we can see that F_3 does not affect the decision-making procedure at all! More specifically F_3 only affects $\hat{Y}_1$ if F_1 or F_2 are already present, but if those features are present, then increasing the score does not affect the decision because $\hat{Y}_1 \geq 1$ (implying that $C_1 = 1$ regardless of F_3). Therefore, the large "importance" of a feature for a model prediction may not imply an impact on a decision made with that prediction.

As we mentioned at the outset, SHAP was not designed to explain system decisions—so this is not an indictment of SHAP. It is an illustration that explaining model predictions and

Joining orders	**Impact of F_1**	**Impact of F_2**	**Impact of F_3**
F_1, F_2, F_3	1	1	20
F_1, F_3, F_2	1	11	10
F_2, F_1, F_3	1	1	20
F_2, F_3, F_1	11	1	10
F_3, F_1, F_2	11	11	0
F_3, F_2, F_1	11	11	0
Shapley values	6	6	10

Table 2: Shapley values for $\hat{Y}_1$ and all the joining orders used in their computation.

Joining orders	**Impact of F_1**	**Impact of F_2**	**Impact of F_3**
F_1, F_2, F_3	1	0	0
F_1, F_3, F_2	1	0	0
F_2, F_1, F_3	0	1	0
F_2, F_3, F_1	0	1	0
F_3, F_1, F_2	1	0	0
F_3, F_2, F_1	0	1	0
Shapley values	0.5	0.5	0
There is a single counterfactual explanation: $\{F_1, F_2\}$			

Table 3: Shapley values and joining orders for C_1, as well as all counterfactual explanations for this decision.

explaining system decisions are two different tasks. We might conclude then that we could adapt SHAP to compute feature importance weights for system decisions, for example, by transforming the output of the decision system into a "scoring function" that returns 1 if the decision is the same after removing features and returns 0 otherwise. This transformation, originally introduced by Moeyersoms et al. (2016) (also in the context of using Shapley values for instance-level explanations), would allow us to use SHAP to obtain importance weights for the system decision—even decisions with multiple, unordered alternatives that cannot normally be represented as a single numeric score.

Table 3 shows the Shapley values of the features with respect to the decision-making procedure C_1 (when applying the suggested transformation). It illustrates that F_3 indeed does not affect the decision at all. However, the next examples show that, even when

Joining orders	Impact of F_1	Impact of F_2	Impact of F_3
F_1, F_2, F_3	0	1	0
F_1, F_3, F_2	0	1	0
F_2, F_1, F_3	1	0	0
F_2, F_3, F_1	1	0	0
F_3, F_1, F_2	0	1	0
F_3, F_2, F_1	1	0	0
Shapley values	0.5	0.5	0
There are two counterfactual explanations: $\{F_1\}$ and $\{F_2\}$			

Table 4: Shapley values for C_2, as well as all counterfactual explanations for this decision.

importance weights are computed with respect to the decision-making procedure rather than the model predictions, the weights do not capture well how features affect decisions.

4.2 Example 2: Multiple interpretations for the same weights

In Example 1, the decision changes when we remove (or change) F_1 and F_2 simultaneously, and removing any of the features individually does not change the decision. So, according to our definition in Section 3.2, there is a single counterfactual explanation, $\{F_1, F_2\}$. However, suppose we were to use the following scoring function to make decisions instead:

$$\hat{Y}_2 = F_1 F_2 \tag{7}$$

Table 4 shows the Shapley values for C_2, which are the same as for C_1 (see Table 3) because features F_1 and F_2 are equally important in both cases. However, the decision-making procedure is different because the new scoring function implies that removing either feature would change the decision. Therefore, with the new scoring function, there would be two counterfactual explanations, $\{F_1\}$ and $\{F_2\}$, but the importance weights do not capture this. This implies that (in general) importance weights do not communicate how removing (or changing) the features may change the decision.[6]

4.3 Example 3: Positive impact of non-positive weights

In Example 1, we showed that even if a feature has a large, positive importance weight for a model's instance-level prediction, changing the feature may have no effect on the decision made for that instance. Importance weights can also be misleading if we use them to explain system decisions, because a feature with an importance weight of zero may have a positive

6. Note that Ramon et al. (2019) show a way to use importance weighting methods (such as LIME and SHAP) to search for counterfactual explanations; this is different from computing importance weights for system decisions.

Joining orders	Impact of F_1	Impact of F_2	Impact of F_3
F_1, F_2, F_3	1	-1	1
F_1, F_3, F_2	1	1	-1
F_2, F_1, F_3	-1	1	1
F_2, F_3, F_1	1	1	-1
F_3, F_1, F_2	0	1	0
F_3, F_2, F_1	1	0	0
Shapley values	0.5	0.5	0
There are three counterfactual explanations: $\{F_1\}, \{F_2\}$, and $\{F_3\}$			

Table 5: Shapley values for C_3, as well as all counterfactual explanations for this decision.

effect on the decision! We illustrate this with a third example, for which we use the following scoring function:

$$\hat{Y}_3 = F_1 + F_2 - 2F_1F_2 - F_1F_3 - F_2F_3 + 3F_1F_2F_3 \tag{8}$$

Table 5 shows the Shapley values with respect to C_3, and we can see that the values are the same as in the previous examples, but the decision-making process has changed once again. Notably, removing (or changing) F_3 can change the decision from $C_3 = 1$ to $C_3 = 0$, as evidenced by the impact of F_3 in the first and third joining orders, but the importance weight of F_3 is 0. The counterfactual explanation framework, on the other hand, reveals that there are three counterfactual explanations in this example: $\{F_1\}$, $\{F_2\}$, and $\{F_3\}$. Thus, a feature that we might mistakenly deem as irrelevant due to its non-positive weight, is in fact as important as the other features with positive weights (at least for the purposes of explaining the decision $C_3(I) = 1$).

4.4 Drawbacks of using averages

While the previous examples were deliberately constructed to illustrate the limitations of importance weights (and thus may seem contrived), they reveal an important insight: it is difficult to capture the impact of features on decisions with a single number, especially when features interact with each other. This is particularly relevant when explaining black-box models (such as neural networks), which are well-known for learning complex interactions between features. Moreover, we will show in Section 5 how the hypothetical examples we illustrated in this section also occur in real-world scenarios.

The main reason why importance weights are problematic for explaining system decisions is that they essentially aggregate across potential explanations (i.e., feature sets) to provide a single explanation per decision. Thus, each decision is explained using a single vector of weights. Typically, the importance weighting methods summarize the impact of features in a single vector by averaging across multiple feature orderings. The problem is that the average impact of a feature is not fine-grained enough to describe dynamics between features, and more importantly, it is difficult to interpret: why should the average across feature orderings be relevant to explain a decision? After all, it might not be representative of the potential impact that features have (as in the case of F_3 in Example 3).

Counterfactual explanations circumvent the drawbacks of using averages because the explanations are defined at the counterfactual level, meaning that each explanation represents a counterfactual world in which the decision would be different. This allows a single decision to have multiple explanations, allowing a richer interpretation of how the features may influence the decision.

5. Case Studies

We now present three case studies to illustrate the phenomena discussed above using real-world data. The first case study contrasts counterfactual explanations with explanations based on importance weights, showing fundamental differences. The second case study showcases the power of counterfactual explanations for very high-dimensional data and shows how the heuristic procedure that generates counterfactual explanations may be adjusted to search and sort explanations according to their relevance to the decision maker. The third case study shows the application of counterfactual explanations to AI systems that are more complex than just applying a threshold to the output of a single predictive model—specifically, to systems that integrate multiple models predicting different things. In all case studies, we use SHAP to compute importance weights with respect to the decision-making procedure rather than model predictions (as discussed above).

5.1 Study 1: Importance Weights vs Counterfactual Explanations

To showcase the advantages of counterfactual explanations over feature importance weights when explaining data-driven decisions, we explain decisions made by a system that makes decisions to accept or deny credit, based on real data from Lending Club, a peer lending platform. The data is publicly available and contains comprehensive information on all loans issued starting in 2007. The data set includes hundreds of features for each loan, including the interest rate, the loan amount, the monthly installment, the loan status (e.g., fully paid, charged-off), and several other attributes related to the borrower, such as type of house ownership and annual income. To simplify the setting, we use a sample of the data used by Cohen et al. (2018) and focus on loans with a 13% annual interest rate and a duration of three years (the most common loans), resulting in 71,938 loans. The loan decision making is simulated but is in line with consumer credit decision making as described in the literature (see Baesens et al., 2003).[7]

We use 70% of this data set to train a logistic regression model that predicts the probability of borrowers defaulting using the following features: loan amount (loan_amnt), monthly installment (installment), annual income (annual_inc), debt-to-income ratio (dti), revolving balance (revol_bal), incidences of delinquency (delinq_2yrs), number of open credit lines (open_acc), number of derogatory public records (pub_rec), upper boundary range of FICO

7. Note that the Lending Club data contains a substantial number of loans for which traditional models estimate moderately high likelihoods of default, despite these all being issued loans. This may be due to Lending Clubs particular business model, where external parties choose to fund (invest in) the loans.

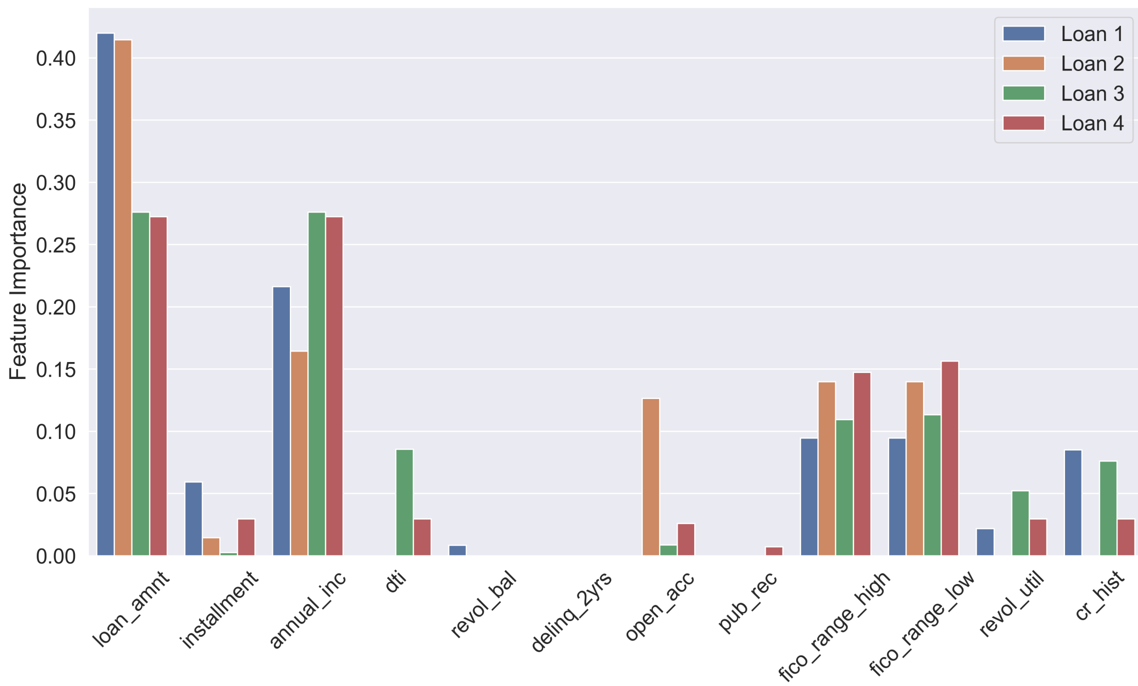


Figure 2: Feature importance weights according to SHAP

score (fico_range_high), lower boundary range of FICO score (fico_range_low), revolving line utilization rate (revol_util), and months of credit history (cr_hist). The model is used by a (simulated) system that denies credit to loan applicants with a probability of default above 20%. We use the system to decide which of the held-out 30% of loans should be approved.

By comparing counterfactual explanations to explanations based on feature importance weights, we can see counterfactual explanations have several advantages. First, importance weights do not communicate which features would need to change in order for the decision to changeso their role as explanations for decisions is incomplete. Figure 2 shows the feature importance weights assigned by SHAP to four loans (different colors) that are denied credit by the system. For instance, according to SHAP, loan_amnt was the most important feature for the credit denial of all four loans. However, this information does not fully explain any of the decisions. The credit applicant of Loan 1, for example, cannot use the explanation to understand what would need to be different to obtain credit; the feature importance weights do not explain why he or she was denied credit. Was it the amount of the loan? The annual income? Both?

Table 6, in contrast, shows all counterfactual explanations for the credit denial decision of Loan 1. Each column represents an explanation, and the arrows in each cell show which features are present in each explanation (recall that a counterfactual explanation is a set of features). The last column shows the difference between the original value of each feature and the value that was imputed to simulate missingness (the mean in our case), illustrating how our generalized counterfactual explanations may be applied to numeric features.

Features	Explanations						Distance from mean
	1	**2**	**3**	**4**	**5**	**6**	
loan_amnt	↑						+$16,122
installment					↑		+$540
annual_inc		↓	↓	↓	↓	↓	-$9,065
revol_bal						↓	-$4,825
fico_range_high			↓				-16
fico_range_low		↓					-16
revol_util						↑	+12%
cr_hist				↓			-92 months
↑ means feature is **too large** to grant credit.							
↓ means feature is **too small** to grant credit.							

Table 6: Counterfactual explanations for Loan 1

For example, as shown in column 1, one possible explanation for the credit denial of Loan 1 is that the loan amount is too large (or more specifically, $16,122 larger than the average) given the other aspects of the application. The data indeed shows that the amount for Loan 1 is $28,000, but the average loan amount in our sample is $11,878. In this instance, one could explain the decision in several other ways. The explanation in column 4 suggests that the $28,000 credit would be approved if the applicant had a higher annual income and a longer credit history, which are below average in the case of the applicant. Therefore, from these explanations, it is immediately apparent how the features influenced the decision. This highlights two additional advantages of counterfactual explanations: they give a deeper insight into why the credit was denied and provide various alternatives that could change the decision.

Table 7 shows the counterfactual explanations of Loan 4 to emphasize this last point. From Figure 2, we can see that Loan 1 and Loan 4 have similar importance weights. Thus, from this figure alone, one may conclude that these two credit denial decisions should have similar counterfactual explanations. Yet, comparing Table 6 and Table 7 reveals this in fact is not the case. Loan 4 has many more explanations, and even though the explanations in both loans have similar features, the only explanation that the loans have in common is the first one (i.e., loan amount is too large); there is no other match.

Importantly, the number of potential counterfactual explanations grows exponentially with respect to the number of features, and we know of no algorithm with better than exponential worst-case time complexity for finding all explanations. Therefore, finding all

Features	**Explanations**															**Distance from mean**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
loan_amnt	↑															+$16,122
installment						↑						↑			↑	+$540
annual_inc		↓														-$9,065
dti				↑						↑					↑	+5
open_acc								↑						↑		+1
pub_rec									↑							+1
fico_range_high			↓							↓	↓	↓	↓	↓		-16
fico_range_low			↓	↓	↓	↓	↓	↓	↓							-16
revol_util							↑						↑		↑	+12%
cr_hist					↓						↓				↓	-92 months
↑ means feature is **too large** to grant credit.																
↓ means feature is **too small** to grant credit.																

Table 7: Counterfactual explanations for Loan 4

counterfactual explanations may be intractable when the number of features is large.[8] In the case of the loans discussed in this case study, we were able to conduct an exhaustive search because the number of features is relatively small; thus Tables 6-7 show all possible counterfactual explanations for the credit denials of Loan 1 and Loan 4. In other settings, we may need to be satisfied with an approximation to the set of all explanations.

In cases where the number of explanations is large, additional steps to improve interpretability may be helpful, such as defining measures to rank explanations according to their usefulness. One such measure is the number of features present in the explanation (the fewer, the better). In fact, the heuristic we used to find explanations in this example, the same introduced by Martens and Provost (2014), tries to find the shortest explanations first. However, there could be other more relevant measures depending on the particular decision-making problem—such as the individual's ability to change the features in the explanation. As mentioned above, our generalized framework would allow incorporating the cost of changing features as part of the heuristic procedure, resulting in an algorithm designed to (try to) find the cheapest or more relevant explanations first. Because finding all possible explanations was tractable in this case, we did not incorporate costs in the heuristic we used to find explanations in this empirical example, but we do so in the next case study.

8. Ramon et al. (2019) demonstrates the effectiveness of starting the importance weights in order to efficiently generate a counterfactual explanation, but this does not reduce the worst case complexity for finding all explanations. Furthermore, as noted above, computing the importance weights itself is computationally expensive.

Nonetheless, one can see that not all features shown in Figure 2 and Tables 6-7 would be relevant for loan applicants looking for recommendations to get their credit approved. So, SHAP may be adjusted further to compute weights only for a subset of features. Since SHAP deals with missing features by imputing default values, we can easily extend SHAP to only consider certain (relevant) features by setting the default values of the irrelevant features equal to the current values of the instance. Then, SHAP will compute importance weights only for the features that have a value different from the default. We do this for Loan 4 and define loan amount and annual income as the only relevant features. This would make sense in our context if customers can only ask for less money or show additional sources of income to get their credit approved.

After doing this, SHAP computes an importance weight of 0.5 for both the loan amount and the annual income, and there are two counterfactual explanations: the applicant can either reduce the loan amount or increase the annual income to get the loan approved (columns 1 and 2 in Table 7). However, consider a different scenario. Suppose the bank were stricter with the loans it approves and used a decision threshold 2.5 percentage points lower. Now, in order to get credit approved, the applicant of Loan 4 would need both to reduce the loan amount and to increase her (or his) annual income.

This situation is directly analogous to Example 2 in Section 4.2. With this different decision system, there is a single counterfactual explanation (instead of two) consisting of both features, so the counterfactual framework captures the fact that the decision-making procedure changed. However, SHAP would still show an importance weight of 0.5 for each feature. Thus, the counterfactual explanations and the SHAP explanations exhibit different behavior. SHAP explanations suggest that the two decisions are essentially the same. The counterfactual explanations suggest that they are quite different. We argue that the latter is preferable in many settings. It may well be that the former is preferable in some settings, but we havent found a credible and compelling example.

5.2 Study 2: High-dimensional and Context-specific Explanations

We use Facebook data to showcase the advantages of counterfactual explanations when explaining data-driven decisions in high-dimensional settings. The data, which was collected through a Facebook application called myPersonality,[9] has also been used by other researchers to compare the performance of various counterfactual explanation methods (Ramon et al., 2019). We use a sample that contains information on 587,745 individuals from the United States, including their Facebook Likes and a subset of their Facebook profiles. In general, Facebook users do not necessarily reveal all their personal characteristics, but their Facebook Likes are available to the platform. For this case study, in order to simulate a decision-making system, we assume there is a (fictitious) firm that wants to launch a marketing campaign to promote a new product to users who are more than 50 years old. Given that not all users share their age in their Facebook profile, the firm could use a predictive model to predict who is over-50 (using Facebook Likes) and use the predictions to decide whom to target with the campaign.

The Facebook Likes of a user are the set of Facebook pages that the user chose to "Like" on the platform (we capitalize "Like", as have prior authors, to distinguish the act

9. Thanks to the authors of the prior study, Kosinski et al. (2013), for sharing the data.

on Facebook). So, we represent each Facebook page as a binary feature that takes a value of 1 if the user Liked the page and a value of 0 otherwise. We kept only the pages that were Liked by at least 1,000 users, leaving us with 10,822 binary features. The target variable for modeling is also binary and takes a value of 1 if the user is more than 50 years old, and a value of 0 otherwise. We use 70% of the data to train a logistic regression model. In our fictitious setting, the model is used by a decision system that targets the top 1% of users with the highest probability of being an older person, which (in our sample) implies sending promotional content to the users with a probability greater than 41.1%. We use the system to decide which of the held out 30% of users to target.

Importantly, while the system could generate a lot of value to the firm, we need to consider users sense of privacy and how they might feel about being targeted with the promotional campaign. For example, some users may feel threatened by highly personalized offers ("How do they know this about me?") and thus may be interested in knowing why they were targeted (see Chen et al. (2017) for a more detailed discussion). Such users may be unlikely to be interested in the intricacies of the model but rather in the data about their behavior that was used to target them with promotional content. If that is the case, framing explanations in terms of comprehensible input features (e.g., Facebook Likes) is critical.

One approach is to use importance weights to rank Facebook pages according to their feature importance (as computed by a technique such as SHAP) and then show the user the topmost predictive pages that she (or he) Liked. However, given the large number of features (Facebook pages), computing weights in a deterministic fashion is intractable. SHAP circumvents this issue by sampling the space of feature combinations, resulting in sampling-based approximations of the influence of each feature on the prediction. However, the downside is that the estimates may be far from the real values, which may lead to inconsistent results. For example, if we were to use the topmost important features to explain a decision, we should consider whether different runs of a non-deterministic method repeatedly rank the same pages as the most important ones. Unfortunately, as we will show, the set of the topmost important features becomes increasingly inconsistent (across different runs of SHAP) as the number of features increases.

For instance, in our holdout data set there is a 34-year-old user who would be targeted with an ad for older persons (the model predicts a 42% probability that this user is at least 50 years old). So, as an example, suppose this user wants to know why he or she is being targeted. Let's say that we have determined that showing the top-3 most important features makes sense for this application. Table 8 shows the top-3 most predictive pages according to their SHAP values (importance weights) for the system decision. The table shows the result of running SHAP five times to compute the importance weights, each time sampling 4,100 observations of the space of feature combinations.[10] Because SHAP uses sampling-based approximations, we can see that SHAP values vary every time we compute them, resulting in different topmost predictive pages. Importantly, while some pages appear recurrently, only Paul McCartney appears in all 5 approximations.

10. We use the SHAP implementation provided here: `https://github.com/slundberg/shap/`. At the moment of writing, the default sample size is $2048+2m$, where m is the number of features with a non-default value. Our choice of $4,100$ is larger than the SHAP implementation's default sample size for all of the experiments we run.

Approximation 1	**Approximation 2**	**Approximation 3**	**Approximation 4**	**Approximation 5**
Elvis Presley (0.1446)	Paul McCartney (0.1471)	Paul McCartney (0.1823)	Paul McCartney (0.1541)	Elvis Presley (0.1582)
Bruce Springsteen (0.1302)	William Shakespeare (0.1321)	Neil Young (0.1676)	Elvis Presley (0.1425)	Paul McCartney (0.1489)
Paul McCartney (0.1268)	Brain Pickings (0.1319)	The Hobbit (0.1417)	Leonard Cohen (0.1359)	Bruce Springsteen (0.1303)
Importance weights (SHAP values) shown in parentheses.				

Table 8: Topmost predictive pages and their SHAP values for a single decision to target our example user with the over-50 ad.

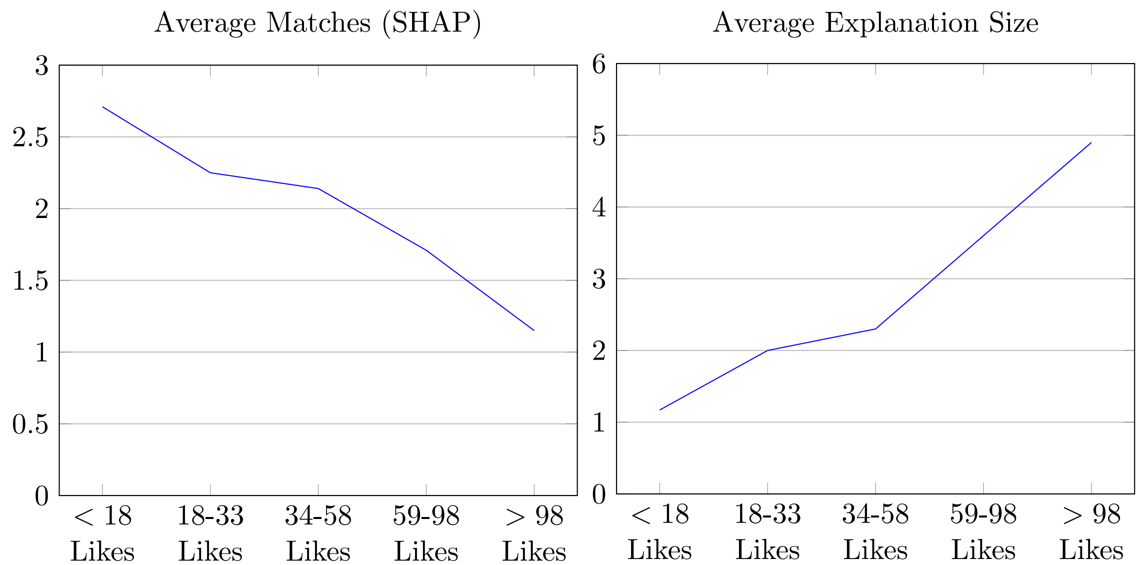
As we will show in more detail below, this inconsistency is the consequence of using SHAP to estimate importance weights for too many features. This specific user Liked 64 pages, which is not an unusually large number of Likes—more than a third of the targeted users in the holdout data set have at least that many Likes. There are (at most) 64 non-zero SHAP values to estimate, making the task significantly simpler than if we had to estimate importance weights for all 10,822 features. However, SHAP proves unreliable to find the most predictive pages (let alone to estimate the importance weights for each page). We increased the sample size for SHAP to observe when the estimates became stable for this particular task (note that we already were running SHAP with a larger sample size than the default). For this specific user, it took 8 times more samples from the feature space for the same topmost pages to show consistently across all approximations, increasing computation time substantially (from 3 to 21 seconds per approximation on a standard laptop). This time would increase dramatically for data settings with hundreds of non-zero features, which are not uncommon (e.g., see Chen et al., 2017; Perlich et al., 2014).

In contrast, counterfactual explanations were found in a tenth of a second (on the same laptop), five of which we show in Table 9. Each explanation consists of a subset of Facebook pages that would change the targeting decision if it were removed from the set of pages Liked by the user. In other words, each of the sets shown in Table 9 is an explanation in its own right, representing a minimum amount of evidence that (if removed) changes the decision. Importantly, these explanations are short, consistent (because they are generated in a deterministic fashion), and directly tied to the decision-making procedure.

As an additional systematic demonstration of the negative impact that an increasing number of features may have on the consistency of sampling-based feature-importance approximations, we show how the more pages a user has Liked, the more inconsistent the set of the top three most important pages becomes. The process we used is as follows. First, we picked a random sample of 500 users in the holdout data that would be targeted by the system (as described above). Then, we applied SHAP five times to approximate the importance weights of the features used for each of the 500 targeting decisions (sampling 4,100 observations of the feature space each time). Finally, for each targeting decision, we counted the number of pages that appeared consistently in the top three most important pages across all five approximations. We call this the number of matches. Thus, if the

Explanation 1	The user would not be targeted if {Paul McCarney} were removed.
Explanation 2	The user would not be targeted if {Elvis Presley} were removed.
Explanation 3	The user would not be targeted if {Neil Young} were removed.
Explanation 4	The user would not be targeted if {Leonard Cohen} were removed.
Explanation 5	The user would not be targeted if {Brain Pickings} were removed.

Table 9: Counterfactual explanations for a single decision to target our example user with the over-50 ad.



(a) Average matches by quantile

(b) Average size by quantile

Figure 3: Variations in explanations by number of Likes

approximations were consistent, we would expect the same three pages to appear in the top three pages of all approximations, and there would be three matches. In contrast, if the approximations were completely inconsistent, no pages would appear in the top three pages of all five approximations and there would be no matches. It took about an hour to run this experiment on a standard laptop.

The result of the experiment is in Figure 3a, which shows the average number of matches by quantile. As predicted, SHAP approximations are not consistent for users who have Liked many pages. For the largest instances, most cases have only one page that appears in all five SHAP runs. In order to contrast SHAP with counterfactual explanations, we ran our algorithm to find one counterfactual explanation for each of the 500 targeting decisions, which took 15 seconds on a standard laptop. The results are shown in Figure 3b, which shows the average size of counterfactual explanations by quantile.[11] From the figure, we can see that explanations are larger for users who Liked many pages but remain relatively small considering the number of features present, which concurs with the findings by Chen et al. (2017).

Finally, in this case study we also adjust our method to incorporate domain-specific preferences ("costs") and showcase how they can lead to more comprehensible explanations. The explanations we have shown so far (in both case studies) were generated using the heuristic search procedure proposed by Martens and Provost (2014), which does not consider the relevance of the various possible explanations and was designed to find the smallest explanations first. Nonetheless, short explanations may include Likes of relatively uncommon pages, which may be unfamiliar to the person analyzing the explanation. To illustrate how domain preferences can be taken into account when generating explanations of decisions, let's say that for our problem, explanations with highly specific Likes are problematic for a feature-based explanation. The recipient of the explanation is much less likely to know these pages, so he or she would be better served with explanations using popular pages. To this end, we can adjust the heuristic search (as discussed in Section 3.4) to find explanations that include more relevant—viz., more popular—pages by associating lower costs to their removal from an instance's input data. Specifically, we adjust the heuristic search so that it penalizes less-popular pages (those with fewer total Likes) by assigning them a higher cost.

Table 10 shows some examples of how the first explanation found by the algorithm changes depending on whether the relevance heuristic is used. As expected, the explanations found when using the relevance heuristic can include more pages than the "shortest first" search; however, those pages are also more popular (as evidenced by their total number of Likes). Importantly, these examples show how the search procedure can be easily adapted to find context-specific explanations. In this case, the user may be interested in finding explanations with popular pages, but the search could also be adjusted to show first the explanations with pages that were recently Liked by the user or that have pages more closely related to the advertised product.

5.3 Study 3: System Decisions with Multiple Models

For our third case study, we illustrate the advantages of our proposed approach when applied to complex systems, including ones that use multiple models to make decisions. We use the data set from the KDD Cup 1998, which is available at the UCI Machine Learning Repository. The data set was originally provided by a national veterans organization that wanted to maximize the profits of a direct mailing campaign requesting donations. Therefore, the business problem consisted of deciding which households to target with direct mails. Importantly, one could approach this problem in several ways, such as:

1. Using a regression model to predict the amount that a potential target will donate so that we can target her if that amount is larger than the break-even point.

11. Recall that targeting decisions may have several counterfactual explanations. The numbers we report here are the average sizes of the first explanation we found for each targeting decision.

User ID	First explanation found (WITHOUT the relevance heuristic)	First explanation found (WITH the relevance heuristic)
11	‘It’s a Wonderful Life’ (1,181 Likes) ‘JESUS IS LORD!!!!!!!!!!!!!!!!!!!!!!!!!!!! if you know this is true press like. :)’ (1,291 Likes)	‘Reading’ (47,288 Likes) ‘American Idol’ (15,792 Likes) ‘Classical’ (8,632 Likes)
38	‘The Hollywood Gossip’ (1,353 Likes) ‘Remember those who have passed. Press Like if you’ve lost a loved one’ (2,248 Likes)	‘Pink Floyd’ (43,045 Likes) ‘Dancing With The Stars’ (5,379 Likes) ‘The Ellen DeGeneres Show’ (16,944 Likes) ‘American Idol’ (15,792 Likes)
108	‘Six Degrees Of Separation - The Experiment’ (3,373 Likes) ‘They’re, Their, and There have 3 distinct meanings. Learn Them.’ (3,842 Likes)	‘Star Trek’ (11,683 Likes) ‘Turn Facebook Pink For 1 Week For Breast Cancer Awareness’ (12,942 Likes)
413	‘Sarcasm as a second language’ (1,540 Likes) ‘RightChange’ (3,842 Likes)	‘Reading’ (47,288 Likes) ‘Pink Floyd’ (43,045 Likes) ‘Where the Wild Things Are’ (13,781 Likes) ‘Proud to be an American’ (3,938 Likes)

Table 10: First counterfactual explanations found

2. Using a classification model to predict whether a potential target will donate more than the break-even point so that we can target her if this is the case.

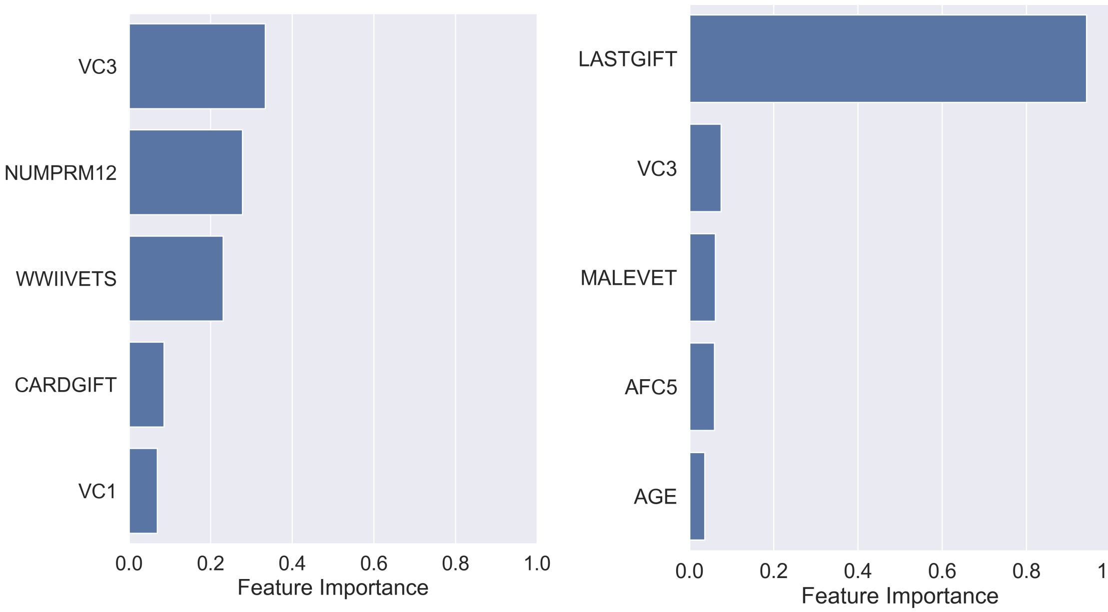
3. Using a classification model to predict the probability that a potential target will donate and a regression model to predict the amount if the potential target were to donate. By multiplying together the results of these two models, one could obtain the expected donation amount and send a direct mail if the expected donation is larger than the break-even point.

To showcase system decisions that incorporate multiple models, we illustrate our generalized framework using the third approach, which is also the one that was used by the winners of the KDD Cup 1998.

We use XGBoost for both regression and classification using 70% of the data and the following subset of features: Age of Household Head (AGE), Wealth Rating (WEALTH2), Mail Order Response (HIT), Male active in the Military (MALEMILI), Male Veteran (MALEVET), Vietnam Veteran (VIETVETS), World War two Veteran (WWIIVETS), Employed by Local Government (LOCALGOV), Employed by State Government (STATEGOV), Employed by Federal Government (FEDGOV), Percent Japanese (ETH7), Percent Korean (ETH10), Percent Vietnamese (ETH11), Percent Adult in Active Military Service (AFC1), Percent Male in Active Military Service (AFC2), Percent Female in Active Military Service (AFC3), Percent Adult Veteran Age 16+ (AFC4), Percent Male Veteran Age 16+ (AFC5), Percent Female Veteran Age 16+ (AFC6), Percent Vietnam Veteran Age 16+ (VC1), Percent Korean Veteran Age 16+ (VC2), Percent WW2 Veteran Age 16+ (VC3), Percent Veteran Serving After May 1975 Only (VC4), Number of promotions received in the last 12 months (NUMPRM12), Number of lifetime gifts to card promotions to date (CARDGIFT), Number of months between first and second gift (TIMELAG), Average dollar amount of gifts to date (AVGGIFT), and Dollar amount of most recent gift (LASTGIFT).

In order to motivate the problem, suppose that a system uses the classification and regression models on the holdout 30% of data to target the 5% of households with the largest (estimated) expected donations, essentially targeting the most profitable households with a limited budget. In this case, both the targeters and the targeted may be interested in explanations for why the system decided to send any particular direct mail. This is a particularly challenging problem for methods designed to explain model predictions (not decisions), since the system makes decisions using more than one model. Therefore, it is possible that the most important features for predicting the probability of donation are not the same as the most important features for predicting the donation amount, and so determining which features led to the targeting decision is not straightforward.

To illustrate this better, consider one targeted household in the holdout data, for which we computed SHAP values for its predicted probability of donating (given by the classification model) and its predicted donation amount (given by the regression model). We normalized the SHAP values for each model prediction so that the sum of the values adds up to 1. The top 5 most important features for the probability prediction and the regression prediction are shown in Figure 4a and Figure 4b respectively. Interestingly, only VC3 (percent of 16+ WW2 veterans in the household) is part of the most important features for both the classification model and the regression model. Importantly, we cannot explain



(a) Top features for probability

(b) Top features for amount

Figure 4: Features with largest importance weights

the targeting decision from these figures alone: even though we know the most important features for each prediction, there is no way of telling what was actually vital for the system to make the targeting decision. Was the household targeted because of the size of the last gift (LASTGIFT)? Or would the households high probability of donating justify the targeting decision even if LASTGIFT had a smaller value?

As per our earlier discussions, SHAP may be repurposed to compute feature importance weights for system decisions that incorporate multiple models by transforming the output of the system into a scoring function that returns 1 if the household is targeted and returns 0 otherwise. However, as we have similarly shown for other problems, acquiring feature importance weights for decisions made based on expected donations (rather than amounts or probabilities) would still not explain the system decisions. In contrast, counterfactual explanations can transparently be applied to system decisions that involve more than one model. Specifically, by defining the predicted expected donation as a scoring function (which is the result of multiplying the predictions of the two models), we can use the same procedures showcased in the previous examples to find explanations for targeting decisions. Table 11 shows the explanations found for the targeted household discussed above.

Interestingly, some of the highest-scoring SHAP features, shown in Figures 4, are not present in any of the explanations (e.g., MALEVET), whereas some features that are present in some explanations do not have large SHAP values (e.g., AVGGIFT). In fact, AVGGIFT had a negative SHAP value in the regression model (meaning we would expect its impact on the non-default decision to be negative), but it appears in all explanations! This example illustrates the importance of defining explanations in terms of decisions and not predictions, particularly when dealing with complex, non-linear models, such as XGBoost.

Features	Explanations					
	1	2	3	4	5	6
AGE						↓
WWIIVETS	↑					
VC1			↓			
VC2					↑	
VC3		↑				
NUMPRM12		↑	↑	↑	↑	↑
CARDGIFT				↑		
AVGGIFT	↑	↑	↑	↑	↑	↑
LASTGIFT	↑	↑	↑	↑	↑	↑
↑ means household was targeted because feature is **above** average.						
↓ means household was targeted because feature is **below** average.						

Table 11: Explanations for targeting decision

More specifically, because SHAP attempts to evaluate the overall impact of features on the model prediction, it averages out the negative and positive impacts that features have on the prediction when removed alongside all other feature combinations. Hence, if a feature has a large negative impact in one case and several small positive impacts in other cases, that feature may have a negative SHAP value (if the single negative impact is greater than the sum of the small positive impacts). This behavior is the same that we illustrated in Section 4.3 (Example 3), which of course would be counterproductive when trying to understand the influence of features on the decision making. Averaging out the impact of features over all feature combinations hides the fact that (in non-linear models) features may provide evidence in favor or against a decision depending on what other features are removed, which explains why AVGGIFT had a negative SHAP value but is present in the explanations shown in Table 11.

6. Discussion

The previous studies illustrate various advantages of counterfactual explanations over importance weighting methods. The first study shows that knowing the importance weight of features is not enough to determine how the features affect system decisions. The second study demonstrates the strengths of counterfactual explanations in the presence of high-dimensional data. In particular, the study shows that sampling-based approximations of importance weights get worse as the number of features increases. Counterfactual explanations sidestep this issue because small subsets of features are usually enough to explain decisions. Moreover, the study showcased a heuristic procedure to search and sort counterfactual explanations according to their relevance. Finally, the third study shows that importance weights may be misleading when decisions are made using multiple (and complex) models. More specifically, we see a real instance of the phenomenon we showed in Section 4.3, in which features with negative SHAP weights may in fact have a positive effect on system decisions.

It has been argued that a disadvantage of counterfactual explanations is that each instance (decision) usually has multiple explanations (Molnar, 2019); this is also referred to as the Rashomon effect. The argument is that this is inconvenient because people may prefer simple explanations over the complexity of the real world. This issue may be exacerbated as the number of features increases because the number of counterfactual explanations may grow exponentially. In contrast, most importance weighting methods converge to a unique solution (e.g., Shapley values in the case of SHAP), regardless of the number of features.

However, our second case study suggests that importance weighting methods may actually not scale well when the number of features increases because their approximations may become inconsistent. Moreover, objective measures of relevance (e.g., number of Likes in our Facebook case study) may be incorporated as part of the heuristic procedures used to find counterfactual explanations. Thus, the fact that the number of counterfactual explanations may grow exponentially is not necessarily problematic. Our study shows that short, consistent, and relevant explanations are significantly faster to find than computing importance weights, even when the number of features is large.

Something that was not explored in the case studies was the sensitivity of the counterfactual explanations to the method used to deal with missing values. This is an interesting direction for future research, as we would expect distinct alternatives for dealing with missing features to affect explanations differently. For example, if features are correlated, mean imputation and retraining the model without the removed feature may produce different results. For instance, a decision may change when imputing the mean for a removed feature, but if instead the missing feature is dealt with by using a model trained without that feature (Saar-Tsechansky and Provost, 2007), the decision may not change when removing the feature because other features may capture most of the information given by the removed feature. Therefore, while our proposed framework would work with either approach, future research should assess the advantages of each approach in different settings.

Moreover, this study compared importance weights with a specific type of counterfactual explanations (formally defined in Section 3.2). Specifically, our explanations are defined in terms of counterfactual worlds in which some of the features are absent when making decisions. Nonetheless, there are other types of counterfactual worlds that may be of interest when explaining decisions. For example, in our first case study, we showed that some loan applicants were denied credit because the amount they requested was too large (i.e., the decision changed when we removed the loan amount feature). While this explains the credit denial decision, these applicants may instead be interested in the maximum amount they could ask for, so that they are no longer denied credit. Such a counterfactual explanation could be defined as a set of "minimal" feature adjustments that changes the decision.

Other researchers have proposed various methods to obtain such counterfactual explanations. For example, in the context of explaining predictions (not decisions), Wachter et al. (2017) define counterfactual explanations as the smallest change to feature values that changes the prediction to a predefined output. Thus, they address explanations as a minimization problem in which larger (user-defined) distances between counterfactual instances and the original instance are penalized more. Their method, however, focuses on gradient-based models, does not work with categorical features, and may require access to the machine learning method used to learn the model (which usually is not available for deployed systems). Tolomei et al. (2017) define counterfactual explanations in a similar way, but instead propose how to find such explanations when using tree-based methods. Other counterfactual methods have also been implemented in the Python package Alibi.[12] The package includes a simple counterfactual method loosely based on Wachter et al. (2017), as well as an extended method that uses class prototypes to improve the interpretability and convergence of the algorithm (Van Looveren and Klaise, 2019).

Another key assumption behind all the instance-level explanation methods discussed in this paper (feature importance as well as counterfactual) is that examining an instance's features will make sense to the user. This presumes at least that the features themselves are comprehensible. This would not be the case, for example, if the features are too low-level or for cases where the features have been obfuscated, for example to address privacy concerns (see e.g., the discussion of "doubly deidentified data" by Provost et al. (2009)).

Relatedly, another promising direction for future research is to study how users actually perceive these different sorts of explanations in practice. In particular, it would be interesting to analyze the impact that various types of explanations have on users' adoption of AI systems and their decision-making performance. Settings where the decisions made by deployed AI systems are closely monitored by users (see Lebovitz et al. (2019) for a clear example) would be ideal for such a study.

7. Conclusion

This paper examines the problem of explaining data-driven decisions made by AI decision-making systems from a causal perspective: if the question we seek to answer is why did the system make a specific decision, we can ask which inputs caused the system to make its decision? This approach is advantageous because (a) it standardizes the form that an explanation can take; (b) it does not require all features to be part of the explanation, and (c) the explanations can be separated from the specifics of the model. Thus, we define a (counterfactual) explanation as a set of features that is causal (meaning that removing the set from the instance changes the decision) and irreducible (meaning that removing any subset of the features in the explanation would not change the decision).

Importantly, this paper shows that explaining model predictions is not the same as explaining system decisions, because features that have a large impact on predictions may not have an important influence on decisions. Moreover, we show through various examples and case studies that the increasingly popular approach of explaining model predictions using importance weights has significant drawbacks when repurposed to explain system decisions. In particular, we demonstrate that importance weights may be ambiguous or even misleading when the goal is to understand how features affect a specific decision.

Our work generalizes previous work on counterfactual explanations in at least three important ways: (i) we explain system decisions (which may incorporate predictions from

12. See `https://github.com/SeldonIO/alibi`

several predictive models) rather than model predictions, (ii) we do not enforce any specific method to remove features, and (iii) our explanations can deal with feature sets with arbitrary dimensionality and data types. Finally, we also propose a heuristic procedure that allows the tailoring of explanations to domain needs by introducing costs—for example, the costs of changing the features responsible for the decision.

References

Robert Andrews, Joachim Diederich, and Alan B Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8 (6):373–389, 1995.

Vicky Arnold, Nicole Clark, Philip A Collier, Stewart A Leech, and Steve G Sutton. The differential use and effect of knowledge-based system explanations in novice and expert judgment decisions. *MIS Quarterly*, pages 79–97, 2006.

Bart Baesens, Tony Van Gestel, Stijn Viaene, Maria Stepanova, Johan Suykens, and Jan Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6):627–635, 2003.

Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, volume 8, page 1, 2017.

Daizhuo Chen, Samuel P Fraiberger, Robert Moakler, and Foster Provost. Enhancing transparency and control when drawing data-driven inferences about individuals. *Big Data*, 5(3):197–212, 2017.

Maxime C Cohen, C Daniel Guetta, Kevin Jiao, and Foster Provost. Data-driven investment strategies for peer-to-peer lending: A case study for teaching data science. *Big Data*, 6 (3):191–213, 2018.

Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems*, pages 24–30, 1996.

Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 598–617. IEEE, 2016.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, pages 497–530, 1999.

Henrik Jacobsson. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

Ujwal Kayande, Arnaud De Bruyn, Gary L Lilien, Arvind Rangaswamy, and Gerrit H Van Bruggen. How incorporating feedback mechanisms in a DSS affects DSS evaluations. *Information Systems Research*, 20(4):527–546, 2009.

Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

Michael T Lash, Qihang Lin, W Nick Street, and Jennifer G Robinson. A budget-constrained inverse classification framework for smooth classifiers. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1184–1193. IEEE, 2017.

Steve LaValle, Eric Lesser, Rebecca Shockley, Michael S Hopkins, and Nina Kruschwitz. Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 52(2):21–32, 2011.

Sarah Lebovitz, Natalia Levina, and Hila Lifshitz-Assaf. Doubting the diagnosis: How artificial intelligence increases ambiguity during professional decision making. *Available at SSRN 3480593*, 2019.

Vincent Lemaire, Raphael Féraud, and Nicolas Voisine. Contact personalization using a score understanding method. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 649–654. IEEE, 2008.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

David Martens and Foster Provost. Explaining data-driven document classifications. *MIS Quarterly*, 38(1):73–100, 2014.

David Martens, Bart Baesens, Tony Van Gestel, and Jan Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1466–1476, 2007.

Julie Moeyersoms, Brian d'Alessandro, Foster Provost, and David Martens. Explaining classification models built on high-dimensional sparse data. In *Workshop on Human Interpretability in Machine Learning: WHI 2016, June 23, 2016, New York, USA/Kim, Been [edit.]*, pages 36–40, 2016.

Christoph Molnar. Interpretable machine learning, see 18.1 counterfactual explanations. `https://christophm.github.io/interpretable-ml-book/counterfactual.html`, 2019. Accessed: 2019-12-11.

Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine Learning*, 95(1):103–127, 2014.

Foster Provost. Understanding decisions driven by big data, 2014. URL `https://www.youtube.com/watch?v=l7XPN0XUCkw`.

Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking.* O'Reilly Media, Inc., 2013.

Foster Provost, Brian Dalessandro, Rod Hook, Xiaohan Zhang, and Alan Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 707–716. ACM, 2009.

Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. Counterfactual explanation algorithms for behavioral and textual data. *arXiv preprint arXiv:1912.01819*, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of Machine Learning Eesearch*, 8(Jul):1623–1657, 2007.

Ethan L Schreiber, Richard E Korf, and Michael D Moffitt. Optimal multi-way number partitioning. *Journal of the ACM (JACM)*, 65(4):24, 2018.

Galit Shmueli and Otto R Koppius. Predictive analytics in Information Systems research. *MIS Quarterly*, pages 553–572, 2011.

Erik Štrumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18, 2010.

Erik Štrumbelj, Igor Kononenko, and M Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, 68(10): 886–904, 2009.

Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 465–474. ACM, 2017.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GPDR. *Harv. JL & Tech.*, 31:841, 2017.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022, 2015.