Data Augmentation Methods for Reject Inference in Credit Risk Models

Jingxian Liao[1,3], **Wei Wang**[1], **Jason Xue**[1], **Anthony Lei**[2]
[1]Intuit AI, Intuit, Inc.
[2]QuickBooks Capital, Intuit, Inc.
[3]Department of Computer Science, University of California Davis
{jingxian_liao, wei_wang3, jason_xue, anthony_lei}@intuit.com

Abstract

A significant challenge in credit risk models for underwriting is data representativeness. When credit scoring models are built using only applicants who have been accepted for credit which is the common strategy in the industry, such non-random sampling mainly influenced by credit policy makers and previous loan performances may introduce sampling bias to the estimated credit models and accordingly influence the models' prediction of default on loan payment when screening applications from all borrowers. In this paper, we proposed two data augmentation methods that aim to identify and pseudo-label parts of the declined loan applications based on the confidence level of the estimated labels to mitigate sampling bias in the training data. Besides prevalent model performance metrics, we also reported loan application approval rates at various loan default rate intervals from the business perspective. Our proposed methods were compared to the original supervised model and the traditional reject inference method using fuzzy augmentation. The results showed that self-training model with calibrated probability as data augmentation selection criteria improved the ability of credit score to differentiate good/bad loan applications and, more importantly, increased loan approval rate by 2.6% while keeping similar default rate comparing to the KGB model. The results demonstrate practical implications on how future underwriting model development process should follow.

1 Introduction

Credit scoring models are tools that financial institutions design to guide lending decisions for businesses or individuals. The model predicts the probability of default, i.e., applicants' probability of not repaying their debts, from collected financial information during the application stage. It is a binary classification model that separate bad borrowers from good ones. Traditional credit scoring models are trained with only a part of loan applicants that are approved by institutes, since repayment performances only exist for funded loans. Accepted applicants are already screened by the risk scoring models and manual checks during the underwriting process. In comparison, the entire application population includes rejected applicants whose actual repayments are unknown and potential applicants who never apply. Therefore, from the perspective of data sampling, the training samples from accepted applicants are biased from the through-the-door population at the time of credit underwriting. Though it is hard to consider potential applicants since no financial information provided, this paper proposes approaches to address the sampling bias issue by inferring rejected applicants and augmenting representitiveness of training samples. This technique in lending domain is referred as reject inference (Siddiqi 2003; Montrichard 2007).

Reject inference (RI) are techniques that combine accepted applicants with their repayment and rejected applicants with estimated performance into inferred data sets and generate reject inference scoring models. Previous studies have introduced different strategies to estimate the rejected applications. One common practice is to obtain external loan performance data from credit bureaus for rejected applicants, though it is relatively costly. Another well-known strategy, fuzzy augmentation, assigns labels to the rejects based on the scoring model trained by accepted applicants with adjustment made on sample weights, and then retrains the scoring model (Montrichard 2007). Recent model-based techniques have proposed new models to assign labels to the rejects from the angle of semi-supervised learning, such as semi-supervised SVMs, self-learning, and K-prototype clustering (Li et al. 2017, 2020; Kozodoi et al. 2019). However, some methods like clustering only have good performances on low dimensional data according to theoretical findings (Bellman 2015). Moreover, datasets used in these experiments are usually oversimplified in low dimension and relatively small in size as well (Li et al. 2017, 2020).

The contribution of this paper is two-fold. First, we propose two novel reject inference techniques to estimate the performance of applicants whose actual default statuses are unknown. One method includes a self-training method with variation on the choice of most confident of unlabeled predictions that are added to the training set. We introduce probability calibration and Trust Score as confidence models to select the most confident predictions (Triguero, García, and Herrera 2015; Jiang et al. 2018; Niculescu-Mizil and Caruana 2005). Another method uses the idea of data programming, and initializes multiple weak classifiers to jointly label the data along with Snorkel generative models (Ratner et al. 2017). Second, we introduce a new business-related measure (denoted as *approval rate*) to evaluate the performance of reject inference methods. By controlling potential default

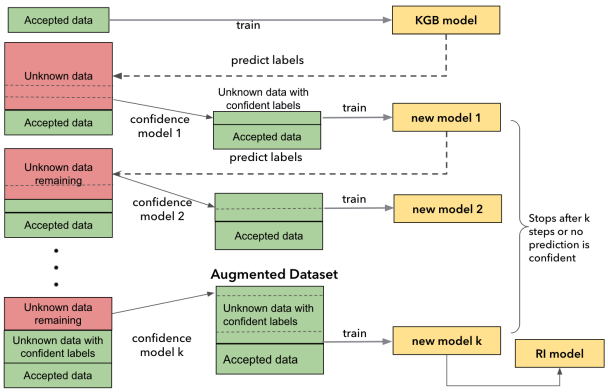

Figure 1: Flowchart of self-training method with confidence model for reject inference.

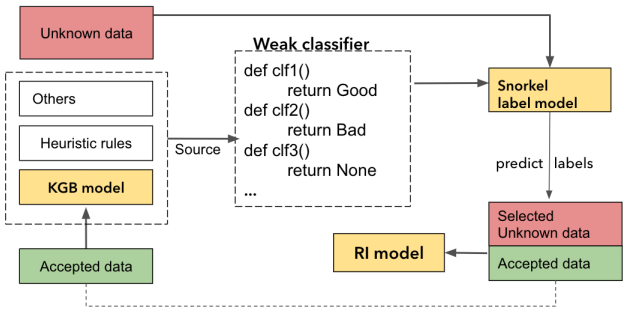

Figure 2: Flowchart of weak supervision for reject inference.

rate, estimated approval rate measures the percentage of applicants that can be approved as an estimated business Key Performance Indicator (KPI). This measure considers both accepted label accuracy and also the application population. It provides us a unique metric for domain-specific evaluation.

2 Methods

In this section, we present two reject inference methods. Self-training method combines a self-training algorithm and a pseudo-label confidence model. And we introduce another method using multiple weak classifiers and Snorkel (Ratner et al. 2017) to predict the default status of loan applications whose performances are unlabeled.

Consider a set of n loan applications $x_1, x_2, \cdots, x_n \in R^k$ where k is the number of features. This set includes m accepted applications $x_1, x_2, \cdots, x_m \in X_a$ with corresponding labels $y_1, y_2, \cdots, y_m \in \{\text{Good, Bad}\}$ and consists of $x_{m+1}, \cdots, x_n \in X_u$ whose labels are unknown. The credit scoring model trained with X_a only is denoted as Known Good/Bad (KGB) model. To mitigate sampling bias, reject inference techniques assign labels to unlabeled applications, and combine accepted data and pseudo-labeled data into inferred data sets to represent the whole application population and update credit scoring models. The scoring model with inferred data as training set is denoted as reject inference model (RI model).

2.1 Self-training with confidence model

Figure 1 is an overview of our proposed self-training method pipeline. It starts with training an initial model (also the KGB model in the first iteration) on the accepted data X_a and uses it to predict all the unlabeled data. Then, a confidence model is introduced to filter the most confident predictions whose labels are either good or bad in unlabeled data X_u with a fine-tuned threshold. The selected unlabeled data are labeled in accordance with the predictions, and the training set is augmented with new labeled data, denoted as X_a^1. Then RI model is retrained with labeled data X_a^1. This process is repeated, and RI model and confidence model update every iteration along with labeled data X_a^j. It will stop until no confident applications are identified from confidence model or reaching a pre-defined number of rounds as stopping criteria.

We applied two confidence models to accommodate the attributes of different algorithms for reject inference: Trust Score (Jiang et al. 2018) and probability calibration (Niculescu-Mizil and Caruana 2005). The traditional self-training selects confident predictions whose prediction probabilities p satisfy $p > \alpha$ or $p < 1 - \alpha$, where a is a probability threshold. However, many popular algorithms, such as Naive Bayes, SVM and Random forest, tend to yield a characteristic sigmoid-shaped distortion in predicted probabilities. Our probability calibration confidence model adds isotonic probability calibration (Niculescu-Mizil and Caruana 2005) and uses calibrated probabilities to filter confident predictions. Trust Score model, on the other hand, provides prediction accuracy from the nearest neighboring approach (Jiang et al. 2018). It pre-selects a high-density data range for each class. Then a trust score is defined as follows to evaluate the prediction: for a predicted test label, the trust score is the ratio between the distance from the test label to the nearest class different from the predicted label class and the distance to the predicted label class within the data range. In this work, the score is based on 5% of the instances from each class. A high score implies high prediction accuracy since the predicted case is close to labeled data with the same label class.

From empirical results, probability calibration shows a significant improvement in maximum margin methods, such as XGBoost, the one used in our experiment. And Trust Score works as an alternative of algorithms' own confidence scores from initial feature space and validation set.

2.2 Weak supervision with data programming

In recent years, data programming has been widely discussed and developed to generate labeled training sets. One such successful example is Snorkel project (Ratner et al. 2017). Taking advantage of Snorkel flow and generative models, we initialize rule-based weak classifiers and estimate the labels of unknown data. Figure 2 is the pipeline of our weak supervision method.

We collected common heuristic rules to detect potential default applicants from interviews with underwriting and credit policy experts. Furthermore, based on weight of evidence of variables from KGB model, we selected features with high information value and monotonic pattern of default rate (Wang et al. 2020). For each selected feature, we identified a feature value threshold when the bad loan rate climbs and labeled applications in high bad loan rate bins as bad. For example, business tenure is a popular index to measure the stability of applicants. We created a weak classifier that applicants whose business history is less than one year will be labeled as bad. It is worth noting that these rules are based on limited features of labeled accepted data and expert experiences, and their accuracy are weak compared to the KGB model. The accuracy and correlations of weak classifiers are learned with Snorkel generative model. And Snorkel reports final probabilistic labels for part of unlabeled data that it can predict. The reject inference model is trained with accepted data and data labeled by weak supervision with confidence. For the details of Snorkel system, please check their papers and website (Snorkel 2020).

3 Experiments

3.1 Data

Our research was carried out using loan data from Intuit lending business which has offered business loans to its small business accounting software users since 2017. These loans are repaid weekly, bi-weekly or monthly over a period of six, nine, or twelve months.

Between 2017 and 2020, hundreds of thousands of loan applications have been submitted, and tens of thousands of loans have been issued. Over a quarter of issued loans have reached maturity. Those issued loans still in the process of repayment and those we declined previously, representing vast majority of all loan application population, are not included in the credit risk models due to lack of loan performance history. 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, et cetera.

For this research, Intuit provided us a random and anonymous sample of loan applications with a size of around 20,000 to ensure the repsentativeness of the population.

We will not discuss here hundreds of features that are extracted from bank transactions and how users' bureau data was processed through our internal data pipeline, apart for noting that this kind of data is intrinsically noisy. Some of the noises are introduced by information representation and transmission of bank data, inaccurate recording of business bureau data, and significant variability due to the differences in the nature of business among loan applicants.

After feature engineering, the entire dataset was split into a training set and a test set according to the loan application date. For a better evaluation on more representative test applications, we augmented the labeled test data by assigning labels to part of rejected applications with external loan history from bureaus as ground truth. So in the test set, the labeled subset is a combination of internal loans with their performances and rejected applications with estimated labels from their bureau credit history. We set stringent matching criteria in order to maximally eliminate false positive matches, such as requiring a relatively narrow matching window, matching credit accounts whose types and days past due are similar to our loan population only. Eventually, about 13% of data in the test set are labeled by the bureau credit account data.

Table 1: Data matching between Bureau and Intuit*

	Good	Bad
Good	91%	$\sim 0\%$
Bad	3%	6%

*Bureau data are shown in rows and Intuit data are shown in columns. Percentages are shown in as % of total number of loans with known outcomes.

To further validate the quality of data matching between bureau credit accounts and loan data, we calculated the confusion matrix between bureau data and existing labeled loan data. Results show that the matching quality is satisfactory as shown in Table 1 - about 97% of the data were matched correctly.

3.2 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* (labeled as Good) 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:

- *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 charge off).
- *Unknown Outcome* – loans are those in good standing which will mature in more than 30 days, approved but not taken by applicants or declined due to applicants' creditworthiness.

3.3 Choice of credit risk model

Our previous work (Wang et al. 2020) found that gradient boosted tree algorithm (XGBoost) provided the best model performance among several candidate algorithms for credit risk scoring, and simultaneously monotonic constraints (DMLC/xgboost 2016) on inputs can provide explanations on the predicted score in conjunction with Shapley values (Lundberg and Lee 2017). Best hyperparameters used in XGBoost is determined by Amazon Sagemaker XGBoost hyperparameter tunning using Bayesian search (Amazon-Sagemaker 2020). For comparison purposes, we will select

XGBoost as the choice of credit risk models for all the methods throughout the experiment.

3.4 Benchmark models

We adopted two benchmark models in this experiment: a Known Good/Bad model that does not have any sampling correction, and a fuzzy argumentation method as representative of current reject inference techniques. The Known Good/bad XGBoost model is trained with only accepted and funded applicants.

Fuzzy argumentation involves assigning labels to unlabeled data based on the KGB model and retrain to get RI model (Montrichard 2007). It assigns unknown data as being partial Good and partial Bad by labels and weights. Every application in X_u is duplicated as two records with two labels y: (1) y_1 = Good with weight $p(\text{Good})$; and (2) y_2 = Bad with weight $p(\text{Bad})$. The weights $p(\text{Good})$ and $p(\text{Bad})$ are predicted probabilities based on KGB model. The sum of two weights is equal to 1. And accepted applications are also weighted by 1. Then the RI model is constructed on weighted data.

3.5 Evaluation metrics

Both the benchmarks and our new methods are tested on the same test set to ensure a fair comparison.

AUC-ROC (AUC) and K-S are used to compare the performances. Besides commonly used AUC for binary classification models, K-S is a metric between 0 and 1 that measures the maximum separation between the cumulative distribution of the two classes (Bradley 1997; Smirnov 1948; Kolmogorov 1933). Note that both metrics do not depend on the selection of classification thresholds, making them attractive as evaluation metric in the context of credit risk domain.

Approval rate Besides domain-independent evaluation metrics, we introduce a novel evaluation metric from a business KPI perspective, **approval rate**. In general, when more applications are approved, more loans with bad outcome will be introduced. For the maximum profit and risk control, lending institutions prefer to extend their customer population while keeping controllable potential loan default. Therefore, for a given risk score threshold $t(p)$ where p is the pre-defined bad rate, the approval rate is calculated as

$$\text{Approval rate} = \frac{\text{number of applications with score} \leq t(p)}{\text{nubmer of applications}}$$

$$\text{bad rate } p = \frac{\text{number of loans with label Bad}}{\text{number of loans with labels in } \{\text{Good, Bad}\}}$$

Note that calculation of approval rate is based on both labeled test set and the unlabeled test data. Refer to Figure 3 for an illustration of risk score distribution and how it is related to approval rate calculation. Given the test data, predicted risk score, and a specific risk score threshold, the numerator of approval rate is the size of data whose risk score is lower than the threshold. The corresponding bad rate is the ratio of bad loans in the combined labeled loans in the augmented data set.

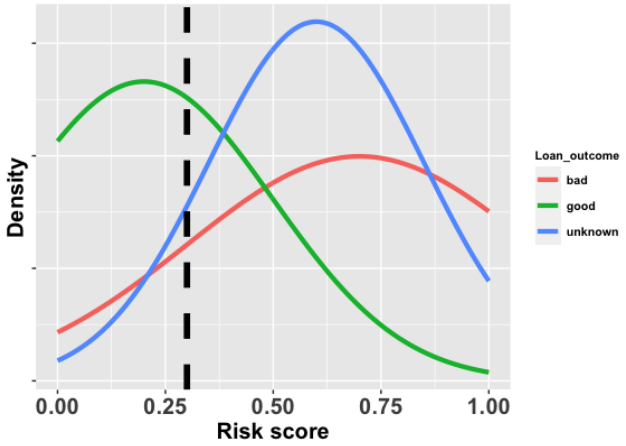

Figure 3: Illustration of approval rate calculation. The dashed line is risk score approval threshold $t(p)$: applications with lower risk scores are approved.

To take unlabeled data into consideration, the bad rate thresholds need to be set lower than normal business bad loan rates that financial institutes could take. Therefore, we report multiple approval rate estimates on different low bad rates, including 2.5%, 3%, and 3.5% in the results.

4 Results and Discussion

4.1 Experiment results

Table 2 summarizes the performances of our reject inference methods and benchmarks on the test set. The training size reports the final training size applied in each method compared to the original accepted data size. Fuzzy augmentation uses the full unknown data for the training, while all the other methods selectively include part of unlabeled dataset into their training set. The best of each metric is highlighted. For the two benchmarks, fuzzy augmentation does not improve the performance compared to KGB model on most of the metrics. For self-training method, XGBoost algorithm works better with calibrated probability as confidence model than Trust Score. Self-training with probability calibration confidence model outperforms all other methods in terms of AUC, K-S statistics, and approval rate @2.5% bad rate. Compared to the KGB model, the approval rate increases from 52.9% to 54.3%, and K-S statistics improves from 0.367 to 0.381. In contrast, self-training with TrustScore only performs better than fuzzy augmentation on the approval rates. For the second method we proposed – weak supervision, the results are more mixed. The K-S statistics, AUC, and approval rate at low bad rate (2.5%) of weak supervision are the lowest among all methods, but approval rates at higher bad rates (3% and 3.5%) are the highest.

Performance gain are relatively modest, consistent with the prior literature (Hand and Henley 1993). Friedman's rank sum test reports that not all the methods perform the same ($p < 0.05$), but pairwise comparisons do not show significant differences between most of methods. However,

Table 2: Comparison of Performance of Reject Inference Methods and Benchmarks

Method*	Training size	AUC	K-S	Approval rate** @ 2.5% bad rate	Approval rate @ 3% bad rate	Approval rate @ 3.5% bad rate
KGB	100%	0.737	0.367	0.529	0.618	0.696
FA	238%	0.738	0.365	0.510	0.603	0.684
ST - TS	126%	0.732	0.356	0.514	0.617	0.704
ST - CP	115%	**0.740**	**0.381**	**0.543**	0.602	0.706
WS	167%	0.727	0.346	0.471	**0.624**	**0.714**

* KGB: Known Good/Bad model; FA: Fuzzy Augmentation; ST-TS: Self-training based on Trust Score; ST-CP: Self-training based on calibrated probability; and WS: Weak supervision.

** Approval rate: approval rate estimate based on labeled test dataset and unlabeled test set.

considering the large loan volume involved, it is still considered as a significant difference for business purposes.

4.2 Discussion

In self-training method, we observed that calibrated probability works better with XGBoost algorithm compared to Trust Score. The prediction uncertainty estimation is crucial for base learners in self-training methods. In our case, we applied two simple uncertainty measures based on previous findings about Gradient boosting machines: Isotonic regression tries to correct boosting models' prediction probabilities while Trust Score provides uncertainty information about the relative positions of the data points (Jiang et al. 2018; Niculescu-Mizil and Caruana 2005). The final training set sizes among different methods in Table 2 also implies the need for prediction accuracy measure and selection for inferred training set as discussed in other literature (Li et al. 2020). And the introduction of data that are far from decision boundaries of classifiers may not help with classification performance. As prediction uncertainty studies develop, one future work direction is to consider new uncertainty estimates to improve the self-training performance for RI models.

Weak supervision method shows mixed results comparing to other methods. We observed highest approval rate at higher bad rate. This is likely partially due to the fact that unlabeled data are in general more risky than selected labeled loans and weak supervision model covers much more unlabeled data when bad rate is set higher. Most of weak classifiers are devised towards precision in identifying bad applicants among relatively high credit risk applicants rather than covering all population. Therefore when the bad rates are higher and close to the bad rates threshold business institutes use, weak supervision is more likely to resemble the pattern of the features/labels. In order for this method to perform well consistently on the low bad rate scenarios, one future work is to create more diverse labeling functions to cover, for example, precisely identifying good cases with various degree of coverage. The other reason we think weak supervision has high applicability in lending is its ability to even "cold-start" training credit risk models without having the access to any labels, thanks to the generative/discriminative nature of Snorkel labelling models.

We proposed a domain-specific and business related KPI *Approval rate* to evaluate reject inference models in addition to ROC-AUC and K-S which are domain independent. Approval rate is closely related to hypothesis testing framework - it controls type II error rate and reports the fraction under the null hypothesis (in our case, loans with good label). From the lending business perspective, it predicts the business revenue and potential customer population it can serve with default loss control. This measure could be extended and apply to other financial domain such as insurance and consumer lending which have strict and specific prediction error limitation as domain-specific evaluation metric.

5 Conclusion and Future Work

In this paper, we have shared application of several data augmentation methods which can help alleviating sampling bias in credit risk models and how we should evaluate these methods based not only on the traditional model performance metric but also on business KPI related metric, i.e., application approval rate.

We had empirically shown that including data selectively from the loan population with unknown outcome can effectively improve credit risk models, in terms of their performance on the general population. The traditional reject inference method (Fuzzy Augmentation) tends to yield worse performance across all the evaluated metrics. Varying degree of effectiveness of the proposed methods on improving the approval rate seems to depend on the level of sample bad rate. This warrants further research to understand the difference in the augmentation methods and directs us to think about combining them into a hybrid method.

References

Amazon-Sagemaker. 2020. How Hyperparameter Tuning Works. URL https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-how-it-works.

Bellman, R. E. 2015. *Adaptive control processes: a guided tour*, volume 2045. Princeton university press.

Bradley, A. P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30(7): 1145–1159.

DMLC/xgboost. 2016. [New Feature] Monotonic Constraints in Tree Construction. https://github.com/dmlc/xgboost/issues/1514.

Hand, D. J.; and Henley, W. E. 1993. Can reject inference ever work? *IMA Journal of Management Mathematics* 5(1): 45–55.

Jiang, H.; Kim, B.; Guan, M.; and Gupta, M. 2018. To trust or not to trust a classifier. In *Advances in neural information processing systems*, 5541–5552.

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

Kozodoi, N.; Katsas, P.; Lessmann, S.; Moreira-Matias, L.; and Papakonstantinou, K. 2019. Shallow Self-Learning for Reject Inference in Credit Scoring. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 516–532. Springer.

Li, Z.; Hu, X.; Li, K.; Zhou, F.; and Shen, F. 2020. Inferring the outcomes of rejected loans: an application of semisupervised clustering. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* .

Li, Z.; Tian, Y.; Li, K.; Zhou, F.; and Yang, W. 2017. Reject inference in credit scoring using semi-supervised support vector machines. *Expert Systems with Applications* 74: 105–114.

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*, 4765–4774. Curran Associates, Inc. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Montrichard, D. 2007. Reject inference methodologies in credit risk modeling. *SESUG 2008* .

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, 625–632.

Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, 269. NIH Public Access.

Siddiqi, N. 2003. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Princeton University Press. ISBN ISBN 0-691-09046-7.

Smirnov, N. 1948. Table for Estimating the Goodness of Fit of Empirical Distributions. *Ann. Math. Statist.* 19(2): 279–281. doi:10.1214/aoms/1177730256. URL https://doi.org/10.1214/aoms/1177730256.

Snorkel. 2020. Snorkel: Programmatically Build and Manage Training Data. https://www.snorkel.org/.

Triguero, I.; García, S.; and Herrera, F. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems* 42(2): 245–284.

Wang, W.; Lesner, C.; Ran, A.; Rukonic, M.; Xue, J.; and Shiu, E. 2020. Using Small Business Banking Data for Explainable Credit Risk Scoring. In *The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, New York, NY, USA, February 7-12, 2020*, 13396–13401. AAAI Press. URL https://aaai.org/ojs/index.php/AAAI/article/view/7055.