



A Monte Carlo simulation framework for reject inference

Billie Anderson, Mark A. Newman, Philip A. Grim II & J. Michael Hardin

To cite this article: Billie Anderson, Mark A. Newman, Philip A. Grim II & J. Michael Hardin (2023) A Monte Carlo simulation framework for reject inference, Journal of the Operational Research Society, 74:4, 1133-1149, DOI: [10.1080/01605682.2022.2057819](https://doi.org/10.1080/01605682.2022.2057819)

To link to this article: <https://doi.org/10.1080/01605682.2022.2057819>



Published online: 09 Apr 2022.



Submit your article to this journal [↗](#)



Article views: 184



View related articles [↗](#)







View Crossmark data [↗](#)

RESEARCH ARTICLE



A Monte Carlo simulation framework for reject inference

Billie Anderson^a , Mark A. Newman^b , Philip A. Grim II^b  and J. Michael Hardin^c 

^aUniversity of Missouri Kansas City (UMKC), Kansas City, MO, USA; ^bHarrisburg University of Science and Technology, Harrisburg, PA, USA; ^cSamford University, Birmingham, AL, USA

ABSTRACT

Credit scoring is the process of determining whether applicants should be granted a financial loan. When a financial institution decides to create a credit scoring model for all applicants, the institution only has the known good/bad loan outcomes for accepted applicants. This causes inherent bias in the model. We address a gap in the reject inference literature by developing a methodology to simulate rejected applicants. A methodology to illustrate how to simulate rejected applicants must be developed so that the reject inference techniques can be studied and appropriate reject inference techniques can be selected. This study uses a peer-to-peer financial loan information from accepted and rejected financial loan applicants to perform Monte Carlo simulation of rejected applicants. Using simulated data, the researchers compare the performance of three widely used reject inference techniques.

ARTICLE HISTORY

Received 5 April 2021
Accepted 19 March 2022

KEYWORDS

Credit scoring; reject inference; Monte Carlo simulation

1. Introduction

Credit scoring is one of the most successful applications of quantitative analysis in business and is a key analytical decision-making tool. Credit scoring is the process of determining how likely an applicant can repay a loan. Many economic facets utilize credit scoring, such as banks, credit card companies, mortgage lenders, insurance carriers, and retailers. Lenders build credit scoring models estimating how likely an applicant will repay the loan (Lessman et al., 2015). Applicants with a high probability of repayment based on the model are deemed to have good credit risks and will be awarded a loan. In contrast, those determined to have low probability of repayment are deemed to be bad credit risks and will not be awarded a loan. Developing credit scoring models are becoming more dependent on statistical and machine-learning models (Dastile et al., 2020).

The financial institution can observe the performance of the accepted applicants, and the outcome will be confirmation if a loan is either good or bad. The outcomes of rejected applicants were not observed. Because of changing economic environments, competition, and new credit products being constantly introduced into the financial marketplace, financial institutions are constantly updating their credit scoring models (Baesens, 2003; Hand, 2001; Nikolaidis et al., 2017). The updated credit scoring model must include accepted and rejected applicants so the model will

be representative of the applicant population of the financial institution. Reject inference is the process of inferring good or bad loan status of rejected applicants so the updated credit scoring model will be representative, including accepted and rejected applicants, when applied to a new population of credit applicants. Most accepted applicants do not resemble the general population of credit applicants. This potential bias is well known and understood in the credit industry (Bucker et al., 2013; Crook & Banasik, 2004; Shen et al., 2020). To avoid bias in the updated credit scoring model, rejected applicants with their inferred good or bad loan status were incorporated into the model-building process.

In recent years, peer-to-peer (P2P) lending has become a popular financial lending model. LendingClub, founded in 2007, is the largest peer-to-peer online financial loan service facilitating lending between individuals (Vallee & Zeng, 2019). The LendingClub was established during the 2008 financial crisis, when the public was losing trust in the traditional financial system. LendingClub was revolutionary in the sense that it allowed individuals to obtain credit by disintermediating banks and other financial lenders. LendingClub provides available information regarding the applicants that have been accepted and rejected for financial loans.

LendingClub is known to have a higher rejection rate of applicants than other financial institutions

because the loan granting process is completely performed in an online lending market (Iyer et al., 2016; Tian et al., 2018). Part of the reason for the high rejection rate is that the LendingClub can only obtain information about potential applicants in an online lending market. Therefore, the LendingClub sets higher thresholds for granting loans compared to traditional lending institutions. LendingClub has publicly available data from 2007 to the first quarter of 2019. The rejection rate for this period was approximately 92%. This high rejection rate provided authors with an opportunity to study the reject inference problem.

1.1. Contribution

In practice, analysts use various ad hoc reject inference methods and never really have a full understanding of their impact on the final model. One of the seminal works in reject inference noted a lack of understanding of what can be achieved using rejected applicants' data (Hand & Henley, 1994). The novel contribution of this work is presenting a Monte Carlo simulation framework to generate different scenarios of the distribution of inferred good and bad loan outcomes for a set of rejected credit applicants using the LendingClub dataset. By simulating rejected applicants, this research provides a framework that can be used to gain insight into which reject inference techniques would be appropriate for a given set of characteristics from a rejected applicant pool.

This study performs a Monte Carlo simulation using logistic regression as the underlying statistical model for performing reject inference techniques. Despite the rise of machine learning models for credit scoring and reject inference, the logistic regression model remains widely used among practitioners because of its ease of interpretation (Dastile et al., 2020). Many of the studies in the literature aimed at employing sophisticated predictive models such as neural networks, support vector machines, and random forests, which only provide a marginal gain in terms of accurately predicting good and bad credit risks over logistic regression (Oskarsdottir et al., 2019).

This study provides a general Monte Carlo simulation framework that can be utilized to more thoroughly understand the distribution of rejected applicants so the proper reject inference model may be selected.

2. Related work

One of the main claims often made is performing reject inference helps prevent sample bias. Eisenbeis (1978) found that using a credit scoring

model based on accepted applicants would frequently provide misrepresentative results. Other authors have also reported the necessity to incorporate the inferred good/bad loan status of rejected applicants into the redeveloped model to avoid bias (Crook & Banasik, 2004; Joannes, 1993; Verstraeten & Van den Poel, 2005). In reject inference, the major assumption is that the credit scoring model developed with both the accepted and rejected applicants would be superior as the model is more representative of the new applicants the financial institution would prefer to assess for a loan.

Many rejected inference techniques are a form of extrapolation (Crook et al., 2007; Hand & Henley, 1994). Extrapolation uses data from accepted applicants to build a model predicting the probability of loan default. Thereafter, the model built on the accepted applicants only is extrapolated to the rejected applicant's region. If the characteristics of bad loans from the accepted applicant pool are similar to those of bad loans from the rejected applicant pool, extrapolation is a valid reject inference technique (Hand & Henley, 1994).

From a statistical perspective, the reject inference problem is a missing variable problem. There are three types of missingness in the context of missing data: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR) (Feelders, 1999).

MCAR presumes that the missingness is due to randomness; that is, whether the applicant is accepted or rejected for credit is not based on the loan applicant's characteristic variables or repayment history. The decision to offer a loan to the applicant is simply due to random chance. MCAR is not a realistic method to presume for the reject inference problem since a financial institution would not remain profitable if MCAR is presumed to be the missing data mechanism.

MNAR is another missing data mechanism that can be presumed for the reject inference problem. MNAR presumes that the missing loan outcome is influenced by the loan characteristic variables and the missing loan outcome. Therefore, MNAR rests upon the assumption that the missing loan outcome is affected by the unobserved loan outcome. There have been several empirical studies that have investigated reject inference under the MNAR assumption. Chen and Astebro (2012) proposed a method to generate the missingness under a MNAR assumption using a bound and collapse Bayesian model that improved classification based on the empirical results. Bucker et al. (2013) studied the MNAR reject inference problem. These authors concluded, based on an empirical study, that when data is

available on the missing loan outcome for the rejected applicants, the credit scoring models can be improved. However, in practical cases, information on the missing loan outcome status for the rejected applicants is not known, as is the case for the Lending Club data used in this study. MNAR means that the financial institution has the right to change the credit-granting decision based on the overall impression of an applicant (Tian et al., 2018). MNAR would be realistic to presume if financial institution uses a completely manual decision-making process in the credit-granting decision. Otherwise, if a credit scoring system is in place, MNAR would require heavy overrides.

In this study, the data is presumed to MAR. MAR means that the missingness of the loan outcome status for the rejected applicants is due solely to the observed characteristic variables and some decision criterion. Unless a financial institution can develop a surrogate for the missing loan outcome status of the rejected applicants, MAR is a reasonable and practical assumption for the missingness for the reject inference problem. In practice, most rejected applicant loan applicants belong to the MAR category (Li et al., 2017; Tian et al., 2018). Feelders (1999) also noted that MAR is a reasonable assumption since most financial institutions use a formal statistical model for making the credit-rating decision.

Since this study presumes the data to be MAR, the presumption is that the reason for the missingness is due to the loan characteristics of the rejected applicants and a decision criterion that is imposed by the financial institutions. The Monte Carlo simulation performed in this study is used to select a reject inference technique when the loan characteristics of the rejected applicants have been modified to a certain degree under the MAR assumption. With MAR data, the missingness is related to the values of the characteristic variables observed. The Monte Carlo procedure randomly generates a distribution that mimics, as closely as possible, the observed distribution of the characteristic variables plus the designed perturbations in the shifts of the distributions of the characteristic variables. Thus, the Monte Carlo simulation preserves the MAR condition of the data.

There have been past Monte Carlo simulation studies performed using MAR data scenarios. Enders and Bandalos (2001) performed a Monte Carlo simulation that examined four different imputation techniques using MAR data for structural equation models. The Monte Carlo simulation studied how the different missing data mechanisms affected the bias of the parameter estimates, and model goodness of fit. Lai et al. (2017) performed a

Monte Carlo simulation for a logistic regression model when the missing outcome variable was MAR. The simulation compared the MAR scenario to the full data (no missing outcome) and examined the effects of selecting predictor variables under different varying sample sizes and number of predictors. The authors note that when MAR is the missing data mechanism, it is valid to assume that only a few of the characteristic variables contribute to why the outcome variable is missing. There are two characteristic variables in this study that are being simulated, and they are both related to the reason that the loan applicants were rejected.

Liu and Kost (2017) illustrate several applications of Monte Carlo simulation methods of handling missing data issues for longitudinal clinical trials under the presumption that the missing data is MAR. In particular, the authors examined a logistic regression model. The authors note that the random variation from the simulations can be a disadvantage for using Monte Carlo simulations. Enough replications need to be carried out to reduce random variation. In this study, 1000 replications were performed to guard against the random variation problem.

Most recently, researchers in the reject inference field have been applying more contemporary statistical and machine-learning techniques. Some of these contemporary techniques allow the avoidance of the extrapolation problem. Recently, an increase in the application of SVMs to reject inference because support vector machines have been shown to be successful in credit scoring applications (Bellotti & Crook, 2009; Goh & Lee, 2019; Teles et al., 2020). Particular, interest in applying semi-supervised techniques to the reject inference problem has increased.

Semi-supervised techniques combine supervised and unsupervised methods in the field of machine learning to build a model (van Engelen & Hoos, 2020). The LendingClub data is particularly applicable for semi-supervised techniques because semi-supervised techniques work best when a large amount of unlabeled data exists (Levatic et al., 2017). Semi-supervised techniques have been cited as being naturally suited to the reject inference problem (Liu et al., 2020).

Li et al. (2017) used the LendingClub data and applied semi-supervised support vector machine to illustrate the value of incorporating rejected applicants into a credit scoring model. These authors use LendingClub data to show that the semi-supervised support vector machine outperforms traditional logistic regression as a reject inference technique. Part of the improvement of using the support vector machine was that rejected applicants were

incorporated into the model building process, so the extrapolation problem was avoided.

Tian et al. (2018) used the LendingClub data and developed a kernel-free semi-supervised support vector machine application for reject inference. These authors attempt to solve the issue of not rejecting credit applicants that would have resulted in good loans. Ultimately, credit scoring models reject some applicants that would have turned out to be good credit risks. These applicants that should have been accepted for credit but rejected are known as outliers. The authors showed that using kernel-free support vector machine can improve the classification accuracy of the credit scoring model when these outliers can be identified.

Shen et al. (2020) developed a novel unsupervised method for performing reject inference in which the outcome of the rejected applicants does not have to be inferred. Instead, a self-taught learning (STL) mechanism is employed that is based on unsupervised techniques. The benefit from this method is that it can avoid the traditional extrapolation problem. That is, the authors developed a reject inference technique that does not have to assume that the rejected applicants have the same distributions as the accepted applicants.

Anderson (2019) used a Bayesian network to perform reject inference. The innovation in this work was a technique was applied that attempted to avoid the extrapolation problem that plagues the reject inference problem. Anderson (2019) applied a Bayesian network to perform reject inference and compared the results of the Bayesian network to three classic reject inference techniques that do require extrapolation. Since a Bayesian network avoids the extrapolation problem because there is no functional form that must be estimated on the accepted applicants and applied to the rejected applicants, an empirical study showed the Bayesian network outperformed three classic reject inference techniques that require extrapolation.

Xia et al. (2018) conducted a study that approached the reject inference problem using a semi-supervised approach. These authors combined a tree-based ensemble method and a contrastive pessimistic likelihood estimation (CPLE) framework. The authors applied their reject inference technique on two P2P lending datasets. The empirical results of the study showed that the authors semi-supervised reject inference technique outperformed the traditional extrapolation techniques on both P2P datasets.

Mancisidor et al. (2020) developed two novel Bayesian models for reject inference by combining Gaussian mixtures and auxiliary variables in a semi-supervised framework using deep learning neural

networks. The models used posterior distribution of the loan outcome to infer the unknown status of the rejected applications by enumerating the two possible outcomes of the rejected applicants exactly. Using the LendingClub data, the authors showed that these deep learning models performed better than the extrapolation methods.

Ehrhardt et al. (2021) argued that many of the more modern-day techniques such as semi-supervised methods that attempt to allow the rejected applicants data to participate simultaneously with the accepted applicants are based only on empirical studies and are not based on theoretical outcomes. The authors reexamine several of the classic reject inference techniques that are based on logistic regression and discuss the mathematical properties of each. The authors do perform an empirical study using real-world data from their financial institution. The authors conclude that none of the classical reject inference methods uniformly performs best.

The innovation of this research is that the Monte Carlo simulation framework, normally developed for evaluating reject inference models, is applied using any statistical model that can be used to perform reject inference, including the more contemporary machine learning models applied to reject inference in the recent years.

3. Materials and methods

3.1. Reject inference models

The three main reject inference techniques widely used among practitioners are based on logistic regression (Siddiqi, 2017). The logistic regression model is defined as

$$P(y = 1|x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}. \quad (1)$$

and

$$\begin{aligned} P(y = 0|x) &= 1 - P(y = +1|x) \\ &= \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)} \end{aligned} \quad (2)$$

where x is the characteristic variable vector, $P(y = 1|x)$ is the probability of classifying applicant x as a bad loan, $P(y = 0|x)$ is the probability of classifying applicant x as a good loan, and β, β^T are model parameters estimated using maximum likelihood estimation.

3.1.1. Simple augmentation

The simplest reject inference method is hard cutoff augmentation, also known as simple augmentation (Siddiqi, 2017). Simple augmentation builds a model on accepted applicants and applies this model to rejected applicants. Once model parameters are

Table 1. Parceling.

Parcel	Predicted probability bad	Number of goods—accepts	Number of bads—accepts	Proportion of bads—accepts	Number of goods—rejects	Number of bads—rejects	Proportion of bads—rejects
1	$0 < p \leq 0.10$	750	250	25%	4500	1500	25%
2	$0.10 < p \leq 0.20$	800	200	20%	4800	1200	20%
3	$0.20 < p \leq 0.30$
.
10	$0.90 < p \leq 1.0$	900	100	10%	5400	600	10%

estimated, the decision on the rejected applicant's characteristic variable x is classified as follows:

$$\hat{y} = \begin{cases} 1, & \text{if } P(y = 1|x) \geq 1/2 \text{ bad rate accepted applicants} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Classifying rejected applicants as bad loans using a cutoff value that is half the bad rate among the accepted applicants is standard in reject inference practice (Siddiqi, 2017). For this study, the LendingClub data had a 13% bad rate among the accepted applicants, so a cutoff value of 6.5% was used to classify rejected loans as bad loans.

3.1.2. Parceling

Parceling is a hybrid method combining proportional assignment and augmentation (Siddiqi, 2017). Proportional assignment is the random partitioning of the rejects into good and bad classifications based on the number of good and bad accepted applicants within the scoring parcels. Table 1 lists the parceling method. The predicted probabilities of default from the model built using accepted applicants were binned and placed into parcels. The proportional assignment of rejected applicants is applied within each parcel. Best practice dictates that rejected applicants should be randomly allocated within the parcels using a bad rate multiplicative factor of two to five times greater than the bad rate in the equivalent parcel of accepted applicants (Siddiqi, 2017).

Because of the high proportion of bad accepts in the higher parcels, the recommended multiplicative bad rate factor for rejected applicants was not feasible. Instead, the multiplicative factor used for allocating the proportion of bad rejected applicants to each parcel was half the proportion of good acceptance added to the proportion of bad accepts. For example, if a parcel had 87% of bad accepted applicants, then the proportion of bad rejected applicants allocated to this parcel would be 93.5%.

3.1.3. Fuzzy augmentation

Fuzzy augmentation uses rejected applicants twice in the final model. The rejected applicant is decomposed into two components: a partial good and a partial bad. For each rejected applicant, two observations are generated in the form of a weight variable; one observation has a good outcome loan

status, and another has a bad outcome loan status. The weight used for the partial-good applicants was $P(y = 0|x)$. The weight used for the partially bad applicants was $P(y = 1|x)$. A weighted logistic regression was performed using the accepted and rejected applicants for the final model.

3.2. Data and variables

Data in this study used publicly available data from the LendingClub for the period 2007 to the first quarter of 2019. During this period, LendingClub evaluated 32,614,385 credit applicants for P2P loans. The accepted loans include 2,376,343 applicants and 144 variables describing the applicants, loan characteristics, and the known good/bad loan outcome status. Rejected loans contain 30,238,042 loans and only nine variables. The analysis used five overlapping variables: employment length, region, loan amount, debt-to-income ratio, and loan status. The employment length variable was discretized into the following values: < 1 year, 1–5 years, 6–9 years, 10+ years, and a separate bin for unknown values. The region variable was discretized into four standard US regions with unknown regions removed from the final consideration. Similarly, the loan amount and debt to income ratio variables that were unknown, negative, or above three times the inter-quartile range (IQR) were deemed outliers and were removed from the final consideration. This filtering reduced the final dataset size to 2,370,242 accepted loans and 27,128,116 rejected loans.

The target variable that indicates whether a loan was good or bad was determined using the loan status variable in the accepted applicant's data. The categories "fully paid," "current," "in grace period," and "does not meet the credit policy: fully paid" were classified as good loans. The categories "charged off," "default," "late (16–30 days)," "late (31–120 days)," and "does not meet the credit policy: fully charged off" were classified as bad loans. After deriving the good and bad status for the accepted loans, there were 308,131 bad loans and 2,062,111 good loans. The default rate for the accepted applicants was 13%.

Because the number of classes in the target variable is unequal, the Lending Club dataset is unbalanced. Traditionally, an unbalanced data set in

Table 2. Yearly statistics 2007-Q1 2019.

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Q1 2019
Accepts	585	2,370	5,239	12,480	21,655	53,331	134,770	235,552	420,998	434,090	442,108	492,071	114,993
Goods	435	1,877	4,523	10,735	18,368	44,698	113,749	193,853	342,731	358,195	387,658	465,957	113,778
Bads	150	493	716	1,745	3,287	8,637	21,021	41,699	78,267	77,895	54,450	26,114	1,215
Default Rate	25.64%	20.80%	13.67%	13.98%	15.18%	16.20%	15.6%	17.70%	18.59%	17.48%	12.32%	5.31%	1.06%
Rejects	4,596	20,026	41,976	95,189	185,139	303,491	676,024	1,572,701	2,461,236	4,183,539	6,299,987	8,838,990	2,445,222
Acceptance Rate	11.29%	10.58%	11.10%	11.59%	10.47%	14.95%	16.62%	13.03%	14.61%	9.4%	6.56%	5.27%	4.49%
Rejection Rate	88.71%	89.42%	88.90%	88.41%	89.53%	85.05%	83.38%	86.97%	85.39%	90.6%	93.44%	94.73%	95.51%

credit scoring is handled by undersampling or oversampling in practice (Anderson, 2007; Mays, 2001; Siddiqi, 2017). However, in prior seminal credit scoring academic publications that performed empirical studies, the datasets were left unbalanced (Crone & Finlay, 2012). Crone and Finlay (2012) pointed out that many of the published empirical credit scoring studies focused on applying and comparing algorithms to different credit scoring datasets, not on determining the appropriate ratio for the number of goods and bads that the target variable should contain. In a similar fashion, this study will not rebalance the Lending Club data, but will use the original distribution of observed good and bad loans in the data.

Table 2 presents yearly statistics of the data from 2007 to the first quarter of 2019. LendingClub's default rates range from 5.31% to 25.64%. Li et al. (2017) suggest that the large variation in the default rates could be because of loan demand trends, the LendingClub's internal loan policies, and the general economic environment.

Figure 1 shows an interesting trend in the default and acceptance rates. As the default rate increases and decreases, the acceptance rates follow the same pattern. The first quarter of 2019 is not displayed in Figure 1 because we do not have a complete year of data.

Table 3 displays the descriptive statistics for the debt to income and loan amount variables from 2007 to the first quarter of 2019. The rejected applicants have higher average debt-to-income ratio than accepted applicants for each year. Up until 2012, rejected applicants requested more money than accepted applicants, on average. In 2013 and beyond, accepted applicants requested more money, on average, than rejected applicants.

The distribution of loan amounts for good and bad loans for accepted applicants and the distribution for rejected applicants is shown in Figure 2. The range of the loan amount was the same for all three groups. The loan amount requested by accepted applicants for bad loans is slightly higher than that for good loans. More rejected applicants requested loan amounts in multiples of five-and ten-thousand-dollar increments. Interestingly, accepted applicants requested more loan amounts that were not in multiples of five-and ten-thousand-dollar increments.

The distribution of debt to income for good and bad loans for accepted applicants and the distribution for rejected applicants is shown in Figure 3. The debt-to-income variability for good and bad loans is fairly similar. As one would imagine, more variability is found in the rejected applicant's debt to income values. The large spike at 100 indicates that some of the rejected applicants had the same amount of debt as their income.

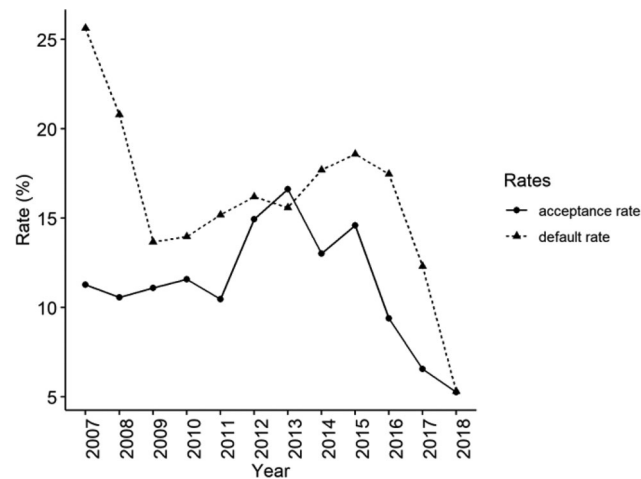


Figure 1. Acceptance and default rates 2007–2018.

Table 3. Descriptive statistics 2007–Q1 2019.

		Mean		Standard deviation	
		Debt to income	Loan amount	Debt to income	Loan amount
2007	Accepts	11.04	8,296	7.16	6,127
	Rejects	21.71	8,827	20.59	7,275
	Both	21.00	8,767	19.83	7,156
2008	Accepts	13.33	8,847	7.28	5,742
	Rejects	21.82	8,975	19.16	6,867
	Both	20.92	8,962	18.46	6,757
2009	Accepts	12.57	8,847	6.54	6,006
	Rejects	22.48	10,749	18.51	7,658
	Both	21.38	10,650	17.86	7,498
2010	Accepts	13.15	10,545	6.51	6,603
	Rejects	22.68	11,390	18.23	8,465
	Both	21.57	11,292	17.55	8,275
2011	Accepts	13.89	12,056	6.68	8,170
	Rejects	22.68	13,412	17.62	10,671
	Both	20.48	13,270	16.96	10,445
2012	Accepts	16.67	13,464	7.58	8,085
	Rejects	22.40	14,740	17.55	11,041
	Both	21.55	14,549	16.67	10,661
2013	Accepts	17.22	14,709	7.59	8,098
	Rejects	23.25	13,776	17.74	10,800
	Both	22.25	13,931	17.22	10,405
2014	Accepts	18.05	14,871	8.02	8,438
	Rejects	23.09	12,317	18.41	10,475
	Both	22.43	12,469	18.05	10,268
2015	Accepts	19.15	15,241	8.67	8,571
	Rejects	23.25	13,776	17.74	10,800
	Both	23.62	13,081	19.15	10,116
2016	Accepts	18.81	14,734	8.79	8,991
	Rejects	25.15	12,545	19.87	10,370
	Both	24.55	12,751	18.81	10,598
2017	Accepts	18.84	14,834	9.90	9,627
	Rejects	28.66	12,284	15.29	10,683
	Both	28.02	12,451	18.84	10,636
2018	Accepts	19.06	16,008	11.11	10,126
	Rejects	31.55	12,878	19.87	10,730
	Both	30.89	13,043	19.06	10,727
Q1 2019	Accepts	19.83	16,650	11.25	8,170
	Rejects	31.69	13,816	29.53	11,217
	Both	31.16	13,944	19.83	11,194

3.3. Model performance measures

Generally, a valid way for measuring model performance is to use its classification accuracy and discriminatory power. Crook et al. (2007) suggests that the confusion matrix, which compares the number of actual good and bad loans to the predicted number of good and bad loans. From the confusion matrix, several model diagnostic statistics

can be computed for evaluating the reject inference models. These statistics are listed as follows

- Accuracy: Proportion of correctly classified good and bad loans.

$$Accuracy = \frac{TN + TP}{(TN + FN + FP + TP)}$$

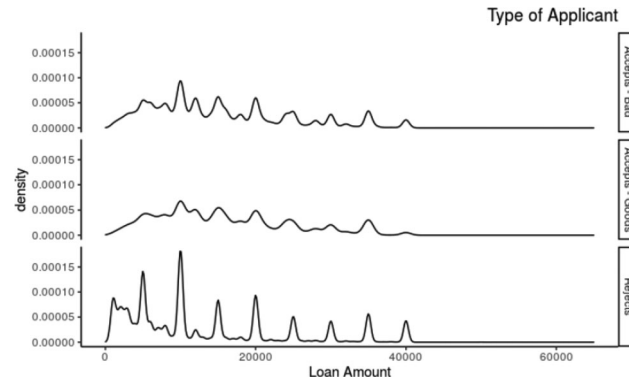


Figure 2. Distribution of loan amount.

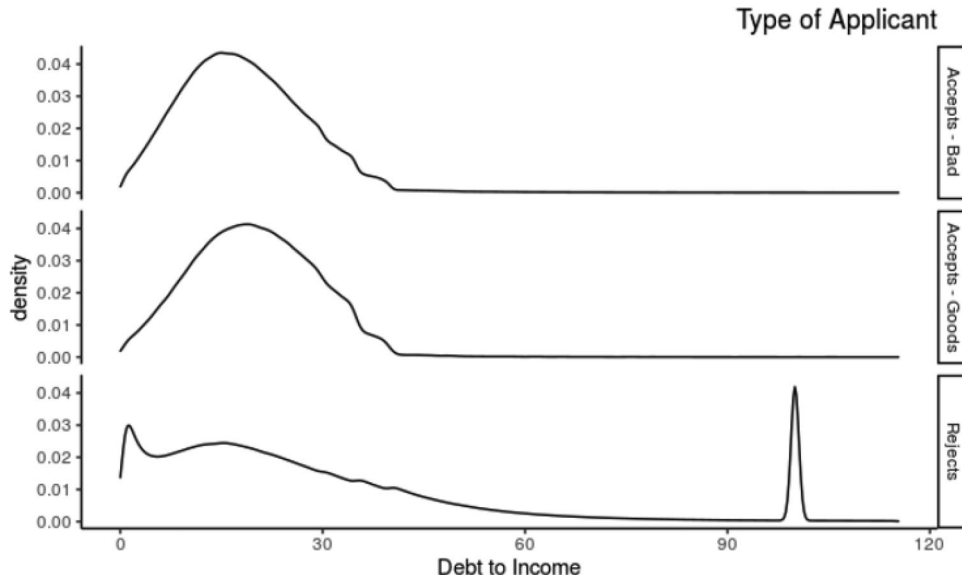


Figure 3. Distribution of debt to income.

where TN = true negative, TP = true positive, FN = false negative, and FP = false positive

- i. Sensitivity: Proportion of bad loans correctly classified as bad loans.

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}$$

- ii. 1-specificity: Proportion of good loans incorrectly classified as bad loans.

$$1 - \text{specificity} = \frac{FP}{(TN + FP)}$$

- iii. Area under the ROC curve (AUC): area under the receiver operating characteristic curve (ROC) The ROC curve is a graph of sensitivity versus 1-specificity for all cut-off values. A cut-off value is the posterior predicted probability from the models determining which applicants are classified as good or bad loans. AUC values greater than 0.50 indicate that the model performs better than if the financial agency randomly assigns applicants as good or bad credit risks.

The question arises: which of these model performance measures would be preferred in the reject

inference/credit scoring application? Accuracy is a popular metric in financial applications such as credit scoring because it can signal significant loss or profits (Li & Chen, 2020). However, it should be noted that a disadvantage of accuracy is that it is one metric developed from using one cutoff value of the posterior probabilities from the models. Accuracy can change if the cutoff value changes. While accuracy is the performance of the model at one cutoff value, AUC is the performance of a model across all cutoff values (Ballings & Van den Poel, 2013). Several authors have argued that AUC is a more objective measure of a model's performance and should be preferred instead of accuracy (Baecke & Van den Poel, 2011; Coussement & Van den Poel, 2008).

Siddiqi (2017) suggests that the choice of which model performance measure to choose is ultimately based on what the business objective of the credit scoring model. For example, if the model is being built to minimize losses, then sensitivity should be maximized and (1-specificity) should be minimized. Hand (2012) pointed out that none of the model performance measures are "right" or "wrong." They each simply assess different components of the models.

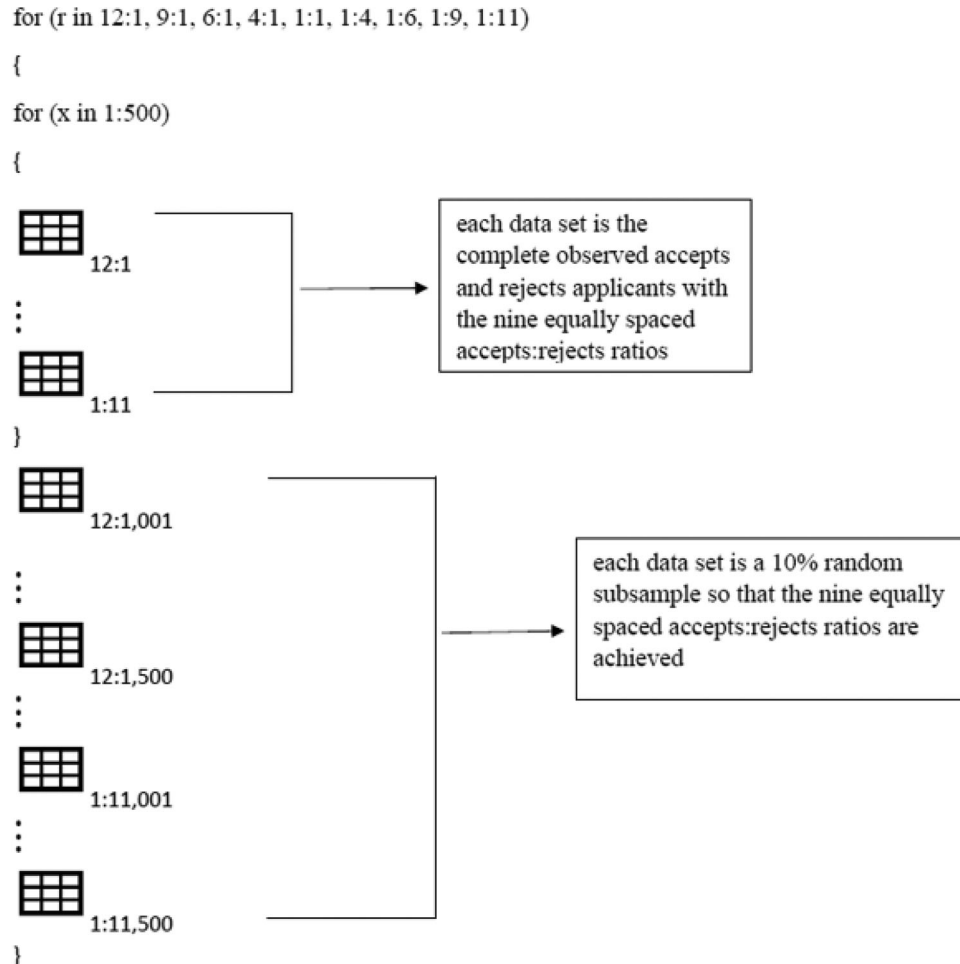


Figure 4. Monte Carlo simulation accepts:rejects methodology.

The focus of selecting a specific model performance measure highlights the importance of deciding on an objective of the credit scoring model that will be implemented (Hand, 2006; Siddiqi, 2017).

3.4. Monte Carlo simulation methodology

In this study, two Monte Carlo simulations were performed. Each of these Monte Carlo simulations are explained in the next two sub-sections.

3.4.1. Accepts:rejects ratio

The 92% rejection rate in the LendingClub data leads to an unbalanced number of accepted compared to rejected applicants. This rejection rate equates to a ratio of 1 to 11; that is, for every accepted applicant, 11 applicants are rejected. To assess the effects for each of the reject inference techniques on an unbalanced accepts:rejects (read as “accepts to rejects”) ratio, the first Monte Carlo simulation determines an accepts:rejects ratio for each reject inference technique. The second simulation uses the accepts:rejects ratio to perform another Monte Carlo simulation to provide suggestions for the best reject inference technique that should be employed under certain scenarios.

Figure 4 illustrates the simulation methodology for the first Monte Carlo simulation determining the accepts:rejects ratio using a grid search over nine equally spaced accepts:rejects ratios. The ratios 12:1, 9:1, 6:1, 4:1, 1:1, 1:4, 1:6, 1:9, 1:11 are the different combinations of the accepts:rejects ratio employed by the grid search. To determine an accepts:rejects ratio for each of the reject inference techniques, 500 10% subsampled data sets were generated using nine equally spaced accepts:rejects ratios. For each of the 4,500 sampled datasets, simple augmentation, parceling, and fuzzy reject inference methods were applied to each of the sampled datasets.

Accuracy, area under the ROC curve (AUC), sensitivity, and (1-specificity) were used for evaluating each of the reject inference techniques for selecting the accepts:rejects ratio. In practice, analysts use a variety of models to compare the performance of classification models (Hand, 2012). Normally, these model diagnostics are reserved for evaluating the performance of reject inference techniques (Crook et al., 2007). In the first simulation, these model diagnostics are used for selecting the accepts:rejects ratio.

The accepts:rejects ratio for each reject inference technique will be used in the second Monte Carlo

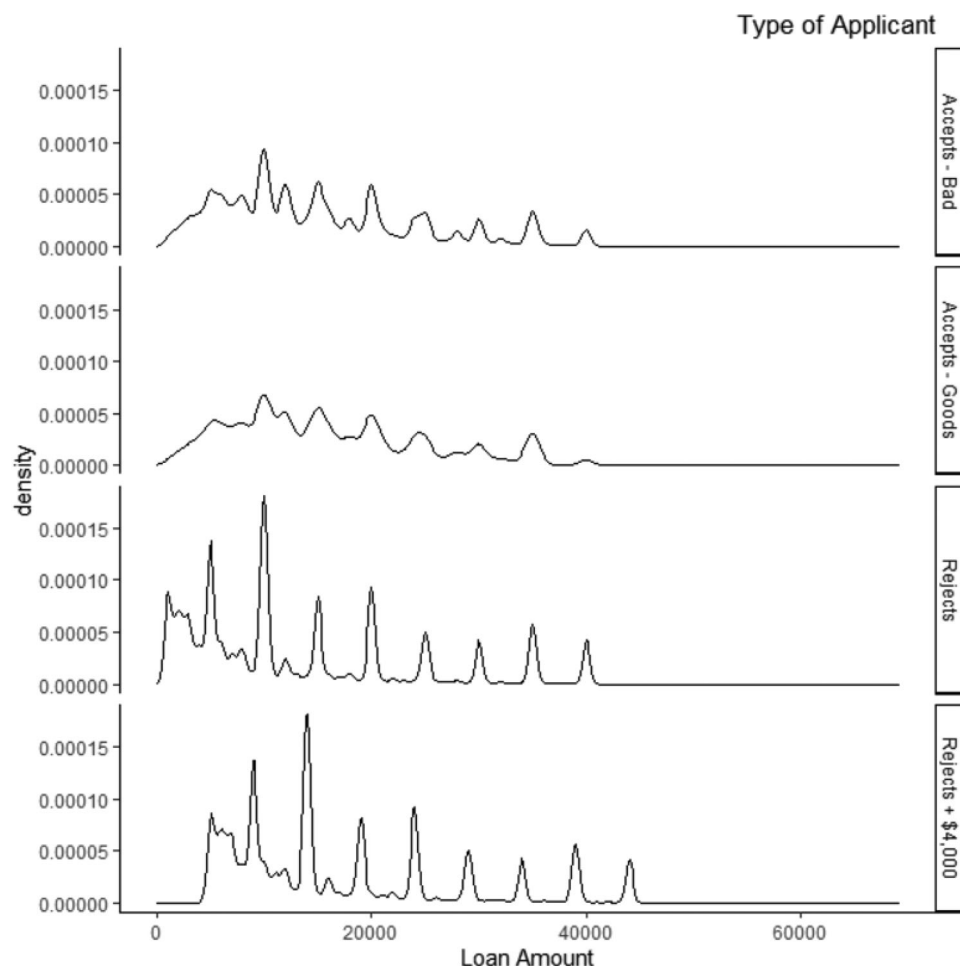


Figure 5. Loan amount for rejected applicants shifted \$4000.

simulation to explore different distributions of the variables of the rejected applicants. The second simulation determines the reject inference technique that should be selected under varying values for the interval-level variables of rejected applicants.

3.4.2. Reject inference models

The second simulation uses the accepts:rejects ratio, selected using the methodology in Section 3.4.1 (Accepts:Rejects Ratio), to perform a second Monte Carlo simulation to study the performance of the three reject inference methods. Perturbations are introduced to the distributions of the two interval variables for rejected applicants, debt to income, and loan amount. Debt to income is increased and decreased by up to eight percent in increments of two. The loan amount variable increases and decreases by up to \$4,000 in increments of \$1,000. The employment length and region variables were kept constant as these are non-interval variables. To illustrate how the perturbations in the variables are performed, Figure 5 compares the observed distribution of loan amount to a maximum \$4,000 perturbation of loan amount for the rejected applicants. Thus, the loan amount for the rejected applicants' distribution has been increased by \$4,000 more than what has been empirically observed, while keeping

the accepted applicants' distribution constant. To illustrate the perturbations further, Figure 6 compares the observed distribution of debt-to-income to a maximum 8% perturbation of debt-to-income for the rejected applicants. Therefore, debt to income for the rejected applicants' distribution has been increased by 8% more than has been empirically observed while holding accepted applicants' distribution constant.

As with all Monte Carlo simulations, the perturbations will possibly cause values outside of the domain. In this case, negative debt-to-income ratios and negative loan amounts could be produced. After the Monte Carlo cell was determined, all the cells in the original distribution were fully generated. Subsequently, out of domain values were removed. This process ensures that the edges of the search space maintain their empirically determined shapes as close as possible.

1,000 replications in total were performed for each of the 81 (nine loan amounts by nine debt to income values) Monte Carlo cells. The three reject inference models, as described in Section 3.1 (Reject Inference Models), were fitted for each replication. This procedure exceeded our capacity to generate and store each of the 81 Monte Carlo cells. Each of the 81 Monte Carlo cells was generated individually

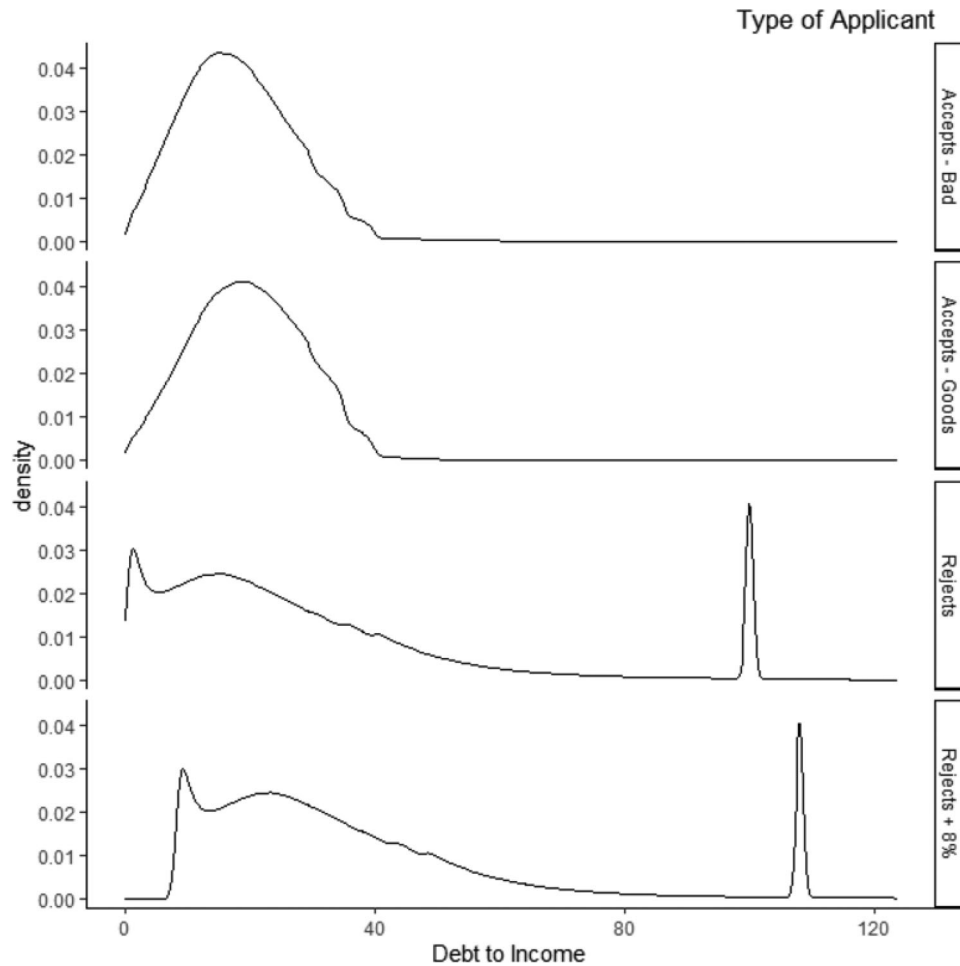


Figure 6. Debt to income for rejected applicants shifted 8%.

based on the seed values. [Appendix A](#) provides an analysis to determine the number of simulations based on 95% confidence intervals for the four model diagnostics discussed in this study. The 95% confidence intervals could be used to determine a reasonable compromise between the accuracy of the desired model diagnostic and computational effort. For this study, the authors chose to use 1,000 replications to obtain the most robust model diagnostic estimates.

To ensure model validity, k-fold cross validation was performed on each of the 1,000 replications in each Monte Carlo cell to verify reject inference model performance. The reject inference model was trained and tested k times for each simulated sample. Cross-validation consists of dividing the sample into k subgroups. Each subgroup was tested using the classification rule from the remaining (k-1) groups (Hand et al., 2001). The k different test results were obtained for each train-test configuration. This study used a 10-fold cross-validation. The reject inference models were built on training data sets and evaluated on test datasets. The model performance measures were computed from the average of the 1,000 replications from the test data for each of the 81 Monte Carlo cells.

4. Results

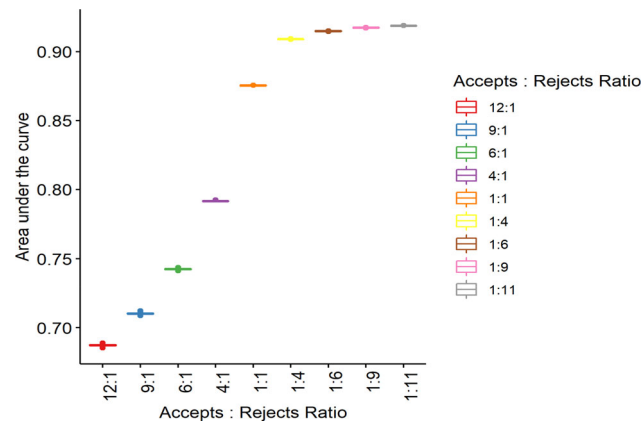
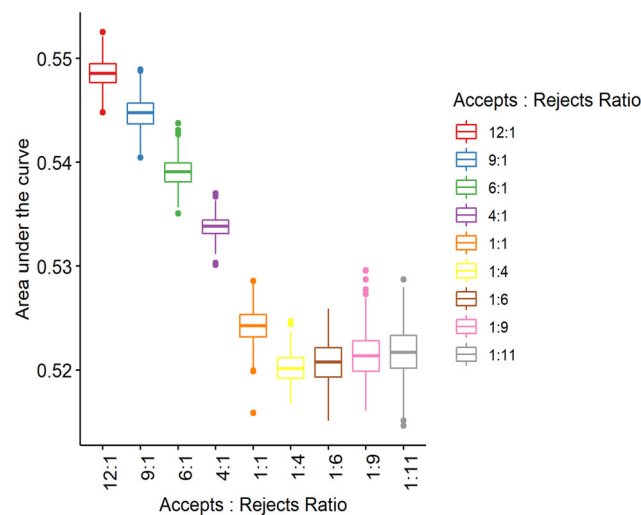
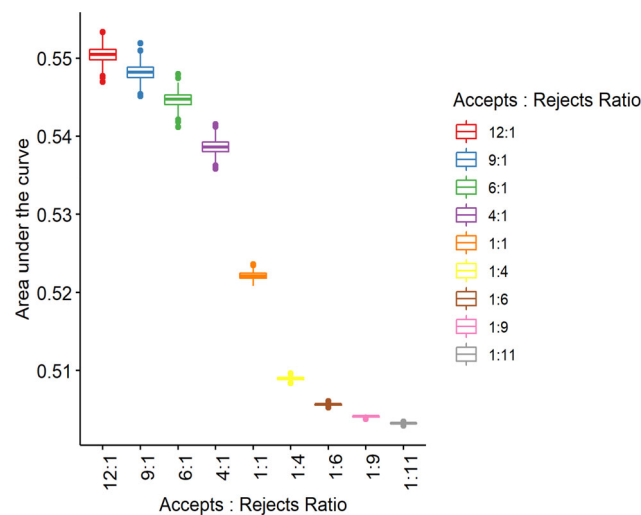
4.1. Selecting the accepts:rejects ratio

The model diagnostics from the first Monte Carlo simulation are presented in [Table 4](#) along with the accepts:rejects ratio. For simple augmentation, AUC, accuracy, and sensitivity produce the same accepts:rejects ratio with the number of rejects being close to what the ratio was in the observed data. If reducing the number of false positives is important for simple augmentation, then the ratio should reflect the opposite of the accepts:rejects ratio in the observed data. That is, the number of accepts should be 11 for every rejected applicant. For parceling and fuzzy, we observed an opposite trend in the accepts:rejects ratio for AUC as compared to simple augmentation. [Figure 7](#) illustrates that the more rejected applicants are used in simple augmentation, the better the AUC. [Figures 8](#) and [9](#) display that the more rejected applicants used in the parceling and fuzzy techniques, the more the AUC declines.

For simple augmentation, because all four model diagnostics produce the same accepts:rejects ratio, a ratio of 1:11 will be used in the second Monte Carlo simulation. Two model diagnostics, accuracy and 1-specificity, produce the same accepts:rejects ratios;

Table 4. Accepts:rejects ratio model diagnostics for sampled data sets.

Reject inference technique	Maximum AUC accepts:rejects ratio	Maximum accuracy accepts:rejects ratio	Maximum sensitivity accepts:rejects ratio	Minimum 1-specificity accepts:rejects ratio
Simple Augmentation	0.92 1:11	0.85 1:11	0.84 1:11	0.09 1:11
Parcelling	0.55 12:1	0.54 4:1	0.95 1:4	0.46 4:1
Fuzzy	0.55 12:1	0.69 6:1	0.35 12:1	0.18 4:1

**Figure 7.** Simple augmentation: effects of adding more rejected applicants.**Figure 8.** Parceling: effects of adding more rejected applicants.**Figure 9.** Fuzzy: effects of adding more rejected applicants.

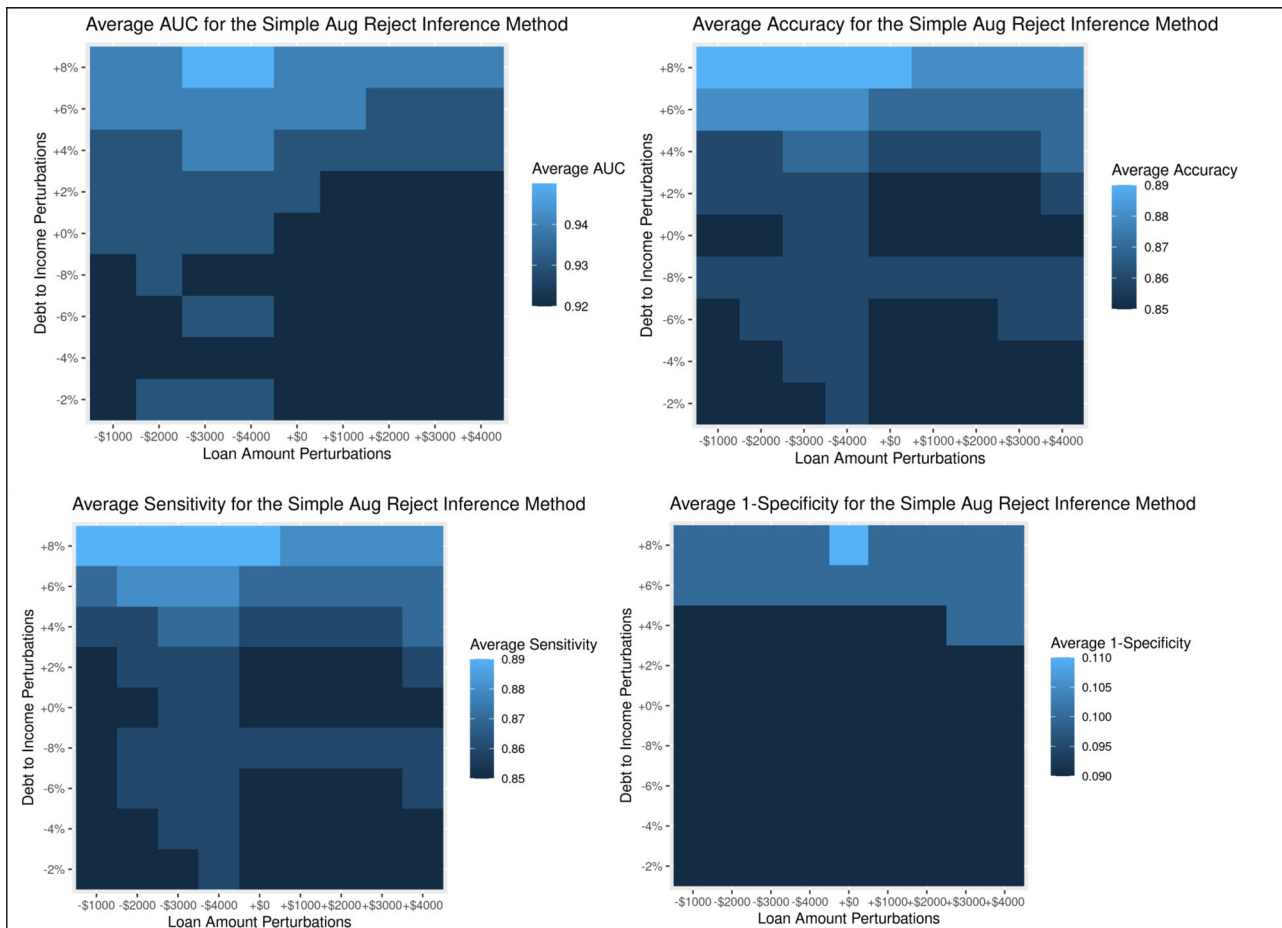


Figure 10. Average model diagnostics from simple augmentation Monte Carlo simulations.

thus, parceling will employ an accepts:rejects ratio of 4:1 in the second Monte Carlo simulation. The fuzzy method also has two model diagnostics, AUC and sensitivity, that produce the same accepts:rejects ratios, so fuzzy will employ an accepts:rejects ratio of 12:1 in the second Monte Carlo simulation.

4.2. Heatmaps

Heatmaps were constructed for all model diagnostics discussed in Section 4 for all three reject inference techniques under study for the different perturbations of the interval variables. Each cell in the heat map represents the average model diagnostics over 1,000 replications from the test dataset.

Figure 10 displays the average model diagnostics for the simple augmentation reference method. Simple augmentation produced the best model diagnostics among all three reject inference techniques. AUC performs best when loan amount is decreased by \$3,000 and \$4,000 and debt to income is increased by 8%.

The accuracy and sensitivity exhibit similar patterns in the heatmap distributions in Figure 10. Both metrics achieved optimum performance when loan amount is decreased by \$3,000 and \$4,000 and debt to income is increased by 8%. Sensitivity is also at its peak for all decreased shifts in loan amount.

This pattern indicates that as the shifts in loan amount decrease and the shifts in debt to income increase, AUC, accuracy and sensitivity are improving. (1-specificity) has a distinct pattern for a large majority of the heatmap. The metric performs best for all shifts in loan amount and for all changes in debt to income from -2% to 4%.

Figure 11 displays the average model diagnostics for the parceling reject inference method. The best AUC values occur when loan amount is decreased by \$3,000 or \$4,000 and debt to income is increased by 6% or 8%. Accuracy is optimized when debt to income is increased by 8%, and the loan amount decreases by \$4000. Model sensitivity performs best for all shifts in loan amount (including no shift) and when debt to income is increased by 8%. (1-specificity) is best when debt to income is decreased by 6% and 8%, and the loan amount is decreased by \$4000.

Figure 12 displays the average model diagnostics for the fuzzy-reject inference method. The best AUC and sensitivity values occur when loan amounts decreased by \$4,000 and debt to income increased by 8%. The peak accuracy values occur with the same \$4,000 decrease in loan amount and when debt to income is decreased by 6% and 8%. The lower portion of the heatmap for (1-specificity) indicate low values of (1-specificity) for all decreases in debt to income across all adjustments of

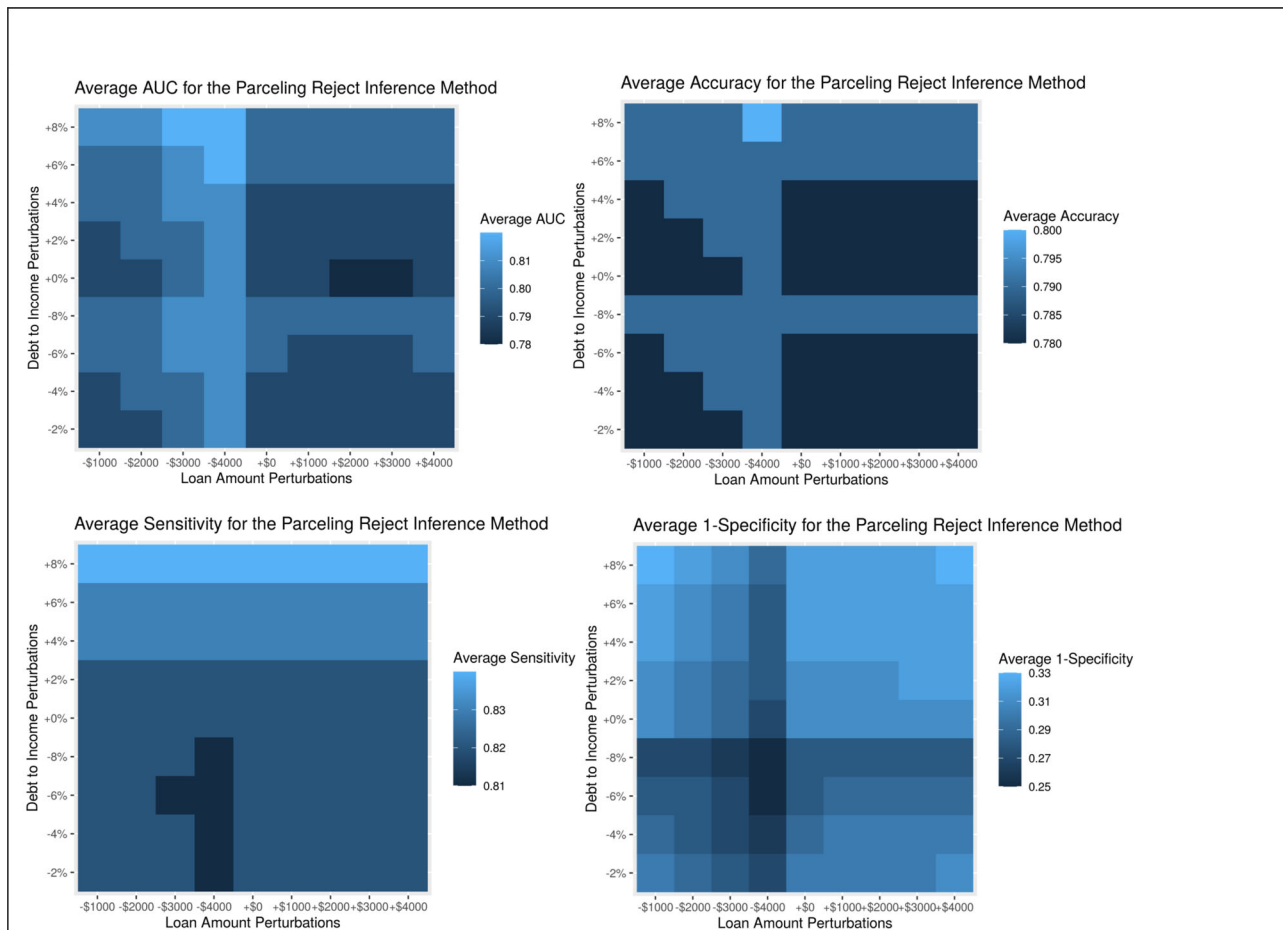


Figure 11. Average model diagnostics from parceling Monte Carlo Simulations.

loan amount. The only exception would be for the loan amount decreases of \$3,000 and \$4,000 which do indicate a slightly higher (1-specificity) amount.

5. Conclusion

In this study, we provide a Monte Carlo framework that can be utilized for assessing reject inference models. This research used traditional reject inference techniques, which are known to be used in practice (Siddiqi, 2017). For this particular study, different reject inference model diagnostics performed better using different shifts in the two interval variables under study. An analyst should expect similar differing model diagnostic results using a larger portfolio with more interval-level variables. The results of Monte Carlo simulation study depend on the particular rejected applicant's characteristic data.

From the heatmap analysis for simple augmentation regarding AUC, accuracy, and sensitivity, when debt to income is increased to its maximum (8%) and loan amount is decreased by \$3,000 or \$4,000, these model diagnostics performed optimally. This could indicate that for the LendingClub data, for the simple augmentation method and for these model diagnostics, rejected applicants should be incorporated into the model with increased debt to income values and

decreased loan amounts. Therefore, a rejected applicant with these variable shifts is required for optimum values for these model diagnostics for simple augmentation. This is the type of decision-making conclusion that this Monte Carlo simulation study can offer to a practitioner. The results of the Monte Carlo simulation can provide guidance on the exact size of shifts/perturbations in the rejected applicant's variables that are required to achieve optimal model performance for different reject inference methods.

The Monte Carlo framework that has been developed is general enough to provide a means of conducting similar experiments using other reject inference techniques. Generally, this method provides an approximation on the appearance of rejected applicants when actually observing rejected applicants is not possible with certain distributional structures. This Monte Carlo simulation can facilitate decision-making for financial institutions by providing insight into what the distributions of rejected applicants should look like for optimal model performance.

One area of future research arising from this work is extending the Monte Carlo framework to incorporate categorical variables into the simulation. Li and Racine (2003) established the theoretical framework for estimating the joint distributions of interval and categorical variables using a nonparametric kernel via

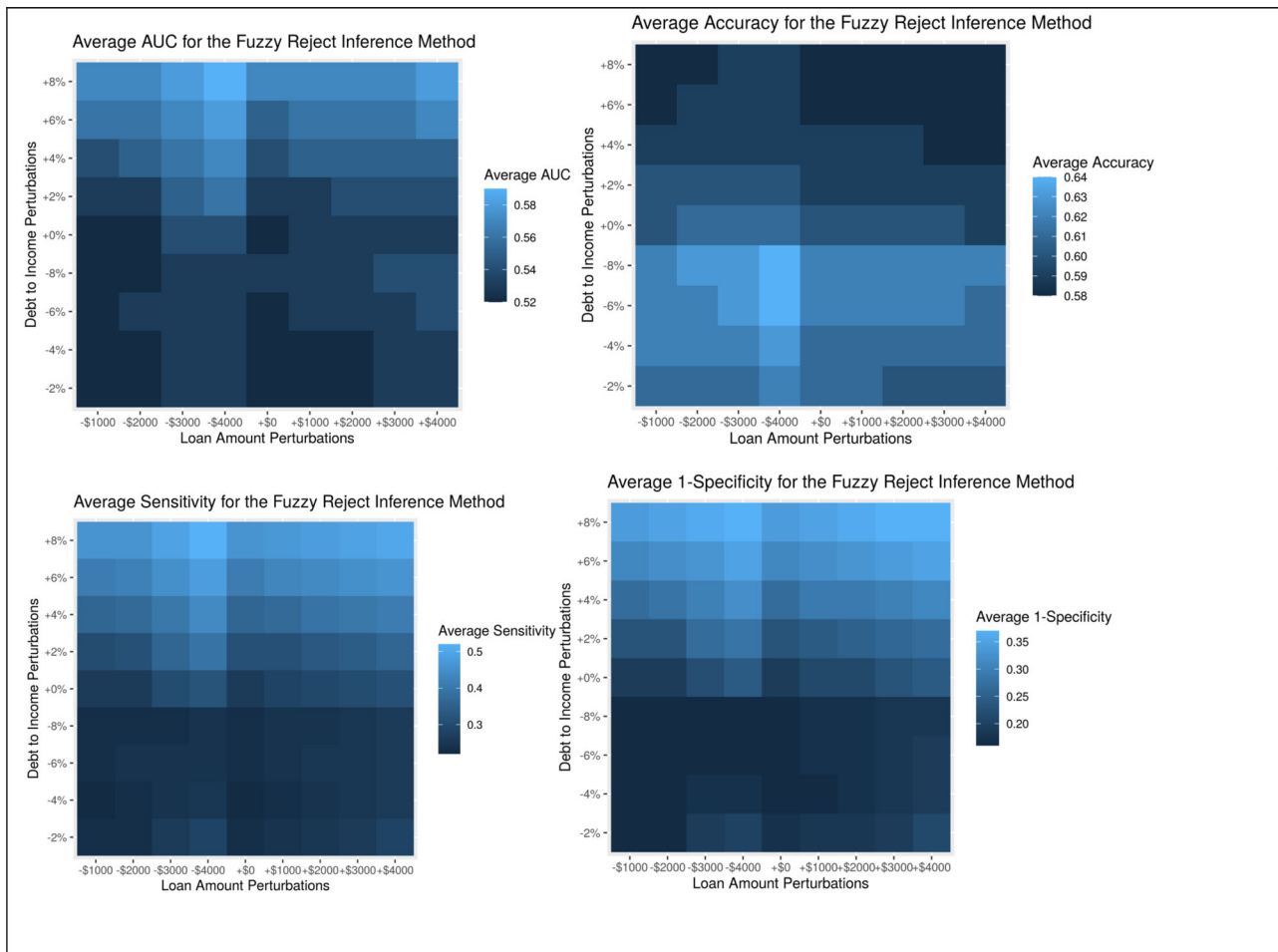


Figure 12. Average model diagnostics from Fuzzy Monte Carlo Simulations.

Monte Carlo simulation. Perhaps applying a nonparametric kernel method for estimating joint distributions over intervals and categorical variables could be employed in future work.

There is no doubt a growing trend of using models from machine learning as credit scoring models that do not have logistic regression as their underlying statistical model. The Monte Carlo framework developed in this study can be applied to the reject inference problem using machine learning models. It would be quite interesting to compare different models such as support vector machines, neural networks, and decision trees to determine if there are certain distributional shifts in the rejected applicant's variables that are needed for these types of models.

Another future area of research would be to determine if an optimal accepts:rejects ratio can be found. This study used traditional credit scoring model performance measures to determine which accepts:rejects ratio would be used in the second Monte Carlo simulation that perturbed the distributions of the rejected applicant's variables. However, future work should examine a methodology for finding the optimal accepts:rejects ratio. Then, the research could be extended to ask whether the optimal accepts:rejects ratio can guarantee the best performance of the second Monte Carlo simulation.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Billie Anderson  <http://orcid.org/0000-0002-1327-7004>

Mark A. Newman  <http://orcid.org/0000-0001-6155-438X>

Philip A. Grim II  <http://orcid.org/0000-0002-0543-7979>

J. Michael Hardin  <http://orcid.org/0000-0003-4468-0661>

References

- Anderson, B. (2019). Using Bayesian networks to perform reject inference. *Expert Systems with Applications*, 137, 349–356. <https://doi.org/10.1016/j.eswa.2019.07.011>
- Anderson, R. (2007). *The credit scoring toolkit: Theory and practice for retail credit risk management and decision automation*. Oxford University Press.
- Baecke, P., & Van den Poel, D. (2011). Data augmentation by predicting spending pleasure using commercially available external data. *Journal of Intelligent Information Systems*, 36(3), 367–383. <https://doi.org/10.1007/s10844-009-0111-x>
- Baensens, B. (2003). *Developing intelligent systems for credit scoring using machine learning techniques* [Unpublished doctoral dissertation]. Katholieke Universiteit Leuven.

- Ballings, M., & Van den Poel, D. (2013). Kernel factory: An ensemble of kernel machines. *Expert Systems with Applications*, 40(8), 2904–2913. <https://doi.org/10.1016/j.eswa.2012.12.007>
- Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications*, 36(2), 3302–3308. <https://doi.org/10.1016/j.eswa.2008.01.005>
- Bucker, M., van Kampen, M., & Kramer, W. (2013). Reject inference in consumer credit scoring with nonignorable missing data. *Journal of Banking & Finance*, 37(3), 1040–1045. <https://doi.org/10.1016/j.jbankfin.2012.11.002>
- Chen, G. G., & Astebro, T. (2012). Bound and collapse Bayesian reject inference for credit scoring. *Journal of the Operational Research Society*, 63(10), 1374–1387. <https://doi.org/10.1057/jors.2011.149>
- Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services. An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34(1), 313–327. <https://doi.org/10.1016/j.eswa.2006.09.038>
- Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, 28(1), 224–238. <https://doi.org/10.1016/j.ijforecast.2011.07.006>
- Crook, J., & Banasik, J. (2004). Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance*, 28(4), 857–874. [https://doi.org/10.1016/S0378-4266\(03\)00203-6](https://doi.org/10.1016/S0378-4266(03)00203-6)
- Crook, J., Edelman, D. B., & Thomas, L. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183(3), 1447–1465. <https://doi.org/10.1016/j.ejor.2006.09.100>
- Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91, 106263. <https://doi.org/10.1016/j.asoc.2020.106263>
- Eisenbeis, R. A. (1978). Problems in applying discriminate analysis in credit scoring models. *Journal of Banking & Finance*, 2(3), 205–219. [https://doi.org/10.1016/0378-4266\(78\)90012-2](https://doi.org/10.1016/0378-4266(78)90012-2)
- Ehrhardt, A., Biernacki, C., Vandewalle, V., Heinrich, P., & Beben, S. (2021). Reject inference methods in credit scoring. *Journal of Applied Statistics*, 48(13–15), 2734–2754.
- Enders, C. K., & Bandalos, D. L. (2001). The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, 8(3), 430–457. https://doi.org/10.1207/S15328007SEM0803_5
- Feelders, A. J. (1999). Credit scoring and reject inference with mixture models. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 8(4), 271–279. [https://doi.org/10.1002/\(SICI\)1099-1174\(199912\)8:4<271::AID-ISAF170>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1099-1174(199912)8:4<271::AID-ISAF170>3.0.CO;2-P)
- Goh, R. Y., & Lee, L. S. (2019). Credit scoring: A review on support vector machines and metaheuristic approaches. *Advances in Operations Research*, 2019, 1–30. <https://doi.org/10.1155/2019/1974794>
- Hand, D. J. (2001). Modeling consumer credit. *IMA Journal of Management Mathematics*, 12(2), 139–155. <https://doi.org/10.1093/imaman/12.2.139>
- Hand, D. J. (2006). Classifier technology and the illusion of progress (with discussion). *Statistical Science*, 21, 1–34.
- Hand, D. J. (2012). Assessing the performance of classification methods. *International Statistical Review*, 80(3), 400–414. <https://doi.org/10.1111/j.1751-5823.2012.00183.x>
- Hand, D. J., & Henley, W. (1994). Can reject inference ever work? *IMA Journal of Mathematics Applied in Business & Industry*, 5(4), 45–55.
- Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. MIT Press.
- Iyer, R., Khwaja, A. I., Luttmer, E. F. P., & Shue, K. (2016). Screening peers softly: Inferring the quality of small borrowers. *Management Science*, 62(6), 1554–1577. <https://doi.org/10.1287/mnsc.2015.2181>
- Ioannes, D. N. (1993). Reject inference applied to logistic regression for credit scoring. *IMA Journal of Mathematics Applied in Business & Industry*, 5(1), 35–43.
- Lai, P., Liu, Y., Liu, Z., & Wan, Y. (2017). Model free feature screening for ultrahigh dimensional data with responses missing at random. *Computational Statistics & Data Analysis*, 105, 201–216. <https://doi.org/10.1016/j.csda.2016.08.008>
- Lessman, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1), 124–136. <https://doi.org/10.1016/j.ejor.2015.05.030>
- Levatic, J., Ceci, M., Kocev, D., & Dzeroski, S. (2017). Semi-supervised classification trees. *Journal of Intelligent Information Systems*, 49(3), 461–486. <https://doi.org/10.1007/s10844-017-0457-4>
- Li, Y., & Chen, W. (2020). A comparative performance assessment of ensemble learning for credit scoring. *Mathematics*, 8(10), 1756. <https://doi.org/10.3390/math8101756>
- Li, Q., & Racine, J. (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis*, 86(2), 266–292. [https://doi.org/10.1016/S0047-259X\(02\)00025-8](https://doi.org/10.1016/S0047-259X(02)00025-8)
- Li, Z., Tian, Y., Li, K., Zhou, F., & Yang, W. (2017). Reject inference in credit scoring using semi-supervised support vector machines. *Expert Systems with Applications*, 74, 105–114. <https://doi.org/10.1016/j.eswa.2017.01.011>
- Liu, F. G., & Kost, J. (2017). Applications of simulations for missing data issues in longitudinal clinical trials. In D. Chen & J.D. Chen (Eds.), *Monte-Carlo simulation-based statistical modeling* (pp. 211–232). Springer.
- Liu, Y., Li, X., & Zhang, Z. (2020). A new approach in reject inference of using ensemble learning based on global semi-supervised framework. *Future Generation Computer Systems*, 109, 382–391. <https://doi.org/10.1016/j.future.2020.03.047>
- Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2020). Deep generative models for reject inference in credit scoring. *Knowledge-Based Systems*, 196, 105758. <https://doi.org/10.1016/j.knosys.2020.105758>
- Mays, E. (2001). *Handbook of credit scoring*. Glenlake Publishing Company.
- Nikolaïdis, D., Doumpos, D., & Zopounidis, C. (2017). Exploring population drift on consumer credit behavioral scoring. In E. Grigoroudis & M. Doumpos (Eds.), *Operations research in business and economics* (pp.145–165). Springer.
- Oskarsdottir, M., Bravo, C., Sarraute, C., Vanthienen, J., & Baesens, B. (2019). The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analysis. *Applied Soft Computing*, 74, 26–39. <https://doi.org/10.1016/j.asoc.2018.10.004>
- Shen, F., Zhao, X., & Kou, G. (2020). Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems*, 137, 113366. <https://doi.org/10.1016/j.dss.2020.113366>

- Siddiqi, N. (2017). *Intelligent credit scoring: Building and implementing better credit risk scorecards* (2nd ed.). Wiley.
- Teles, G., Rodrigues, J. J. P. C., Rabelo, R. A. L., & Kozlov, S. A. (2020). Comparative study of support vector machines and random forests machine learning algorithms on credit operations. *Journal of Software: Practice and Experience*. <https://doi.org/10.1002/spe.2842>.
- Tian, Y., Yong, Z., & Luo, J. (2018). A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines. *Applied Soft Computing*, 73, 96–105. <https://doi.org/10.1016/j.asoc.2018.08.021>
- Vallee, B., & Zeng, Y. (2019). Marketplace lending: A new paradigm? *The Review of Financial Studies*, 32(5), 1939–1982. <https://doi.org/10.1093/rfs/hhy100>
- van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440. <https://doi.org/10.1007/s10994-019-05855-6>
- Verstraeten, G., & Van den Poel, D. (2005). The impact of sample bias on consumer credit scoring and profitability. *Journal of the Operational Research Society*, 56(8), 981–992. <https://doi.org/10.1057/palgrave.jors.2601920>
- Xia, Y., Yang, X., & Zhang, Y. (2018). A reject inference technique based on contrastive pessimistic likelihood estimation for P2P lending. *Electronic Commerce Research and Applications*, 30, 111–124. <https://doi.org/10.1016/j.elerap.2018.05.011>

Appendix A

Tables A1–A3 show 95% confidence intervals for the four model diagnostics discussed in Section 3.3 for each of the reject inference models. The confidence intervals were computed with no distributional shift in the loan amount and no distributional shift in the debt-to-income variables.

For the parceling and fuzzy methods, as the number of simulations increases, the confidence intervals for all the model diagnostics become narrower. These results indicate that if more precision for one of the model diagnostics is desired, the more simulations should be performed for these two reject inference models.

Overall, for simple augmentation, less variability in the confidence intervals was observed. For most of the confidence intervals, over all the different number of simulation values, the confidence intervals were constant. This result indicates, for this data, 50 simulations would have been sufficient.

The authors recognize that for the parceling and fuzzy methods, the difference in the confidence intervals occurred in the fourth decimal place. Therefore, a practitioner would have to decide, based on the model diagnostic chosen, if the gains achieved by obtaining a more precise confidence interval would be worth the computational effort for increasing the number of simulations.

Table A1. 95% confidence interval for different model diagnostics for parceling reject inference method.

Number of simulations	Model diagnostic			
	AUC	Accuracy	Sensitivity	(1-specificity)
50	(0.5236, 0.5242)	(0.6003, 0.6024)	(0.2574, 0.2661)	(0.1902, 0.1988)
100	(0.5235, 0.5240)	(0.6013, 0.6025)	(0.2564, 0.2624)	(0.1894, 0.1951)
250	(0.5237, 0.5241)	(0.6011, 0.6020)	(0.2589, 0.2629)	(0.1917, 0.1955)
500	(0.5237, 0.5240)	(0.6015, 0.6021)	(0.2588, 0.2613)	(0.1915, 0.1939)
1000	(0.5238, 0.5239)	(0.6013, 0.6018)	(0.2601, 0.2619)	(0.1928, 0.1946)

Table A2. 95% confidence interval for different model diagnostics for fuzzy reject inference method.

Number of simulations	Model diagnostic			
	AUC	Accuracy	Sensitivity	(1-specificity)
50	(0.5238, 0.5245)	(0.6003, 0.6021)	(0.2595, 0.2669)	(0.1918, 0.1992)
100	(0.5233, 0.5238)	(0.6008, 0.6022)	(0.2577, 0.2639)	(0.1907, 0.1966)
250	(0.5237, 0.5240)	(0.6008, 0.6017)	(0.2606, 0.2642)	(0.1933, 0.1967)
500	(0.5237, 0.5239)	(0.6012, 0.6018)	(0.2596, 0.2622)	(0.1924, 0.1949)
1000	(0.5237, 0.5239)	(0.6013, 0.6018)	(0.2601, 0.2619)	(0.1928, 0.1946)

Table A3. 95% confidence interval for different model diagnostics for the simple augmentation reject inference method.

Number of simulations	Model diagnostic			
	AUC	Accuracy	Sensitivity	(1-specificity)
50	(0.9231, 0.9232)	(0.8498, 0.8501)	(0.8494, 0.8497)	(0.0889, 0.0893)
100	(0.9231, 0.9232)	(0.8498, 0.8499)	(0.8494, 0.8496)	(0.0891, 0.0893)
250	(0.9231, 0.9232)	(0.8498, 0.8499)	(0.8494, 0.8495)	(0.0891, 0.0892)
500	(0.9231, 0.9232)	(0.8498, 0.8499)	(0.8494, 0.8495)	(0.0891, 0.0892)
1000	(0.9231, 0.9232)	(0.8498, 0.8499)	(0.8494, 0.8495)	(0.0891, 0.0892)