Counterfactual Explanations for Data-Driven Decisions

Carlos Fernandez
New York University, cmf548@nyu.edu

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

Xintian Han
New York University, xh1007@nyu.edu

Counterfactual Explanations for Data-Driven Decisions

Short Paper

Carlos Fernandez, Foster Provost
New York University
New York, NY, USA
{cfernand|fprovost}@stern.nyu.edu

Xintian Han
New York University
New York, NY, USA
xh1007@nyu.edu

Abstract

Users' lack of understanding of systems that use predictive models to make automated decisions is one of the main barriers for their adoption. We adopt the increasingly accepted view of a counterfactual explanation for a system decision: a set of the system inputs that is causal (meaning that removing them changes the decision) and irreducible (meaning that removing any subset of the inputs in the explanation does not change the decision). We generalize previous work on counterfactual explanations in three ways: we explain system decisions rather than model predictions; we do not enforce any specific method for removing inputs, and our explanations can incorporate inputs with arbitrary data structures. We also show how model-agnostic algorithms can be tweaked to find the most useful explanations depending on the context. Finally, we showcase our approach using a real data set to illustrate its advantages over other explanation methods when the goal is to understand system decisions better.

Keywords: Explanations, system decisions, predictive models

Introduction

Data and predictive models are increasingly used to make better decisions. Yet, most data-rich companies still face barriers to adopting advanced predictive analytics, mainly because of managerial and cultural reasons rather than issues related to data and technology (LaValle *et al.*, 2011). In fact, as models become more complex and difficult to understand, users often become more skeptical and reluctant to use them, even if the models are known to improve decision-making performance (Arnold *et al.*, 2006; Kayande *et al.*, 2009). Thus, researchers in recent years have explored methods that aim to reduce the gap in users' understanding of predictive models, most notably by providing explanations of model predictions.

Approaches to explain model predictions include extracting rules that represent the inner workings of the model (e.g., Craven & Shavlik 1996; Martens et al. 2007; Jacobsson 2005) and associating weights to each feature according to their importance (e.g., Lundberg & Lee 2017; Ribeiro et al. 2016). Unfortunately, such approaches often become intractable or produce incomprehensible explanations as models become more complex. For instance, in applications such as text classification, predictions often come from black box models that use thousands of parameters and features. In these cases, it's hard to imagine that a list with thousands of rules or a vector with thousands of weights would be much more comprehensible than the predictive model being explained. More importantly, such explanations are not descriptive of the *decisions* made by the model. For example, suppose that a credit scoring system that uses a predictive model denies credit to a loan applicant, and that feature importance weights reveal that the two most important features in the model prediction were the annual income and the loan amount. While informative, this does not explain what was it 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? Is it necessary to change both? Or perhaps even changing both is not enough.

Some researchers (Martens and Provost, 2014; Wachter, Mittelstadt and Russell, 2017) have proposed to explain models from a causal perspective: when the question we seek to answer is "why did the model-based

system make a specific decision?", we ask specifically, "which data inputs caused the system to make its decision?" This approach is advantageous because it standardizes the form that an explanation can take; it does not require all features to be part of the explanation, and the explanations can be separated from the specifics of the model. The main contribution of this paper is to provide a multi-faceted generalization of this perspective to providing explanations for data-driven decisions rather than model predictions. Our framework for explanations (i) can address features with arbitrary data types, (ii) is model-agnostic, (iii) is scalable to thousands of features, and (iv) can take into consideration the potential cost of changing features as a result of the explanation. We present the framework and an associated explanation-finding algorithm. Then, to showcase situations in which counterfactual explanations explain data-driven decisions better than feature importance weights, we apply the algorithm to data on credit-investment decisions.

Predictive Models and Explanations

Predictive models are often used in systems that either support human decision making or make decisions automatically. The focus of this paper is on the latter, specifically on discrete decision-making, which is closely related to classification tasks. Classification tasks are usually modeled as scoring problems, where we want observations with the "correct" class to have a higher score than observations with the "incorrect" class; the scores rank observations, and classifications are made using a chosen threshold appropriate for the problem at hand (Provost and Fawcett, 2013). Many decision-making problems are modeled as classification tasks by associating a class with each decision. Typically, estimated probabilities of class membership are used as scores, but scores do not necessarily have to be probabilities, and the final output of the system (i.e., the decision) may not correspond to the labels in the training data. For 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 (conditional on the customer making a purchase); the expected profits would be the multiplication of these two predictions. Yet, the final output of the decision-making system would be whether the customer should be targeted with a promotion. Thus, our focus is on explaining system decisions rather than predictive models.

Many researchers have worked on explaining predictive models over the past several decades. Because symbolic models, such as decision trees, are often considered straightforward to explain when they are small enough,[1] most research has focused on explaining non-symbolic (black box) models or large models. Rule-based explanations are a popular approach to do so. These explanations are particularly useful when every decision that the model could possibly make must be understood. For example, in many credit scoring applications, banking regulatory entities require banks to implement globally comprehensive 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, Diederich and Tickle, 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.

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

One common way of assessing feature importance is based on simulating lack of knowledge about features (Lemaire, Féraud and Voisine, 2008; Robnik-Šikonja and Kononenko, 2008). For instance, one could

[1] However, explaining decisions may not be straightforward even with small symbolic models. For example, suppose the model is a binary decision tree that receives two inputs and predicts 1 when either of the inputs is 1. Then, by examining the tree, one may come to the wrong conclusion that (for one of the leaves) one of the inputs needs to be different from 1 for the tree to predict 1.

compare the original model's output with the output obtained when removing a specific feature. If the output changes, it means that the feature was important for the model prediction. Methods that use this approach often decompose each prediction into the individual contributions of each feature and use the decompositions as explanations, thus enabling the visualization of each instance-decision explanation separately. A notable challenge, however, is that interactions between features may lead to ambiguous explanations because the order in which features are removed may affect the importance attributed to each feature. Therefore, subsequent work proposed to assess feature importance by removing all possible subsets of features (rather than only one feature at a time), retraining models without the removed features, and comparing how predictions change (Štrumbelj, Kononenko and Robnik-Šikonja, 2009). However, such approaches may take hours of computation time and have been reported to handle only up to about 200 features. Alternative formulations have attempted to reduce computation time by sampling the space of feature combinations and by using imputation to deal with removed features, resulting in sampling-based approximations of the influence of each feature on the prediction (Strumbelj and Kononenko, 2010; Ribeiro, Singh and Guestrin, 2016; Datta, Sen and Zick, 2017; Lundberg and Lee, 2017).

Nonetheless, importance weights may not be good explanations because they don't communicate what would need to be different for the system decision to change. For example, as mentioned, systems may use predictions from multiple models to make decisions, yet all prior methods were designed to explain score outputs from a single predictive model. In these cases, it is unclear how to combine the importance weights of each model to assess their impact on the system decision. 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 changing only a handful of the features may be enough to change the system's decisions (Martens and Provost, 2014; Chen *et al.*, 2016).

System Decisions and Counterfactual Explanations

We generalize the counterfactual explanations proposed by Martens & Provost (2014). As with the prior work, we define explanations in terms of information that if not present would change the decision. For illustration, suppose a credit card transaction was deemed fraudulent by a system after it was registered as occurring outside the country where the cardholder lives,[2] and suppose the system would have deemed the transaction to be legitimate absent this location. In this case, it is intuitive to consider the location 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 "don't flag". Both explanations consist of information without which the system would have made a different decision.

We define a *counterfactual explanation* as a set of features that is *weakly causal* and *irreducible*. Weakly causal means that removing the set of features from the instance causes the model decision to change.[3] Irreducible means that removing any proper subset of the explanation would not change the model decision. The importance of an explanation being weakly causal is straightforward: the decision would have been different if not for the presence of this set of features. The "weakness" comes from the fact that all the features in the set may not actually be necessary. The irreducible condition serves to avoid including features that are superfluous. More formally, consider instance I consisting of m features $I_i,\ i = \{1,2,\dots,m\}$, for which the decision-making system $C: I \rightarrow \{1,2,\dots,k\}$ gives decision c. Then, a set of features E is a counterfactual explanation for $C(I) = c$ if and only if:

1. $E \subseteq I$ (the features are present in the instance)
2. $C(I - E) \neq c$ (the explanation is weakly causal)
3. $\forall E' \subset E: C(I - E') = c$ (the explanation is irreducible)

[2] We should keep in mind the decision-rather-than-classification perspective. We will use the shorthand "deeming the transaction to be fraudulent." However, in fact, the decision is to flag the transaction as likely fraud, which then will trigger other actions, such as sending a message to the account holder to verify. Flagging may be based on a threshold on the estimated likelihood of fraud but may also consider the existence of evidence from other transactions and the potential loss if the transaction were indeed fraudulent.

[3] 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. Our definition of counterfactual explanations relates to the former.

As mentioned, our approach builds on the explanations proposed by Martens & 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.[4] We generalize counterfactual explanations in three important ways. First, we generalize to broader system decisions, which may incorporate predictions from multiple predictive models. Second, their explanations consist of removing features by setting them to zero, whereas we generalize to arbitrary methods for removing features (and note the important relationship to methods for dealing with missing data). Third, while their approach has been applied in other contexts beyond document classification (Chen *et al.*, 2016; Moeyersoms *et al.*, 2016), these applications all have the same data structure: high-dimensional, sparse, binary features. Our generalization applies to data with arbitrary data types.

Note that counterfactual explanations are generally applied to "non-default" decisions. Data-driven systems usually make default decisions in the absence of information. In our example, a transaction would be considered legitimate unless there is enough evidence suggesting fraud. As a result, explaining default decisions often corresponds to saying, "because there was not enough evidence of a non-default class".[5] Therefore, the focus of this paper is primarily on explaining non-default decisions.

A practical question that our generalization raises is what does it mean to "remove" features from a data instance that will be input to a model-based decision-making procedure? Multiple solutions have been proposed for dealing with features missing from an instance when applying predictive models (Saar-Tsechansky and Provost, 2007),[6] such as imputing default values for the missing features, retraining models with only the available features, etc. The generalized framework is agnostic to which method is used to deal with the removed features—taking the position that this decision is domain- and problem-dependent. What matters is that the decision may change when some of the features are not present at the time of decision making, and thus the change in the decision can be attributed to these missing features. In the empirical example presented below, we use mean imputation for continuous variables and mode imputation for categorical variables. We chose this approach because it is usually far cheaper than retraining models, and it is one of the most commonly used and recommended techniques to deal with missing values. Saar-Tsechansky and Provost (2007) discuss other alternatives and their implications for model predictions; any of them could be used in conjunction with this framework.

Our generalization to system decisions allows us to define a heuristic procedure to find the most useful explanations depending on the context. The heuristic algorithm proposed by Martens & Provost (2014) finds counterfactual explanations by using 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 used by the search algorithm to first consider features that when removed reduce the score of the predicted class the most. This heuristic may be desirable when the goal is to find the smallest (and so arguably the simplest) explanations, such as when explaining the decisions of models that use thousands of features. However, it is often the case that users want to use the explanations as recommendations of what to change in order to affect the system decision. For 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 female" is of very little use as a guide for what to change. In practice, some features are easier to change than others, and some may be (practically) impossible to change.

Therefore, we allow the incorporation of the costs of removing (i.e., changing) each feature as part of the heuristic procedure. This cost may be fixed (e.g., when setting a binary feature to zero), may be contingent on the magnitude of the change (e.g., when adjusting a continuous variable), or may even be infinite. Subsequently, instead of searching for the feature combinations that change the score of the predicted class the most, the heuristic could search for the feature combinations for which the output score changes the most per unit of cost. Returning to the heart attack example, if we assign an infinite cost to changing the gender feature, the heuristic would not search for feature combinations that include it, regardless of its high

---

[4] Martens and Provost also note the straightforward extension to other word-like features.

[5] 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.

[6] Note that this is different from what to do when feature values are missing when learning such models from data.

impact on the output score. Instead, the heuristic would search first for 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 & Provost (2014) could be easily adjusted to find the most useful explanations for the problem at hand. A similar approach has also been suggested for classifiers that have a known and differentiable scoring function (Lash *et al.*, 2017).

Counterfactual explanations have other benefits as well. First, same as importance weights, they are defined in terms of domain knowledge (i.e., features) rather than in terms of modeling techniques. As mentioned before, this is of critical importance to explain individual decisions made by such models to users. More importantly, these explanations can be used to understand which features would need to change for the decision to change, which is not captured by feature importance methods. Also, because only a fraction of the features will be present in any single explanation, our 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 few dozen of features at the most (Martens and Provost, 2014; Chen *et al.*, 2016; Moeyersoms *et al.*, 2016).

Empirical Example

We use data from Lending Club, a lending platform, to showcase the advantages of counterfactual explanations over feature importance weights when explaining data-driven decisions. The data is publicly available and contains comprehensive information on all loans issued between 2007 and 2019 (the data is updated every quarter). 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 things, we use a sample of the data used by Cohen et al. (2018) and focus on loans with a 13% annual interest and a duration of three years (the most common loans), leaving us with 71,938 loans.

We use 70% of this reduced 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 system that denies credit to loan applicants with a probability of default above 20%. We then use the system to decide which of the held-out 30% of loans should be approved.

We use SHAP (Lundberg and Lee, 2017), a popular approach to explain the output of machine learning models, to compute importance weights for system decisions and compare them to counterfactual explanations. SHAP ties instance-level explanations to game theory, unites several feature importance weighting methods (such as the relatively well-known LIME (Ribeiro, Singh and Guestrin, 2016)), and presents the only possible consistent and locally accurate attribution method (see (Lundberg and Lee, 2017) for details). Importantly, SHAP was designed to explain the output of scoring functions, not system decisions.[7] We repurpose SHAP to compute feature importance weights for non-default system decisions by transforming the output of the system into a scoring function that returns 1 if the credit is denied and returns 0 otherwise. More generally, if we define a scoring function that returns 1 if the decision is the same after removing features and returns 0 otherwise, we can thus use SHAP to obtain importance weights for any system decision–even decisions with multiple, unordered alternatives, that cannot normally be represented as a single numeric score.

We now can compare counterfactual explanations to explanations based on feature importance weights, to illustrate the advantages of the former. First, weights do not communicate what features would need to change for the decision to change. Figure 1 shows the feature importance weights assigned by SHAP to four loans that would be denied credit by the system. According to SHAP, *loan_amnt* was the most important

[7] This is also true for all other methods that compute feature importance weights.

feature for the credit denial of the four loans. However, this information does not actually explain any of the decisions. If the credit applicant of Loan 1 wants to understand what to do better in order to obtain credit next time, we cannot use the feature importance information to explain *why* he or she was denied credit. Was it the amount of the loan? The low annual income? Both?

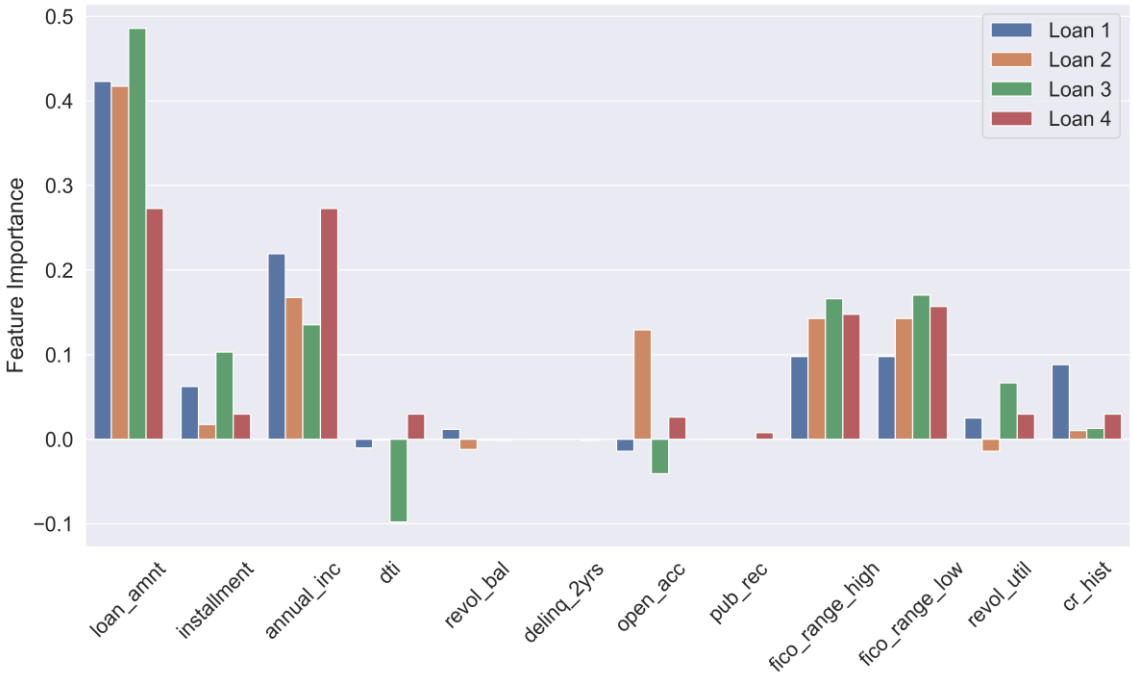

Figure 1. Feature importance weights according to SHAP

Table 1, in contrast, shows all the counterfactual explanations for the credit denial decision of Loan 1. Each column represents a possible explanation, and the arrows in each cell show which features are present in each explanation. The last column shows by how much each feature in the explanation would need to change for the system to approve Loan 1, 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 too large) given the other aspects of the application. The data indeed shows that the amount for Loan 1 is $28,000, but the average loan amount in our sample is $11,878. In this instance, one could explain the decision in several other ways. The explanation in column 4 suggests that the $28,000 credit would be approved if the applicant had a higher annual income and a longer credit history, which are below average in the case of the applicant. Therefore, from these explanations, it is immediately apparent how the features influenced the decision. This highlights two additional advantages of counterfactual explanations: they give a deeper insight into why the credit was denied and provide various alternatives to change the decision.

Table 2 shows the counterfactual explanations of Loan 4 to emphasize this last point. From Figure 1, we can see that Loan 1 and Loan 4 have somewhat similar importance weights. Thus, from this figure alone, one may conclude that these two credit denial decisions should have similar counterfactual explanations. Yet, comparing Table 1 and Table 2 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 counterfactual explanations may grow exponentially with respect to the number of features, so finding all possible counterfactual explanations is generally intractable when the number of features is large. In the case of the loans discussed in this empirical example, we did an exhaustive search because the number of features was relatively small, thus Tables 1-2 show all possible counterfactual explanations for the credit denials of Loan 1 and Loan 4. In most settings, however, the algorithm would need to be restricted to a maximum number of iterations or explanations to be tractable. Nonetheless, the

number of explanations may still be large, so additional steps to improve interpretability could be helpful, such as defining measures to rank explanations according to how "good" they are.

Table 1. Counterfactual explanations for Loan 1

Features	Explanations						Required Change
	1	2	3	4	5	6	
loan_amn	↑						-$16,122
installment					↑		-$540
annual_inc		↓	↓	↓	↓	↓	+$9,065
revol_bal						↓	+$4,825
fico_range_high			↓				+16
fico_range_low		↓					+21
revol_util						↑	-12%
cr_hist				↓			+92 months
↑ means feature is too large to grant credit.							
↓ means feature is too small to grant credit.							

Table 2. Counterfactual explanations for Loan 4

Features	Explanations															Required Change
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
loan_amn	↑															-$12,372
installment						↑						↑			↑	-$418
annual_inc		↓														+$15,065
dti				↑						↑					↑	-5
open_acc								↑						↑		-1
pub_rec						↓			↑							-1
fico_range_high			↓								↓	↓	↓	↓		+21
fico_range_low			↓	↓	↓	↓	↓	↓	↓							+21
revol_util							↑						↑		↑	-13%
cr_hist					↓						↓				↓	+39 months
↑ means feature is too large to grant credit.																
↓ means feature is too small to grant credit.																

One such measure could be the number of features present in the explanation (the fewer, the better). In fact, the heuristic we used to find explanations in this empirical example, the same proposed by Martens & Provost (2014), was designed to find the shortest explanations first (and the tables corroborate that the first explanations are indeed shorter). However, there could be other more relevant measures depending on the domain, such as the individual's ability to change the features in the explanation. As mentioned in the previous section, our generalized framework would allow incorporating the cost of changing features as part of the heuristic procedure, resulting in an algorithm designed to find the cheapest explanations first. Due to space constraints (and because finding all possible explanations was tractable in this case), we did not incorporate costs in the heuristic we used to find explanations in this empirical example. However, future versions of this work will include additional empirical examples to illustrate this.

Nonetheless, one can see that not all features shown in Figure 1 and Tables 1-2 would be relevant for loan applicants looking for recommendations to get their credit approved. So, SHAP may be adjusted further to compute weights only for a subset of features. Since SHAP deals with missing features by imputing default

values, one can delimit the set of 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 assuming most customers can only ask for less money or show additional sources of income to get their credit approved.

After doing this, SHAP computes an importance weight of 0.5 for both the loan amount and the annual income, and there are two counterfactual explanations: the applicant can either reduce the loan amount or increase the annual income to get the loan approved (columns 1 and 2 in Table 2). Now, 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 approved the applicant of Loan 4 would need both to reduce the loan amount *and* to increase his/her annual income. In other words, with this different decision system, there is only one counterfactual explanation consisting of both features. However, SHAP would still show an importance weight of 0.5 for each feature, heedless of the fact that the decision-making process changed.

Conclusion

This paper tackles the problem of explaining data-driven system decisions from a causal perspective: if the question we seek to answer is "why did the system make a specific decision", we can instead 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 an explanation as a set of features that is weakly causal (meaning that removing the set from the instance changes the decision) and irreducible (meaning that removing any subset of the features in the explanation would not change the decision). Our work generalizes previous work in counterfactual explanations in at least four important ways: (i) we explain system decisions (which may incorporate predictions from several predictive models) rather than model predictions, (ii) we do not enforce any specific method to remove features, (iii) our explanations can deal with feature sets with arbitrary dimensionality and data types, and (iv) we allow the tailoring of explanations to domain needs by introducing costs—for example, the costs of changing the features responsible for the decision. Finally, we showcase the approach using a real data set to illustrate its advantages over feature importance weights when the goal is to understand system decisions better. Future versions of this work will incorporate an extended section of the empirical analysis in: (1) high-dimensional settings, (2) settings in which explanations are expected to be used as recommendations with associated costs, and (3) settings where the decision-making procedure draws on more than one model.

Understanding how different alternatives for dealing with missing features may affect explanations is another interesting direction for future research. For example, if features are correlated, mean imputation and retraining the model without the removed feature may produce different results. For instance, a decision may change when imputing the mean for a removed feature, but if a new model without the feature is trained, the same decision may not change when removing the feature if the remaining features capture most of the information given by the removed feature. Therefore, while our proposed framework can work with either approach, future research should discuss the advantages of each approach in various settings.

References

Andrews, R., Diederich, J. and Tickle, A. B. (1995) 'Survey and critique of techniques for extracting rules from trained artificial neural networks', *Knowledge-based systems*, 8(6), pp. 373–389.

Arnold, V. *et al.* (2006) 'The Differential Use and Effect of Knowledge-Based System Explanations in Novice and Expert Judgement Decisions', *MIS Quarterly*, 30(1), pp. 79–97.

Chen, D. *et al.* (2016) 'Enhancing Transparency and Control when Drawing Data-Driven Inferences about Individuals', *ICML Workshop on Human Interpretability in Machine Learning*, (2), pp. 1–25. Available at: http://arxiv.org/abs/1606.08063.

Cohen, M. C. *et al.* (2018) 'Data-Driven Investment Strategies for Peer-to-Peer Lending: A Case Study for Teaching Data Science', *Big Data*, 6(3), pp. 191–213.

Craven, M. W. and Shavlik, J. W. (1996) 'Extracting Tree-Structured Representations of Trained Networks',

Advances in Neural Information …, 8, p. 7.

Datta, A., Sen, S. and Zick, Y. (2017) 'Algorithmic Transparency via Quantitative Input Influence', *Transparent Data Mining for Big and Small Data*, pp. 71–94.

Jacobsson, H. (2005) 'Rule extraction from recurrent neural networks: A taxonomy and review', *Neural Computation*, 17(6).

Kayande, U. *et al.* (2009) 'How incorporating feedback mechanisms in a DSS affects DSS evaluations', *Information Systems Research*, 20(4), pp. 527–546.

Lash, M. T. *et al.* (2017) 'A Budget-Constrained Inverse Classification Framework for Smooth Classifiers', in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, pp. 1184–1193.

LaValle, S. *et al.* (2011) 'Big Data, Analytics and the Path From Insights to Value', *MIT Sloan Management Review*, 52(2), p. 21.

Lemaire, V., Féraud, R. and Voisine, N. (2008) 'Contact personalization using a score understanding method', in *Proceedings of international joint conference on neural networks*.

Lundberg, S. and Lee, S.-I. (2017) 'A Unified Approach to Interpreting Model Predictions', *Advances in Neural Information Processing Systems*. Available at: http://arxiv.org/abs/1705.07874.

Martens, D. *et al.* (2007) 'Comprehensible credit scoring models using rule extraction from support vector machines', *European Journal of Operational Research*, 183(13), pp. 1466–1476.

Martens, D. and Provost, F. (2014) 'Explaining Data-Driven Document Classifications.', *MIS Quarterly*, 38(1), pp. 73-A6.

Moeyersoms, J. *et al.* (2016) 'Explaining Classification Models Built on High-Dimensional Sparse Data', *ICML Workshop on Human Interpretability in Machine Learning*. Available at: http://arxiv.org/abs/1607.06280.

Provost, F. and Fawcett, T. (2013) *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media, Inc.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '"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*. Available at: http://arxiv.org/abs/1602.04938.

Robnik-Šikonja, M. and Kononenko, I. (2008) 'Explaining classifications for individual instances', *IEEE Transactions on Knowledge and Data Engineering*, 20(5), pp. 589–600.

Saar-Tsechansky, M. and Provost, F. (2007) 'Handling Missing Values when Applying Classification Models', *Journal of Machine Learning Research*, 8(Jul), pp. 1625–1657.

Strumbelj, E. and Kononenko, I. (2010) 'An Efficient Explanation of Individual Classifications using Game Theory', *Journal of Machine Learning Research*, 11, pp. 1–18.

Štrumbelj, E., Kononenko, I. and Robnik-Šikonja, M. (2009) 'Explaining instance classifications with interactions of subsets of feature values', *Data & Knowledge Engineering*, 68(10), pp. 886–904.

Wachter, S., Mittelstadt, B. and Russell, C. (2017) 'Counterfactual Explanations Without Opening the Black Box : Automated Decisions and the Gdpr', *Harvard Journal of Law & Technology*, pp. 1–52.