Using Small Business Banking Data for Explainable Credit Risk Scoring

Wei Wang, Christopher Lesner, Alexander Ran, Marko Rukonic, Jason Xue, Eric Shiu
Intuit Inc., Mountain View, CA

Abstract

Machine learning applied to financial transaction records can predict how likely a small business is to repay a loan. For this purpose we compared a traditional scorecard credit risk model against various machine learning models and found that XGBoost with monotonic constraints outperformed scorecard model by 7% in K-S statistic. To deploy such a machine learning model in production for loan application risk scoring it must comply with lending industry regulations that require lenders to provide understandable and specific reasons for credit decisions. Thus we also developed a loan decision explanation technique based on the ideas of WoE and SHAP. Our research was carried out using a historical dataset of tens of thousands of loans and millions of associated financial transactions. The credit risk scoring model based on XGBoost with monotonic constraints and SHAP explanations described in this paper have been deployed by QuickBooks Capital to assess incoming loan applications since July 2019.

1 Introduction

Intuit QuickBooks Online (QBO) offers software solutions for millions of small businesses in US for accounting, recording sales and bill payments, running payroll, etc. Our research shows that the financial records generated by these activities can predict how likely a small business is to repay a loan. This is economically important because accurate loan outcome predictions, by reducing information asymmetry (Akerlof 1970), benefit both lenders and borrowers. Lenders benefit because when loan outcome predictions are more accurate, losses due to loan defaults are reduced and with lower losses lenders can offer more competitive loan interest rates and expand their business to borrowers who would otherwise be denied loans. Small businesses benefit because good loan candidates can now access loans that were unavailable before and/or at better interest rates, while poor loan candidates are prevented from taking loans that are likely to harm both them and their lenders.

A unique challenge of using machine learning for credit risk decisions is that many countries (including USA) have regulations that require lenders to clearly explain their decisions. These regulations are motivated by findings that errors in the information collected by credit reporting agencies adversely impacts millions. For example, America's Federal Trade Commission (FTC 2013; 2015) found that: *"One in four consumers identified errors on their credit reports that might affect their credit scores;one in five consumers had an error that was corrected by a credit reporting agency after it was disputed, on at least one of their three credit reports; Four out of five consumers who filed disputes experienced some modification to their credit report."*

Almost all lenders use financial information about the borrower to assess their financial health. For business borrowers, underwriters want to know the type of business, the purpose of the loan and liquidity/solvency/profitability of the business as assessed using financial health ratios such as: Debt Ratio, Debt-to-Income Ratio, Debt-to-Equity Ratio, Loan-to-Value Ratio, Debt Service Coverage Ratio, Current Ratio, Inventory Ratio, etc. When a loan application is rejected using these eligibility rules and ratios, it is relatively straightforward to explain the reason for rejection by listing the set of failed criteria. However modern statistical and machine learning techniques (Perlis 2011; Duffie 2003; Chen and Guestrin 2016) can model credit risk using much richer information extracted from transactions directly downloaded from borrowers' financial accounts. When loan decisions are the results of machine learning models tracking thousands of relationships across hundreds of interdependent factors, explaining credit decision requires a more sophisticated approach.

2 Outline

The rest of the paper is organized as follows: First, we discuss our loan outcomes dataset and how credit risk models are evaluated. Traditional scorecard risk models are described followed by risk models that use machine learning algorithms such as XGBoost (Chen and Guestrin 2016). Second, we discuss the generation of decision explanations and compare results from two different approaches. Next, we discuss some practical aspects of explaining credit risk decisions. Then we describe the impact of our research on how QuickBooks Capital issues loans. Finally, we conclude the

paper with a summary of what we have learned.

3 Data, Features, Outcomes and Models

Our research was carried out using loan performance data from Intuit QuickBooks Capital which has offered business loans to users of QBO since 2017. These loans are repaid weekly or monthly over a period of six, nine or 12 months. The average loan size is $19,000 with APRs from 12% to 32%. See (Intuit 2019) for details. QBO users are eligible for these loans if they can show good financial health.

Between 2017 and 2019, tens of thousands of loans have been issued, and over a quarter of issued loans have reached maturity. Each loan applicant supplies about six months of business banking data downloaded directly from their financial institutions. Thus alongside the tens of thousands of applications our dataset has hundreds of millions of business banking transactions. A number of features are derived corresponding to: account balance patterns, cash flow trends, composition of recurring liabilities, seasonality and other spending patterns, frequency of negative financial events such as overdrafts and late payments, etc.

We will not discuss here the hundreds of features that can be extracted from banking transactions apart for noting that this kind of data is intrinsically noisy. Some of the noise is generated during download: transactions may include web scraping errors, some transactions may be duplicated, others may entirely disappear. Noise also comes from the processes of financial systems that generate transaction descriptions as representation of the actual business events. Dedicated machine learning systems process these noisy representation in order to recognize what they represent (income vs refunds vs account transfers, expenses vs loan payments vs late fees, etc). Despite the noisy channels that generate and transmit transaction information, these machine learning systems today are ˜80% accurate when predicting where each financial transaction belongs in the personal chart of accounts of each small business. Our development of these dedicated systems has been discussed in (Lesner et al. 2019). In addition to noise introduced by information representation and transmission, our dataset exhibits significant variability due to the differences in the nature of business among loan applicants. QuickBooks Capital serves a widely varied population of small businesses: building contractors, flower shops, cement factories, delivery services, etc.

Loan Outcomes

The outcome of a credit decision is not fully known until the loan has matured and either the full amount due is repaid in the expected time or what is repaid is a partial amount and/or over a much longer period of time. We define a loan to be in *good standing* when timely payments are being made or payments are less than 60 days past due. Using this definition for our discussion, we will simplify loan outcomes as follows:

- *Indeterminate Outcome* loans are all those in good standing which mature in more than 30 days – these loans we exclude from further consideration.
- *Good Outcome* loans are all those loans still in good standing which will mature in 30 days plus all those loans already repaid in full.
- *Bad Outcome* loans are all the rest – the ones that are delinquent (60+ days past due) plus the loans not fully repaid (write-offs due to default).

Scorecard Risk Models

Credit risk models predict how likely a given loan will end in a *bad outcome*. A long-established and commonly used type of risk model, called a Scorecard (Perlis 2011), can be thought of as a table listing loan application features each having a number of feature bins with corresponding weights. In such a table there may be a feature called "credit history depth" with the following possible bins: less than one year, less than two years, less than four, less than eight, more than eight years. The weights associated with these five bins might be -100, 0, 20, 40, 100 and only one of these weights would be selected when that application feature is being scored. Thus each feature contributes one feature bin weight and the overall application risk score is computed as the sum of these feature bin weights in log-odds space and then back-transformed to score space.

Construction of scorecard risk models requires two concepts: **weight of evidence (WoE)** and **information value (IV)**:

- WoE tells the predictive power of a given feature bin. It measures how well that one feature value separates *good outcome* loans from *bad outcome* loans.
- IV ranks the relative importance of features and feature bins, i.e., higher IV indicates higher predictive power which makes it possible to construct a scorecard risk model from just those features that matter most.

WoE and IV are computed as follows:

$$WoE = \ln(\frac{\Pr(Good)}{\Pr(Bad)}) \tag{1}$$

$$IV = \sum_{i=1}^{n}(\Pr(Good_i) - \Pr(Bad_i)) \times \ln(\frac{\Pr(Good_i)}{\Pr(Bad_i)}) \tag{2}$$

Table 1 shows WoE and IV for a sample feature. While it is trivial to bin categorical features, binning continuous variables needs more sophistication. For our scorecard risk model, we use the Break and Heal (XENO 2008; Perlis 2011) algorithm in order to maximize feature IV. Manual adjustment of bins is then applied to satisfy monotonic relationship requirement (see discussion in section 4).

Scorecard risk models are often designed by subjecting loan application features and historical loan outcomes to fitting techniques such as logistic regression. Missing and extreme feature values can be binned separately and both categorical and continuous variables can be handled with optimized binning (XENO 2008). During the construction, the goal is to find bins and bin weights so that resulting scorecards separate good and bad outcomes as evaluated using portions of data held back and not used for optimizing bins and weights.

Table 1: WoE and IV calculated for a sample feature

Interval Levels	% Good	% Bad	% Total	P(Bad)	P(Good)	WoE	IV
0 - 0.007	60.2%	87.6%	84.7%	0.07	0.93	-0.375	0.103
0.007 - 518.8	3.4%	0.2%	0.6%	0.62	0.38	2.617	0.082
518.8 - 1732.0	5.5%	0.7%	1.2%	0.47	0.53	2.035	0.097
1732.0 - 6070.6	9.1%	1.9%	2.7%	0.36	0.64	1.561	0.113
6070.6 - 10103.1	5.1%	1.3%	1.7%	0.32	0.68	1.383	0.053
10103.1 - 17505.6	4.4%	1.7%	2.0%	0.24	0.76	0.981	0.027
17505.6 - 89016.7	8.2%	4.3%	4.7%	0.18	0.82	0.647	0.025
89016.7 - Inf	4.1%	2.3%	2.5%	0.17	0.83	0.578	0.010
Total	100.0%	100.0%	100.0%	0.10	0.90		0.510

Table 2: Risk Model K-S Performance

Model Type	K-S Statistic
Logistic Regression	0.335
Neural Network	0.385
Random Forest	0.390
GBDT (sklearn)	0.408
Scorecard	**0.410**
XGBoost	**0.437**

Due to their simplicity, scorecard models are easy to use and easy to understand and explain. To use one requires just a sheet of paper and it is relatively straightforward to explain decisions based on a scorecard model. The main disadvantage of scorecard models is that their simplicity can also limit their predictive power.

Non Linear Risk Models

Non Linear Risk models (such as those based on decision trees, random forests, boosted trees and neural networks) may perform better than scorecard models when feature interactions are complex. We used our lending dataset (See section 3) to benchmark a number of different models and found that XGBoost (Chen and Guestrin 2016) outperformed all others by 7% maximum separation in cumulative distribution between loans with good and bad outcomes. This performance measure is known as the K-S statistic (Kolmogorov 1933; Smirnov 1948). Our benchmark results are summarized in Table 2. Models were built on 75% of the available data and their K-S performance was measured on the remaining held back 25%.

Surprisingly Neural Network risk model performed relatively poorly on our loan outcomes dataset, however this confirms others' findings (Guegan, Addo, and Hassani 2018) and might be due to a large number of features compared to the limited number of training examples (Hughes 1968). Also notable is that despite its relative simplicity Scorecard ranks second in performance. Based on these benchmark results we focus further discussion to just Scorecard and XGBoost risk models.

4 Explaining Risk Model Results

Adverse Action (AA) Codes

In the USA the Fair Credit Reporting Act (FCRA 1970) requires lenders to explain all unfavourable credit decisions. This is commonly done using Adverse Action (AA) codes. About 100 different codes are used so just a small sample is shown in Table 3. To comply with this requirement we associate each AA code with a group of related risk model features. As part of this association we indicate whether high values of a given feature increase or decrease the credit risk. This is done to ensure that features in each group are semantically related and that each reason code is assigned a mutually exclusive set of features. Table 4 shows what this association looks like.

Table 3: Adverse Action (AA) Reason Codes

AA#	Reason
1	Amount owed on accounts is too high
2	Level of delinquency on accounts
3	Too few bank revolving accounts
...	...
99	Lack of recent account information

Table 4: AA Codes Assigned to Features

AA#	Feature#	Rejection Reason
1	12	High feature 12 value lowers risk
1	5	High feature 5 value lowers risk
1	10	High feature 10 value lowers risk
2	9	High feature 9 value raises risk
...	...	...
13	6	High feature 6 value raises risk

Explaining Scorecard Model Decisions

To explain why a borrower was rejected due to their scorecard risk score being too high, all one needs to do is identify the feature value bins with the largest contribution to that borrower's scorecard risk score. For example if the feature

bin *Average bank balance is less than $200* has the greatest impact on a scorecard risk score the adverse action AA explanation could be: *Your average bank balance is too low.*

Risk Model Monotonicity

Construction of Scorecard risk models that give acceptable explanations requires that as long as a feature's value continues to move in the same direction (either increasing or decreasing) the resulting risk score must not change direction (it also continues to increase or decrease). For example if a feature like "increased borrower income" reduces loan risk there should be no point beyond which greater income raises loan risk. Only features that satisfy this monotonicity requirement can be used when building scorecard risk models.

Explaining XGBoost Model Decisions

As shown above, when a loan application is rejected using a Scorecard it is relatively straightforward to explain the reason for rejection. However when relationships are nonlinear and feature interactions are complex and methods like XGBoost are used, the generation of credit decision explanations requires a more sophisticated approach. We know of two that are suitable:

1. LIME generates random neighborhood samples to weigh features according to the distance from the record in question (Tulio Ribeiro, Singh, and Guestrin 2016).
2. SHAP calculates feature attribution using Shapley values (Lundberg and Lee 2017).

Since SHAP has better consistency with human intuition (Lundberg and Lee 2017), this was the technique we selected.

To explain the risk score for a given loan application using SHAP we compute the Shapley value (using SHAP package (Lundberg and Lee 2017)) for each feature of that loan application. The feature with the largest Shapley value is then mapped to the AA reason code using table 4.

Monotonicity with XGBoost

The monotonicity requirement that ensures credit decisions have acceptable explanations (as described in section 4) was satisfied using a feature in XGBoost (DMLC/xgboost 2016) which forces predictions to monotonically increase or decrease with respect to each feature when other features are unchanged. For a tree based model, the right child's value is constrained to be higher than the left child's value for each split of a particular feature. Without this constraint, the tree algorithm ignores this feature and finds another feature to split.

To measure the effect with and without monotonic constraints, we used repeated 5-fold cross validation. Figure 1 shows that models with monotonic constraints improve K-S performance by 8% on average. As part of this evaluation we also computed ROC-AUC and as shown in figure 2 monotonic constraints improve ROC-AUC performance by about 3%.

We suspect monotonic constraints improve performance because they reduce the influence of noise much like regularization constraints do for Neural Networks (Hinton et al. 2012; Kukacka, Golkov, and Cremers 2017). Some noise in our feature dataset is introduced during transaction download and some during transaction understanding (see section 3), most of it however we believe exists due to natural variation between loan applications.

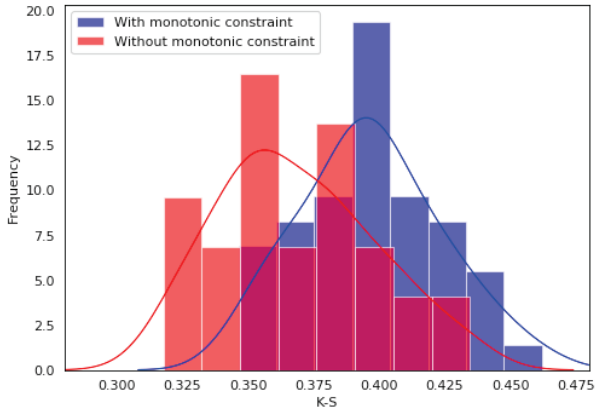

Figure 1: Comparison of K-S between XGBoost with and without monotonic constrains.

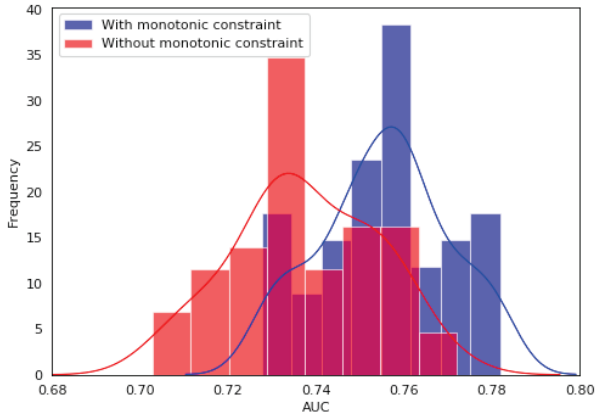

Figure 2: Comparison of AUC between XGBoost with and without monotonic constrains.

Monotonic SHAP Pitfall

When using SHAP with monotonic constraints one has to pay close attention to identified relationships between features and risk score. It is important to verify the observed direction of impact is reasonable because features with a highly unbalanced distribution over the bins can cause problems. For example when missing features are encoded, using an arbitrary feature value like zero, the resulting distribution can be skewed towards the zero value bin. Now if the real relationship between a feature and the outcome is weak that arbitrary decision to use zero for missing features may cause

a reverse interpretation by the model of the feature/outcome dependence. Situations like this lead to generation of incorrect explanations of credit decisions and thus must be prevented.

Figures 3 and 4 illustrate this pitfall using two diagrams. In the top diagram SHAP values are plotted in red and blue horizontally with one plot per feature stacked from top to bottom, most important to least important. Red denotes large feature values (e.g. average account balance is a large number). Blue denotes small feature values (e.g. average account balance is a small number). A gray circle selects the plot of the feature at the top which by its high position we know is the most important. The bottom diagram shows the skewed 84.7% popularity of this feature's 0–0.007 bin and its negative WoE, yet large values of this one feature work to lower risk for all feature values above 0. Since this feature (when present) is very predictive we can continue to use it as long as the missing information situation (which happens 84.7% of the time) is handled properly – for example by creating a separate dummy feature for it.

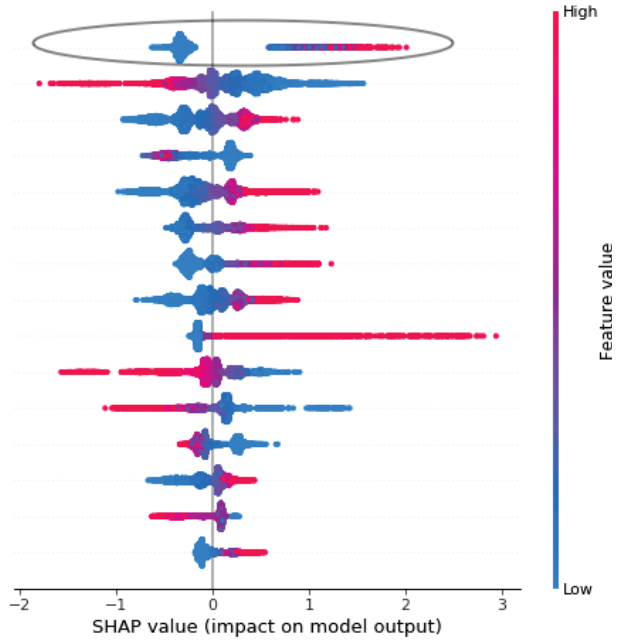

Figure 3: SHAP global summary plot.

Scorecard vs. XGBoost Explanations

Figure 5 shows how often Scorecard and XGBoost risk model decision explanations agree about AA reason codes at different risk score levels. For risk scores falling in the top 10% , the models agree ˜52% of the time. Agreement falls as risk scores decrease, yet even in the lowest 10% of risk scores the two models still agree 42% of the time.

The 42% - 52% agreement is reasonable because the decision boundary in XBGoost is more complex than scorecard, and feature attribution in the two models is also different. We suspect that disagreement occurs in situations where no single reason dominates. For example the drop in agreement

Interval scaled level	% Total	WOE
0 -< 0.007	84.7%	-0.375
0.007 -< 518.8	0.6%	2.617
518.8 -< 1732.0	1.2%	2.035
1732.0 -< 6070.6	2.7%	1.561
6070.6 -< 10103.1	1.7%	1.383
10103.1 -< 17505.6	2.0%	0.981
17505.6 -< 89016.7	4.7%	0.647
89016.7 - Inf	2.5%	0.578

Figure 4: WoE pattern of the feature circled in Figure 3.

when moving from high to low risk scores makes intuitive sense since riskier groups tend to have the same feature attribution regardless of the model – the more risky the borrower the easier it is to identify features that are indicative of high risk and to provide matching explanations.

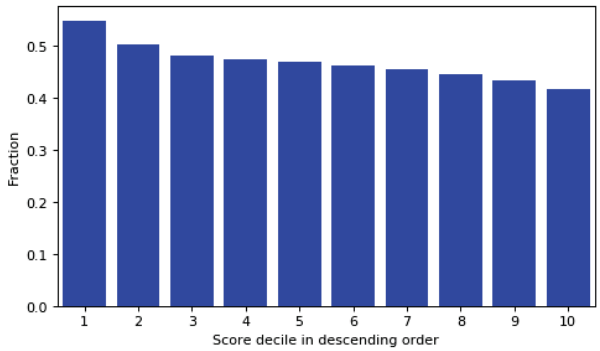

Figure 5: Fraction of loans that would have been rejected with the same reason from scorecard and XGBoost model. X-axis shows descending risk score decile.

5 Deployment and Impact

The machine learning XGBoost risk model and SHAP explanations here described have been deployed by QuickBooks Capital to evaluate incoming loan applications since July 2019.

So far the main benefits include: (1) increased transparency into automated loan decisions (2) ability to offer loans to wider range of customers

Over time, as the outcomes of loans issued so far become known, we expect machine learning risk models to cut QuickBooks Capital loan default rates by 20%. Figure 6 shows that with the scores predicted from our XGBoost model, rejecting top 20% loan applications with highest risk scores will result in a decrease in loan bad outcome rate by 42% (from 8.5% down to 4.9%). And with some investment we expect the number of loan applications that can be handled entirely by our machine learning models to grow further. Loan application review is a major business cost thus effective automation has a large economic payoff.

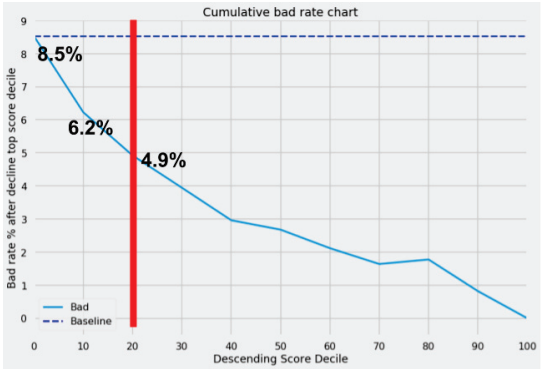

Figure 6: Cumulative bad outcome rate vs. descending score percentile. X-axis is on descending score percentile.

6 Conclusions

In this paper, we have shared how small business loan decisions can be made using machine learning risk models applied to financial transactions and how decision explanations can be generated to comply with lending regulations which is essential for production deployment.

We benchmarked risk models constructed using scorecard, Random Forest, Neural Networks, GBDT and XGBoost. Notably risk models designed using XGBoost achieved K-S 7% better than anything else on our dataset of tens of thousands of small business loans issued over the last two years.

To comply with lending industry regulations that require lenders to provide understandable and specific reasons for credit decisions, we developed a technique that combines the ideas of WoE and Shapley values for local model-agnostic explanations with monotonic constraints in XGBoost. Using this technique we empirically evaluated the AA reason codes generated from risk models using scorecard and XGBoost, showing how their explanations differ across various levels of risk score. The observed differences require further research including how to automate the evaluation of tens of thousands of generated explanations.

References

Akerlof, G. A. 1970. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics* 84:488–500.

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. *CoRR* abs/1603.02754.

DMLC/xgboost. 2016. [new feature] monotonic constraints in tree construction. https://github.com/dmlc/xgboost/issues/1514.

Duffie, D. K. J. S. 2003. *Credit Risk: Pricing, Measurement, and Management.* Princeton University Press.

FCRA. 1970. Fair credit reporting act. https://en.wikipedia.org/wiki/Fair_Credit_Reporting_Act.

FTC. 2013. In FTC study, five percent of consumers had errors on their credit reports that could result in less favorable terms for loans. https://www.ftc.gov/news-events/press-releases/2013/02/ftc-study-five-percent-consumers-had-errors-their-credit-reports.

FTC. 2015. FTC issues follow-up study on credit report accuracy. https://www.ftc.gov/news-events/press-releases/2015/01/ftc-issues-follow-study-credit-report-accuracy.

Guegan, D.; Addo, P. M.; and Hassani, B. 2018. Credit Risk Analysis Using Machine and Deep Learning Models. *Risks* 6(2):38.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580.

Hughes, G. 1968. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory* 14(1):55–63.

Intuit. 2019. Intuit quickbooks capital. https://quickbooks.intuit.com/capital/.

Kolmogorov, A. 1933. Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari.* 4:83–91.

Kukacka, J.; Golkov, V.; and Cremers, D. 2017. Regularization for deep learning: A taxonomy. *CoRR* abs/1710.10686.

Lesner, C.; Ran, A.; Rukonic, M.; and Wang, W. 2019. Large scale personalized categorization of financial transactions. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01):9365–9372.

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

Perlis, J. H. 2011. Scorecard modeling in xeno. *INFOCENTRICITY WHITE PAPER*.

Smirnov, N. 1948. Table for estimating the goodness of fit of empirical distributions. *Ann. Math. Statist.* 19(2):279–281.

Tulio Ribeiro, M.; Singh, S.; and Guestrin, C. 2016. Model-Agnostic Interpretability of Machine Learning. *arXiv e-prints* arXiv:1606.05386.

XENO. 2008. Automated binning in xeno: The break and heal algorithm. *INFOCENTRICITY WHITE PAPER*.