Interpretable Credit Application Predictions With Counterfactual Explanations

Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, Freddy Lecue

Interpretable Credit Application Predictions With Counterfactual Explanations

Rory Mc Grath[1], Luca Costabello[1], Chan Le Van[1],
Paul Sweeney[2], Farbod Kamiab[2], Zhao Shen[2], Freddy Lécué[1,3]
[1]Accenture Labs {rory.m.mc.grath, luca.costabello, chan.v.le.van}@accenture.com
[2]Accenture The Dock {paul.p.sweeney, farbod.kamiab, zhao.shen}@accenture.com
[3]Inria freddy.lecue@inria.fr

Abstract

We predict credit applications with off-the-shelf, interchangeable black-box classifiers and we explain single predictions with counterfactual explanations. Counterfactual explanations expose the minimal changes required on the input data to obtain a different result e.g., approved vs rejected application. Despite their effectiveness, counterfactuals are mainly designed for changing an undesired outcome of a prediction i.e. loan rejected. Counterfactuals, however, can be difficult to interpret, especially when a high number of features are involved in the explanation. Our contribution is two-fold: i) we propose *positive counterfactuals*, i.e. we adapt counterfactual explanations to also explain accepted loan applications, and ii) we propose two weighting strategies to generate more interpretable counterfactuals. Experiments on the HELOC loan applications dataset show that our contribution outperforms the baseline counterfactual generation strategy, by leading to smaller and hence more interpretable counterfactuals.

1 Introduction

Explaining predictions of black box models is of uttermost importance in the domain of credit risk assessment Bruckner [2018]. The problem is even more prominent given the recent right to explanation introduced by the European General Data Protection Regulation Goodman and Flaxman [2016], and a must due to regulation in the financial domain. A common approach to explain black box predictions focuses on generating local approximations of decisions. If f is a machine learning model taking the features X and mapping them to targets Y, then the goal is to find a subdomain of the feature variables and over that domain approximate $f \sim g$, where g is an interpretable and easy to understand function. There has been recent interest in model-agnostic methods of explainability. These methods look to create an *explainer* that should be able to explain any model treating the underlying model as a black box. Ribeiro et al. [2016a].

This paper focuses on *Counterfactual Explanations* Wachter et al. [2017], one of these model-agnostic methods. A counterfactual explanation may justify a rejected loan application as follows:

Your application was denied because your annual income is $30,000 and your current balance is $200. If your income had instead been $35,000 and your current balance had been $400 and all other values remained constant, your application would have been approved.

The explanation describes the required minimum change in inputs to flip the decision of the black box classifier. Note that the latter remains a black box: it is only through changing inputs and outputs that an explanation is obtained.

NIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, Montréal, Canada.

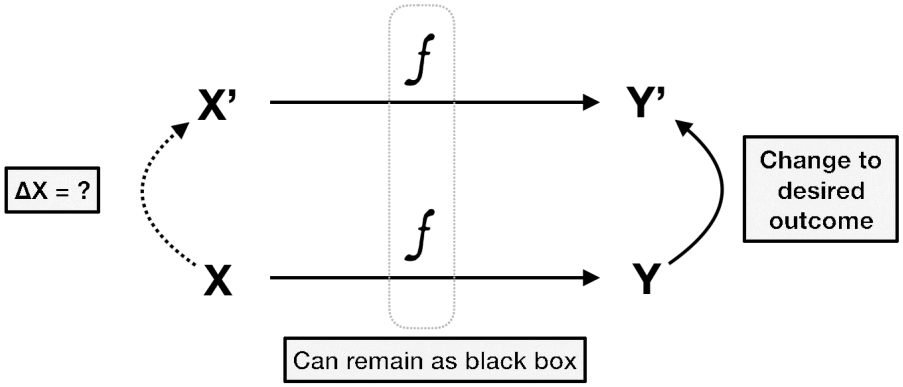


Figure 1: Explaining black box predictions with counterfactuals

Despite their effectiveness, two problems arise: on one hand counterfactuals are inherently designed to describe what it takes to flip the decision of a classifier, hence they poorly address the case in which the decision was satisfactory from an end user perspective (e.g. loan approved). Problems also arise when assessing the interpretability of counterfactuals: the generated counterfactuals often suggest to change a high number of features, therefore leading to less intelligible explanations. For example, counterfactuals generation strategies do not take into account the *importance* of the dataset features, thus underestimating or overestimating certain dimensions. This problem is of particular importance given that it has been showed that human short-term memory is unable to retain a large number of information units Vogel et al. [2001], Alvarez and Cavanagh [2004]. This remains as problematic in the finance domains of loan approval when data scientis and regulators are in the loop.

We predict loan applications with off-the-shelf, interchangeable black-box estimators, and we explain their predictions with counterfactual explanations. To overcome the aforementioned problems, we present the following contribution:

- **Positive Counterfactuals**: in case of a desired outcome, we interpret counterfactuals as a *safety margin*, i.e. a tolerance from the decision boundary. Such counterfactual explanations for positive predictions answer the question *"How much was I accepted by?"*.
- **Weighted Counterfactuals**: inspired by Huysmans et al. Huysmans et al. [2011], we use the size of explanations as a proxy to measure their interpretability. To obtain more compact (and hence more intelligible) counterfactuals we introduce weights in the their generation strategy. We propose two weighting strategies: one based on global feature importance, the other based on nearest neighbours.

We experiment on a credit application dataset and show that our weighted counterfactuals generation strategies lead to smaller counterfactuals (i.e. counterfactuals that suggest to change a smaller number of features), thus delivering more interpretable explanations.

2 Related Work

Local Interpretability A number of works focus on explaining *single predictions* of machine learning models, rather than the model as a whole. This task is also known as *local interpretability*. White box models come with local explanations by design: traditional transparent design approaches include decision trees and rule extraction Guidotti et al. [2018], Molnar [2018]. In that respect some works such as Craven and Shavlik [1995] built surrogate models by interfacing complex models such as deep neural networks with more interpertable models such as decision trees. The authors aim at mimicing the behaviour of a complex model with a much simpler model for interpretability purpose. Other approaches are instead model-agnostic, and also address explanations of predictions of black box models. LIME generates local explanations from randomly generated neighbours of a

record. Features are weighted according to distances from the record.Ribeiro et al. [2016b]. SHAP is another approach based on feature importance for each record Lundberg and Lee [2017]. Other recent lines of research rely on *example-based explanations*: Prototype Bien and Tibshirani [2011] and Criticism Kim et al. [2016] Selection are two recent examples of this. Prototypes are tuples representative of the dataset, whereas criticisms are examples which are not well-explained by prototypes. Adversarial examples Kurakin et al. [2016] are another example-based approach, but they are designed to flip the decision of a black-box predictor rather than explaining it. Counterfactual explanations are also an example-based strategy, but unlike adversarial examples, they inform on how a record features must change to radically influence the outcome of the prediction.

Interpretable Credit Risk Prediction Providing a comprehensive review of more than 20 years of research in credit risk prediction models is out of the scope of this paper. The survey by Lyn et al. Thomas et al. [2017] gives a comprehensive and up-to-date overview). Huang et al. Huang et al. [2004] briefly mention an explanation of predictive models for credit rating, but their survey limits to ranking features by importance with variance analysis. A more recent survey by Louzada et al. focuses on predictive powerLouzada et al. [2016] only. Instead, we list works that consider interpretability as a first-class citizen. A number of works rely on white box machine learning pipeline, mostly using decision trees and rules inference: Khandani et al. Khandani et al. [2010] propose a pipeline to predict consumer credit risk with manual feature engineering and decision trees. The latter being an explainable model, we can consider this work as a rather interpretable approach. Florez-Lopez et al. adopt an ensemble of decision trees and explain predictions with rules López and Ramon-Jeronimo [2015]. Predictive power is encouraging, but there is no comparison against neural architectures. Martens at al. combine rule extraction with SVMs Martens et al. [2007]. Obermann et al. compare decision tree performance to grey and black boxes approaches on an insolvency prediction scenario Obermann and Waack [2015, 2016]. Other white-box approaches include Markov models for discrimination Volkov et al. [2017] and rule inference Xu et al. [2017]. Black box approaches show the most promising predictive power, to the detriment of interpretability. Danenas et al adopt SVM classifiers Danenas and Garsva [2015]. Addo et al. leverage a number of black-box models, including gradient boosting and deep neural architectures Addo et al. [2018]. Although they evaluate the predictive power of their models they do not attempt to explain their predictions, either locally or globally. To the best of our knowledge, no work in literature focuses on local interpretability for black box models applied to credit risk prediction.

3 Preliminaries: Counterfactual Explanations

A counterfactual explanation describes a generic causal situation in the form:

> *Score y was returned because variables X had values $(x_1, x_2...)$ associated with them. If X instead had values $(x'_1, x'_2, ...)$, and all other variables had remained constant, score y' would have been returned.*

Counterfactuals do not need to know the internal structure or state of model or system (e.g. neural network, logistic regression, support vector machine, etc). We treat f as a black box that takes the feature vector x and generates the outcome y, and we determine what is the *closest* x' to x that would change the outcome of the model from y to the desired y' (Figure 1). When generating counterfactuals it is assumed that the model f, the feature vector x and the desired output y' are provided. The challenge is finding x', i.e. an hypothetical input vector which falls close to x but also for which $f(x')$ falls sufficiently close to y'.

Generating Counterfactuals. We generate counterfactual explanations by calculating the smallest possible change (ΔX) that can be made to the input X, such that the outcome flips from y to y'. We generate counterfactuals by optimizing the following loss function $\mathcal{L}$, as proposed by Wachter et al. Wachter et al. [2017]:

$$\mathcal{L}(x, x', y', \lambda) = \lambda(\hat{f}(x') - y')^2 + d(x, x') \tag{1}$$

$$arg \min_{x'} \max_{\lambda} \mathcal{L}(x, x', y', \lambda) \tag{2}$$

where x is the actual input vector, x' is counterfactual vector, y' is the desired output state, $\hat{f}(...)$ is the trained model, λ is the balance weight. λ balances the counterfactual between obtaining the exact desired output and making the smallest possible changes to the input vector x. Larger values for λ favor counterfactuals x' which result in a $\hat{f}(x')$ that comes close to the desired output y', while smaller values lead to counterfactuals x' that are very similar to x. The distance metric $d(x, x')$ measures Δx, i.e. the amount of change between x and x'. We use the Manhattan distance weighted feature-wise with the inverse median absolute deviation (MAD) 3. Such metric is robust to outliers, and introduces sparse solutions where most entries are zero Wachter et al. [2017]. Indeed, the ideal counterfactual is one in which only a small number of features change and the majority of them remain constant. The distance metric $d(x, x')$ can be written as:

$$d(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j}, \tag{3}$$

$$MAD_j = median_{i \in \{1,...,n\}}\Big(\Big|x_{i,j} - median_{l \in \{1,...,n\}}(x_{l,j})\Big|\Big). \tag{4}$$

To generate counterfactuals we adopt the iterative approach described in Algorithm 1. We optimize $\mathcal{L}$ with the Nelder-Mead algorithm, as suggested in Molnar [2018]. We constrain the optimisation with a tolerance ε s.t $|\hat{f}(x') - y'| \leq \varepsilon$. The value for ε depends on the problem space and is determined by the range and scale of y. Step 3 iterates over λ until the ε constraint is satisfied. A check is performed for a value greater than ε as increasing λ will place more weight on obtaining an $\hat{f}(x')$ closer to the given desired output y'. Once an acceptable value for λ is obtained for the given x and y' a set of counterfactuals can be obtained by repeating steps 1 and 2 with the calculated λ. Note that we constrain the features manually, since the heuristic in Algorithm 1 and the adopted optimization algorithm are designed for unconstrained optimization.

Algorithm 1: Counterfactual generation heuristic

1 sample a random instance as the initial x'
2 optimise $L(x, x', y', \lambda)$ with initial x'
3 **while** $|\hat{f}(x') - y'| > \varepsilon$ **do**
4 increase λ by step-size α
5 optimise $L(x, x', y', \lambda)$ 1 with new x'
6 return x'

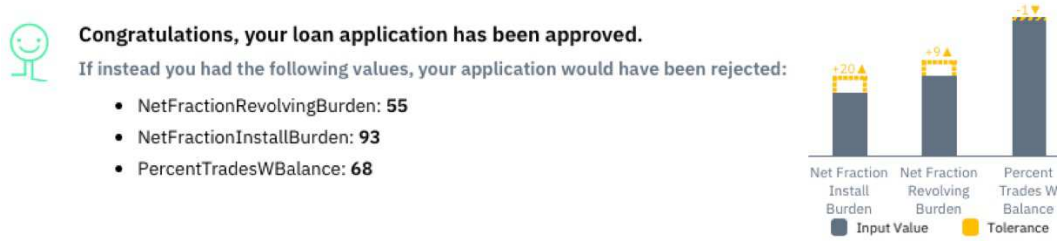
4 Contribution

In this section we describe our two main contributions made towards the explainability of black box machine learning pipelines that predict credit decisions. Using *positive counterfactuals* we explain why a loan was accepted and provide details that help inform an individual when making future financial decisions. Next we present weighted counterfactuals that aim at personalizing the counterfactual recommendations that are provided to individuals that received an undesirable outcome (i.e. their loan was denied).

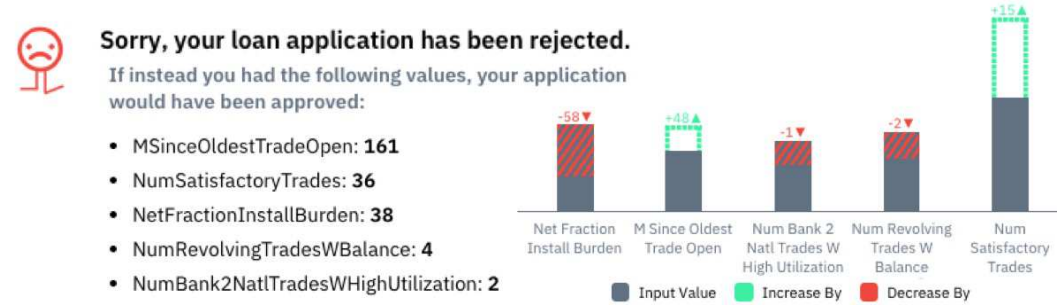
4.1 Counterfactuals for positive predictions

In order to explain why applications were accepted, we applied counterfactuals to the scenario where the individual received the desired outcome, i.e *positive counterfactuals*. Here instead of answering the question "Why wasn't I accepted?" we focus on the question of "How much was I accepted by?". Such approach informs the individual about the features and value ranges that were important for their specific application, thus favouring more informed decisions about potential future financial activities. For example, if the individual is considering an action that may temporarily increase their number of delinquencies then, armed with *positive counterfactuals*, they will have a better understanding of the impact on future loan applications.

In the binary classification case we achieve *positive counterfactuals* by setting the target y' to be the decision boundary i.e $P(y = 1) = 0.5$. This allows us to identify the locally important features that



(a) Positive counterfactual explanation



(b) Counterfactual explanation

Figure 2: Graphical depictions of a positive (a) and negative (b) counterfactual explanation. Note (a) answers the question *"How much was I accepted by?"* - thus leading to tolerances (highlighted in yellow), whereas (b) explains why the credit application was rejected. In this case the counterfactual explanation suggests how to increase (green, dashed) or decrease (red, striped) each feature.

would push the individual to the threshold of being accepted. Another way of viewing this is that these are the features that locally contribute to the desired outcome.

We present this information to the individual and display it as *tolerance*. In Figure 2 this is illustrated by a dashed line. Given that future actions do not reduce the indicated features below the dashed line, and all other features remain constant, then the individuals application should remain likely to be approved.

4.2 Weighted Counterfactuals

The general implementation of counterfactuals described in Section 3 assumes all features are equally important and changing each feature is equally viable. This, however, is not necessarily the case. For each feature its ability to change and the magnitude of the change may vary on a case by case bases. In order to capture this information and create more interpretable actionable recommendations, the generated counterfactuals need to take this into consideration. For example some individuals may be able to increase their savings, while others instead may find it easier to reduce their current expenses. There are also cases where some features may be fixed or immutable. Features like the number of delinquencies in the last six months is historical and fixed. Recommending to change these types of features would be of little use. Our intuition is that promoting highly discriminative features during the generation of counterfactuals leads to more compact, hence better interpretable explanations Huysmans et al. [2011].

We address these issues by introducing a weight vector θ to the distance metric defined in Equation 3. This vector promotes highly discriminative features.

$$d_2(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j} \theta_j, \tag{5}$$

We propose two different strategies to generate these weight vectors. The first relies on the global feature importance, the second relies on a Nearest Neighbors approach. The goal is obtaining coun-

terfactuals that suggest a smaller number of changes or focus on values that are relevant to the individual and have historically been shown to vary.

Global feature importance. We compute global feature importance using analysis of variance (ANOVA F-values) between each feature and the target, and we create a weight vector that promotes highly discriminative features. Our goal is obtaining a smaller set of features in the resulting counterfactual recommendation, thus obtaining more compact explanations.

K-Nearest Neighbors. The second approach uses K-Nearest Neighbors to find cases that are close to the individual but have achieved the desired results. Looking at the nearest neighbors and aggregating over the relative changes we build a weight vector θ that captures the locally important features for this individual that have historically been shown to change. Here we aim to find counterfactuals containing features that are more actionable by the individual. By using K-Nearest Neighbors approach these weights can be automatically learned when applied to new problem spaces.

5 Experiments

We perform a a binary classification task on a credit application dataset. We train a range of black box models and we explain their predictions with counterfactuals, the goal being explaining the classifier decision to reject or accept a loan application. We perform two separate experiments: first, we carry out a preliminary evaluation of the predictive power of our pipeline. This is not the primary focus of this paper, but it is a required step to gauge the quality of predictions. In a second experiment, we assess the size of counterfactuals generated by our weighted counterfactuals generation.

5.1 Experimental Settings

Dataset. We experiment with the HELOC (Home Equity Line of Credit) credit application dataset. Used in the FICO 2018 xML Challenge[1], it includes anonymized credit applications made by real homeowners. We drop highly correlated features and filter duplicate records. After pre-processing we obtain 9,870 records (of which 5,000 positives, i.e. accepted credit applications), and 22 distinct features.

Implementation Details. Our machine learning pipeline is written in Python 3.6. This includes preprocessing, training, counterfactuals generation, and performance evaluation. We use scikit-learn 0.20 for the black box classifiers[2]. All experiments were run under Ubuntu 16.04 on an Intel Xeon E5-2620 v4 2.10 GHz workstation with 32 GB of system memory.

5.2 Results

Predictive Power As preliminary experiment, we assess the predictive power of four classifiers: logistic regression (LogReg), gradient boosting (GradBoost), support vector machine with linear kernel (SVC), and multi-layer perceptron (MLP). Logistic regression apart, the others fall within the black box category. We perform 3-fold, cross-validated grid search model selection over a number of hyperparameters. We adopt balanced class weights for logistic regression, exponential loss for gradient boosting, for each dataset. SVM uses balanced weights, $C = .001$. The neural network uses one hidden layer with 22 units. We use the logistic activation function. Where not specified, we rely on scikit-learn defaults. Results in Table 1 show the predictive power of the best models. Metrics are 3-fold cross-validated.

Counterfactuals Size The preliminarily results of the different weighting strategies as described in Section 4.2 are presented in Table 2. We experiment with 5,000 loan applications in the dataset: we generate a counterfactual explanation for each of them, and compute the average counterfactuals size. Results show that both weighted strategies bring counterfactuals that have a smaller mean and standard deviation. We also observed that in general the average size of the counterfactual recommendations can vary dramatically for the same data given the underlying model.

[1] `https://community.fico.com/s/explainable-machine-learning-challenge`

[2] `http://scikit-learn.org/`

Table 1: Predictive power of the adopted black box classifiers. Best results in bold.

	HELOC	
Model	F1	Acc
LogReg	**0.72**	0.73
MLP	0.70	0.71
GradBoost	**0.72**	**0.74**
SVC	**0.72**	0.73

Table 2: Average size (i.e. average number of features) of generated counterfactual explanations, for each adopted black box classifier. Smaller counterfactuals mean more interpretable explanations. *Importance*=global feature importance strategy, *KNN*=k-nearest neighbours. KNN uses $k = 20$. Best results in bold.

	HELOC		
Model	Baseline	Importance	KNN
LogReg	4.86±1.84	**3.95±1.69**	4.71±1.72
MLP	8.88±2.54	**8.34±2.58**	8.45±2.53
GradBoost	1.5±0.6	**1.49±0.58**	1.5±0.58
SVC	2.5±1.32	**2.01±1.14**	2.44±1.27

In general the global feature importance results in features with a lower mean and standard deviation. We obtain explanations which are 11.2% smaller on average using the global feature importance strategy against the baseline. This is to be expected, as we promote more discriminative features and as a consequence less ancillary features are required. The benefit in the KNN approach is that the counterfactuals are weighted on the features that are locally important. Here we see that while they may not be the best approach they never perform worse than the baseline. The benefit of the weighting strategies comes with helping the optimization process converge on a local optimum when the underlying space is complex. We look to investigate this claim in future work.

6 Conclusion

We explain credit application predictions obtained with black box models with counterfactuals. In case of positive prediction, we show how counterfactuals can be interpreted as a safety margin from the decision boundary. We propose two weighted strategies to generate counterfactuals: one derives weights from features importance, the other relies on nearest neighbours. Experiments on the HELOC loan applications dataset show that weights generated from feature importance lead to more compact counterfactuals, therefore offering more compact and intelligible explanations for end users. Future work will focus on validating the effectiveness of our counterfactual explanations against human-grounded and application-grounded evaluation protocols (including the claim that smaller counterfactuals are indeed more interpretable). We will also experiment with weighting strategies that rely on model-specific feature importance, i.e. effect of feature perturbation on entropy of changes in predictions.

References

Matthew A Bruckner. Regulating fintech lending. *Banking & Financial Services Policy Report*, 37, 2018.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *stat*, 1050:31, 2016.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016a.

Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017.

Edward K Vogel, Geoffrey F Woodman, and Steven J Luck. Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):92, 2001.

George A Alvarez and Patrick Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological science*, 15(2):106–111, 2004.

Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933, 2018.

Christoph Molnar. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/, 2018.

Mark Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pages 24–30, 1995.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *CoRR*, abs/1606.05386, 2016b. URL `http://arxiv.org/abs/1606.05386`.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

Lyn Thomas, Jonathan Crook, and David Edelman. *Credit scoring and its applications*, volume 2. Siam, 2017.

Zan Huang, Hsinchun Chen, Chia-Jung Hsu, Wun-Hwa Chen, and Soushan Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision support systems*, 37(4):543–558, 2004.

Francisco Louzada, Anderson Ara, and Guilherme B Fernandes. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science*, 21(2):117–134, 2016.

Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34:2767–2787, 2010.

Raquel Flórez López and Juan Manuel Ramon-Jeronimo. Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. *Expert Systems with Applications*, 42(13):5737–5753, 2015. doi: 10.1016/j.eswa.2015.02.042. URL `https://doi.org/10.1016/j.eswa.2015.02.042`.

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.

Lennart Obermann and Stephan Waack. Demonstrating non-inferiority of easy interpretable methods for insolvency prediction. *Expert Systems with Applications*, 42(23):9117–9128, 2015.

Lennart Obermann and Stephan Waack. Interpretable multiclass models for corporate credit rating capable of expressing doubt. *Frontiers in Applied Mathematics and Statistics*, 2:16, 2016.

Andrey Volkov, Dries F Benoit, and Dirk Van den Poel. Incorporating sequential information in bankruptcy prediction with predictors based on markov for discrimination. *Decision Support Systems*, 98:59–68, 2017.

Pu Xu, Zhijun Ding, and MeiQin Pan. An improved credit card users default prediction model based on ripper. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1785–1789. IEEE, 2017.

Paulius Danenas and Gintautas Garsva. Selection of support vector machines based classifiers for credit risk domain. *Expert Systems with Applications: An International Journal*, 42(6):3194–3204, 2015.

Peter Martey Addo, Dominique Guegan, and Bertrand Hassani. Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38, 2018.