An Explainable Model for Credit Risk Performance

Chaofan Chen
cfchen@cs.duke.edu

Kangcheng Lin
kangcheng.lin@duke.edu

Cynthia Rudin
cynthia@cs.duke.edu

Yaron Shaposhnik
yaron@simon.rochester.edu

Sijia Wang
sijia.wang@duke.edu

Tong Wang
tong-wang@uiowa.edu

October 1st, 2018

Note: our entry can be found at http://dukedatasciencefico.cs.duke.edu
You will need a password to enter, which we provided separately.
Video: https://www.youtube.com/watch?v=QL7D2TMj7ck&feature=youtu.be

1 Introduction to our FICO Competition Entry

Our vision for this competition entry was to create a fully interpretable global model that combines the best of a traditional credit scoring system and a feedforward neural network, without sacrificing any accuracy to gain interpretability (and indeed we did not[1]). To achieve this end, we created a model that is subdivided into 10 smaller interpretable isotonic regression models (subscales), that are combined at the end into a globally interpretable model. This construction resembles a traditional additive credit scoring model; however, unlike traditional models, nonlinear transformations were used to generate subscale outputs. The final prediction thus arises from a two-layer feedforward neural network. 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.

Even though the global model is itself interpretable, we wanted to create simpler automatically-generated explanations for each prediction. 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 to generate rule-based explanations that are consistent with all training cases. We will discuss this method in more depth later. Third, we provide case-based explanations. Our algorithm chooses cases that are similar on important features to any current case that the user inputs.

2 Description of Global Model

Our global model is what we call a *two-layer additive risk model*, expressed as a two-layer network. The full model is displayed in an interactive graphic in Figure 1. After entering data about an individual (using the sliders or input-boxes on the left) and pressing "Run Model", the network is updated and displayed on the right. The network consists of the following layers:

- **Initial transformation**: The 23 original features are transformed into 182 piecewise constant features, where missing values were converted to binary features. Each original feature contributes to one of the 10 subscales.

  Figure 2 shows how the (original) feature "ExternalRiskEstimate" is transformed into 8 binary features (e.g., one of the binary variables is an indicator specifying whether $0 \leq ExternalRiskEstimate \leq 63$).

[1]The accuracy achieved by our global model in a cross-validation test is 75%, similar to other frequently used machine learning models.

The individual whose score is being displayed in the figure has ExternalRiskEstimate between 0 and 63, and thus received 1.799 points for this feature.

- **First layer (subscale creation)**: 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. In Figure 3, we see the four original features that constitute the "Delinquency" subscale. With the addition of a simple bias term, they are nonlinearly transformed through a sigmoid function to produce a probability, which is 79.8%. Users can click on the subscales to view the calculation, including the nonlinear transformation of features, their combination, and the nonlinear transformation of this combination to form a probabilistic risk estimate. Monotonicity constraints are enforced during learning; we discuss the formulation that enabled monotonicity with respect to these step functions shortly. L_2 regularization prevents overfitting. There are two important sources of nonlinearity: the initial transformation of features into step functions, and the sigmoid transformation of the linear combination of features.

- **Second layer (final risk estimation)**: The subscale results are linearly combined and again nonlinearly transformed into a final probability of failure to repay a loan. Figure 4 shows the final prediction of the model based on all subscales. Users can click on the output to view this calculation.

Design principles

Our model is a feedforward neural network (NN). While in general, fully connected NNs are hard to comprehend even when sparsity regularization is applied, our model is a two-layer additive model with carefully designed sparsity and monotonicity constraints, which makes the calculations easier to comprehend. We applied a customized training procedure to provide interpretability through sparsity and monotonicity:

- Sparsity – The resulting network is very sparse, with each node in the input layer feeding into a single node in the first layer. This structure aggregates the initial 23 features into 10 subscales that capture distinct aspects of a customer's behavior. There are only a small number of discretization levels associated with each feature. The contribution of each subscale to the final prediction can be easily observed by its weighted output (which is colored red to green in the visualization).

- Monotonicity – 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_j, the following features could be created: $b_{j,1} = \mathbb{1}[x_j < 10]$, $b_{j,2} = \mathbb{1}[x_j < 50]$, $b_{j,3} = \mathbb{1}[x_j < 75]$, and $b_{j,4} = \mathbb{1}[x_j < \infty]$. Note that all of these features use one-sided intervals. The trick is that convex combinations of these step functions yield monotonic functions. By enforcing the constraints that the coefficients for $b_{j,1}$, $b_{j,2}$, and $b_{j,3}$ must be non-negative, we shall guarantee a monotonically decreasing relationship between the original continuous feature x_j and the predicted subscale probability of default. From there, the one-sided intervals were split into small two-sided intervals (e.g., $10 \leq x_j < 50$) in which the sum of one-sided intervals is piecewise constant. If a feature x_j is monotonically increasing, we reverse the above one-sided inequality $<$ into $\geq$ (and also replace $\mathbb{1}[x_j < \infty]$ with $\mathbb{1}[x_j \geq 0]$). Finally, if a feature x_j has no desired monotonicity, we drop non-negativity constraints.

3 Explaining Individual Predictions Using Local Models

Our system provides three types of intuitive explanations for the predictions made by the global model.

- **Most important contributing factors:** We identify a list of factors that contribute most heavily to the final prediction. For example, Figure 5 presents the most important four factors 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 points, 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.

The factors are displayed in decreasing order of importance to the global model's predictions.

- **Consistent rule-based explanations:** We designed an algorithm called `SetCoverExplanation`, which generates a set of rules that are consistent with the global model's predictions on all previous customers. Thus, the explanations are 100% consistent with the global model on past data. For example, Figure 6 shows the consistent ruled-based explanation for observation "Demo 1". `SetCoverExplanation` asserts that our global model predicts high-risk for *all* of the 594 previous cases that satisfy these rules (including the examined case). 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.

  These rules are generated by solving a Mixed Integer Program (MIP) using Gurobi[(c)]. The MIP formulation relies on key observations about binary feature spaces:

  1. A subset of features and respective binary values forms a conjunctive rule.
  2. The conjunctive rule must agree with the examined case.
  3. All past cases with opposite predictions by the model must not satisfy the rule so that the explanation would remain globally consistent.

  These properties imply that finding the minimal set of rules is in fact a *minimal set cover problem*:

  1. Nodes correspond to cases.
  2. Sets are collection of cases that do not satisfy the rule. Each set is defined using a binary feature and its respective value.
  3. The objective is to minimize the number of features that comprise the rules.

  After solving an initial minimal set cover problem, `SetCoverExplanation` seeks additional solutions that improve *support* (that is, the number of previous cases that satisfy these rules) subject to constraints on the sparsity of the explanation. This is accomplished using a modification of the minimal set cover formulation. Maximizing support while constraining sparsity improves coverage and reliability of the explanation while maintaining interpretability.

  Computing consistent explanations requires solving a MIP which could lead to running-times of up to a minute (this could be tuned to improve running time or quality of the resulting solution). Nevertheless, computing explanations instantly is only required for testing purposes, and we expect that in an actual deployed system, explanations for most cases one would encounter online could be pre-computed offline, stored, and retrieved (memoized).

- **Case-based explanations:** The 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). All of the presented cases satisfy the rule-based explanations from `SetCoverExplanation`. Cases are sorted according to a distance metric that counts the number of identical features in the binary representation. See Figure 7 for example case-based explanations.

4 Summary

Our approach has several unique advantages that lead to increased understanding and trust. Specifically:

- The *global model compartmentalizes*, so that the subscales represent meaningful concepts and can be more easily verified. It is an interpretable neural network.

- The *subscales are flexible nonlinear functions* of the original variables.
- The *subscales themselves are meaningful*; each subscale is its own miniature risk model.
- *Monotonicity is enforced through positivity constraints* on the increments of feature values.
- *Rule-based explanations* are constructed using a novel optimization method, `SetCoverExplanation`. Its rules are *completely consistent* with the global model, and are well-supported in the database.
- *Case-based explanations* are given, where the distance metric that determines nearby cases reflects the *important factors for explaining the prediction.*
- An *interactive web-interface* allows users to investigate the model and see how it makes each individual prediction. The full model is shown; no calculations are hidden from the user in the global model. The display *highlights what factors are important* for prediction.

Thank you for the opportunity to enter this competition. We enjoyed the chance to participate.

Acknowledgments: Thank you to Joe Shamblin, who has provided tremendous assistance to our team in the creation of the website.

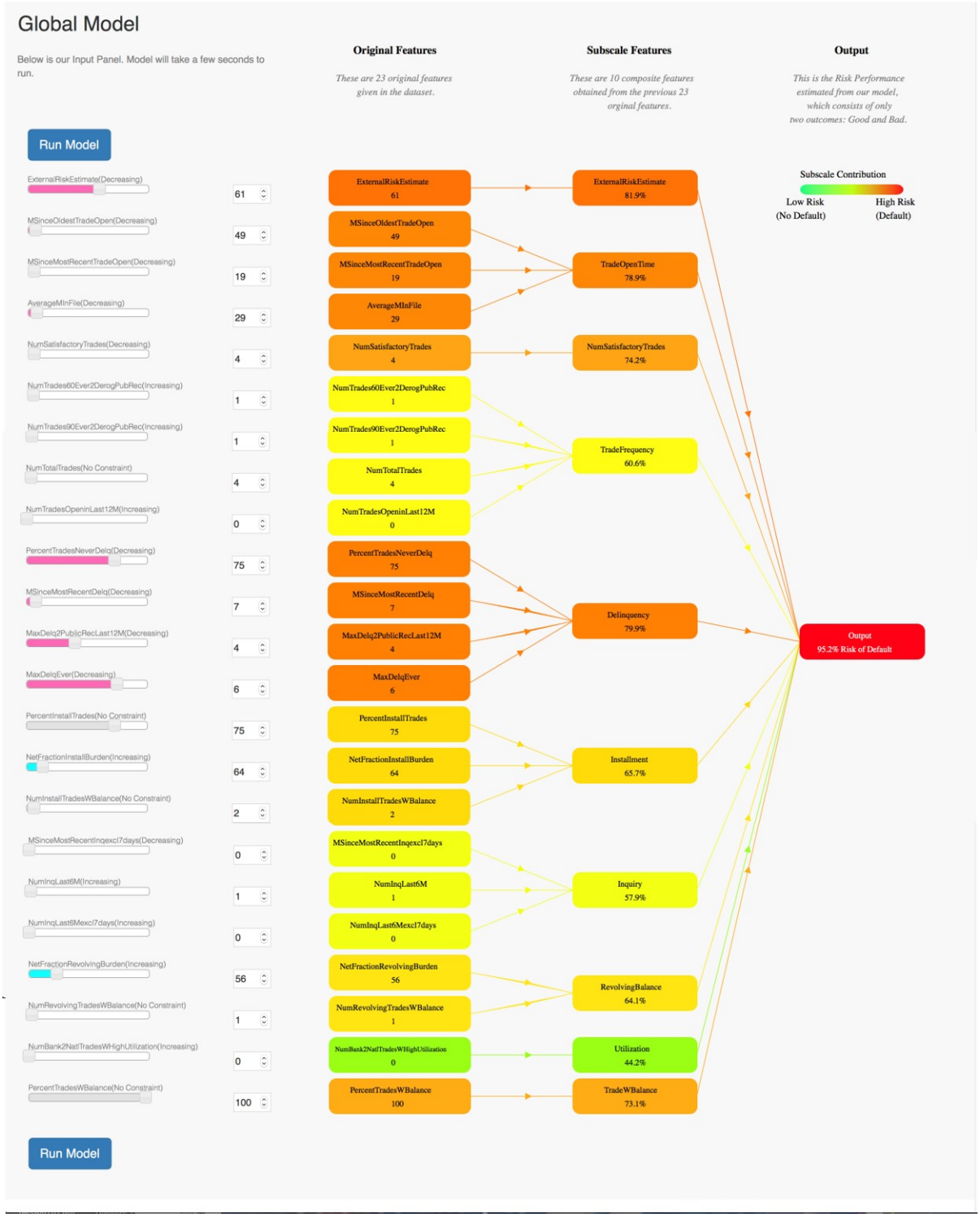

Figure 1: The global model.

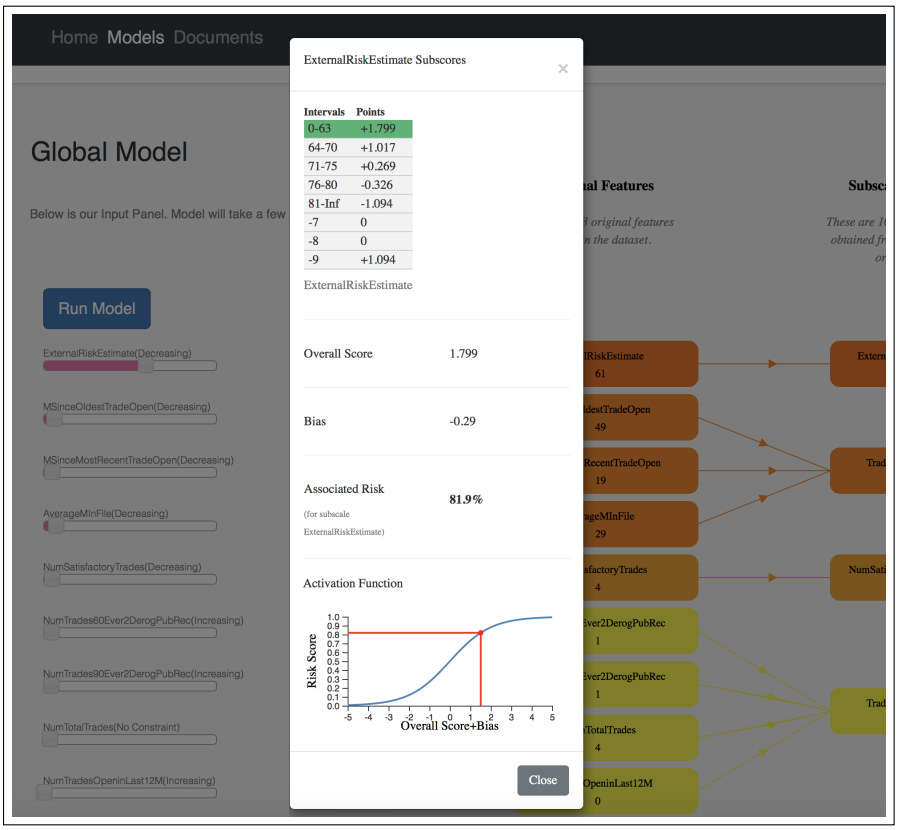

Figure 2: The subscale "ExternalRiskEstimate."

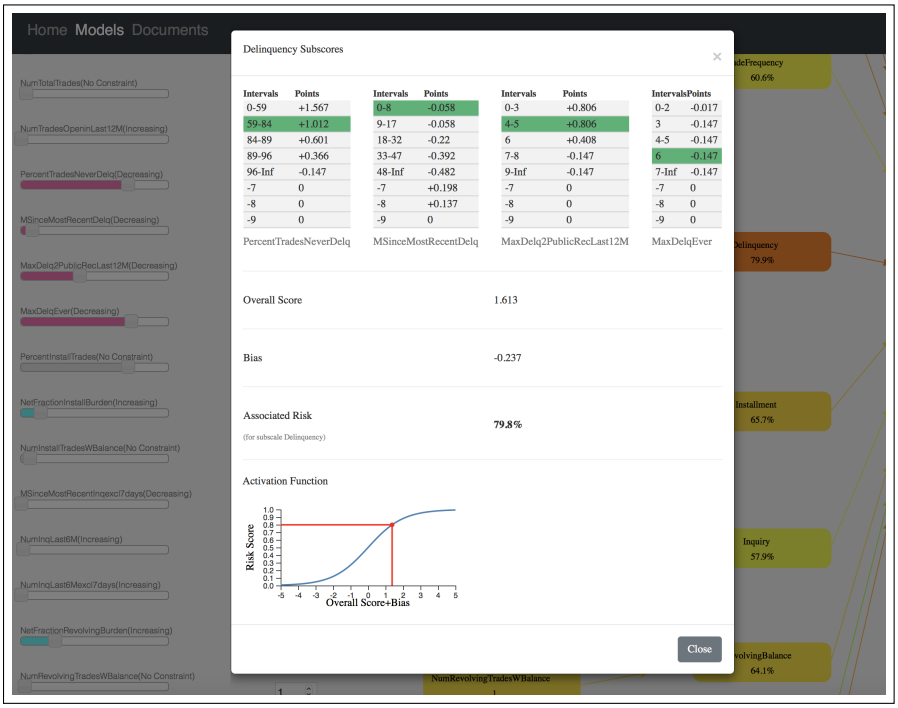

Figure 3: The subscale "Delinquency".

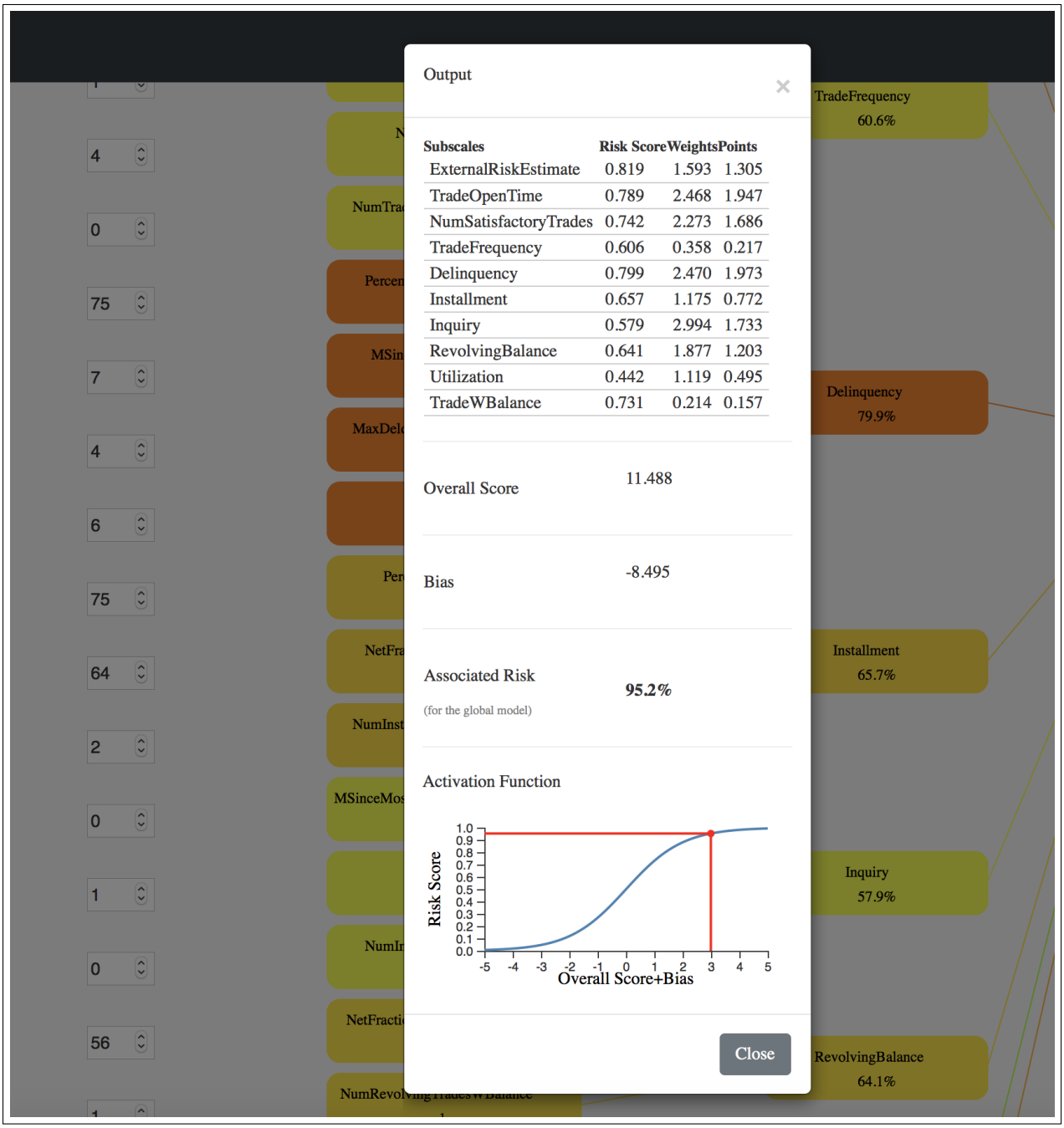

Figure 4: Output of global model.

Local Models

Most Important Contributing Factors

This is a list of factors that contribute most heavily to the final prediction in our model.

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)

Figure 5: Most important contributing factors.

Consistent Rule-based Explanations

The system solves an optimization problem (using Gurobi(c)) to compute the smallest set of rules that guarantees identical prediction by our global model.

For all 594 people whose:

- ExternalRiskEstimate is 63 or less

  and
- AverageMInFile is 48 or less

all of them were predicted to default.

Figure 6: Rule-based explanations.

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 7: Case-based explanations.