An Interpretable Model with Globally Consistent Explanations for Credit Risk

Chaofan Chen[1], **Kangcheng Lin**[2], **Cynthia Rudin**[1,2], **Yaron Shaposhnik**[3], **Sijia Wang**[2], **Tong Wang**[4]
[1] Department of Computer Science, Duke University
[2] Department of Statistical Science, Duke University
[3] Simon Business School, University of Rochester
[4] Department of Management Sciences, Henry B. Tippie College of Business, The University of Iowa

Abstract

We propose a possible solution to a public challenge posed by the Fair Isaac Corporation (FICO), which is to provide an explainable model for credit risk assessment. Rather than present a black box model and explain it afterwards, we provide a globally interpretable model that is as accurate as other neural networks. Our “two-layer additive risk model" is decomposable into subscales, where each node in the second layer represents a meaningful subscale, and all of the nonlinearities are transparent. We provide three types of explanations that are simpler than, but consistent with, the global model. One of these explanation methods involves solving a minimum set cover problem to find high-support globally-consistent explanations. We present a new online visualization tool to allow users to explore the global model and its explanations.

1 Introduction

In 2018, the Fair Isaac Corporation (FICO) proposed a challenge to data science researchers: present an explainable model for the risk of defaulting on a loan. FICO provided a dataset for the challenge, and asked researchers to provide explanations for the global model, as well as local explanations (explanations for a given prediction). They also requested that the global model respect monotonicity constraints on several of the variables.

In responding to this challenge, we considered the specific aspects of the dataset in our modeling approaches: the data are balanced between the two classes, most of the features are real-valued, and most importantly, each of the 23 features is itself interpretable. Because the features already come with a good representation, the algorithm does not need to construct the representation. Generally, for data having this particular property, perhaps with a small amount of feature engineering, most machine learning algorithms tend to have almost the same performance, including algorithms that produce globally interpretable models. Thus, we aimed to create a model that was fully and globally interpretable, rather than to construct a black box.

The globally interpretable model that we constructed is called a *two-layer additive risk model*. It was designed to resemble traditional subscale models, where the features are partitioned into meaningful subgroups, and the subgroup scores are later combined into a global model. Traditional subscale models are generally interpretable because they are decomposable into meaningful components, and because these models are usually linear with coefficients whose sign is positive for factors of increasing risk. Our model preserves these classical elements (decomposable, uses linear modeling, positive coefficients for risk factors), but inserts (interpretable) nonlinearies in several places to makes the model more flexible and accurate. In particular, the algorithm transforms the original features into piecewise constant functions that monotonically increase (or decrease) if we constrain them to do so. Combinations of these piecewise constant functions form the subscales in the second layer of

Preprint. Work in progress.

the network. The subscales are fed through a sigmoid nonlinearity, which has the effect of adding more perceptron-like flexibility to the model, but also makes the subscale scores more meaningful as their own "mini-models;" each subscale produces its own probability of defaulting on a loan. The subscales are combined linearly and sent through a sigmoid function to produce the final probability of defaulting on a loan.

While working on this challenge, it is important to note that the guidelines asked for an explanation for each class of the global model, such as variable importance information. It did not necessarily ask for a model that is globally interpretable. Perhaps it could benefit FICO to have a global model that is *not* interpretable (a "secret sauce")? It is not clear that even if there exists a globally interpretable model, such as the one we found, that it would be desirable for FICO to release it. Thus, we are not certain that we responded as much to FICO's needs as we did to its customers' needs.

The issue raised above, about explainability of a black box model, versus providing a globally interpretable model, is important. With the new General Data Protection Regulation (GDPR) [12] regulations such as "right to an explanation," there could still be little incentive for companies to provide anything more than an modestly local explanation, even if a globally interpretable model existed. However, explanations can be problematic for several reasons. First, unless the global model is uniformly equal to the local model, the explanations will be sometimes incorrect, which makes it difficult to trust either the explanations or the global model itself. Even if an explanation is correct, it could be inconsistent with the global model's actual calculations (i.e., low fidelity [5]), for instance providing reasons that may be true for some cases but not others (e.g., a reason of "too many accounts open" may be used to deny someone a loan even if there is another person with the same number of open accounts who was offered a loan by the same global model). Explanations could also be correct but misleading, offering reasons that are true but incomplete [6] – missing key information. All of these problems with explanation-for-black-box methods are reasons that consumers would benefit from globally interpretable models. Again, however, we fully recognize that creating a globally interpretable model is not usually desirable from a business perspective. [1]

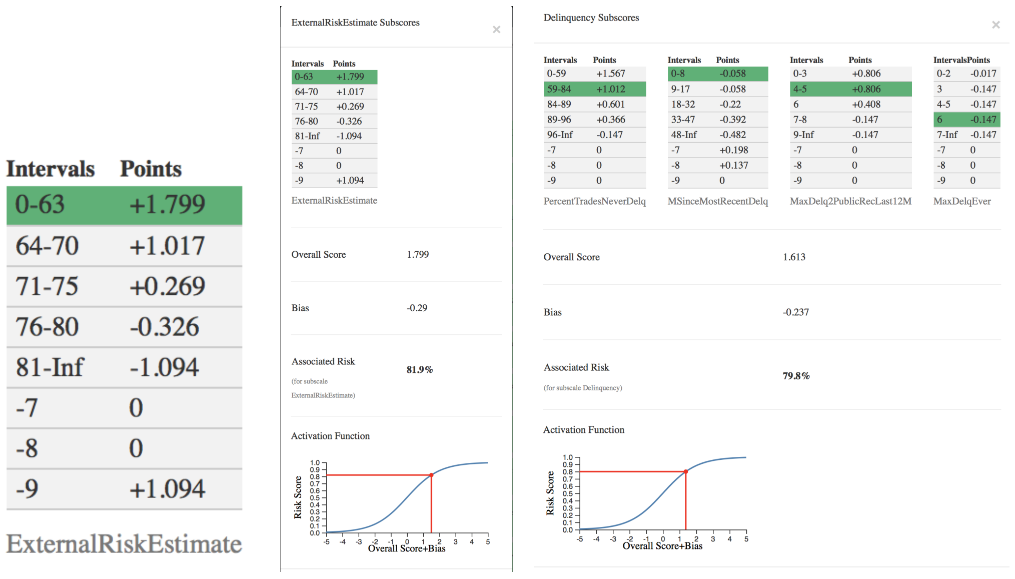
Attempts to create globally interpretable models for financial applications use mainly standard machine learning approaches (e.g. decision trees and support vector machines for bank direct marketing [10, 9]), which cannot accommodate FICO's monotonicity constraints. Some work finds optimal rules [1], but rules are not natural for datasets with many real valued features, like FICO's data. Additive models are natural for real-valued features and can easily preserve monotonicity.

Even though our global model is interpretable, we can produce optional "explanations," which now simply become *summaries* of general trends in the model, or other aspects of predictions. To "explain" individual predictions, our interactive display first highlights the factors that contribute most heavily to the final prediction of the global model. The second explanation method (called `SetCoverExplanation`) is a model-agnostic explanation algorithm that is novel to this work. `SetCoverExplanation` solves a minimal set cover optimization problem [3, 2] to generate conjunctive rules that are consistent with all training cases. Third, we provide case-based explanations. Our algorithm finds cases that are similar on important features to any current case that the user inputs.

We created an interactive display that shows the full computation of the model from beginning to end, without hiding any nonlinearities or computations from the user. Factors are colored according to their contribution to the global model. The form of our global model lends itself naturally to variable importance analysis, and understanding monotonicity constraints, through the visualization.

The novel elements of the work are (i) the form of the two-layer additive risk model, which lends naturally to sparsity, decomposibility, visualization, case-based reasoning, feature importance, and monotonicity constraints, (ii) the interactive visualization tool for the model, (iii) the `SetCoverExplanation` algorithm for high-support local conjunctive explanations, and (iv) the application to finance, indicating that black boxes may not be necessary in the case of credit-risk assessment.

[1]Companies such as DivePlane are now grappling with this issue, where they do not, as of this writing, release code or demonstrations of their models, which are claimed not to be black boxes.



Intervals	Points
0-63	+1.799
64-70	+1.017
71-75	+0.269
76-80	-0.326
81-Inf	-1.094
-7	0
-8	0
-9	+1.094

ExternalRiskEstimate

ExternalRiskEstimate Subscores

Intervals	Points
0-63	+1.799
64-70	+1.017
71-75	+0.269
76-80	-0.326
81-Inf	-1.094
-7	0
-8	0
-9	+1.094

ExternalRiskEstimate

Overall Score	1.799
Bias	-0.29
Associated Risk (for subscale ExternalRiskEstimate)	81.9%

Activation Function

Delinquency Subscores

Intervals	Points	Intervals	Points	Intervals	Points	Intervals	Points
0-59	+1.567	0-8	-0.058	0-3	+0.806	0-2	-0.017
59-84	+1.012	9-17	-0.058	4-5	+0.806	3	-0.147
84-89	+0.601	18-32	-0.22	6	+0.408	4-5	-0.147
89-96	+0.366	33-47	-0.392	7-8	-0.147	6	-0.147
96-Inf	-0.147	48-Inf	-0.482	9-Inf	-0.147	7-Inf	-0.147
-7	0	-7	+0.198	-7	0	-7	0
-8	0	-8	+0.137	-8	0	-8	0
-9	0	-9	0	-9	0	-9	0
PercentTradesNeverDelq		MSinceMostRecentDelq		MaxDelq2PublicRecLast12M		MaxDelqEver	

Overall Score	1.613
Bias	-0.237
Associated Risk (for subscale Delinquency)	79.8%

Activation Function

Figure 1: External Risk Estimate Subscale. *Left*: sum of thresholds, as a piecewise constant function, written as a scoring system. *Middle*: The transformation of points into a risk estimate. The risk estimate for the subscale is meaningful as its own "mini-model" of risk, based only on the ExternalRiskEstimate feature. *Right*: A subscale that uses multiple features.

2 Two-Layer Additive Risk Model and its Visualization

We work with a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^P$ is a vector of features where the categorical features are binarized. The labels $y_i \in \{0, 1\}$. Let $\mathcal{P}$ represent the set of features and $|\mathcal{P}| = P$. We present the structure of our two-layer additive risk model (ARM). While in general, fully connected neural networks are hard to comprehend even with sparsity regularization applied, our model has carefully designed sparsity and monotonicity constraints that make the calculations easier to comprehend. We did not require quadratic terms, as [8] did.

First, to ensure monotonicity of the model with respect to any given feature, we used step functions as our initial transformations of the features, and constrained the coefficients of that feature's step functions to be non-negative. For instance, for a monotonically decreasing feature $x_{\cdot,p}$, the following features could be created: $b_{p,1}(x_{\cdot,p}) = \mathbb{1}[x_{\cdot,p} < 10]$, $b_{p,2}(x_{\cdot,p}) = \mathbb{1}[x_{\cdot,p} < 50]$, $b_{p,3}(x_{\cdot,p}) = \mathbb{1}[x_{\cdot,p} < 75]$, and $b_{p,0}(x_{\cdot,p}) = \mathbb{1}[x_{\cdot,p}$ is not missing$]$[2]. Note that all of these features use one-sided intervals. This choice was made because convex combinations of these step functions yield *monotonic* piecewise linear functions. Specifically, by enforcing the constraints that the coefficients for $b_{p,1}$, $b_{p,2}$, and $b_{p,3}$ must be non-negative, we shall guarantee a monotonically decreasing relationship between the original continuous feature $x_{\cdot,p}$ and the subscale's predicted probability of default. Once the model is learned, the sum of the one-sided intervals becomes a piecewise constant function. For instance,

$$f_p(x_{\cdot,p}) = \beta_{p,1} b_{p,1}(x_{\cdot,p}) + \beta_{p,2} b_{p,2}(x_{\cdot,p}) + \beta_{p,3} b_{p,3}(x_{\cdot,p}) + \beta_{p,0} b_{p,0}(x_{\cdot,p})$$

can be equivalently written

$$\begin{aligned} f_p(x_{\cdot,p}) &= (\beta_{p,1} + \beta_{p,2} + \beta_{p,3} + \beta_{p,0})\mathbb{1}[x_{\cdot,p} < 10] + (\beta_{p,2} + \beta_{p,3} + \beta_{p,0})\mathbb{1}[10 \leq x_{\cdot,p} < 50] \\ &\quad + (\beta_{p,3} + \beta_{p,0})\mathbb{1}[50 \leq x_{\cdot,p} < 75] + \beta_{p,0}\mathbb{1}[75 \leq x_{\cdot,p}], \end{aligned}$$

which can be displayed as a traditional scoring system, as shown in Figure 1 (left panel) for the ExternalRiskEstimate subscale. If coefficients $\beta_{p,1}$, $\beta_{p,2}$, and $\beta_{p,3}$, are nonnegative, the function f_p is nonincreasing. If instead we would like to constrain f_p to be monotonically increasing, we reverse the above one-sided inequalities $<$ into $>$. Of course, if a feature x_p has no desired monotonicity, we drop non-negativity constraints.

[2]We handle missing values also by creating binary features.

Table 1: An example of most important contributing factors learned for observation "Demo 1"

	Most important contributing factors
1	MaxDelq2PublicRecLast12M is 6 or less (from the most important subscale, Delinquency)
2	PercentTradesNeverDelq is 95 or less (from the most important subscale, Delinquency)
3	AverageMInFile is 48 or less (from the second most important subscale, TradeOpenTime)
4	AverageMInFile is 69 or less (from the second most important subscale, TradeOpenTime)

Using domain knowledge obtained from the data description, we partitioned the features $\mathcal{P}$ into different sets for the subscales, inducing sparsity in the first layer of the network. Each subset of features is sent to one node which computes a subscale. Denote the feature subsets as

$$\mathcal{P} = \cup_{k=1}^{K} \mathcal{P}^{[k]},$$

where subset of features $\mathcal{P}^{[k]}$ is sent to a subscale. There are between 1 and 4 of the original features combined per subscale, yielding a total of 10 subscales to represent the original 23 features. Each subscale can be interpreted as a miniature model for predicting the probability of failure to repay a loan, using only the features designated for the subscale. The output of the subscale is a probability, denoted by $r^{[k]}$ for subscale k, which is simply a sigmoid transformation of the score.

$$r^{[k]}(\mathbf{x}) = \sigma\left(\sum_{p\in\mathcal{P}^{[k]}} f_p(x_{\cdot,p})\right) = \sigma\left(\sum_{p\in\mathcal{P}^{[k]}} \sum_{l=0}^{L_p} \beta_{p,l} b_{p,l}(x_{\cdot,p})\right),$$

where $\sigma(\cdot)$ represents a sigmoid function and L_p is the number of binary features created for feature p (not counting the special indicator for non-missing values).

Finally, the subscale results are linearly combined and again nonlinearly transformed into a final probability of failure to repay a loan. The contribution of each subscale to the final prediction can be easily observed by its weighted output.

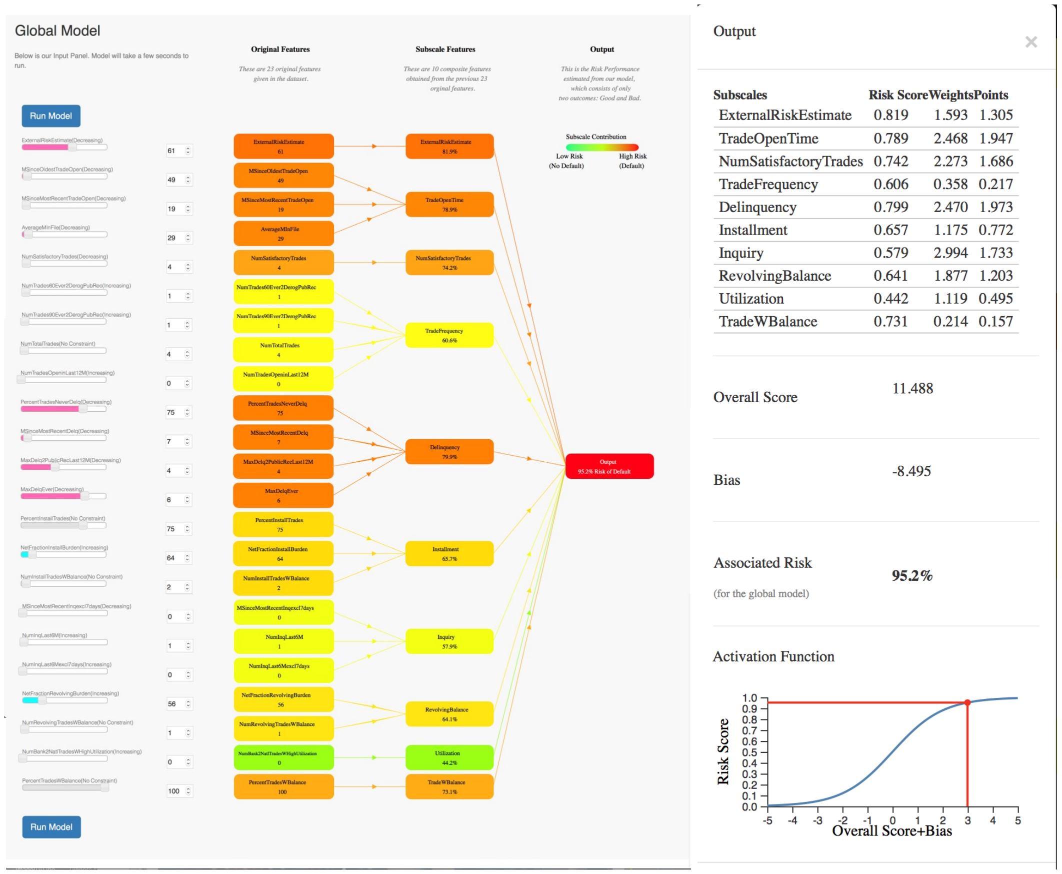
The simplest way to train coefficients $\boldsymbol{\beta}^{[k]} = \{\beta_{p,l} \text{ for } p \in \mathcal{P}^{[k]}, l \in [0, L_p]\}$ is to treat each $r^{[k]}$ as an independent classification model with the $\{y_i\}_{i=1}^{N}$ being the target variables, using regularization (e.g., ℓ_2) to prevent overfitting, and positivity constraints on the thresholds to enforce monotonicity. A slightly more complicated way to train is to optimize for a combination of accuracy for the subscales and accuracy for the global model. On our data, these two methods tended to produce almost identical results. An image of the full model is shown in Figure 2, where the colors indicate the final contribution to the combined score. Red indicates more likely to default on the loan. The 23 feature values can be entered on the left, and clicking on any of the 10 subscales (in the second colored layer) reveals a pop-up window with the calculation, as in Figure 1 (middle and right panels). The final combination of features is shown in Figure 2 (right panel). Figure 3 shows that our global model does not lose accuracy over other machine learning techniques, despite being constrained to be interpretable.

Variable importance

Our two-layer additive risk model comes naturally with a way to identify a list of factors that contribute most heavily to the final prediction. Table 1 shows a list of four factors that are important for predicting observation "Demo 1" to have 95.2% risk of default (i.e., bad risk performance).

To identify the factors, we first identify the most important two subscales and then the most important factors within each subscale. The importance of each subscale in the final model is determined by its weighted score, which is the product of the subscale's output and its coefficient – the larger the product, the larger the contribution of the term in the final risk.

For example, for "Demo 1", the two most important subscales are Delinquency (with points of 1.973) and TradeOpenTime (with points of 1.947). Then, within each of the two subscales, we find two factors that contribute the most to that particular subscale's risk score. The two most important factors for each subscale are determined likewise by the product of the coefficient of each binary feature and the value of the binary feature itself. We finally output those binary features and their corresponding values as the most important contributing factors to the prediction made by our model.


Original Features
These are 23 original features given in the dataset.
Subscale Features
These are 10 composite features obtained from the previous 23 orginal features.
Output
This is the Risk Performance estimated from our model, which consists of only two outcomes: Good and Bad.
Subscale Contribution
Low Risk (No Default)
High Risk (Default)
ExternalRiskEstimate 81.9%
TradeOpenTime 78.9%
NumSatisfactoryTrades 74.2%
TradeFrequency 60.6%
Delinquency 79.9%
Installment 65.7%
Inquiry 57.9%
RevolvingBalance 64.1%
Utilization 44.2%
TradeWBalance 73.1%
Output 95.2% Risk of Default
Run Model
Risk Score
Overall Score+Bias

Global Model

Below is our Input Panel. Model will take a few seconds to run.

ExternalRiskEstimate(Decreasing)	61
MSinceOldestTradeOpen(Decreasing)	49
MSinceMostRecentTradeOpen(Decreasing)	19
AverageMInFile(Decreasing)	29
NumSatisfactoryTrades(Decreasing)	4
NumTrades60Ever2DerogPubRec(Increasing)	1
NumTrades90Ever2DerogPubRec(Increasing)	1
NumTotalTrades(No Constraint)	4
NumTradesOpeninLast12M(Increasing)	0
PercentTradesNeverDelq(Decreasing)	75
MSinceMostRecentDelq(Decreasing)	7
MaxDelq2PublicRecLast12M(Decreasing)	4
MaxDelqEver(Decreasing)	6
PercentInstallTrades(No Constraint)	75
NetFractionInstallBurden(Increasing)	64
NumInstallTradesWBalance(No Constraint)	2
MSinceMostRecentInqexcl7days(Decreasing)	0
NumInqLast6M(Increasing)	1
NumInqLast6Mexcl7days(Increasing)	0
NetFractionRevolvingBurden(Increasing)	56
NumRevolvingTradesWBalance(No Constraint)	1
NumBank2NatlTradesWHighUtilization(Increasing)	0
PercentTradesWBalance(No Constraint)	100

Output

Subscales	Risk Score	Weights	Points
ExternalRiskEstimate	0.819	1.593	1.305
TradeOpenTime	0.789	2.468	1.947
NumSatisfactoryTrades	0.742	2.273	1.686
TradeFrequency	0.606	0.358	0.217
Delinquency	0.799	2.470	1.973
Installment	0.657	1.175	0.772
Inquiry	0.579	2.994	1.733
RevolvingBalance	0.641	1.877	1.203
Utilization	0.442	1.119	0.495
TradeWBalance	0.731	0.214	0.157

Overall Score 11.488

Bias -8.495

Associated Risk (for the global model) **95.2%**

Activation Function

Figure 2: *Left*: Snapshot of visualization tool showing the global model. Colors indicate contribution to the final score. The 23 feature values are entered on the left. *Right*: Final combined score pop-up.

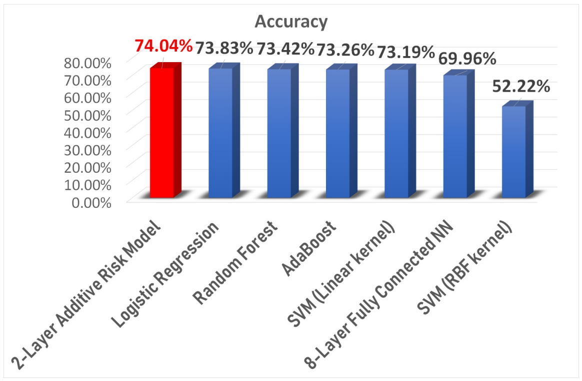


Figure 3: The accuracy of our model compared to other common machine learning models.

The factors are grouped by subscales and are displayed in decreasing order of importance (within each important subscale) to the global model's predictions.

3 Consistent rule-based explanations with `SetCoverExplanation`

As noted earlier, in addition to predicting risk using a globally interpretable model, we generate consistent rules that summarize broad patterns of the classifier with respect to the data. These are not strictly "explanations" in the sense that they do not explain the global model's computations; instead they provide useful patterns, which has been a popular form of explanation in the state-of-the-art literature on model explanations [7, 4, 13]. As an example, consider Observation 6 in the HELOC dataset for which the global model predicts a high risk of default. `SetCoverExplanation` returns the following rule-based explanation:

For *all* 700 people where:

- $ExternalRiskEstimate \leq 63$, and
- $NetFractionRevolvingBurden \geq 73$,

the global model predicts a high risk of default.

`SetCoverExplanation` asserts that our global model predicts high-risk for *all* of the 700 previous cases that satisfy these rules. Therefore these rules are *globally consistent*. In contrast, explanations (from other methods) that are not consistent may hold for one customer but not for another, which could eventually jeopardize trust.

In what follows, we formalize the discussion on consistent rules and put it into concrete mathematical terms. After defining rules in Section 3.1, we address aspects of optimization in Section 3.2. In Section 3.3 we describe how to use consistent rules to identify similar cases for case-based explanations.

3.1 Notation and definitions

Consider a training set $\mathbf{X}, \mathbf{y}$ comprising of a design matrix $\mathbf{X} \in \{0,1\}^{N \times P}$ (that is, N observations and P binary features), and a corresponding vector of binary labels $\mathbf{y} \in \{0,1\}^N$. Let $h^M : \{0,1\}^P \rightarrow \{0,1\}$ denote a classifier that was trained using the dataset $\mathbf{X}, \mathbf{y}$, and let $\mathbf{y}^M$ denote the vector of labels generated by the model h^M (the super-script M stands for model).

Assume $\mathcal{P}' \subseteq \mathcal{P}$ is a subset of features and $y \in \{0,1\}$ is a label. The *rule* $\mathcal{P}' \Rightarrow y$ describes the following binary function/classifier:

$$h^{\mathcal{P}' \Rightarrow y}(\mathbf{x}) = \begin{cases} y & \text{if } \left(\prod_{p \in \mathcal{P}'} x_p\right) = 1 \\ 1 - y & \text{otherwise.} \end{cases}$$

That is, for a given observation $\mathbf{x}$, the rule $\mathcal{P}' \rightarrow y$ predicts y based on the projection of the observation onto the subspace of features $\mathcal{P}'$; specifically, by applying a logical AND operator on the subset of features in $\mathcal{P}'$.

Let $\mathbf{x}_e, y_e$ denote an observation and the respective model prediction that we wish to create a rule for. We say that the rule $\mathcal{P}' \Rightarrow y_e$ provides a *consistent explanation* for $\mathbf{x}_e, y_e$ if the following conditions are met:

1. (Relevance) $x_{e,p} = 1$ for every $p \in \mathcal{P}'$.
2. (Consistency) When $x_{e,p} = 1$ for every $p \in \mathcal{P}'$, it must also hold that $y_i^M = y_e$ for every observation $\mathbf{x}_i \in \mathbf{X}$ where $x_{i,p} = 1$ for every $p \in \mathcal{P}'$.

The second condition establishes consistency by enforcing all observations in the dataset to agree with the rule, in the sense that all observations for which the binary variables in $\mathcal{P}'$ are true have the same label.

We measure the quality of a rule $\mathcal{P}' \rightarrow y$ using two criteria:

- *Sparsity* – the cardinally of $\mathcal{P}'$, that is, $|\mathcal{P}'|$. This captures to a certain degree the level of interpretability of the rule.
- *Support* – the number of observations in the dataset that satisfy the rule, that is, $|\{\mathbf{x}_i : \left(\prod_{p \in \mathcal{P}'} x_p\right) = 1\}|$. This serves as a measure of coverage for the applicability of the rule.

Note that the fact that the above definitions apply only to rules where features are equal to 1 may seem to be a limitation. However, one can easily extend the feature space by adding binary features that are equal to the complements on the original features, that is, by adding $\mathbf{X}^c = 1 - \mathbf{X}$. In this case, rules that contain complement features can be interpreted as rules where an original feature is equal to 0. In what follows, we assume that the design matrix $\mathbf{X}$ is of dimensions $N \times 2P$ and includes both the original and complement matrices: $[\mathbf{X}, \mathbf{X}^c]$.

3.2 Optimization

We now describe the formulation of multiple algorithms to generate rules that achieve the objectives of high sparsity and support.

Optimizing sparsity. Let b_p denote a binary decision variable that indicates whether $p \in \mathcal{P}'$. Denoting $P_e \triangleq \{p : x_{e,p} = 1\}$ the largest set of features that "agree" with observation x_e allows us to write Condition 1 as $\mathcal{P}' \subseteq P_e$. Therefore, we only consider variables b_p for which $p \in P_e$. In order for Condition 2 to hold, observations with labels different from y_e must not satisfy the rule $\mathcal{P}'$. That is, for each such observation i, a feature $p \in P_e$ must be selected for which $x_{i,p} = 0$. Each feature p therefore covers a set of oppositely labeled observations, and a feasible solution must cover all observations whose labels are different from y_e. This is an instance of the *Minimal Set Cover Problem* [3, 2].

More formally, let $A_p \subseteq \{1, \ldots, N\}$ denote the (constant) set of observations i that satisfy $x_{i,p} = 0$. Finding a rule with optimal sparsity is the solution to the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{p \in P_e} b_p & \\ \text{subject to} \quad & \sum_{p \in P_e} b_p \cdot \mathbb{1}[\mathbf{x}_i \in A_p] \geq 1, \quad & i \in \{i : y_i^M \neq y_e\} \\ & b_p, \in \{0,1\} & p \in P_e. \end{aligned} \tag{1}$$

We briefly note that we conducted a computational study on the HELOC dataset where sparse explanations were generated for each of the 10K observations, based on all other observations. The running time consistently took less than 7 seconds, and the average sparsity was under 3 features.

Optimizing support. Formulation (1) can be extended to incorporate support by adding a binary decision variable r_i (and an appropriate linear constraint) for each observation that indicates that the rule applies to the respective observation. An additional constraint was added to limit the number of features to a predefined constant MAX_SPARSITY in the resulting rule.

We experimented with our formulations by generating explanations on the HELOC dataset. We first solved Formulation 1, and set its solution as the value of MAX_SPARSITY in the modified formulation that optimizes support. We then increased the value of MAX_SPARSITY by 1 and 2 to relax the respective constraint, in order to improve the support size (at the cost of reduced sparsity).

We generated additional explanations where we maximized support for all observations in the HELOC dataset. We found that the average number of features in each explanation was unchanged and equal to 2.9 when MAX_SPARSITY was set to the solution of Formulation 1; sparsity slightly increased to 3.6 and 4.4 when MAX_SPARSITY was increased by 1 and 2, respectively. We also found that the number of explanations for which the support is less than 10 was 9.7% of all observations for the solution given by Formulation 1, and was equal to 4.7%, 1.2%, and 0.2% of all observations for the maximal support solution where MAX_SPARSITY was increased by 0, 1, and 2, respectively. Clearly, this indicates a tradeoff between support and sparsity; when the constraint on MAX_SPARSITY is relaxed, the cardinality of the rule increases and the rule becomes less interpretable, however, at the same time support increases which improves the confidence in the resulting rule. Overall, the average cardinality is nicely low and the support is reasonably high.

Optimization procedure for the challenge. In an actual deployed system, generating rules could be done offline for each user, since users' credit information changes infrequently over time. In contrast, in our code, we wanted to allow the user the ability to interact with the system to see how explanations are generated for any possible new observation. Therefore, we wanted to limit the running time for generating explanations to provide a reasonably short response.

To this end, we first created a database of explanations using the HELOC dataset. We created explanations for maximal sparsity, and maximal support subject to sparsity constraints of +0, +1,

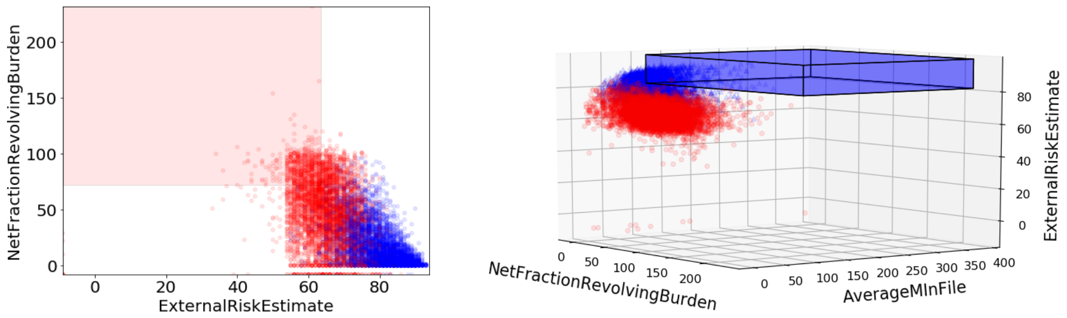
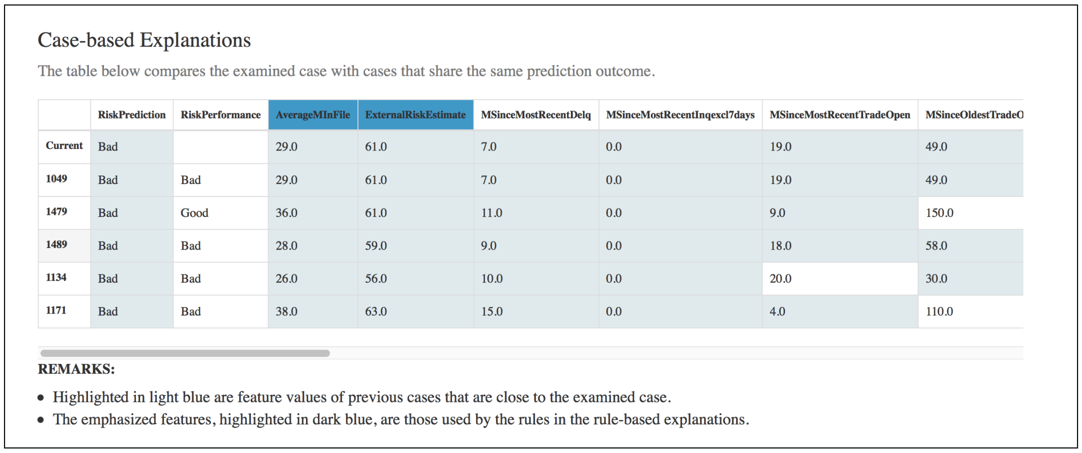


Figure 4: *Left:* an explanation for Observation 6 that has support of 700 observations; *Right:* an explanation for Observation 1005 that has support of 990 observations.



Case-based Explanations

The table below compares the examined case with cases that share the same prediction outcome.

	RiskPrediction	RiskPerformance	AverageMInFile	ExternalRiskEstimate	MSinceMostRecentDelq	MSinceMostRecentInqexcl7days	MSinceMostRecentTradeOpen	MSinceOldestTradeO
Current	Bad		29.0	61.0	7.0	0.0	19.0	49.0
1049	Bad	Bad	29.0	61.0	7.0	0.0	19.0	49.0
1479	Bad	Good	36.0	61.0	11.0	0.0	9.0	150.0
1489	Bad	Bad	28.0	59.0	9.0	0.0	18.0	58.0
1134	Bad	Bad	26.0	56.0	10.0	0.0	20.0	30.0
1171	Bad	Bad	38.0	63.0	15.0	0.0	4.0	110.0

REMARKS:

- Highlighted in light blue are feature values of previous cases that are close to the examined case.
- The emphasized features, highlighted in dark blue, are those used by the rules in the rule-based explanations.

Figure 5: Case-based explanations.

and +2 of optimal. We did the same for additional 10K random observations. When a user clicks to generate an explanation, the database is scanned and the sparsest rule whose support is greater than 10 is returned. Otherwise, if no rules were found, the maximal sparsity optimization problem is solved. If the support of the returned rule is greater than 10, that solution is returned. Otherwise, the maximal support optimization is solved, consecutively relaxing the constraint on the maximal sparsity. If rules with sufficiently large support were not found, the procedure is repeated with a threshold for stopping being 5. An error message is displayed if no explanations were found.

Figure 4 illustrates the tradeoff between simplicity and coverage for two rules generated by `SetCoverExplanation`. The rule on the left is 2-dimensional and the rule on the right involves 3 dimensions.

3.3 Case-based explanations

Here we present another popular form of explanation, case-based explanations [14, 11]. Given a previously unseen case, we can identify similar cases from the provided dataset to justify the prediction made by our model for the unseen case. To do this, we find all the cases in the dataset that satisfy the consistent rule-based explanation for the unseen case (obtained by `SetCoverExplanation`). We then rank these cases according to how many binary features (see Section 2 for how we obtain the binary features) they share with the unseen case, and present the five highest-ranked similar cases to the user. Figure 5 gives an example of a case-based explanation: our visualization contains a table showing the current case (top row) and previous cases that are most closely related to the current case (all other rows).

4 Conclusion

Since the FICO dataset does not seem to require a black box for good performance, perhaps many other applications in finance also do not require a black box. This answer to this remains unclear. However, challenges like this one can help us to determine the answer to this important question.

Appendix: Login Information

The web interface for our system can be found at:http://dukedatasciencefico.cs.duke.edu
with username dukedatascience and password OxNaUTsSjH0GQ

References

[1] C. Chen and C. Rudin. An optimization approach to learning falling rule lists. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 604–612, 2018.

[2] U. Feige. A threshold of ln n for approximating set cover (preliminary version). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 314–318. ACM, 1996.

[3] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[4] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):93, 2018.

[6] C. Hernández and S. Tecpan. Correct answers with wrong justifications? analysis of explanations in classical mechanics with fci test. In *Journal of Physics: Conference Series*, volume 1043, page 012056. IOP Publishing, 2018.

[7] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.

[8] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM, 2013.

[9] S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[10] S. Moro, R. Laureano, and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In *Proceedings of European Simulation and Modelling Conference (ESM'2011)*, pages 117–121. Eurosis, 2011.

[11] A. Peet and E. Pitcovski. Normal knowledge: Toward an explanation-based theory of knowledge. *The Journal of Philosophy*, 115(3):141–157, 2018.

[12] P. Regulation. General data protection regulation. *Official Journal of the European Union*, 59:1–88, 2016.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

[14] R. C. Schank, A. Kass, and C. K. Riesbeck. *Inside case-based explanation*. Psychology Press, 2014.