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

Carlos Fernández-Loría 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

We examine counterfactual explanations for explaining the decisions made by model-based AI systems. The counterfactual approach we consider defines an explanation as a set of the system's data inputs that causally drives the decision (i.e., changing the inputs in the set changes the decision) and is irreducible (i.e., changing any subset of the inputs does not change the decision). We (1) demonstrate how this framework may be used to provide explanations for decisions made by general, data-driven AI systems that may incorporate features with arbitrary data types and multiple predictive models, and (2) propose a heuristic procedure to find the most useful explanations depending on the context. We then contrast counterfactual explanations with methods that explain model predictions by weighting features according to their importance (e.g., SHAP, LIME) and present two fundamental reasons why we should carefully consider whether importance-weight explanations are well-suited to explain system decisions. Specifically, we show that (i) features that have a large importance weight for a model prediction may not affect the corresponding decision, and (ii) importance weights are insufficient to communicate whether and how features influence decisions. We demonstrate this with several concise examples and three detailed case studies that compare the counterfactual approach with SHAP to illustrate various conditions under which counterfactual explanations explain data-driven decisions better than 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, stakeholders are often skeptical and reluctant to adopt systems without the ability to explain system decisions, even if the systems 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. For example, explanations may help customers understand the reasoning behind decisions that affect them. Other individuals, 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. Thus, 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 of the model (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 of newly introduced "model-agnostic" methods that can produce importance weights for any predictive model: 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.

As a main contribution, this paper presents two fundamental reasons why importance-weight explanations are not well-suited to explain data-driven decisions made by AI systems despite their popularity. 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 explain model predictions may portray an inaccurate picture of how input data influences 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 prior work that clarifies this for research or for practice.

An alternative to importance-weight explanations are counterfactual explanations—explanations explicitly designed to explain system decisions. 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.

To our knowledge, the first framework for counterfactual explanations for decisions was introduced in this journal to explain document classifications (Martens and Provost, 2014). The framework, which is model-agnostic and has been shown to scale with very large numbers of features, has since been applied to other sparse high-dimensional settings (Moeyersoms et al., 2016; Chen et al., 2017; Ramon et al., 2020), but 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). Therefore, as a second contribution, we extend the framework as introduced by Martens and Provost to provide explanations for decisions made by general, data-driven AI systems that may incorporate features with arbitrary data types and multiple predictive models. In addition, we propose and showcase a heuristic procedure that may be used to search and sort counterfactual explanations according to their context-specific relevance.

Finally, and as a third contribution, we demonstrate these extensions and illustrate the advantages of the 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. We present three simple examples showing the advantages, and then we present three business case studies using real-world data to show that the differences between the approaches are not purely academic.

2. AI Systems and Explanations

We focus 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. These AI systems may or may not have been built using machine learning—this paper studies explaining the decisions of a system-in-practice, not on how the system was built.[1]

1. However, explaining the decisions of the system-in-practice can also help to understand the system-building process, for example, by debugging training data (Martens and Provost, 2014).

2.1 Explaining models and their predictions

Over the past several decades, many researchers have worked on explaining predictive models—which is not the same as explaining the decisions made with such models, as we discuss in detail in subsequent sections. Because symbolic models, such as decision trees, are often considered straightforward to explain when they are small, 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 attempt to explain the model as a whole, rather than explaining particular decisions made. As Martens and Provost (2014) point out, this can be viewed as explaining every possible decision the model might make—but the methods are not designed to explain individual decisions. Furthermore, the model itself being explainable does not necessarily imply that individual decisions made by the model are explainable.

Another related approach is to produce models explicitly designed to be both accurate and comprehensible (Wang and Rudin, 2015; Angelino et al., 2017). However, these studies are about building intelligible systems, not explaining the decisions of a system currently in use. Therefore, the methods proposed in these studies are meant to replace existing black-box models with new models that are easier to interpret. This may not always be possible, particularly if the goal is to explain the decisions of a system that incorporates multiple models or subsystems.

A fundamentally different approach (and one of the primary explanation techniques we analyze in this paper) is to explain the predictions of complex models by associating a weight to each feature in the model. 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 estimated probability of default (based on one of the case studies we present below). These importance weights were generated using SHAP (Lundberg and Lee, 2017), which we will discuss in more detail in the following sections. As the example shows, each weight in the explanation represents the attributed impact of its respective feature on the prediction. Thus, the weight associated with the loan amount feature ('loan_amnt') implies that the feature is attributed an increase of (roughly) 2.5% in the estimated probability of default for that individual.

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

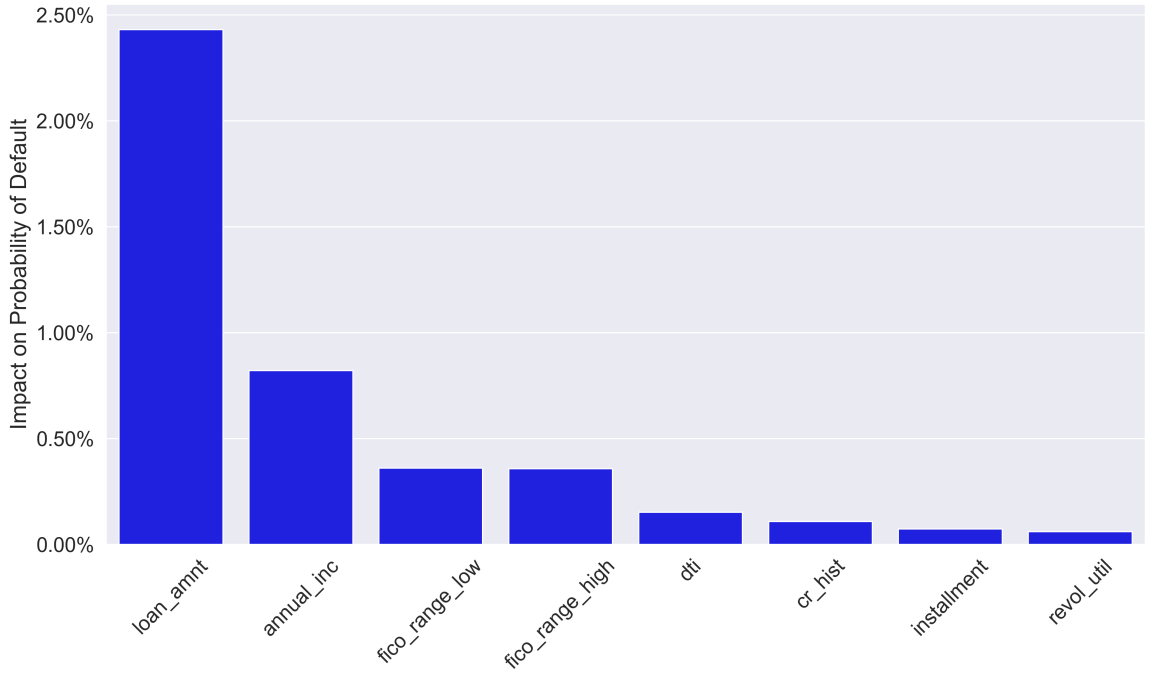


Figure 1: Example of an importance-weight explanation for a model prediction

(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. These are some of the reasons why importance-weight methods have become one of the most popular approaches to explain model predictions in the last few years.

One notable challenge, however, is the computation of the weights. For example, 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), typically by comparing the original model's output with the output obtained when the information given by a specific feature is removed (e.g., by imputing a default value for the feature). Unfortunately, interactions between features may lead to ambiguous explanations because the order in which features are removed may affect the importance attributed to each feature. Researchers have proposed to address this issue by comparing the model predictions when removing all possible subsets of features (Štrumbelj et al., 2009), but this is intractable when the number of features is large. Therefore, recent formulations (such as SHAP) have attempted to reduce computation time by sampling the space of feature combinations, 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 may not be adequate to explain system decisions (as opposed to model predictions) because they don't communicate how the features actually influence decisions, as we illustrate with several examples in this study. For instance, Figure 1 does not communicate how the different features could influence the decision to

grant credit. 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 (as we show below) only a handful of the features may be critical for understanding the system's decisions.

2.2 Explaining system decisions

As mentioned, the focus of our study is on AI systems that make, support, or recommend discrete decisions. Discrete decision making is closely related to classification, and indeed the subtle distinction can often 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. For example, scores may correspond to estimated probabilities of defaulting on credit, and individuals may be classified as defaulting or not based on their probability of defaulting. Decision-making problems may also be modeled as "classification tasks" by associating a class with each decision (e.g., "grant credit" and "do not grant credit"), which is related to (but usually not the same as) classifying individuals according to labels in the data. For example, estimated probabilities of class membership are often combined with application-specific information on costs and benefits to produce a next stage of more nuanced scores (e.g., the expected profits of granting credit). These scores may then be used by a system to make decisions using a chosen threshold appropriate for the problem at hand (Provost and Fawcett, 2013). Thus, for example, a credit scoring system may decide to extend credit to an individual with a relatively high probability of default if the interest rate is high.

Critically, this implies that the final output of the system (i.e., the decision) may not correspond to the labels in the training data. As an additional 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—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, which is not the same as predicting whether a customer will make a purchase (and because of selection bias and other complications, we often patently would not want to learn models based on training data of who was targeted).

Explaining the decisions made by intelligent systems has received both practical and research attention from the IS community for decades (Gregor and Benbasat, 1999). Martens and Provost (2014) provide an overview of the IS literature that frames, motivates, and explains the importance of explaining system decisions for system adoption, improvement,

and use. Notably, prior work shows 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. This result highlights the need for methods to make the decisions of such AI systems more transparent.

3. Counterfactual explanations

We now present counterfactual explanations for system decisions in detail, showing where we generalize from prior work. In general, counterfactual explanations describe a counterfactual situation in the form: "if X had not occurred, Y would not have occurred." Thus, counterfactual explanations can be used to explain system behavior in the following narrow[2] and causal sense. The "outcome" (Y) is the focal decision made by the system, and the "causes" (X) are the data inputs that resulted in the system making the decision. Therefore, in our context, a counterfactual explanation consists of a set of data inputs that, when changed, result in a different system decision. For instance, in credit scoring, one could explain a credit denial decision by saying "if the applicant had a higher annual income, the system would have granted credit."

The idea of taking a causal perspective to explain system decisions with counterfactuals was first proposed (to our knowledge) in MISQ (Martens and Provost, 2014), and others have followed with similar counterfactual approaches since then (see Molnar, 2019; Verma et al., 2020, for examples). Martens and Provost (2014) define counterfactual explanations in terms of input data that would change the decision if it were not present. Unfortunately, that paper did not seem to make clear the general nature of the counterfactual explanations. The counterfactual explanations originally were proposed for document classification, and while they subsequently have been used in other business settings (Moeyersoms et al., 2016; Chen et al., 2017; Ramon et al., 2020), this initial use has led several researchers to state in their work that the framework is specific to document classification and/or categorical features (see, e.g., Molnar, 2019; Wachter et al., 2017; Biran and Cotton, 2017; Tamagnini et al., 2017). We recast and generalize this framework to be more broadly applicable.

2. We do not intend to add to the philosophy or long debate on counterfactual theories of causation (Menzies and Beebee, 2019). Understanding what causes computer system decisions, albeit narrow as compared to understanding causality of natural phenomena, is facilitated because we can observe directly the counterfactual "if X had not occurred, Y would not have occurred," by changing the system inputs and observing the system output. This of course does not necessarily say anything about causal relationships outside the behavior of the computer system, such as in the data-generating process.

3.1 Framing counterfactual explanations from an evidence-based perspective

Counterfactual explanations consist of hypothetical realities that differ from the observed facts, but in order for these explanations to be useful, these realities must be plausible. This leads to three fundamental challenges when using counterfactual explanations to explain system decisions. First, we must define what plausible means, which will likely vary across contexts. Second, searching for all potential explanations may be intractable. Third, there may be multiple explanations for each decision, so we may need to define a criteria to choose (or at least rank) explanations.

Similarly to importance-weight methods that assess feature importance by simulating lack of knowledge about features, we argue that some of these challenges may be partially addressed by framing counterfactual explanations in terms of "absent evidence": explanations may be framed in terms of features that change the system decision when the evidence they provide is no longer present. 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 not have flagged the transaction absent this location.[3] 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.

This perspective offers several advantages to address the challenges mentioned above. First, absent evidence may be used to define a reasonable set of plausible changes. For instance, in the example above, "removing evidence" from a model-based decision-making procedure may imply replacing the location of the transaction with the country where the cardholder lives or replacing the consumption category with the cardholder's most common consumption category. Second, it narrows the set of potential explanations substantially because the point is not to consider all the different ways in which the features could be changed, but rather what would be the effect of not having some particular evidence. Third, we may rank explanations according to the relevance of the features in them (e.g., location may be easier to communicate than consumption category). We discuss these advantages in more detail below, after formally defining counterfactual explanations using the evidence-based perspective.

Another subtle implication of this perspective is that its 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

3. 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.

was not enough evidence of a non-default class".[4] Thus, as with prior work, we focus 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 setting each feature in the set to some predetermined counterfactual value (e.g., the mean) causes the system decision to change.[5] Irreducible means that no proper subset of the explanation is causal. The importance of an explanation being causal is straightforward: the decision would have been different if not for the specific values of the features in the set (i.e., the "evidence"). 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.

Formally, we define counterfactual explanations as follows. Consider an instance I consisting of a set of m features, $I = \{A_1, ..., A_m\}$, for which the decision-making system $C : I \rightarrow \{1, ..., k\}$ gives decision c. A feature A_i is an attribute taking on a particular value, like `income=$50,000` or `country=FRANCE`, and evidence is "removed" from a feature by setting it to some predetermined counterfactual value that makes sense in the particular application (e.g., the mean or the mode). Then, given a set of features E, $I - E$ represents instance I after setting the features in E to their respective counterfactual values, and 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, this definition 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 two important ways. First, it makes explicit how the explanations may be used for broader system decisions. Second, their practical implementation of explanations (and subsequent work) consists of removing evidence by setting features to zero, whereas we generalize to arbitrary counterfactual values.

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. Figure 2 shows some counterfactual explanations for the credit denial decision.

4. 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.
5. As mentioned, 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. Counterfactual explanations for AI system decisions relate to the former and do not necessarily tell us anything about the latter.

Explanation 1:	Credit denied because {'loan_amnt'} is above average.
Explanation 2:	Credit denied because {'annual_inc'} is below average.
Explanation 3:	Credit denied because {'fico_range_high', 'fico_range_low'} are below average.

Figure 2: Examples of counterfactual explanations for a system decision

3.3 Removing evidence from features

A vital practical question raised by our definition of counterfactual explanations is what counterfactual values should be used to "remove evidence" from features? Most explanation methods—including methods that do not provide counterfactual explanations, such as importance-weight methods—typically simulate lack of knowledge about features by replacing their values with some default value, such as the mean. 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.[6] However, it may be more appropriate to use other strategies for removing evidence in other applications, such as in the cardholder-level perspective discussed in our fraud example.

The explanation framework we present is agnostic to which method is used to define the counterfactual values associated to each feature—taking the position that this decision is domain and problem dependent. For example, Saar-Tsechansky and Provost (2007) discuss various imputation strategies for dealing with missing features when applying predictive models; any of them could be used in conjunction with this framework to define counterfactual values. In our comparison with feature-importance methods, presented below, we use the same approach that those methods use (mean imputation), so that the analysis is indeed a comparison of the two different types of methods and is not confounded by using different strategies for replacing feature values. Nevertheless, in one of the case studies, we illustrate model-based imputation as an alternative approach to demonstrate how it satisfies different needs when producing explanations.

Importantly, within a particular domain and explanation context, the system developers and domain experts should choose the most appropriate method for removing evidence, as a one-size-fits-all strategy is unlikely to work well in practice. The examples in this study are just meant as a broad guideline of when certain methods can work better than others. For example, if a manager wants to understand why a certain credit application was rejected, then using mean imputation to explain that the application was rejected because the applicant's annual income is below average could be reasonable. However, this explanation may not suffice to the applicant, particularly if the explanation is meant to be used as a recommendation to get the credit approved. In such cases, model-based imputation could be used to come up with counterfactual values that match those of similar applicants that got their credit approved.

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

3.4 A procedure for finding useful counterfactual explanations

Our 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). In this paper, and for the examples that follow, we adopt heuristic procedures instead. Algorithm 1 shows how the algorithm proposed by Martens and Provost (2014) may be used to find counterfactual explanations. Algorithm 1 generalizes the original algorithm by using $I - E$ to represent instance I after setting the features in E to their respective counterfactual values, whereas feature values were always set equal to zero in the original algorithm, which would be a specific instance of $I - E$ in our generalized framework. The second generalization will be presented presently: the introduction of a preference function.

This algorithm 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. This scoring function is then used by the search algorithm to first consider features that, when changed to their counterfactual value, 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 consider features according to their overall importance for the prediction, where the importance may be computed by a feature importance explanation technique (Ramon et al., 2020). Both heuristics have been shown to scale well in high-dimensional settings (Martens and Provost, 2014; Ramon et al., 2020).

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 little use as a guide for what to do about it. Generally, some features will lead to better explanations than others depending on the application. For example, some features may be easy to change, while others may be practically impossible to change (e.g., gender)—so while an explanation including a very expensive-to-change feature would indeed explain the decision, it would not give practicable guidance toward what could be done to affect the decision.

Therefore, we allow the incorporation of a preference function as part of the heuristic procedure to search first for the most relevant explanations. We pose the preference function as a cost function on the feature changes: the cost function associates costs to the 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 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.

Algorithm 1: Evidence-Based Explainer (EBE)

Input : $I = \{A_1, A_2, ..., A_m\}$ % Instance consisting of a set of m features
$f_c : I \rightarrow \mathbf{R}$ % Scoring function
$C : I \rightarrow \{1, 2, ..., k\}$ % Decision-making system that uses scoring function f_c
$max\_iteration = 30$ % Maximum number of iterations

Output: *explanation_list* % List of explanations.

$c = C(I)$ % The class predicted by the trained classifier.
$p = f_c(I)$ % Corresponding score of the predicted class.
explanation_list = [ ]
$i = 0$
combinations = Initialize empty list of sets ordered by scores (from lowest to largest)
Insert an empty set with score p to *combinations*
while $i < max\_iteration$ AND *combinations* is not empty **do**
    combination = pop set with the smallest score from *combinations*
    $used\_features = I - combination$
    if $C(used\_features) = c$ **then**
        foreach feature A_j in $used\_features$ **do**
            $E = combination \cup \{A_j\}$
            if E is not a superset of an explanation in *explanation_list* **then**
                $p = f_c(I - E)$
                Insert set E with score p to *combinations*
            end
        end
    else
        % Entering here implies the combination is causal
        explanation = *combination*
        % The following makes sure the combination is irreducible
        foreach set E in the power set of *combination* **do**
            if $C(I - E) \neq c$ AND E is smaller than *explanation* **then**
                explanation = E
            end
        end
        Add *explanation* to *explanation_list*
    end
    $i = i + 1$
end

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 adjusted to find the most useful explanations for the problem at hand. Doing so would only require defining a cost function $c(I, E)$—which represents the "cost" of setting the features in E to their respective counterfactual values—and sorting potential explanations in descending order according to $(f_c(I) - f_c(I - E))/c(I, E)$ (i.e., the score change per unit of cost) instead of in ascending order according to $f_c(I - E)$ (as in line 14 of Algorithm 1). Similar approaches have been suggested for inverse classification (Lash et al., 2017), any of which could be repurposed to find counterfactual explanations as defined in this paper.

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 is discussed in detail in the prior work referenced above, this is of critical importance to explain to users individual decisions made by such models. However, the social science literature on explanations suggests that referring to probabilities or statistical generalizations is usually unhelpful and that we should instead think of explanations as a conversation (Kaur et al., 2019). Thus, following this principle, our definition of counterfactual explanations as sets of features that affect decisions (see Figure 2) is more focused on starting the relevant conversation than explanations involving the importance of features to the predictive model outputs (see Figure 1).

More importantly, counterfactual 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 that importance weights can remain the same despite substantial changes to decision making (Examples 1, 2, and 3) and that features deemed unimportant by the weights can 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:

$$I = \{A_1 = 1, A_2 = 1, A_3 = 1\}, \tag{4}$$

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

where $\{A_1, A_2, A_3\}$ are binary attributes, 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 set feature values equal to zero when considering features as part of a counterfactual explanation.

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. In this context, a joining order is a permutation according to which the impact of the features on a model's prediction is considered (e.g., first A_2, then A_3, and lastly A_1). The impact of a feature corresponds to the change in the model prediction when the feature's default (counterfactual) value is replaced with the value observed for that instance, and the Shapley value consists of the average impact across permutations. We illustrate the computation of Shapley values precisely in the examples below.

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 joining orders, 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 presented in Section 5, where the number of features is much larger.

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:

$$\hat{Y}_1(I) = A_1 + A_2 + 10A_1A_3 + 10A_2A_3, \tag{6}$$

Joining orders	**Impact of A_1**	**Impact of A_2**	**Impact of A_3**
A_1, A_2, A_3	1	1	20
A_1, A_3, A_2	1	11	10
A_2, A_1, A_3	1	1	20
A_2, A_3, A_1	11	1	10
A_3, A_1, A_2	11	11	0
A_3, A_2, A_1	11	11	0
Shapley values	6	6	10

Table 1: Shapley values for $\hat{Y}_1$ and all the joining orders used in their computation.

so the prediction and the decision for instance I are $\hat{Y}_1(I) = 22$ and $C_1(I) = 1$ respectively. Note that in practice we often do not know the exact functional form of $\hat{Y}_i$ or $\hat{C}_i$, so we do not get to peek into the internals of the explained model/system and can only probe it by feeding it different inputs and seeing what the outputs are.

Table 1 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 across the joining orders, which corresponds to the Shapley values.

According to Table 1, SHAP gives A_3 a larger weight than A_1 or A_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 A_3 does not affect the decision-making procedure at all! More specifically A_3 only affects $\hat{Y}_1$ if A_1 or A_2 are already present, but if those features are present, then increasing the score does not affect the decision because $\hat{Y}_1 \geq 1$ already (implying that $C_1 = 1$ regardless of A_3). Therefore, the large "importance" of a feature for a model prediction may not imply any 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 explaining system decisions are two different tasks. We might then conclude that the issue would be solved by using SHAP to compute feature importance weights for system decisions (rather than for model predictions). Table 2 shows the Shapley values of the features with respect to the decision-making procedure C_1 instead of $\hat{Y}_i$.[7] It illustrates that A_3 indeed does not affect the decision at all. However, the next examples show that, even when importance

7. SHAP can also be used to explain non-binary categorical decisions by transforming the output of the decision system into a "scoring function" that returns 1 if the decision is the same after changing the 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 even for decisions with multiple, unordered alternatives that cannot normally be represented as a single numeric score.

Joining orders	**Impact of A_1**	**Impact of A_2**	**Impact of A_3**
A_1, A_2, A_3	1	0	0
A_1, A_3, A_2	1	0	0
A_2, A_1, A_3	0	1	0
A_2, A_3, A_1	0	1	0
A_3, A_1, A_2	1	0	0
A_3, A_2, A_1	0	1	0
Shapley values	0.5	0.5	0
There is a single counterfactual explanation: $\{A_1, A_2\}$			

Table 2: Shapley values for C_1, and all counterfactual explanations for this decision.

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 change A_1 and A_2 simultaneously, and changing any of the features individually does not change the decision. So, according to our definition from Section 3.2, there is a single counterfactual explanation, $\{A_1, A_2\}$. However, suppose we were to use the following scoring function to make decisions instead:

$$\hat{Y}_2 = A_1 A_2 \tag{7}$$

Table 3 shows the Shapley values for C_2, which are the same as for C_1 (see Table 2) because features A_1 and A_2 are equally important in both cases. However, the decision-making procedure is different because the new scoring function implies that changing either feature would change the decision. Therefore, with the new scoring function, there would be two counterfactual explanations, $\{A_1\}$ and $\{A_2\}$, but the importance weights do not capture this. This implies that importance weights do not communicate well how changing the features may change the decision.[8]

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

8. Note that Ramon et al. (2020) 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 A_1**	**Impact of A_2**	**Impact of A_3**
A_1, A_2, A_3	0	1	0
A_1, A_3, A_2	0	1	0
A_2, A_1, A_3	1	0	0
A_2, A_3, A_1	1	0	0
A_3, A_1, A_2	0	1	0
A_3, A_2, A_1	1	0	0
Shapley values	0.5	0.5	0
There are two counterfactual explanations: $\{A_1\}$ and $\{A_2\}$			

Table 3: Shapley values for C_2, and counterfactual explanations for this decision.

system decisions, because the opposite can be true as well: a feature with an importance weight of zero may have a positive effect on the decision! We illustrate this with a third example, for which we use the following scoring function:

$$\hat{Y}_3 = A_1 + A_2 - 2A_1A_2 - A_1A_3 - A_2A_3 + 3A_1A_2A_3 \tag{8}$$

Table 4 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, changing A_3 can change the decision from $C_3 = 1$ to $C_3 = 0$, as evidenced by the impact of A_3 in the first and third joining orders, but the importance weight of A_3 is 0. The counterfactual explanation framework, on the other hand, reveals that there are three counterfactual explanations in this example: $\{A_1\}$, $\{A_2\}$, and $\{A_3\}$. Thus, a feature that we might mistakenly deem as irrelevant due to its non-positive weight, is in fact just 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, 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 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.

Joining orders	**Impact of A_1**	**Impact of A_2**	**Impact of A_3**
A_1, A_2, A_3	1	-1	1
A_1, A_3, A_2	1	1	-1
A_2, A_1, A_3	-1	1	1
A_2, A_3, A_1	1	1	-1
A_3, A_1, A_2	0	1	0
A_3, A_2, A_1	1	0	0
Shapley values	0.5	0.5	0
There are three counterfactual explanations: $\{A_1\}, \{A_2\}$, and $\{A_3\}$			

Table 4: Shapley values for C_3, and counterfactual explanations for this decision.

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 reveal 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 may not be representative of the potential impact of features (as in the case of A_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. Table 5 summarizes the differences between the two approaches to explain system decisions.

5. Case Studies

We now present three case studies to illustrate the phenomena discussed above using real-world data.[9] 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 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

9. The code is available but is blinded for review.

	Importance weights	**Counterfactual explanations (evidence-based framework)**
Unit of analysis	Instance level. There is one explanation for each system decision.	Counterfactual level. There may be multiple or no explanations for each system decision.
Output explained	Model predictions (although the methods can be adapted to explain system decisions). Critically, a feature that affects model predictions may not affect system decisions.	System decisions.
Design Intent	Quantify feature importance. The explanations do not communicate how system decisions change as a result of changing the features.	Explain system decisions. The explanations are defined in terms of how system decisions change as a result of changing features.
Approach	Summarize the impact of each feature in a single number. However, features may affect system decisions in different ways depending on the values of other features.	Identify features that affect the system decision within the context of specific values for the other features.

Table 5: Summary of differences between importance weights and counterfactual explanations to explain system decisions

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 system decisions to accept or deny credit based on real data from Lending Club, a peer lending platform. The data is publicly available to users 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

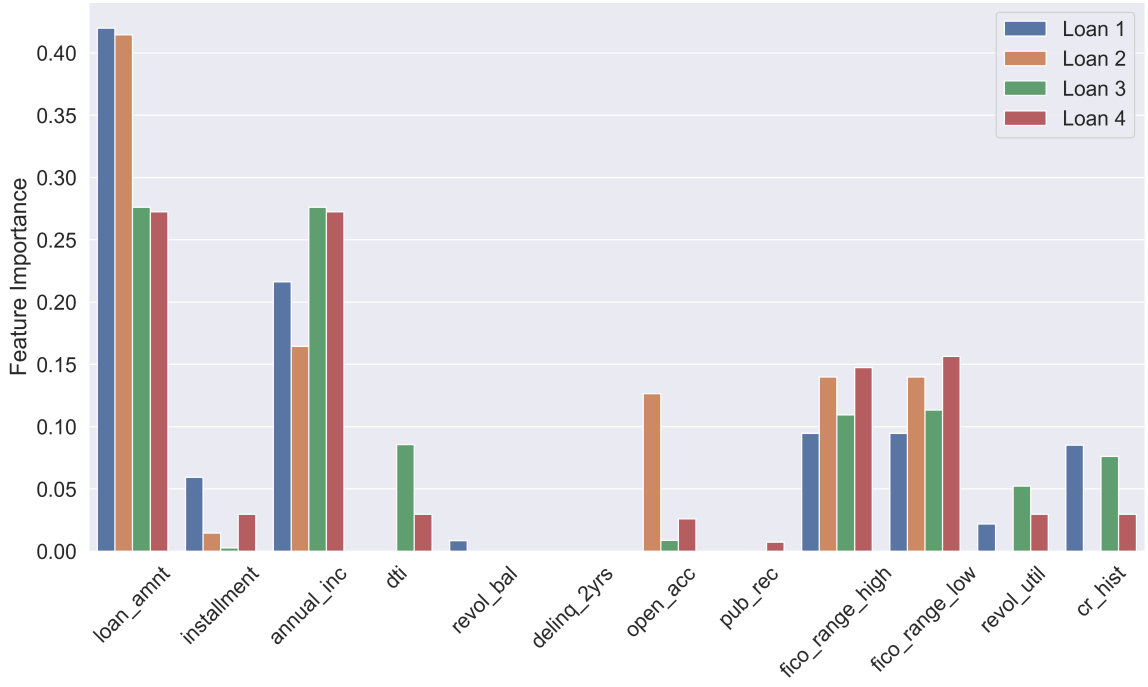


Figure 3: Feature importance weights according to SHAP

is in line with consumer credit decision making as described in the literature (see Baesens et al., 2003).[10]

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 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 23%. 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 that counterfactual explanations have several advantages. First, importance weights do not communicate which features would need to change in order for the decision to change—so their role as explanations for decisions is incomplete. Figure 3 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

10. 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 Club's particular business model, where external parties choose to fund (invest in) the loans.

Features	Explanations						Distance from mean
	1	**2**	**3**	**4**	**5**	**6**	
loan_amnt	↑						+$16,122
installment					↑		+$540
annual_inc		↓	↓	↓	↓	↓	-$9,065
dti							n/a
revol_bal						↓	-$4,825
delinq_2yrs							n/a
open_acc							n/a
pub_rec							n/a
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

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 evidence removal (the mean in this case), illustrating how our generalized counterfactual explanations may be applied to numeric features.

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

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
revol_bal																n/a
delinq_2yrs																n/a
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

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 3, we can see that the most important features for Loan 1 and Loan 4 are the same. 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. So, finding all coun-

terfactual explanations may be intractable when the number of features is large.[11] 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 a subset of all possible 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, this paper's 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 the most relevant explanations first. Because finding all explanations was tractable in this case, we did not incorporate costs in the heuristic we used to find explanations, but we do so in the next study.

Nonetheless, one can see that not all features shown in Figure 3 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 evidence removal 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.

Under these conditions, 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

11. Ramon et al. (2020) demonstrates the effectiveness of starting with 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 can be computationally expensive.

is preferable in many settings. It may well be that the former is preferable in some settings, but we haven't found a credible and compelling example.

Another crucial aspect of counterfactual explanations (as defined in our framework) is the method that is used to 'remove evidence'.[12] As we discussed before, such methods should be carefully chosen according to the domain and the problem. For example, mean imputation may be adequate to explain to a Lending Club investor why she should not invest in a particular loan, but other imputation methods may be more appropriate for explaining the same decision to the credit applicant. For instance, if the applicant is a 20-year old requesting a loan to pay for tuition, then having a short credit history may not be considered an anomaly (in this particular context). Thus, a more appropriate imputation method may consist of replacing the credit history with a value that is typical of individuals requesting student loans.

In line with this example, we illustrate next how counterfactual explanations may be generated using model-based imputation. For each of the 12 available features, we fit a linear regression model using training data of applicants who would be granted credit by the system and the other features as predictors. We then use each of these models to impute feature values. For instance, in our previous example, the credit history model may be used to impute the expected credit history of the 20-year old requesting the student loan.

Table 8 shows how the observed values for the four loans shown in Figure 3 differ from the corresponding default values when using mean imputation and model-based imputation (the entries in the table are these differences). The table shows several interesting results. First, although all the loan amounts are above average (according to mean imputation), these amounts are relatively common among other applicants with similar characteristics, as evidenced by the small differences with respect to the default value under model-based imputation. Therefore, even though the importance weights in Figure 3 hint at the loan amount as the primary reason for credit denial, this feature may not be considered relevant evidence in the context of these applicants. Moreover, the gap in annual income under model-based imputation is substantially larger, which reveals that the annual income of these applicants may be more anomalous than what is suggested by mean imputation. In fact, features that consist of evidence against credit worthiness under mean imputation may actually be evidence in favor under model-based imputation (and vice versa). For example, in the case of Loan 4, the applicant has one more credit line than the average applicant, and as a result this feature is part of two counterfactual explanations in Table 7 (see open_acc). However, according to model-based imputation, this user has one credit line less than other applicants with similar characteristics, which may be considered evidence in favor of credit worthiness depending on the context. Therefore, the method used to deal with evidence removal provides another way in which counterfactual explanations may be tailored to the context.

As an illustration, Table 9 shows all the counterfactual explanations for the four loans shown in Figure 3 when using model-based imputation (rather than mean-imputation). These explanations frame evidence against credit worthiness relative to similar credit applications that have been approved. The table shows that the counterfactual explanations

12. This aspect is also crucial for feature importance methods that use imputation to simulate lack of knowledge about features, such as SHAP. We are not aware of a similar discussion in the literature about those methods.

Features	Mean Imputation				Model-based Imputation			
	Loan 1	**Loan 2**	**Loan 3**	**Loan 4**	**Loan 1**	**Loan 2**	**Loan 3**	**Loan 4**
loan_amnt	16,122	15,722	6,672	12,372	123	79	-13	1
installment	539	528	226	417	-4	-3	1	0
annual_inc	-9,065	-9,065	-27,065	-15,065	-32,568	-85,302	-29,903	-32,926
dti	-1	-1	10	4	2	-14	6	5
revol_bal	-4,825	10,730	506	589	-9,750	-12,829	-10,982	-6,081
delinq_2yrs	0	0	0	0	0	-1	0	0
open_acc	-3	29	7	1	-4	26	6	-1
pub_rec	0	0	0	1	0	0	0	1
fico_range_high	-16	-21	-26	-21	0	0	0	0
fico_range_low	-16	-21	-26	-21	0	0	0	0
revol_util	11	-12	32	12	-3	6	22	1
cr_hist	-92	-22	-104	-39	-91	-68	-113	-58

Table 8: Differences between observed values and default values for our four loans, using mean imputation and model-based imputation

have changed (compared to the ones shown in Tables 6 and 7) to reflect that loan amount is no longer considered evidence against credit worthiness (since similar applicants have applied for similar amounts), whereas annual income is now considered the primary reason for credit denial.

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

For our second case study, 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,[13] has also been used by other researchers to compare the performance of various counterfactual explanation methods (Ramon et al., 2020). 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.

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

Loan 1	**Explanation 1.** Credit denied because: - Annual income is $32,568 less than expected.
Loan 2	**Explanation 1.** Credit denied because: - Annual income is $85,302 less than expected.
Loan 3	**Explanation 1.** Credit denied because: - Annual income is $29,903 less than expected. **Explanation 2.** Credit denied because: - Debt-to-income ratio is 6 units more than expected. - Revolving line utilization rate is 22% more than expected. - Credit history is 113 months less than expected. **Explanation 3.** Credit denied because: - Debt-to-income ratio is 6 units above expected. - Open credit lines are 6 more than expected. - Credit history is 113 months below expected.
Loan 4	**Explanation 1.** Credit denied because: - Annual income is $32,926 below expected.

Table 9: Counterfactual explanations using model-based imputation; "expected" means the value that model-based imputation predicts for this example.

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 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 a user's 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). Furthermore, explanations can lead to higher user engagement resulting from more confidence and transparency in product recommendations (Friedrich and Zanker, 2011). In such settings, users are 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 estimated values 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 10 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 4100 observations of the space of feature combinations.[14] Because SHAP uses sampling-based approximations, we can see that SHAP values vary every time we compute

14. 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

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

Importance weights (SHAP values) shown in parentheses.

Table 10: Topmost predictive pages and their SHAP values for a single decision to target our example user with the over-50 ad.

them, resulting in different topmost predictive pages. Importantly, while some pages appear recurrently, only Paul McCartney appears in all 5 approximations.

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 unsually 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 in order 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 Figure 4. Each explanation consists of a subset of Facebook pages that would change the targeting decision if removed from the set of pages Liked by the user. In other words, each of the sets shown in Figure 4 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

value. Our choice of 4100 is larger than the SHAP implementation's default sample size for all of the experiments we run.

Explanation 1: The user would not be targeted if {Paul McCarney} had not been Liked.

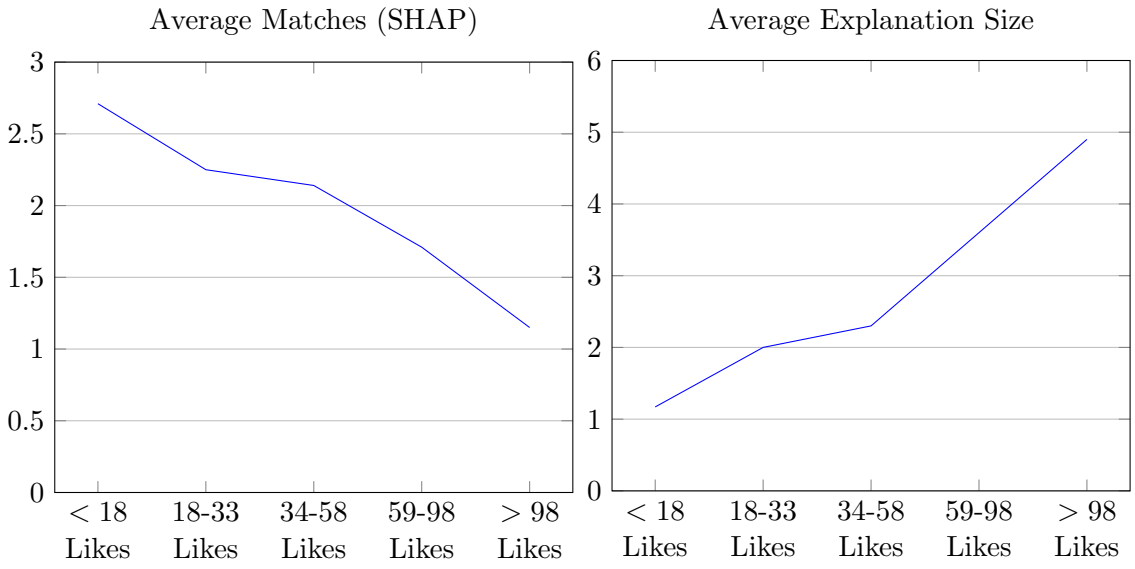
Explanation 2: The user would not be targeted if {Elvis Presley} had not been Liked.

Explanation 3: The user would not be targeted if {Neil Young} had not been Liked.

Explanation 4: The user would not be targeted if {Leonard Cohen} had not been Liked.

Explanation 5: The user would not be targeted if {Brain Pickings} had not been Liked.

Figure 4: 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 5: Variations in explanations by number of Likes

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 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 5a, which shows the average number of matches by quantile. As predicted, SHAP approximations are not consistent for users who have Liked many pages. Recall that SHAP is supposed to be estimating the Shapley values for

the features; thus they ought to be consistent. However, for the largest instances, most cases have only one page that appears in all five SHAP runs. This implies that (for most users) SHAP is not reliable enough to explain decisions by showing the topmost predictive pages.

Another alternative is to explain the targeting decisions using counterfactual explanations, but we may worry about providing explanations that are unnecessarily large. 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 5b, which shows the average size of counterfactual explanations by quantile.[15] 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 of 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 11 shows 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

15. 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 11: First counterfactual explanations found

Repository. The data set was originally provided by a national veteran's 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.

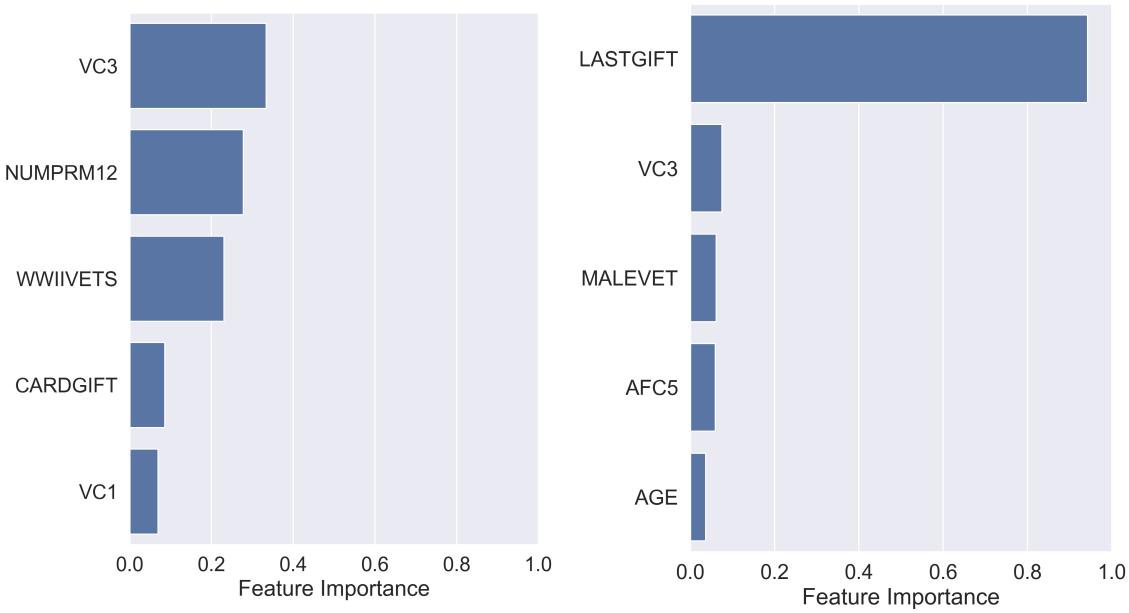
2. Using a classification model to predict whether a potential target will donate more than the break-even point so that we can target her if this is the case.

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 was the one used by the winners of the KDD Cup 1998.

We use XGBoost for both regression and classification training with 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).

Consider the following decision-making setting for our study. The decision-making 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.



(a) Top features for probability

(b) Top features for amount

Figure 6: Features with largest importance weights

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 6a and Figure 6b respectively. Interestingly, only VC3 (percent of 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 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 household's high probability of donating justify the targeting decision even if LASTGIFT had a smaller value?

As discussed earlier, SHAP may be used 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 show-

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 12: Explanations for targeting decision

cased in the previous examples to find explanations for targeting decisions. Table 12 shows the explanations found for the targeted household discussed above.

Interestingly, some of the highest-scoring SHAP features, shown in Figures 6, 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.

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 their values are changed alongside all other feature combinations. Hence, if a feature has a large negative impact in one feature ordering and several small positive impacts in other orderings, 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 orderings hides the fact that (in non-linear models) features may provide evidence in favor or against a decision depending on what other features are changed, which explains why AVGGIFT had a negative SHAP value but is present in the explanations shown in Table 12.

6. Discussion

These studies with real-world data illustrate various advantages of counterfactual explanations over importance weighting methods. The first study shows that the importance weights of features are not enough to determine how they affect system decisions. Furthermore, it shows how different imputation methods may be used to generate and customize counterfactual explanations for various purposes and users. 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 the synthetic example Section 4.3, in which features with negative SHAP weights may 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, as we have shown, instance-weight explanations have serious deficiencies for actually explaining decisions, so we cannot argue that they are preferable because they are simpler. Moreover, our second case study shows that importance weighting methods may actually not scale well when the number of features increases, because their approximations may become inconsistent. In the case of counterfactual explanations, 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 and rank counterfactual explanations. Thus, the fact that there are multiple counterfactual explanations is not necessarily problematic. In our study, short, consistent, and relevant explanations were significantly faster to find than computing importance weights, even with a large number of features.

Something that was just briefly explored in the first case study is the sensitivity of the explanations to the method that is used to remove evidence. This is not a challenge particular to the counterfactual approach; feature importance approaches, such as SHAP, also require the choice of such a method (e.g., mean imputation). We argue that, in the case of counterfactual explanations, the evidence-based perspective is useful to define plausible and relevant counterfactual scenarios, and so the choice of the method should carefully match the domain and the intent behind the explanations. This is an interesting direction for future research, as we would expect distinct alternatives for dealing with evidence removal to affect explanations differently, resulting in different interpretations of the system decision. For example, mean imputation and model-based imputation may produce very different explanations, and each of them may be more appropriate in different settings. For instance, in our credit scoring example, the former may be more useful for general ques-

tions such as 'Why should I be concerned about giving this individual a loan?' because the counterfactual implies comparing the individual to the general population. On the other hand, the latter may be more appropriate for a follow-up question such as 'Given that this individual is applying for a student loan, what additional concerns should I have?' because the counterfactual implies comparing the individual to others with similar characteristics or that have applied for similar loans.

Another option is to forgo imputation strategies altogether and produce counterfactual values using some other alternative procedure. For example, continuous features could be discretized into bins, and the algorithm we propose could be easily adapted to greedily move feature values across bins until the predicted class is changed, as done by Gomez et al. (2020). This procedure could be guided by the preference function introduced in our framework to search first for the most relevant counterfactual values. Similarly, any of the several methods proposed by Lash et al. (2017) for inverse classification could be used to produce counterfactual explanations as well.

Importantly, our study compares importance weights with a specific type of counterfactual explanation. As defined in Section 3.2, our explanations use the evidence-based perspective to simulate counterfactual worlds in which the evidence supporting the decision made by the system is absent. 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 decreased 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 (Verma et al., 2020). 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, (i) focuses on models for which the gradient at the decision point can be computed, (ii) does not work with categorical features, and (iii) may require access to the machine learning method used to learn the model (which usually is not available for deployed decision-making 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.[16] 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

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

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)).

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 and interpretation of AI systems, preferably through user studies (Binns et al., 2018; Dodge et al., 2019). Along this line of research, Kaur et al. (2019) studied data scientists' use of interpretability tools (including SHAP) when uncovering common issues that arise when building and evaluating predictive models. They found that despite being provided with standard tutorials, few of the participants were able to accurately describe what the visualizations were showing. As a result, some participants over-trusted the model because they used the explanations to rationalize suspicious observations, whereas others became skeptical of the visualizations and eventually stopped using them.

Importantly, different users are likely to require different things from explanations, and thus it is unlikely that a particular explanation will always be the best for every objective. Other researchers have recognized this and have proposed general frameworks to define characteristics of good explanations based on user needs (Lu et al., 2019). This study builds on this school of thought by providing a flexible explanation framework that may be used to address a wide and diverse range of needs. We discuss this in more detail below as part of the managerial implications of our study.

Additionally, the way explanations are visualized is also likely to affect how users perceive them. Although in this paper we used text and tables to present counterfactual explanations, interactive tools are probably much better suited for visualizing counterfactual explanations, specially if the goal is to analyze decisions at an aggregate level. What would work best in practice to visualize explanations is also likely to be context dependent, but there is already a nascent stream of research proposing various tools to visualize counterfactual explanations as proposed by Martens and Provost (2014). Examples include Tamagnini et al. (2017); Krause et al. (2017); Gomez et al. (2020). Any of these visualization tools could be easily adapted to visualize counterfactual explanations as proposed by our framework.

Another interesting direction is to learn from data the explanations that work better for different users. In the context of consumer-facing recommendations, McInerney et al. (2018) propose a method that simultaneously learns the best content to recommend for each user and the type of explanation(s) to which each user responds best. They showcase their method with music recommendations and find that personalizing explanations and recommendations together provides a significant increase in user engagement. Therefore, explanations may also have an important role on the outcome that the system decisions seek to optimize. This type of method could be used in conjunction with our framework to learn the imputation strategies or cost functions that are most effective to improve decisions (or other outcomes of interest).

Having said this, much more work is still needed to provide explanations that truly address user needs, and in particular the needs of decision subjects rather than decision makers. Barocas et al. (2020) reveal several easily overlooked assumptions on which uses of counterfactual explanations (and explanation methods in general) often rely and that may

result in the detriment of users affected by system decisions if not acknowledged—specifically when those explanations are then used to suggest concrete actions. For example, feature changes may interact with facts about a person's life that are invisible to the model, and thus the system may recommend something that would interfere with another goal in people's life. Changes that might be inexpensive for one person might be costly for another person, and thus we may need to account not only for how feature changes affect costs, but also for how costs vary across the population. In other cases, highlighting what the person must *not* change might be as important as the explanation itself: think about a credit applicant that switched jobs to increase her annual income just to find out that she still does not qualify for credit due to her short time of employment at her new job.

While our framework could be used to work around some of these problems (e.g., decision subjects could communicate their costs by ranking features according to how easy it would be for them to change them), these are all still largely unresolved challenges that require careful solutions in order to avoid other potential issues (e.g., privacy concerns, excessive complexity, users gaming the system). To address these challenges, future research should seek to understand what actions people actually take when confronted with explanations and how they are affected by those actions.

7. Managerial Implications

Importance weighting methods are rapidly becoming a very popular (if not the most popular) alternative for explaining model predictions. However, this paper shows that these methods may not be appropriate to explain the decisions made by model-based AI systems. Notably, the examples and case studies in this paper illustrate various pitfalls that managers should be aware of if they intend to deploy importance-weight explanations, the most salient being that importance weights are insufficient to communicate whether and how features affect decisions. Our paper proposes a counterfactual framework as an alternative specifically designed for such a task, illustrating its advantages throughout the examples.

We also demonstrate how counterfactual explanations can be applied much more broadly—to more problems and systems—than many prior authors seem to have realized. In our case studies, we use the proposed counterfactual explanations to explain system decisions made (a) using numeric and categorical features, (b) in low- and high-dimensional settings, (c) with linear and non-linear models, and (d) with system decisions based on one or multiple predictive models. The examples and case studies were also motivated by various business settings commonly encountered in practice and in which AI systems may be particularly useful, such as credit scoring and targeted advertising.

Finally, we propose two ways in which managers or other end users of the system may tailor counterfactual explanations to suit their context. The first consists of defining how to deal with the removal of evidence in order to generate explanations. As mentioned before, this is a choice that should be driven by the business context, and thus may change according to stakeholder needs. Continuing with the targeted advertising example, mean and mode imputation may be a reasonable approach for users that want to understand which of their actions led the system to target them. On the other hand, a manager using the system to make those targeting decisions may want to understand which are the features that led to the best targeting decisions in order to decide which features to keep investing

in (assuming he or she is purchasing third-party data for targeting, which is not unusual). Thus, for the use case of the manager, a better approach to deal with evidence removal might be to simulate the system behavior if the manager were to stop purchasing data for that feature—for example, by using a model built without those features.

Second, we also allow end-users to tailor explanations by incorporating context information as part of the heuristic procedure that is used to generate counterfactual explanations. Such information could consist of the cost of acquiring or changing the features, the degree of relevance of the features, or other domain-driven 'rules' (e.g., the feature value for 'is_female' should not be changed if 'is_pregnant=1'). We propose (and illustrate how) to incorporate this information as part of a user-defined cost function that may be used by a heuristic procedure to search first for potential explanations with 'lower costs', resulting in more context-specific explanations.

8. Conclusion

We examine the problem of explaining data-driven decisions made by AI 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 changing the values of these feature changes the decision) and irreducible (meaning that changing the values of any subset of the features in the explanation does 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 two important ways: (i) we explain system decisions (which may incorporate predictions from several predictive models using features with arbitrary data types) rather than model predictions, and (ii) we do not enforce any specific method to produce counterfactual values for the features. 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.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *The Journal of Machine Learning Research*, 18(1):8753–8830, 2017.

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.

Solon Barocas, Andrew D Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89, 2020.

Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 'it's reducing a human being to a percentage' perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 Chi conference on human factors in computing systems*, pages 1–14, 2018.

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.

Jonathan Dodge, Q Vera Liao, Yunfeng Zhang, Rachel KE Bellamy, and Casey Dugan. Explaining models: an empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 275–285, 2019.

Gerhard Friedrich and Markus Zanker. A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98, 2011.

Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. Vice: visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 531–535, 2020.

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.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. Technical report, Working paper, 2019.

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.

Josua Krause, Aritra Dasgupta, Jordan Swartz, Yindalon Aphinyanaphongs, and Enrico Bertini. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 162–172. IEEE, 2017.

Michael T Lash, Qihang Lin, Nick Street, Jennifer G Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 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.

Joy Lu, Dokyun DK Lee, Tae Wan Kim, and David Danks. Good explanation for algorithmic transparency. *Tae Wan and Danks, David, Good Explanation for Algorithmic Transparency (November 11, 2019)*, 2019.

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.

James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 31–39, 2018.

Peter Menzies and Helen Beebee. Counterfactual theories of causation. `https://plato.stanford.edu/entries/causation-counterfactual/`, 2019. Accessed: 2020-09-08.

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. *Advances in Data Analysis and Classification*, 2020.

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.

Paolo Tamagnini, Josua Krause, Aritra Dasgupta, and Enrico Bertini. Interpreting black-box classifiers using instance-level visual explanations. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, pages 1–6, 2017.

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.

Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

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.