Review of Finance, 2021, 561–593
doi: 10.1093/rof/rfaa039
Advance Access Publication Date: 12 September 2020

Fintech for the Poor: Financial Intermediation Without Discrimination*

Prasanna Tantri

Indian School of Business

Abstract

I ask whether machine learning (ML) algorithms improve the efficiency in lending without compromising on equity in a credit environment where soft information dominates. I obtain loan application-level data from an Indian bank. To overcome the problem of the selective labels, I exploit the incentive-driven within officer difference in leniency within a calendar month. I find that the ML algorithm can lend 60% more at loan officers' delinquency rate or achieve a 33% lower delinquency rate at loan officers' approval rate. The efficiency is maintained even when the algorithm is explicitly prevented from discriminating against disadvantaged social classes.

JEL classification: G21; G23; G51

Keywords: Machine Learning, Discrimination

Received May 24, 2020; accepted October 13, 2020 by Editor Amiyatosh Purnanandam.

1. Introduction

The extant literature on fintech finds that the use of machine learning (ML) algorithms in credit decisions improves efficiency.[1] However, concerns have been expressed about the possibility of fintech applications worsening equity.[2] Almost all such studies examine developed economy settings where loan decision-making is mostly based on hard information.

* I thank the editor Amiyatosh Purnanandam, an anonymous editor, and an anonymous referee for their valuable feedback and guidance. I thank Shilpa Aggarwal, Shashwat Alok, Bhagwan Chowdhry, Aaditya Dar, Prachi Deuskar, Madhu Vishwanathan, and seminar participants at the Indian School of Business for their valuable feedback. I thank Aditya Murlidharan for excellent research assistance. I am grateful to the Digital Identity Research Initiative (DIRI) and Center for Analytical Finance, Indian School of Business for providing the data and the necessary financial assistance for this project. Any remaining errors are mine.

1 See Buchak *et al.* (2018); Mishkin and Strahan (1999); Fuster *et al.* (2019); Berg *et al.* (2019); Bianchi, Büchner, and Tamoni (2019); and Dobbie *et al.* (2018).

2 See Fuster *et al.* (2018) and Bartlett *et al.* (2019).

These findings cannot be directly applied to emerging economies, especially when the borrowers are poor, due to the lack of high-quality hard information and consequent reliance on soft information.[3] The question of equity is also important given the economic and social status of the poor in emerging economies. Therefore, the question of whether the use of ML algorithms to assist loan officers can improve credit market efficiency without increasing discrimination has considerable real-world implications.

I define efficiency in two ways: lower default rate at a loan application acceptance rate and higher loan application acceptance rate at a default rate. Equity is measured using the allocation of credit to vulnerable sections of society. Equity is defined in a relative sense by comparing the credit allocation to vulnerable sections with and without the ML algorithm.

The loan officers' production function uses both hard and soft information in assessing the creditworthiness of a borrower (Rajan, Seru, and Vig, 2015; Bursztyn *et al.*, 2019). However, the loan officer may have biases and cognitive limitations.[4] The machine uses only hard information and, hence, is likely to estimate creditworthiness with some error. The machine does not suffer from cognitive limitations and biases but lacks soft information. The key question I ask is whether the machine's superior processing of hard information is sufficient to offset the overall information advantage possessed by the loan officers. I also ask whether the machine achieves efficiency at the cost of equity.

I use loan application processing by loan officers of a listed commercial bank in India as the economic setting. The loan applicants are all individuals who belong to the lower strata of society in terms of income. Most of the loan applicants are either farmers or own informal enterprises such as retail shops, small garment factories, food processing units, among others. Most borrowers lack formal documents. Therefore, it is very difficult for banks to assess the creditworthiness of such borrowers. Not surprisingly, access to formal credit is a major challenge for this segment of the population (Cole, Sampson, and Zia, 2011; Cole et al., 2013; Kirschenmann, 2016). Existing lending is based on a combination of hard and soft information. I obtain information about 17,508 loan applications made by borrowers located in 196 villages/towns of 10 Indian states. Over and above the information disclosed in the loan application, I also have data about the eventual loan decision and repayments. I use the categorization of a loan as a loan in default as a measure of loan performance.

I use the gradient boosted decision trees (Friedman, 2001) by organizing the data at a borrower level. The outcome variable takes the value of 1 when the loan under consideration defaults, and 0 otherwise. Specifically, I use the XGboost algorithm (Chen and Guestrin, 2016). I use borrower and loan application characteristics available at the time of loan application processing as input variables. I use officer and location fixed effects in training. As a basic sanity check, I test and find that the risk scores generated by the ML algorithm are positively associated with the actual loan defaults. The AUC of the algorithm is 72%, which is reasonable (Berg et al., 2019).

As in Kleinberg et al. (2017) (KLLLM, henceforth), I am faced with selective labels problem: I can infer the counter-factual for cases approved by the loan officer but rejected by the machine but not for cases rejected by the loan officer but approved by the machine. Therefore, I can only reject approved cases based on ML algorithm's recommendation but

3 See Agarwal and Hauswald (2010); Skrastins and Vig (2018); Liberti and Mian (2008); Qian, Strahan, and Yan (2015); and Drucker and Puri (2008).

4 Keys *et al.* (2010); Cortés, Duchin, and Sosyura (2016); Berg, Puri, and Rocholl (2013); Griffin and Maturana (2016); Agarwal *et al.* (2014); and Chen, Moskowitz, and Shue (2016).

not approve cases rejected by the loan officers. The fact that the rejected and approved applications may differ in ways not observable by the algorithm aggravates this problem. KLLLM, who compare the bail or jail decisions made by judges in New York with the decisions based on predictions of an ML algorithm, exploit the difference in leniency levels of judges to deal with the selective labels problem. Based on the assumption that cases are randomly assigned to the judges, they compare the actual decisions of strict judges with the decision arrived at using an ML algorithm that uses the decision of the lenient judges as the base and jails additional defendants based on risk scores assigned by the ML algorithm until the release rate equals the release rate of strict judges. They call the process as contraction.

Using insights from the literature on loan officer incentives, I design an empirical strategy that exploits the possibility that a loan officer is likely to be more lenient at the end of the month when compared to the beginning. Many studies show that loan officers tend to relax credit standards as month-end approaches (Tzioumis and Gee, 2013; Cole, Kanz, and Klapper, 2015; Agarwal and Ben-David, 2018). I compare the relative leniency within an officer during the two halves of a month. I find that the officers approve 44% (82%) of loan applications in the first (last) 15 days of a month. The default rate is lower (higher) in the first (second) half. Therefore, I consider a loan officer "strict" during the first half of a month and "lenient" during the second half.

To test whether the ML algorithm does better than the officers, I use loans lent in the second half of the month as a starting point, as in KLLLM. I then ask whether by rejecting some loans made during the second half of the month, which are actually approved and hence whose outcomes are observable, is it possible to achieve a higher efficiency. I reject cases that are assigned the highest risk score by the ML algorithm. My strategy differs from that of KLLLM in two ways. First, the definition of strict and lenient is within an officer. Second, I define the objective function from the bank's point of view.

I then test whether the ML algorithm outperforms the loan officers. I start with the assumption that the bank's objective is to maximize the amount of loan lent at a given level of default or minimize default at a level of lending. The following example explains the procedure employed. Suppose the amount of loan lent in the first (last) 15 days is INR[5] 80 (100) and the amount in default in the first (last) 15 days is INR 10 (20). The thought experiment is to test what happens to the default rate of the last 15 days if the riskiest INR 20 worth of loans is denied. The riskiest loans are identified using the ML algorithm-generated risk scores. If the default amount after reducing the last 15 days' credit to INR 80 is less than INR 10, then it is possible to deduce that the ML algorithm is efficient. The same conclusion can be reached if the ML algorithm can lend more than INR 80 while keeping the default rate at INR 10. I find that the algorithm delivers a 33% reduction in default rate while keeping the loan amount lent unchanged. In an alternative scenario, it delivers a 60% increase in lending while keeping the default rate unchanged. The results hold when I redefine the approval rate based on (i) the proportion of the number of loan applications approved; (ii) the proportion of the value of loan applications approved.

The modified performance of the last 15 days serves as a valid counter-factual for the actual performance of the first 15 days for the following reasons: (i) I test and find that the loan application types do not differ, both observably and unobservably, between the two halves of a month; and (ii) since the same officer is classified as strict and lenient during

5 INR stands for Indian Rupee.

different times, screening ability and soft information remain the same between the two halves.

There could be a concern that the ML portfolio achieves a lower default rate at the cost of profitability. I examine and find that the interest rate charged on cases approved by the lenient officer and rejected by the ML algorithm is not significantly different from the interest rate charged on other loans. Therefore, it is unlikely that the algorithm achieves a lower default rate at the expense of profitability. Further, I directly compare the ML portfolio and the strict period portfolio based on profitability under some assumptions about the cost of funds and the proportion of recovery after default. I find that the algorithm outperforms even in terms of profits.

I then consider the question of equity. Here I ask whether the ML algorithm discriminates against borrowers from the disadvantaged sections of society. The Indian constitution identifies some specific social groups as historically disadvantaged. I first test whether borrowers belonging to such sections are more likely to be rejected by the ML algorithm. The answer is in the negative. Second, I conduct the contraction procedure by imposing an explicit condition that the proportion of selected borrowers from the disadvantaged sections cannot be lower than the same proportion during the strict period. The out-performance of the ML algorithm remains more or less unchanged. Third, the ML algorithm clears the fairness test (Saleiro et al., 2018) concerning lending to the disadvantaged sections of the society. Fourth, the algorithm continues to outperform even when information about the social category is not used for training. Given these results, it is reasonable to conclude that the ML algorithm achieves efficiency without compromising on equity.

One major concern with the empirical analysis is the limited number of observations from a big data perspective. While there are important studies that use similar sample sizes (Spadon et al., 2019; Taninaga et al., 2019), it is important to address this question. First, I test my results using randomly selected smaller subsamples. I find that the out-performance of the ML algorithm, both in terms of efficiency and equity, increases progressively with an increase in sample size. Therefore, an even higher sample size is more likely to strengthen my results. Second, I test the relative efficiency of the algorithm using various methods such as logistic regression, decision trees, random forests, and XG boost. I find a systematic improvement in the performance of the ML portfolio as I move from simpler to more complicated methods: XG boost turns out to be the most efficient.

This article talks to the growing literature that studies the impact of fintech on credit markets (Dobbie *et al.*, 2018; Fuster *et al.*, 2018; Berg *et al.*, 2019; Bianchi, Büchner, and Tamoni, 2019). I study the use of ML algorithms in an emerging market context where soft information dominates. I show that the ML algorithms have the potential to improve access to credit without harming equity. My findings are related to the literature that deals with issues related to access to formal finance (Ayyagari, Demirgüç-Kunt, and Maksimovic, 2010; Khandani, Kim, and Lo, 2010; Cole, Sampson, and Zia, 2011; Cole *et al.*, 2013; Sirignano, Sadhwani, and Giesecke, 2016; Demirguc-Kunt, Klapper, and Singer, 2017; Pierce and Snyder, 2018; Bartlett *et al.*, 2019). I show that efficient use of an ML algorithm can potentially bring down defaults and thereby reduce the cost of lending to the poor.

I contribute to the literature that examines the use of soft information in lending (Chakraborty and Hu, 2006; Drucker and Puri, 2008; Liberti and Mian, 2008; Agarwal and Hauswald, 2010; Hertzberg, Liberti, and Paravisini, 2010; Cheng, Raina, and Xiong, 2014; Iyer *et al.*, 2015; Beck, Ioannidou, and Schäfer, 2017; Du, Yu, and Yu, 2017; Fisman, Paravisini, and Vig, 2017; Skrastins and Vig, 2018). Soft information plays a dominant role in emerging markets especially when the prospective borrowers are poor and do not possess proper documents. To the best of my knowledge, the application of ML techniques to such credit environments is new to the literature.

Finally, I add to the growing economics and finance literature that employs ML techniques to aid human decision-making (Kleinberg *et al.*, 2015, 2017; Mullainathan and Obermeyer, 2017; Mullainathan and Spiess, 2017; Athey, 2018; Agrawal, Gans, and Goldfarb, 2019; Athey and Imbens, 2019; Gathergood *et al.*, 2019). The issue that I study falls into the category of problems where the outcome is generated by a human decision-maker, and hence, I see outcomes only for approved cases and not for unapproved cases—the so-called selective labels problem. Inspired by KLLLM, I develop a way of dealing with the problem of selective labels by exploiting the difference in leniency within a loan officer.

2. Data and Institutional Details

I obtain loan application-level data from a commercial bank in India. The bank operates through business correspondents (BCs) in rural areas. BCs are retail agents appointed by banks to provide basic banking services at locations that are not serviced by bank branches. BCs are typically individuals having a permanent base in a village. The BCs are allowed to perform basic banking functions such as opening bank accounts, collecting deposits, transfer of funds, facilitation of withdrawals, collection of small value loans, selling third-party products, and accepting loan documents to be sent to the bank for processing. The bank provides the necessary technological infrastructure for BCs to operate. BCs are paid fees on the number of successful transactions. Since it is difficult for the banks to deal with the large number of atomistic BCs, they work through BC aggregators, who recruit and manage BCs on behalf of banks. I obtain loan application-level data from one such BC aggregator who covers ten large states of India for the bank. The dataset covers 196 BCs located in as many different towns and villages.

For loan applications, BCs serve as loan application collection points. A designated loan officer from the bank processes and disposes the loan applications. The loan officer physically visits the BC location and interacts with the loan applicants. At times, the loan officers visit the place of work or farm of the loan applicant and also ask for references. Therefore, it is reasonable to infer that the loan officers collect both hard and soft information about the loan applicants. Besides, they also obtain inputs from the BCs.

The dataset combines information about loan applications and loan-related transactions. I have applicant-level information such as occupation, age, gender, individual and family income, postal address, among others. I also have information such as the amount applied for, the collateral, the date of application, and others, which are at the loan application level. I present the summary information about the dataset in Table I. Out of 17,508 loan applications in the data, 11,106 are accepted. Therefore, I have loan transaction data for 11,106 borrowers. The dataset spans a period between January 2017 and February 2020. I have an average of 12 monthly repayment observations for every loan.

The data are at a borrower-month level. I have information about whether a borrower has ever defaulted on the loan under consideration. I have information about the monthly installment due date, the installment amount, overdue amount if any, total loan amount, and the outstanding loan amount as on the due date of every installment. The average loan

Table I. Sample construction and summary

In this table, I report details about the sample used and compare approved and rejected cases in terms of loan application characteristics.

	Data Summary			
	Full sample	Approved	Rejected	p-value
Sample size	17,508	11,106	6,252	
Approval rate (%)	63	100	0	
Loan information				
States	10			
Towns/villages	196			
Number of bank officers	167			
Time period	January 2017–February 2020			
Average repayment data covered	12 months			
Average loan amount lent (INR)	153,650			
Average interest rate	13.70			
Applicant characteristics				
Log of total assets	11.98	12.11	11.73	0.000
Log of annual income	13.08	13.12	13.01	0.000
Occupation				
Business	0.78	0.79	0.76	0.000
Farmer	0.18	0.17	0.20	0.000
Caste				
General	0.61	0.58	0.65	0.000
Other Backward Class	0.32	0.34	0.27	0.000
SC/ST	0.08	0.07	0.08	0.937
Nominee relationship				
Spouse	0.69	0.71	0.67	0.000
Parents	0.15	0.14	0.16	0.001
Others	0.16	0.15	0.17	0.000
Gender (female)	0.27	0.26	0.28	0.007
Customer age	36.32	36.09	36.72	0.259
Nominee age	42.07	43.73	41.43	0.265
Marital status	0.85	0.86	0.84	0.000

amount is INR 153,650 which is close to USD 2,000. The average interest rate charged is close to 13.7%, which is close to 6.5% above the risk-free rate. The loans are lent purely on commercial terms without any subsidy from the government.[6]

I then compare the loan application characteristics of approved and rejected cases. I find that approved and rejected cases systematically differ on observable characteristics such as the value of assets and income. I then present the composition of accepted and rejected cases in terms of occupation, social groups, and nominee details. The approved borrowers are less likely to be farmers when compared to the rejected borrowers. The Indian

6 The interest rate is in the ballpark of the rate charged by other lenders to small borrowers. A comparison of interest rates on loans to small borrowers is presented here https://www.bankbazaar.com/personal-loan/sme-loans.html.

constitution classifies the population into four social groups. The term scheduled caste (SC) refers to people who historically formed the lowest strata of society. They were considered untouchables. Scheduled tribes (ST) refers to indigenous people living in some specified tribal areas. Other backward classes refer to those who were above SCs in the social hierarchy but significantly poor. The remaining population is considered general (Murlidharan, Nishesh, and Tantri, 2020). The loan officers are more (less) likely to lend to other backward classes (general). There is no difference in the proportion of scheduled castes and tribes (SCSTs) between the rejected and accepted pool. The applications where the nominee is the spouse are more likely to be accepted. There seems to be a preference for married loan applicants and men. I do not find economically meaningful differences between the two groups in terms of either the applicant age or the nominee age. The table clearly shows that the accepted and rejected cases systematically differ in terms of many observable characteristics. Table I presents the summary for some selective variables. The full list of variables used for training the algorithm is presented in Table A.1 of the Online Appendix. In addition, I use officer and location fixed effects.

3. Hypothesis

3.1 True Default Function

The true underlying default function with Y_i as the outcome variable representing loan performance, X_i as observables and S_i as unobservables is as follows[7]:

$$f(X_i, S_i) = E(Y_i|X_i, S_i) \tag{1}$$

The above equation shows that the loan performance depends both on soft and hard information. As discussed in the introduction, hard information refers to information that can be verified by third parties and includes information such as past performance of loans, the address of the borrower, the family composition and size, occupation, and others. Soft information cannot be easily verified by third parties and, hence, cannot be easily transmitted. A borrower's character-related information falls into this category. The third category includes information which is by nature hard information but, given the institutional context of emerging economies, it is hard to verify them. Information about income falls into this category. Officers acquire soft information in various ways ranging from using their social networks to participating in community rituals (Bhowal, Subramanian, and Tantri, 2020).

3.2 The ML Algorithm

$$g(X_i) = E(Y_i|X_i) + \varepsilon_{i,g} \tag{2}$$

where $\varepsilon_{i,g}$ is the estimation error because this function has an approximation error.

The algorithm uses only hard information. Therefore, the information content of the soft information part will appear in the error. It is important to note that since the machine is not biased and is unaffected by behavioral issues, it is likely to estimate the relationship between X_i and Y_i efficiently and the error $\varepsilon_{i,g}$ is most likely due to the lack of soft information.

7 I thank the anonymous referee for suggesting a way of formally describing the hypothesis. I have borrowed material from the referee's letter.

3.3 The Loan Officer Production Function

$$h(X_i, S_i) = f(X_i, S_i) + \varepsilon_{i,h} \tag{3}$$

where $\varepsilon_{i,h}$ captures several things, including approximation error and potential biases.

The loan officers can potentially use both hard and soft information that they gather. It should be noted that it is unlikely that the loan officers will have access to all kinds of soft information required to predict loan performance. It is also possible that the loan officers' information is dated. Moreover, the loan officers could be biased, inconsistent, cognitively bounded due to resource constraints, and suffer from other behavioral limitations. Therefore, they may not be able to use the available information efficiently. The error term $\varepsilon_{i,h}$ accounts for such biases in using hard information.

3.4 The Key Questions

I ask three key questions within this framework. The first question is whether the out of sample prediction error of h is greater than that of g. If the answer is no, then the officer outperforms the machine. If the answer is yes, I ask the second question of whether the advantage gained by the loan officer due to additional soft information offsets the loss due to increased error. The value of soft information can be defined as

$$Z_S(S, X) = E(Y_i|X_i, S_i) - E(Y_i|X_i). \tag{4}$$

The second question asked is whether or not the loan officers' advantage due to $Z_S(S, X)$ is sufficient to offset the loss due to incremental error caused by bias and cognitive limitations.

Finally, I also examine the impact on equity. I view equity in a relative sense by comparing the proportion of credit that flows to the vulnerable sections under the loan officer and the machine technologies. I operationalize this comparison by imposing a constraint on Equation (2). The constraint ensures that the proportion of credit given to vulnerable sections by the machine is not lower than that given by the officers. In other words, if P^h (P^g) is the proportion of loans granted to vulnerable sections by the officer (machine), I impose a constraint that $P^h = P^g$ in Equation (2) to arrive at equity. I test whether the machine outperforms in terms of efficiency even after imposing the above constraint.

4. The ML Exercise

For the ML exercise, I organize the data at a borrower level. I can only use 11,106 observations relating to loans as I have outcomes for only those cases where the loans have been sanctioned. Therefore, out of 17,508 applications, 6,252 rejected cases are not considered for training. I use the information provided in loan applications as input variables and a dummy variable representing whether the loan under consideration has defaulted or not as the outcome variable. I then apply a supervised ML algorithm. Typically, ML algorithms follow an 80:20 rule, where the algorithm is trained on 80% of the data and tested on the remaining 20%. The data are randomly selected. Accordingly, I use 80% of the overall sample of loans for training and the remaining for out of sample testing. Thus, my training sample has 8,885 and the testing sample has 2,221 borrower level observations. I use more than 60 variables, and loan officer and location fixed effects as inputs for training.

5. Empirical Strategy

5.1 The Algorithm

I use the XGBoost algorithm, which belongs to the gradient boosted decision trees class. As in the case of any ML algorithm, the objective is to estimate a functional relationship between inputs and outputs. The inputs in my case are various loan application characteristics and dummy variables representing fixed effects. The outcome variable that is fed into the algorithm as a part of supervised learning is a dummy variable that takes the value of 1 if the loan under consideration defaults, and 0 otherwise. The predicted variable can be interpreted as the probability of loan default.

The decision trees work by splitting the input variables into various binary classes. For example, the first split might be on whether the loan applicant is a farmer or not. The subsequent splits may be based on other variables such as income, age, and others. To avoid overfitting, ML algorithms create a small test sample within the training sample and keep checking the performance of predictions out of sample. The adjustment done to the tree based on this within training set out of sample test is known as "regularization" and the process itself is known as "tuning."

I use learning rate and maximum depth weight as tuning parameters over which the model cross validates. The learning rate parameter is set to control the weights of new trees added to the model. I set the learning rate at 0.1. The n estimator is set at 100. It specifies the number of trees to be generated. The maximum depth parameter is set at 6. I have kept it low to reduce complexity and overfitting. I provide details about the model parameters in Table A.2 of the Online Appendix.

A note on gradient boosting (Chen and Guestrin, 2016) is in order. Boosting is a process of improving the predictions made by the algorithm in the initial rounds. It calculates the loss function and attempts to minimize the loss by giving higher weights to observations that are most wrongly predicted. The algorithm creates new models in this manner and stops when no more improvement in prediction accuracy is possible. The gradient boosting technique uses a gradient descent algorithm to minimize the loss when adding new models. To make the model robust, I use five-fold cross-validation. A model created using a part of the training dataset is tuned on the other four parts and the cycle gets repeated until no more improvements are possible. The process leaves me with predicted risk or what can be loosely interpreted as the probability of default for every observation.

5.2 Evaluation of Algorithm Accuracy

5.2.a. AUC–ROC curve

Before proceeding to the comparison between the algorithm and the officers, I test the intrinsic quality of predictions. This step is typical of any ML exercise. I use the commonly used metric known as the area under the ROC curve.[8] The ROC curve is plotted by using two ratios. The first (second) ratio is the ratio of true positive among all actual positives (false positives among all actual negatives). The ratio is plotted at various thresholds for classification of probability score as success or failure. For example, assume that a 60% risk score and above means success or default in this case. Suppose at this threshold, 70% of the actual defaulters are classified as defaulters and 0% of the actual non-defaulters are classified as defaulters, then the outcome is plotted as 0 on the *y*-axis and 70% on the *x*-

8 ROC stands for receiver operating characteristic.

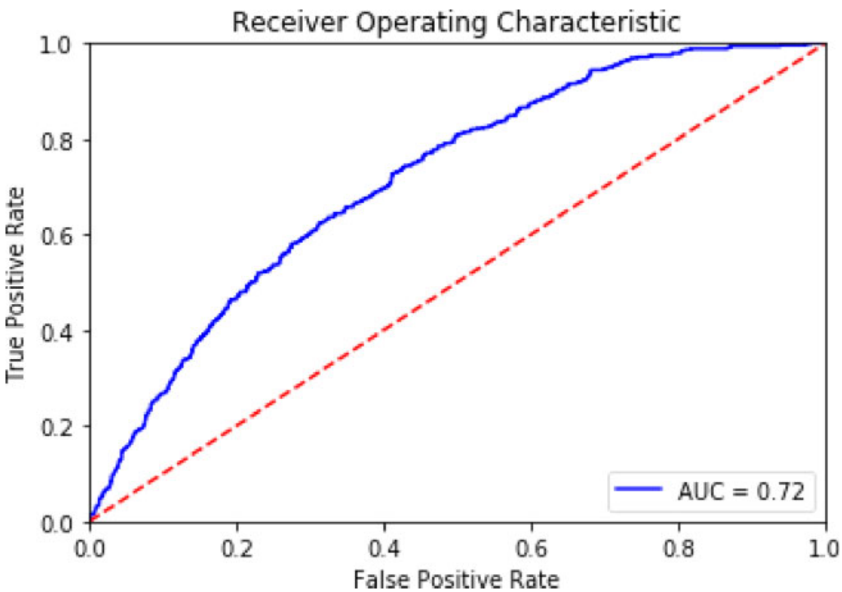

Figure 1. AUC–ROC curve. This figure depicts the AUC–ROC curve.

axis. Suppose, at a threshold of 50%, 80% of the actual defaulters are classified as defaulters and 10% of the actual non-defaulters are classified as defaulters, then the outcome is plotted as 10% on the *y*-axis and 80% on the *x*-axis. The curve thus formed is known as the ROC curve. The area under the ROC curve is the metric used to test the accuracy of the model. It is easy to see that a random guess is likely to lead to a 45 degree ROC curve and an AUC of 50%. I depict the AUC–ROC curve in Figure 1. The AUC works out to close to 72% which is above the conventional thresholds (Berg *et al.*, 2019).[9]

5.2.b. Actual versus predicted

To evaluate the model, I compare the actual and the predicted. Figure 2 presents the comparison. The calibration curve (Solem, 2012) plots the ML algorithm-generated risk scores in the horizontal axis and the predicted default risk in the vertical axis. The test is out of sample. As shown in the figure, a higher predicted risk translates into a higher default. The result shows that the ML algorithm works in the right direction.

I then test whether the loan officer approval rate is related to the risk scores generated by the machine. If the risk scores generated by the machine and the risk assessment by officers are directionally similar, I expect to find a negative relationship between the machine-generated risk score and the loan approval rate. I plot a calibration curve that plots the association between risk scores predicted by the machine and the loan officer approval rate in Figure 3. The risk scores are plotted on the horizontal axis and the approval rates on the vertical axis. I do not find a clear negative relationship between the risk scores and the acceptance rate. The loan officers seem to have a tough time in identifying high-risk

9 In addition, I evaluate the ML algorithm based on other measures such as recall, precision, and F1 score. I find that the algorithm performs reasonably well. I provide a detailed description of the tests in Section A of the Online Appendix and depict the results in Figures A.1, A.2, and A.3 of the Online Appendix.

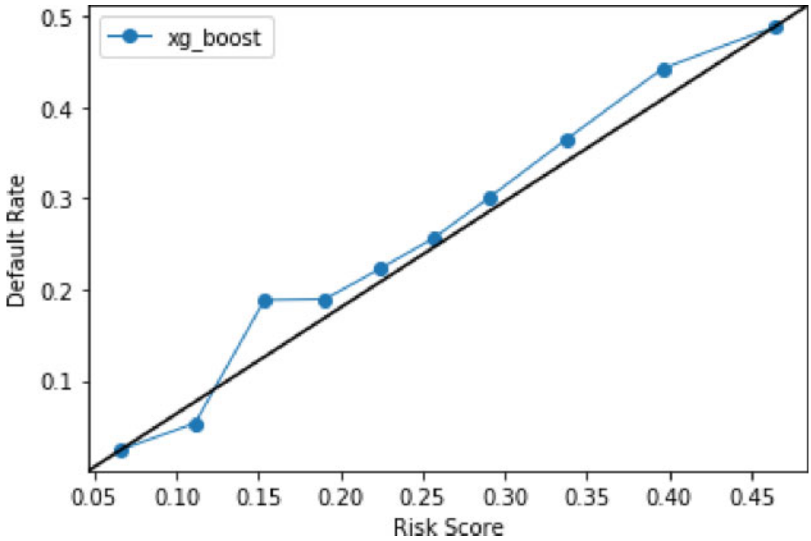

Figure 2. Risk score and actual default. This calibration curve plots the predicted risk scores on the horizontal axis and the default rate on the vertical axis.

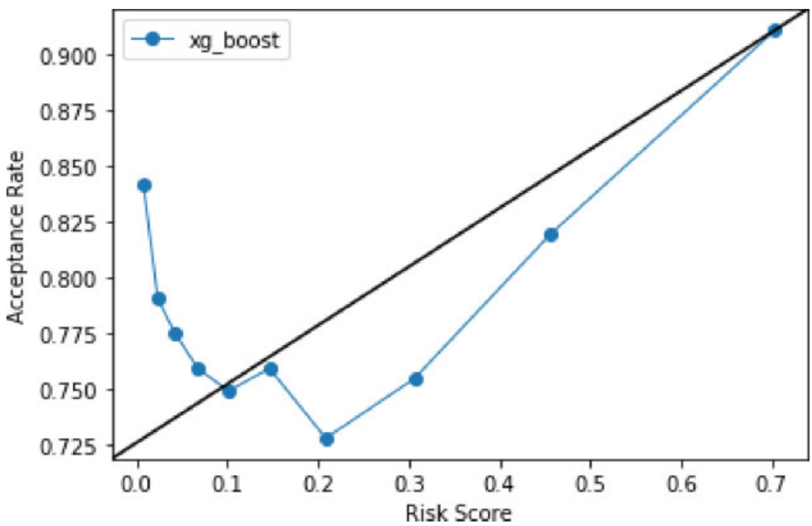

Figure 3. Risk score and loan officers' selection. This calibration curve plots the predicted risk scores on the horizontal axis and the acceptance rates on the vertical axis.

borrowers. The approval rates are high for both low-risk and high-risk borrowers. The risk model that loan officers apply differs significantly from the assessment of the ML program.

5.3 Comparing Loan Decisions

After training and testing the ML algorithm, the next logical step is to compare the performance of the algorithm with that of the loan officers. At the outset, it is important to clearly define the objective function to be optimized. I look at the problem from the bank's point of view and not from that of the loan officers. In other words, I ask whether given the current incentive structure of the loan officers, can ML algorithms outperform the loan officers in furthering the objectives of the bank.

I start with the scenario where the bank's objective is to minimize defaults at a given level of loan amount lent. I then consider the possibility that the bank may want to minimize defaults subject to maintaining a target loan approval rate. The idea here is to lend to the maximum number of borrowers. In both the above cases, I also consider the situation where the bank wants to maximize lending at a given level of defaults. I then look at the impact on profitability. Finally, I consider the possibility that the bank may have some social objectives such as maintaining the amount lent to certain sections of society.

5.3.a. Measuring defaults

A loan is considered overdue if the monthly installment amount is not repaid on or before the due date. Using the overdue status, I define default in five ways. The first definition considers a loan to be in default only if the status of the loan is overdue as at the end of the last installment in the data. To account for possible right censoring, I further modify the definition of default and create additional measures. The second definition of default leaves out all loans whose data coverage is less than twelve installments from the sample and defines a loan to be in default if the loan is in overdue status as at the end of the twelfth month. The third and the fourth definitions consider nine and six months, respectively, in place of twelve months used in the second definition. Finally, my fifth measure considers a loan to be in default if the loan attains overdue status any time during the sample period. The definition is most stringent in identifying non-defaulters and, hence, is useful in identifying the algorithm's ability to identify high-quality borrowers.

5.3.b. The selective labels problem

Note that for comparison purposes, I need to know the actual outcomes under both the systems of loan decision-making. It is possible to use the actual outcomes for cases where loan officers and the ML algorithm agree. The situation becomes complicated when the two ways of loan application appraisals do not agree. The ML decision may differ from that of the loan officers in two ways. The machine may recommend rejection of cases that are accepted by the loan officers. Conversely, the machine may recommend acceptance of cases that are rejected by the loan officers. The first category of disagreements does not pose a challenge as attaching "outcomes" for cases rejected by the machine is straightforward: rejected loan applications do not default. The real challenge comes from the second set of disagreements. It is extremely hard to know what the outcome of cases that are rejected by loan officers but accepted by the ML algorithm would have been. This challenge is known as the "selective labels" problem (Guidotti *et al.*, 2019).

One possible solution is imputation using the actual performance of borrowers with similar risk scores as those who have missing labels. Such imputation assumes that the applicants accepted and rejected by the officers are similar even in terms of unobservable characteristics. Consider a situation where the loan officer rejects a large number of loan applications of middle-aged male applicants who are chronically alcoholic because he/she knows that such borrowers are highly likely to default. This information is not present in the application form. The loan officers get this information from their social networks (Fisman, Paravisini, and Vig, 2017). Such a selection by loan officers makes the accepted and rejected middle-aged men systematically different. The machine on the other hand learns only based on accepted cases. Since it sees only the good among the middle-aged men, it assigns low-risk scores to such people. Therefore, a policy of imputing outcomes based on risk scores of accepted cases is likely to result in machine accepted cases under-performing out of sample.

5.4 Contraction of Applications

I attempt to compare the decisions of the officers with those of the ML algorithm. I use the well-documented fact that loan officers tend to be lenient as the month-end approaches because they have lending targets to be fulfilled (Tzioumis and Gee, 2013; Behr *et al.*, 2014; Agarwal and Ben-David, 2018). My empirical strategy of comparing within officers is based on insights from the literature that examines the impact of nonlinear incentives. A system where employees are asked to complete a minimum fixed quota within a time frame to be eligible for incentive payments is a typical example of nonlinear incentives. The loan officers in my setting face monthly lending targets. De-jure, there is no change in the incentive structure within a month.

Theoretically, Holmstrom and Milgrom (1987) show that within a framework of repeated moral hazard and nonlinear contracts, an agent's incentives change with time and also cumulative output achieved. Tzioumis and Gee (2013) show that when faced with nonlinear incentives in the form of month-end targets, mortgage originators in the USA increase the number of originations at the cost of quality as the month-end approaches. Others have found similar results in diverse contexts such as software sales, manufacturing, recruitment of navy personal, accounting, and others (Healy, 1984; Asch, 1990; Oyer, 1998; Agarwal and Wang, 2009; Larkin, 2014). One rational reason from the agents' point of view could be the possibility of the principal revising the target upwards if agents meet the target early—the so-called ratchet effect (Freixas, Guesnerie, and Tirole, 1985). Therefore, agents have reasons to go slow at the beginning of the month. Alternatively, present-biased preferences of employees leading to procrastination in the early part of a month is a plausible behavioral explanation (Tzioumis and Gee, 2013). In line with the extant findings, I find that the loan officers in my setting increase the amount of loans lent towards the end of the month by adopting liberal screening standards.

I first test the hypothesis that officers tend to be more lenient towards the end of the month. To this end, I compare the acceptance rate and default rate between the first and second half of a month. I use the entire sample for this purpose. I present the results in Table II. A lenient regime is likely to be characterized by high acceptance and default rates. I call the first 15 days "strict" period and the last 15 days "lenient" period. I present the results for the entire sample, the lenient regime, and the strict regime.

I define the acceptance rate in three ways. Definition 1 represents the ratio between the number of loans granted and total loan applications. I find that the ratio is 44% during the strict period and 82% during the lenient period. I then define the ratio based on the relationship between the value of loans granted and applied. When I use the second definition, I find that the ratio is 39% during the strict period and 76% during the lenient period. Besides, I look at the loan amount sanctioned. I find that the total loan amount sanctioned during the strict period is INR 600 million, whereas the same during the lenient period is INR 1,110 million. Finally, I look at the default rates as well. Expectedly, I find that the default rate is higher during the lenient period when compared to the strict period. For instance, under definition 1, the default rate during the strict period is 25%,

Table II. Comparison between strict and lenient periods based on approval rates and loan performance

In this table, I compare the strict and lenient periods in terms of acceptance rate and loan performance. The status of a loan with respect to its last installment in the data is considered to measure loan performance.

Comparison between strict and lenient period in full data

Methods	All		Strict		Lenient	
	Volume	Default	Volume	Default	Volume	Default
Loan approval rate	0.63	0.24	0.44	0.21	0.82	0.25
Loan amount proportion	0.57	0.24	0.39	0.22	0.76	0.25
Loan amount lent (INR million)	1,710	408	600	131	1,110	277

whereas the same during the lenient period is close to 21%. The result is directionally similar even when I use the other two definitions. For this summary, I define default as a loan being overdue as at the due date for the last installment in the sample (definition 1 of default).

The contraction procedure works as follows. I start with the actual portfolio of the lenient period. I ask what will happen to the default rate of the lenient period if the acceptance rate is made the same as that of the strict period by rejecting the riskiest cases identified by the ML algorithm. Notice that I reject loans already lent by the officers. It is possible to answer the counterfactual question of what will happen to loan performance if a loan lent by the officer is rejected by the ML algorithm: such loans will not default. Therefore, I do not face the problem of selective labels here. I reject applications by arranging them in the descending order of risk, as identified by the machine, and start rejecting from the riskiest cases until I hit the target loan approval rate.

6. Results

6.1 The Actual Comparison Using Contraction

I depict the contraction procedure graphically in Figure 4.[10] The horizontal axis depicts the acceptance rate and the vertical axis depicts the default rate. In this figure, I define acceptance rate as the proportion of loan applications approved in terms of numbers. The point X1 (X2) denotes the acceptance and the default rates of the lenient (strict) period. The curve starting from X1 denotes the various default rate-acceptance rate combinations achieved by the machine by using the contraction procedure. Notice that both X1 and X2 are above the curve.

Some findings are noteworthy here. First, X3 (X4) denotes the default (acceptance) rate the machine would achieve at the acceptance (default) rate of the strict period. The machine outperforms on both fronts. Second, a comparison between the curve from X1 to X3 and the line X1 to X2 tells that the trade-off the machine achieves between a reduction in acceptance rate and the default rate is superior to a linear trade-off. In other words, the machine rejects fewer applications to achieve a targeted default rate than a linear trade-off

10 I use the first definition of default as defined in Section 5.3.1.

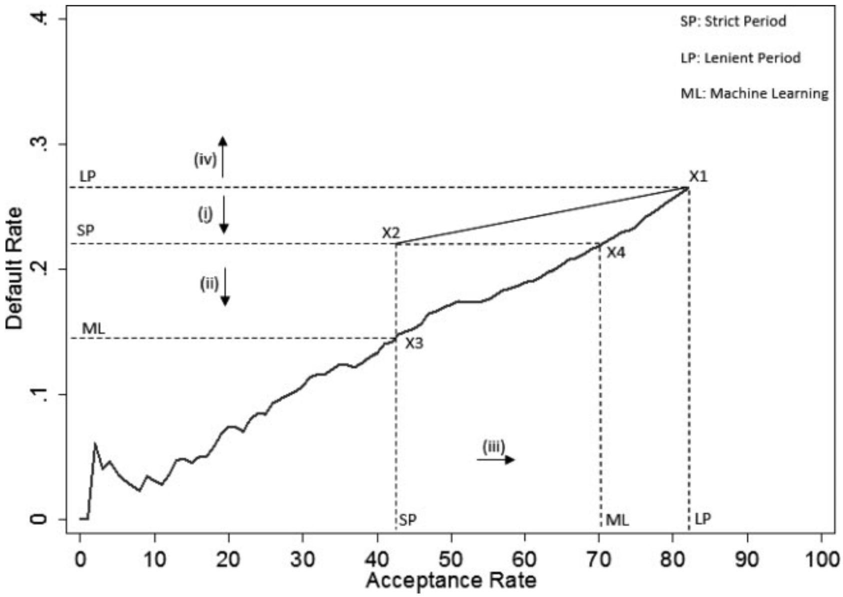

Figure 4. Comparison between the algorithm and the strict period loan approval rate. This figure plots the acceptance rate on the horizontal axis and the default rate on the vertical axis. The point X1 (X2) denotes the acceptance and the default rates of the lenient (strict) period. The curve starting from X1 denotes the various default rate-acceptance rate combinations achieved by the machine by using the contraction procedure.

would have achieved. The figure also implies that the machine adds a lower default rate to achieve a higher acceptance rate than a linear trade-off would have achieved. Finally, notice the four arrows. Arrow (i) denotes the difference between the strict and lenient periods in terms of default rate. Arrow (ii) shows the out-performance of the machine over the strict period in terms of default rate keeping the acceptance rate fixed. Arrow (iii) shows the out-performance of the machine over the strict period in terms of acceptance rate keeping the default rate fixed. Arrow (iv) denotes applications rejected during the lenient period. I leave such applications out of the exercise due to the problem of the selective labels. In Figures A.4 and A.5 of the Online Appendix, I draw similar figures using the other two definitions of loan acceptance rate explained in Section 5.4.

Note that my focus is on points X3 and X4. Also, the figure is based on one definition of default. To understand the economic magnitude of the outperformance and its sensitivity to the definition of default used, I tabulate the results using all three definitions of loan acceptance and all five definitions of loan performance in Table III. As explained in Section 2, the table is based on the test sample of 2,221 borrowers used for the out of sample test. In Panel A, I present the results of the exercise where I start from the lenient portfolio and create an artificial machine portfolio by bringing down the acceptance rate to the level of the strict portfolio (Point X3 in Figure 4). I drop the riskiest cases as identified by the ML algorithm. Therefore, the comparison is between the actual strict portfolio and the machine portfolio created using the lenient portfolio and following the contraction procedure. I present the comparison using five definitions of default as described in Section 5.3.1 and in the

Table III. Comparison based on loan approval rate and default using contraction

In this table, I present the results of the contraction procedure. The first (second) half of the month is considered the strict (lenient) period. In Panel A, I create an ML portfolio starting from the lenient portfolio and eliminating cases in the descending order of risk scores as assigned by the ML algorithm. I stop when the acceptance rate of the machine portfolio equals the acceptance rate of the strict portfolio. I report the reduction in default rate achieved by the machine portfolio over the strict portfolio. Similarly, in Panel B, I create a machine portfolio by equating the default rates and report the incremental acceptance rate. In column 1, a loan is considered to be in default if it is overdue with respect to its last installment in the data. In column 2, a loan is considered to be in default if it is in default at the end of the twelfth month. Loans having coverage for less than twelve installments are excluded from the training sample. In column 3 (4), I leave out loans that entered in the last nine (six) months and call a loan in overdue status as at the end of nine (six) months as a loan in default. Finally, in column 5, a loan is considered to be in default if it defaults at any time during the sample period.

Panel A

Default classification	Last installment	12th installment	9th installment	6th installment	Ever default
Loan approval rate	0.33	0.41	0.43	0.19	0.37
Loan amount proportion	0.40	0.39	0.36	0.23	0.39
Loan amount lent (INR million)	0.35	0.17	0.24	0.05	0.39

Panel B

Default classification	Last installment	12th installment	9th installment	6th installment	Ever default
Loan approval rate	0.60	0.59	0.61	0.11	0.73
Loan amount proportion	0.74	0.55	0.54	0.29	0.78
Loan amount lent (INR million)	0.28	0.12	0.11	0.03	0.30

same order (in different columns), and the three definitions of loan acceptance rate (in rows) as described in Section 5.4.

Consider column 1 and row 1. Here, I define default based on the status of the last installment in the data and define leniency by using the ratio between the number of applications approved and applied. I find that the machine portfolio's default rate is 33% lower than the default rate of the strict portfolio.[11] The outperformance is 40% and 35% when I use the other two definitions of loan acceptance. The difference appears economically meaningful. The result holds across definitions of loan acceptance rate and default. In Panel B, I conduct the contraction procedure by equating the default rates and testing the implication on the loan approval rate (Point X4 of Figure 4). The organization of the Panel mimics the organization of Panel A. As shown in column 1 and row 1 of the table, the acceptance rate of the machine is higher by 60% when compared to the strict portfolio. The outperformance is 74% and 28% when I use the other two definitions of loan acceptance. As in Panel A, the result holds across definitions of loan acceptance rate and default.

11 The difference in default rate between the strict period portfolio and the machine created portfolio is expressed as a percentage of default rate of strict period portfolio.

The results clearly show that the use of the ML algorithm improves the overall efficiency of lending and helps increase lending without an increase in default rates. In terms of the framework described in Section 3, it appears that the advantage the loan officers gain due to soft information ($Z_S(S, X)$ in Equation (4)) is not sufficient to offset their incremental error over the machine (the difference between h and g). Therefore, machine portfolio can achieve a significantly lower (higher) default (acceptance) rate at the same acceptance (default) rate of the officer.

6.1.a. Robustness using the hazard model

As further robustness, I compare the strict period portfolio with the machine aided artificial portfolio carved out of the lenient period using the Cox proportional hazard model. Organizing the data at a borrower level, I identify the spell for each borrower. For non-defaulting loans that have completed their tenure, the spell is the loan tenure. For non-defaulting loans that did not, I use the number of months of coverage in the dataset as the spell. For defaulting loans, the spell ends in the month of default. I estimate the hazard model using default as the final event. I present the results in Table IV. I use all three definitions of acceptance rate as defined in Section 5.4. Treatment is a dummy variable that takes the value of 1 for the machine portfolio, and 0 otherwise. I present both the hazard rate and the coefficients in alternate columns. In line with the results presented in Table III, I find that the default rate of the machine portfolio is significantly lower.[12]

6.2 Possible Concerns

6.2.a. Observable and unobservable differences

The difference in the leniency of loan officers between the first and second half of the month is crucial for identification. Implicit is the assumption that the loan applications that enter the system during the two phases are not systematically different, both in terms of observable and unobservable characteristics. I first test the observable part here. Table V presents a comparison between strict and lenient period loan applications in terms of observable characteristics such as interest rate charged, assets, income, loan amount applied, occupation, composition of social groups, nominee relationship, number of dependents, and the borrower and nominee age. I do not observe significant differences between the two groups.

I then test for unobservable differences. I use the ML algorithm to do so. I train the algorithm using the loans lent during the lenient period and test them on loans lent during the strict period. If the algorithm trained on lenient period loans makes a reasonable prediction for the strict period, then it is likely that the relationship between inputs and outputs is similar in both the periods. Given that the comparison is within an officer, the screening ability of the officer can be assumed to be constant. The differences between the two halves of the month are more likely to be driven by loan officer incentives than any other systematic factor. Therefore, the ML algorithm that is trained using the lenient period loans and tested using the strict period loans can test for unobservable differences.

I find that the ML algorithm so developed has an AUC score of 74%, which is not only high but also comparable to the AUC score of my main test. This shows that the algorithm

12 The number of observations is 1,445 although my test sample has 2,221 observations. This is because, as noted above, I include only those loans from the lenient portfolio that are not rejected by the ML algorithm.

Table IV. Comparison based on hazard model

In this table, I present the results of a Cox proportional hazard model. I organize the data at a borrower level. I define a spell as lower of the entire tenure of the loan and the number of monthly installments covered in the data for non-defaulting loans. Default ends the spell for defaulting loans. I use the first definition of default defined in Table III as an outcome variable for the hazard test. I present the hazard rate in columns 1, 3, and 5 and the respective coefficients in columns 2, 4, and 6. I use the three definitions of acceptance rate as indicated in the table. Treatment is a dummy variable that takes the value of 1 for the ML portfolio arrived at using the contraction procedure. The strict period portfolio is in the base. ***, **, and * represent significance at the 1%, 5%, and 10% levels, respectively.

Dependent variable	Acceptance rate		Loan proportion		Loan amount	
	Hazard rate	Coefficient	Hazard rate	Coefficient	Hazard rate	Coefficient
Treatment	0.81**	−0.20**	0.80**	−0.22**	0.80**	−0.22**
	−(−2.12)	(−2.12)	(−2.26)	(−2.26)	(−2.26)	(−2.26)
Spell	0.95***	−0.05***	0.95***	−0.05***	0.95***	−0.05***
	(−6.78)	(−6.78)	(−6.87)	(−6.87)	(−6.87)	(−6.87)
Observations	1,445	1,445	1,413	1,413	1,413	1,413

Table V. Comparison between strict and lenient regimes based on borrower characteristics

In this table, I compare the lenient and strict periods based on observable characteristics.

Comparison: strict and lenient period				
	All	Strict	Lenient	p-value
Log of total assets	11.98	11.97	11.99	0.531
Log of annual income	13.08	13.09	13.07	0.202
Annual interest rate	0.14	0.14	0.14	0.526
Occupation				
Business	0.78	0.77	0.78	0.091
Farmer	0.18	0.18	0.18	0.790
Caste				
General	0.61	0.61	0.60	0.173
Other backward class	0.32	0.32	0.32	0.540
SC/ST	0.08	0.07	0.08	0.149
Nominee relationship				
Spouse	0.69	0.69	0.70	0.229
Parents	0.15	0.14	0.15	0.634
Gender (female)	0.27	0.26	0.27	0.195
Customer age	36.33	36.60	36.05	0.298
Nominee age	42.11	42.01	42.21	0.916
Marital status	0.85	0.85	0.85	0.194

trained on lenient loans does a good job of predicting outcomes on the sample of strict loans. Further, I plot the above predictions against actuals in Figure A.6 of the Online Appendix. The calibration curve plots the risk factors on the horizontal axis and the actual default rates on the vertical axis. As can be seen, the line is close to 45 degrees. The finding shows that the actuals and the predicted are close. In other words, the relationship between the inputs and outputs is similar during both periods. Given the absence of difference in screening ability, the above result can be attributed to the lack of unobservable differences.

6.2.b. The sample size and the method

ML algorithms work well when the data size is huge. Admittedly, I have only close to 17,000 loan applications and close to 11,000 borrowers. Therefore, it is reasonable for a reader to be skeptical about my inferences. Specifically, a reader may worry that (i) the results may not hold in larger samples (ii) the sample size is too less for the use of XG boost technique. In Section B of the Online Appendix, I describe in detail that my results improve with an increase in sample size from 5,000 to the full sample. I select subsamples randomly for this purpose. The results are presented in Panel A of Table A.4 of the Online Appendix. Similarly, I also show that the XG boost algorithm performs better than other simpler algorithms. I present these results in Panel B of Table A.4.

6.2.c. Directed lending program

The Government mandates all banks including private-sector banks to allocate 40% of their credit to the "priority sector." Several categories of loans such as agricultural loans, loans to small and medium enterprises, small ticket housing loans, and others (Banerjee and Duflo, 2014) are considered "priority sectors." Agriculture has a sub-quota of 18%. Although directed lending exists in many parts of the world (Bhue, Prabhala, and Tantri, 2019), it is important to test whether my results are due to this feature. It is crucial to note that I verify and find that the bank met the priority sector requirement comfortably during my sample period. Besides, as shown in Table I, the bank charges an interest spread of more than 6%. Prima facie, the loans do not appear to be heavily subsidized. Admittedly, I cannot directly identify whether a particular loan is a priority sector loan or not. I have information about the occupation of the borrower and not always about the purpose of the loan. For example, I cannot distinguish between a personal loan and an agricultural loan borrowed by a farmer. Only the latter is considered a part of the priority sector. Similarly to identify whether a business (housing) loan is a part of the priority sector, I lack information about the investment made in plant and machinery (value of the house).

Within these limitations, I design a test to examine whether the ML outperforms among loans that are less likely to be a part of the priority sector. I create a subsample by excluding agricultural loans and those loans which are below the median size, in terms of value, among nonagricultural loans. It is reasonable to assume that the proportion of priority sector loans is likely to be significantly less in this subsample when compared to the overall sample. I perform the contraction procedure on this subsample by equating the loan acceptance rate of the strict period portfolio and the ML portfolio. I present the results in Table VI. The table mimics the organization of Panel A of Table III. Notice that the ML algorithm outperforms the strict period portfolio irrespective of the definition of default and acceptance rate used. Therefore, my results are unlikely to be solely due to the priority sector status of the sample loans.

6.2.d. Are the differences significant?

The differences presented in Table III appear economically meaningful. I attempt to measure the significance of the difference in a statistical sense. Given that the coefficients generated by the ML algorithm may not be consistent (KLLLM), estimating statistical

Table VI. Non-priority sector loans

In this table, I present the results of the contraction procedure. I exclude agricultural loans and loans that are below the median of loan amount among nonagricultural loans. The first (second) half of the month is considered the strict (lenient) period. I create an ML portfolio by starting from the lenient portfolio and eliminating cases in the descending order of risk scores as assigned by the ML algorithm. I stop when the acceptance rate of the machine portfolio equals the acceptance rate of the strict portfolio. I report the reduction in default rate achieved by the machine portfolio over the strict portfolio. In column 1, a loan is considered to be in default if it is overdue with respect to its last installment in the data. In column 2, a loan is considered to be in default if it is in default at the end of the twelfth month. Loans having coverage for less than twelve installments are excluded from the sample. In column 3 (4), I leave out loans that entered in the last nine (six) months and call a loan in overdue status as at the end of nine (six) months as a loan in default. Finally, in column 5, a loan is considered to be in default if it defaults at any time during the sample period.

Default classification	Last installment	12th installment	9th installment	6th installment	Ever default
Loan approval rate	0.29	0.34	0.33	0.20	0.35
Loan amount proportion	0.36	0.34	0.30	0.20	0.39
Loan amount lent	0.39	0.20	0.16	0.07	0.32

significance is a challenge. I ask whether the outperformance of the algorithm could arise by chance. To answer the above question, I create an empirical distribution by bootstrapping. I start with the lenient portfolio and reject applications randomly until the approval rate equals the approval rate of the strict period. Notice that the contraction procedure here is not guided by the ML algorithm. I then calculate the difference between the default rate during the strict period and the artificial portfolio, in each round. I repeat this exercise 1,000 times and plot the differences in Figures A.7, A.8, and A.9 of the Online Appendix. I use the three definitions of the approval rate in the three figures. The dotted line represents the boundary of 95% of the area. The thick line denotes the actual difference obtained using the ML algorithm determined risk factors for contraction. The actual difference is way beyond the 95% cut-off. Therefore, it is likely that my results are significant in a statistical sense.

6.2.e. Is the model stable?

To address concerns regarding the stability of the ML model, I run the model thirty times and report the distribution of the outperformance in Table A.5 of the Online Appendix. I note that (i) the ML algorithm outperforms in all thirty trials; and (ii) the distribution is within a narrow range. Therefore, it is reasonable to conclude that the ML algorithm is stable.

6.3 Impact on Profits

A skeptic may argue that the ML algorithm achieves lower defaults by excluding loans on which the officers would have charged higher interest rates. In other words, the loan officer prices the risk appropriately. In that case, it is not clear whether achieving a lower default rate in itself is beneficial to the bank. In terms of the framework described in Section 3, it is not clear whether the outperformance of the ML algorithm continues to hold if I use profitability as the outcome variable Y_i in place of loan performance. I face two challenges in computing profits. The first one relates to the cost of funds to the bank. Here, I assume that the lending of one division of the bank is unlikely to influence the bank's overall deposit mobilization. Therefore, I consider the risk-free rate as the opportunity cost of funds to calculate the gross margin on each loan. Second, I do not know the loss given default. Therefore, I calculate profits under various assumed recovery rates. I conduct several types of tests to assess the incremental profitability of the ML algorithm.

6.3.a. Does the machine pick up low yielding loans?

I ask whether the loan officers charge a higher interest rate on loans rejected by machines but accepted by the officers by estimating the following regression equation:

$$Interest_{Rat}e_{ijt} = \beta_0 + \beta_j + \beta_t + \beta_1 * ML_{Rejecte}d_{ijt} + \beta \cdot X_{ijt} + \varepsilon_{ijt} \quad (5)$$

Each observation is a loan i lent by an officer j at time t. The dependent variable $Interest_{ijt}$ is the annualized interest rate charged on a loan i lent by an officer j during time t. The explanatory variable, $ML_{Portfoli}o_{ijt}$, is a dummy variable that takes the value of 1 for loans in the ML portfolio, and 0 otherwise. β_j represents officer-level fixed effects, and β_t represents month X year fixed effects. X_{ijt} represents borrower-level characteristics. I include borrower-level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. The standard errors are clustered at the officer level, and adjusted t-statistics are reported in parentheses below the regression estimates.

The results are presented in Table VII. In Panel A, I restrict the data to the ML portfolio and the strict period portfolio. In Panel B, I restrict the sample to loans lent during the lenient period as the contraction exercise is done using the lenient portfolio. In Panel A (B), the comparison is between the ML portfolio and the strict (lenient) portfolio. I do not find a strong association between interest rates and the machine's decision to reject/accept. Further, in Table A.6 of the Online Appendix, I compare the entire strict and lenient period portfolios in terms of the interest rate charged on loans. I do not find any significant difference. Therefore, it is unlikely that the machine achieves a lower level of default by selectively omitting cases on which the loan officer would have charged higher interest rates.

6.3.b. Interest Income and contraction

To address any residual concerns relating to machine scores simply reflecting risks already factored in by the officer, I perform the contraction procedure described in Section 6.1 using interest rates as a proxy for risk scores. I start with the lenient portfolio and eliminate loans with the highest interest rates until the acceptance rate of the above artificial portfolio equals the acceptance rate of the strict period. I report the results in Table A.7 of the Online Appendix. The table mimics the organization of Panel A of Table III. Notice that the first four columns show negative results. In other words, the portfolio created by eliminating high-interest rate loans underperforms the strict period portfolio. While column 5 shows positive results, the values are close to 50% of the values presented in column 5 of Table III. The results reject the view that machine portfolio outperforms by simply eliminating high-interest cases. Finally, I train the model with and without including interest rates. The AUC of the model changes by less than half a percent because of interest rates. As

Table VII. Does the machine select low yielding loans?

I test whether loan officers charge higher interest rates for cases rejected by the machine but approved by the officers. The data are organized at a loan level. I use the test sample here. Interest rate charged is the dependent variable. In panel A, I compare the ML portfolio with the strict period portfolio. In Panel B, I restrict the data to the lenient portfolio and compare loans accepted and rejected by the ML algorithm. The main explanatory variable is a dummy variable that takes the value of 1 if the loan under consideration is accepted by the machine, and 0 otherwise. I include borrower-level characteristics such as the value of assets, annual income, loan amount applied for, and age as controls. I include fixed effects at the officer and at the month-year level. I use the three definitions of acceptance rates, as defined in Table II, in different columns. The standard errors are clustered at the officer level, and adjusted *t*-statistics are reported in parentheses below the regression estimates. ***, **, and * represent statistical significance at the 1%, 5%, and 10% levels, respectively.

Panel A						
	(1)	(2)	(3)	(4)	(5)	(6)
	Loan amount		Acceptance rate		Loan proportion	
Dependent Variable	Interest Rate					
ML portfolio	0.00	−0.00	−0.00	−0.00	0.00	0.00
	(0.16)	(−0.00)	(−0.22)	(−0.30)	(0.52)	(0.26)
Total assets		0.00		0.00		0.00
		(1.03)		(1.03)		(1.04)
Annual income		−0.00		−0.00		−0.00
		(−0.31)		(−0.28)		(−0.16)
Loan amount asked		0.00		0.00		−0.00
		(0.08)		(0.04)		(−0.03)
Borrower age		−0.00		−0.00		−0.00
		(−0.67)		(−0.69)		(−0.64)
Officer fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Month × year fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,430	1,430	1,441	1,441	1,400	1,400
R^2	0.09	0.09	0.09	0.09	0.09	0.10
Panel B						
ML portfolio	−0.00	−0.00	−0.00	−0.00	−0.00	−0.00
	(−0.88)	(−0.80)	(−0.88)	(−0.80)	(−0.86)	(−0.82)
Total assets		0.00		0.00		0.00
		(0.96)		(0.97)		(0.97)
Annual income		−0.00		−0.00		−0.00
		(−0.47)		(−0.46)		(−0.46)
Loan amount asked		0.00		0.00		0.00
		(0.49)		(0.46)		(0.52)
Borrower age		0.00		0.00		0.00
		(1.19)		(1.19)		(1.19)
Officer fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Month × year fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,328	1,325	1,328	1,325	1,328	1,325
R^2	0.08	0.08	0.08	0.08	0.08	0.08

argued by KLLLM, it is possible that a combination of other variables already incorporates the information contained in interest rates. As shown in Figure 3, the results could also be due to officers' inability to identify and price risk correctly.

6.3.c. Difference in profits

I also compute the difference in the profitability of the ML algorithm-based portfolio and the actual strict period portfolio. Note that both the portfolios have the same acceptance rate. I present the results by equating the loan amount sanctioned by both the portfolios. I present the results in Table VIII. I first calculate the net interest income by deducting interest calculated at the risk-free rate over the lifetime of a loan within my dataset from the interest income. I do not consider income when the loan is in default. In each row, I assume a different recovery rate after default. In row 1, I use the average recovery rate after bankruptcy proceedings. I use the national average of 44%. I do not have recovery rates for the bank which gave me the data. In the next three rows, I consider recovery rates of 50%, 75%, and 87.5%. As in Table III, I consider all five definitions of default.

Profit here refers to the gain of the ML portfolio over the strict portfolio. I normalize the gain using the net interest income during the strict period and report the results. The first cell in the table (row 1 and column 1) shows that the ML portfolio earns 15% higher profits when compared to the strict period portfolio when I use the national average recovery rate and the first definition of default as defined in Section 5.3.1. Notice that irrespective of the definition used, the machine outperforms the strict period portfolio in terms of profits.

The comparison in Table VIII is only at two points. To understand the relative performance of the machine in terms of profits at various levels, I plot the relative performance of the machine and the officer at various points in Figure 5.[13] In Figure 5, the horizontal axis depicts the loan amount and the vertical axis depicts the profit amount. The point X1 (X2) denotes the loan and the profit amount of the lenient (strict) period. The curve starting from X1 denotes the various profit-loan amount combinations achieved by the machine by using the contraction procedure. Notice that both X1 and X2 are below the curve.

Some findings are noteworthy here. First, X3 denotes the profit amount the machine would achieve at the acceptance rate of the strict period. The machine outperforms. Second, a comparison between the curve from X1 to X3 and the line X1 to X2 tells that the machine achieves a higher profit rate at every level of loan acceptance rate than a linear combination of X1 and X2. Finally, notice the four arrows. Arrow (i) denotes the difference between the strict and lenient periods in terms of profit. Arrow (ii) shows the outperformance of the machine over the strict period in terms of profit keeping the acceptance rate fixed. Arrow (iii) shows the difference in loan amount lent between the strict and the lenient periods. Arrow (iv) denotes applications rejected during the lenient period. Figure A.10 presented in the Online Appendix depicts the association between the amount in default and profits. I find similar results here as well.

6.4 Impact on Equity

It is critical to ask whether the machine achieves efficiency at the expense of equity. This is an important concern raised in the context of lending in developed economies. Equity is even more important for emerging economies given the existence of large-scale poverty.

13 I use the first definition of default as defined in Section 5.3.1.

Table VIII. Comparison based on profits

In this table, I present the results of comparison based on profits. The table presents the incremental profit achieved by the algorithm over the strict period. The profit number is normalized using the gross interest income of the strict period. I compute profits by assuming the risk-free rate as the opportunity cost of funds and different levels of recovery given default. In row 1, I assume the recovery rate equal to the average recovery rate post-bankruptcy in India. In rows 2–4, I assume different levels of recovery. In column 1, a loan is considered to be in default if it is overdue with respect to its last installment in the data. In column 2, a loan is considered to be in default if it is in default at the end of the twelfth month. Loans having coverage for less than twelve installments are excluded from the training sample. In column 3 (4), I leave out loans that entered in the last nine (six) months and call a loan in overdue status as at the end of nine (six) months as a loan in default. Finally, in column 5, a loan is considered to be in default if it defaults at any time during the sample period.

Panel A

Default classification	Last installment	12th installment	9th installment	6th installment	Ever default
Bankruptcy recovery rate	0.15	0.06	0.41	0.27	0.67
50% recovery rate	0.13	0.05	0.36	0.23	0.57
75% recovery rate	0.10	0.02	0.18	0.12	0.24
87.5% recovery rate	0.08	0.01	0.09	0.07	0.08

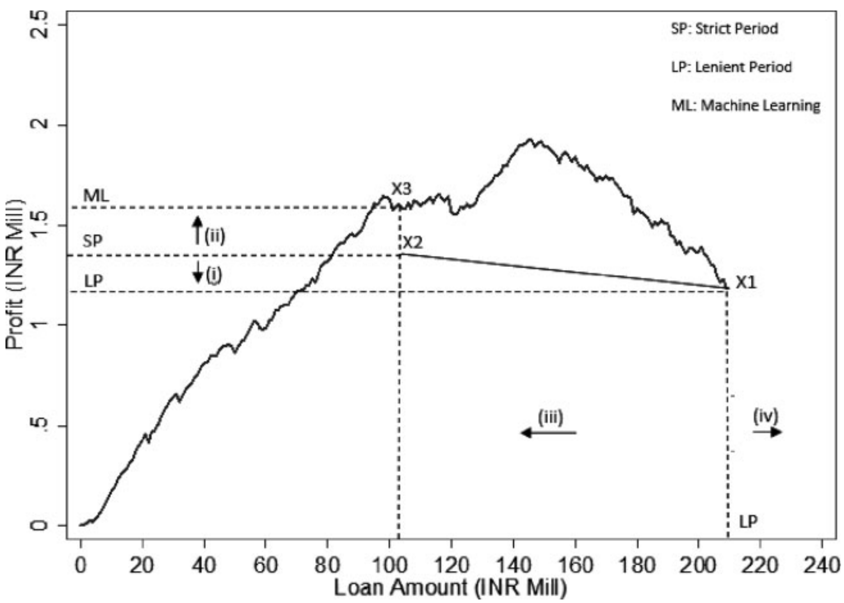

Figure 5. Profit comparison between the algorithm and the strict period based on loan amount lent. This figure plots the loan amount lent on the horizontal axis and the profit amount on the vertical axis. The point X1 (X2) denotes the loan amount and the profit of the lenient (strict) period. The curve starting from X1 denotes the various loan amount-profit combinations achieved by the machine by using the contraction procedure.

Researchers working in developed economy settings have noted that even when the social category information is not used in training, the machine labels disadvantaged groups as highly risky and ends up reducing the flow of credit to them (Fuster *et al.*, 2018). The situation in emerging economies may be different when compared to their developed counterparts. There are a significant number of poor people in almost all social categories. Besides, as noted in Section 3, the officers use a lot of soft information and discretion in lending as both the quality and quantity of hard information are not very high. Therefore, there is a possibility that ML can achieve efficiency even when it is constrained by (i) using the available hard information effectively (ii) correcting loan officer biases (iii) identifying risky borrowers within categories. Therefore, I test this empirical question relating to equity. It is crucial to note that I do not attempt to measure absolute equity. I attempt to test whether the machine discriminates more than the officers do. As noted in Section 2, the constitution of India classifies historically disadvantaged communities SCSTs and provides for affirmative action. I use this classification and ask whether the ML algorithm discriminates against SCSTs.

6.4.a. Are SCSTs more likely to be rejected by machines?

I start by examining whether SCSTs are more likely to be rejected by the machine during the contraction exercise. To this end, I estimate the following regression equation.

$$ML_{Rejected}d_{ijt} = \beta_0 + \beta_j + \beta_t + \beta_1 * SCST_{ijt} + \beta \cdot X_{ijt} + \varepsilon_{ijt} \quad (6)$$

The data are organized at a loan i level. The dependent variable, $ML_{Rejecte}d_{ijt}$, is a dummy variable that takes the value of 1 for loans that are rejected by the ML algorithm but accepted by officers, and 0 otherwise. The explanatory variable $SCST_{ijt}$ is a dummy variable that takes the value of 1 if the borrower under consideration belongs to SC or ST community, and 0 otherwise. Other terms have the same meaning as in Equation (5).

I report the results in Table IX. I find that the social status of the borrower is not significantly associated with the rejection by the machine. Further, I retrain the machine without feeding in information about the social status of the person. I do this to check whether the machine learns about the social status by using other information. I repeat the above analysis and present the results in Table A.8 of the Online Appendix. I do not find any evidence of discrimination. SCSTs are no more likely to be rejected by the machine when compared to other borrowers.

6.4.b. Are SCSTs assigned higher risk scores?

I explicitly test whether the algorithm assigns a higher risk score to SCSTs. The test is intended to capture discrimination that might have gone unnoticed due to the nature of the contraction exercise. In other words, even if SCSTs are assigned a higher risk score than comparable non SCSTs, they will not be eliminated by the algorithm if they are below the cut-off in terms of risk scores. I estimate the regression Equation (6) using the risk scores as the dependent variable. I present the results in Table A.9 of the Online Appendix. I find that the machine is unlikely to assign a higher risk score to the SCST borrowers.

6.4.c. Contraction with constraints

Given the results presented in Sections 6.4.1 and 6.4.2, it appears that a contraction exercise that is explicitly constrained to maintain the SCST proportion should not do

Table IX. Do machines discriminate?

In this table, I test whether the machines discriminate based on social status. The data are organized at a loan level. I use the test sample here and restrict the data to loans lent during the lenient period. The dependent variable is a dummy variable that takes the value of 1 if the loan under consideration is rejected by the machine, and 0 otherwise. SCST is a dummy variable that takes the value of 1 if the borrower belongs to the SC or ST community, and 0 otherwise. I include borrower-level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. I include fixed effects at the officer and month-year level. I use the three definitions of acceptance rates, as defined in Table II, in different columns. The standard errors are clustered at the officer level, and adjusted *t*-statistics are reported in parentheses below the regression estimates. ***, **, and * represent statistical significance at the 1%, 5%, and 10% levels, respectively.

Dependent variable	(1)	(2)	(3)	(4)	(5)	(6)
	Loan amount		Acceptance rate		Loan proportion	
	ML rejected	ML rejected	ML reject	ML rejected	ML reject	ML reject
SC ST	−0.01	−0.01	−0.00	0.00	−0.01	−0.00
	(−0.61)	(−0.34)	(−0.20)	(0.06)	(−0.30)	(−0.02)
Total assets		−0.00		0.00		−0.00
		(−0.25)		(0.04)		(−0.56)
Annual income		−0.00		0.00		0.00
		(−0.06)		(0.27)		(0.77)
Loan amount asked		0.00		0.00		0.00*
		(0.86)		(0.55)		(1.67)
Borrower age		−0.01***		−0.01***		−0.01***
		(−3.92)		(−3.87)		(−3.59)
Officer fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Month-year fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,328	1,325	1,328	1,325	1,328	1,325
R^2	0.63	0.64	0.63	0.64	0.63	0.64

significantly worse than the procedure without such constraints. The constraint can be denoted as $P^b = P^g$ and applies to Equation (2) described in Section 3. Recall that P^b (P^g) refers to the proportion of SCSTs in the officer (machine) portfolio. I present the results in Table X. The organization of the table exactly mimics the organization of Table III. I include a constraint that the proportion of SCSTs selected cannot be lower than the proportion in the actual portfolio of the strict period. The results show that the constraint does not make much difference to the performance of the algorithm. The algorithm continues to achieve a lower default rate at the officers' acceptance rate and a higher acceptance rate at the officers' default rate.[14] Finally, I test and find that the results relating to equity improve with an improvement of sample size from randomly selected 5,000 loan applications to the

14 In Table A.10 of the Online Appendix, I repeat the exercise without using information relating to castes while training the algorithm. I find that the algorithm achieves both efficiency and equity even when use the restricted training dataset. In Figure A.11 of the Online Appendix, I show the performance of the constrained algorithm at all points as in Figure 4.

Table X. Comparison based on loan approval rate and default with constraints

In this table, I present the results of the contraction procedure. The first (second) half of the month is considered a strict (lenient) period. In Panel A, I create an ML portfolio by starting from the lenient portfolio and eliminating cases in the descending order of risk scores as assigned by the ML algorithm. I constrain the algorithm to ensure that the proportion of SCSTs is maintained. I stop when the acceptance rate of the machine portfolio equals the acceptance rate of the strict portfolio while maintaining the SCST proportion. I report the reduction in default rate achieved by the machine portfolio over the strict portfolio. Similarly, in Panel B, I create the machine portfolio by equating the default rates and maintaining the proportion of SCSTs. I report the incremental acceptance rate. In column 1, a loan is considered to be in default if it is overdue with respect to its last installment in the data. In column 2, a loan is considered to be in default if it is in default at the end of the twelfth month. Loans having coverage for less than twelve installments are excluded from the training sample. In column 3 (4), I leave out loans that entered in the last nine (six) months and call a loan in overdue status as at the end of nine (six) months as a loan in default. Finally, in column 5, a loan is considered to be in default if it defaults at any time during the sample period.

Panel A					
Default Classification	Last installment	12th installment	9th installment	6th installment	Ever default
Loan approval rate	0.33	0.45	0.45	0.19	0.38
Loan amount proportion	0.37	0.34	0.35	0.10	0.35
Loan amount lent (INR million)	0.37	0.14	0.23	0.06	0.40

Panel B					
Default Classification	Last installment	12th installment	9th installment	6th installment	Ever default
Loan approval rate	0.29	0.13	0.11	0.04	0.30
Loan amount proportion	0.61	0.60	0.62	0.16	0.73
Loan amount lent (INR million)	0.74	0.55	0.55	0.32	0.79

full sample. I present the results in Table A.11 and describe the same in detail in Section C of the Online Appendix. Therefore, lack of observations does not seem to impact my interpretation of results regarding equity.[15]

6.4.d Comparison in terms of lending to SCSTs

In Table XI, I compare the amount of loan lent to SCSTs and proportion of SCST borrowers in the portfolio of (i) strict officers, (ii) the ML algorithm with no constraints, and (iii) the ML algorithm that seeks to maximize lending to SCSTs by maintaining the default and acceptance rate of the strict period. There is not much difference between the actual strict period portfolio and the unconstrained ML portfolio. Comparing the first two columns, I find that the SCST proportion of the unconstrained ML portfolio in terms of

15 In Table A.12 of the Online Appendix, I constrain the ML algorithm to maintain the proportion of poor borrowers. I find that the algorithm continues to outperform.

Table XI. Comparison based on lending to SCSTs

In this table, I compare the actual strict period (column 1), unconstrained contraction by equating acceptance rates (column 2), and contraction that aims to maximize lending to SCSTs without compromising on portfolio performance in terms of acceptance and default (column 3). In row 1 (2), I consider the loan acceptance rate in terms of numbers (value) of loans. In row 3, I consider the loan amount.

Method	Strict period	ML portfolio	Maximizing SCST
Acceptance rate	0.08	0.07	0.08
Loan proportion	0.06	0.06	0.08
Loan amount (INR million)	6.46	7.56	16.50

numbers is 1% lower. The amount lent is close to INR 1 million higher. In terms of definition 2 (loan amount proportion), there is no significant difference. Column 3 shows that the actual lending to SCSTs can be significantly increased without any change in default rates and also while maintaining the loan acceptance rate. The proportion in terms of the number of SCSTs over the total number of borrowers can be increased from 7% to 8% without any increase in the default rate. In terms of value, the proportion can go up from 6% to 8%. Finally, there is scope to increase the absolute amount of lending from INR 7.56 million to INR 16.5 million, without any increase in default rate.

6.5 Fairness Test

Finally, I conduct a fairness test using the "Aequitas– A Bias and Fairness Audit Toolkit" (Saleiro *et al.*, 2018). I ask whether the algorithm is biased against SCSTs or the poor. I use loan application data. The loan officer's decision to accept or reject an application becomes the outcome of the test. A rejected application is assigned the value of 1 and an accepted application is assigned a score of 0. Using the algorithm trained on loan performance in terms of default, I allocate risk scores to both accepted and rejected applicants in the data. I then use these risk scores to recommend an application to be selected or rejected. Using the proportion of actual acceptance rate by loan officers, I create a cut-off in my risk score for classification. Risk scores above the threshold are recommended to be rejected, taking the value of 1, and the ones below the threshold are recommended to be accepted, taking the value of 0. For example, if in the dataset 30% of the applications are rejected, then I consider the highest 30% cases in terms of risk scores as rejected by the algorithm. I now have both actual outcomes and machine recommendations in terms of acceptance and rejection. I define various error-based metrics in Table XII.

The tool defines bias as a disparity measure across groups when compared with a reference group. Disparity rate is the divergence between the treated and the reference group. I test whether the machine is biased against SCSTs and the poor. I use borrowers belonging to the general category and not poor as reference groups, respectively. The test calculates disparity using several matrices. The tool uses the 80% rule in defining disparity. A deviation below 0.8 and above 1.25 (1/0.8) is classified as bias. Disparity numbers of SCST (poor) are presented in Panel A (B) of Table XII. Using the inbuilt classification of supervised fairness of the model, my model qualifies as a fair model in both classification groups

Table XII. Fairness test

In this table, I present the results of the fairness test based on the method suggested in Saleiro *et al.* (2018). In Panel A (B), I assess fairness in selecting SCST (poor) and use the general category (not poor) as the reference group. The following are the key variables.

(1) False omission rate (FOR): False negative (FN)/predicted negative (PN);

(2) False discovery rate (FDR): False positive (FP)/predicted positive (PP);

(3) False positive rate (FPR): False positive (FP)/labeled negative (LN);

(4) False negative rate (FNR): False negative (FP)/labeled positive (LP);

(5) Predicted positive rate (PPR): Fraction of the entities predicted as positive that belong to a certain group; and

(6) Predicted positive group rate (PPRev): Fraction of entities within a group which were predicted as positive.

Panel A

Attribute name	Caste		
Attribute value	General (reference group)	SCST	Disparity rate
False omission rate (FOR)	0.53	0.60	1.13
False discovery rate (FDR)	0.39	0.49	1.26
False positive rate (FPR)	0.55	0.65	1.18
False negative rate (FNR)	0.37	0.43	1.16
Predicted positive rate (PPR)	0.90	0.10	0.11
Predicted positive group rate (PPREV)	0.59	0.61	1.03
Supervised fairness	True	True	

Panel B

Attribute name	Income		
Attribute value	Not poor (reference group)	Poor	Disparity rate
False omission rate (FOR)	0.46	0.55	1.20
False discovery rate (FDR)	0.42	0.45	1.07
False positive rate (FPR)	0.50	0.64	1.28
False negative rate (FNR)	0.38	0.37	0.97
Predicted positive rate (PPR)	0.52	0.48	0.92
Predicted positive group rate (PPREV)	0.56	0.64	1.14
Supervised fairness	True	True	

of caste and income. Also, I see that my disparity measures are well within the range in most cases.

7. A Discussion about the Role of Monitoring

It is crucial to clearly understand how and where an ML tool can be used and most importantly, what are its limitations. The purpose of this study is to compare decision-making by human beings with and without the ML tool and not to replace humans. My tests do not answer the question of whether ML does better than humans. The performance of a loan depends on both screening and monitoring by the loan officers (Diamond, 1984). The ML tool that I have designed in this study does only the screening part. My tests assume that the monitoring effort of the officer will continue to be the same even when a machine is used. In other words, based on my findings a case for machine outperforming humans in the context of loans can be made only in cases where the monitoring effort is zero. Given my data limitations, such a strong assumption looks unrealistic. In all other cases, the ML algorithm is likely to be a handy tool that is likely to improve the efficiency of the officers.

8. Conclusion

I ask whether ML algorithms can be used to aid loan officers in improving access to formal credit to the poor. Loan application processing by loan officers appears to be a prediction problem involving loan application characteristics as input variables and loan performance as the outcome variable. Therefore, the problem appears to be amenable to the use of ML techniques. However, the literature on financial intermediation shows that the loan officers possess soft information and they use such information in lending. The use of soft information, on which a machine cannot be trained, can hamper the ability of the machine to learn based on information presented in loan applications. Therefore, it is not clear whether an efficient use of hard information by the machines can outperform the actual use of a combination of soft and hard information by loan officers in predicting loan outcomes. Using loan application-level data from a bank in India, I attempt to answer the above question.

I encounter the problem of the selective labels as I have actual outcomes only for approved cases. I resolve the problem by exploiting the fact that, driven by incentives, the officers tend to be relatively lenient towards the end of the month. Using the lenient period portfolio as the base and strict period portfolio as the benchmark, I find that rejecting loan applications which are considered the riskiest by the ML algorithm but accepted by officers improves the performance in terms of default, profitability, and loan amount. Finally, I verify that the ML algorithm maintains its outperformance even after taking into account equity considerations in terms of lending to the vulnerable sections of society.

These findings, I believe, have significant implications for both lending and policy. From a policy point of view, improving access to formal credit to lower strata of the society is a major goal of emerging market governments. Given that the quality of hard information is suspect, the lenders rely heavily on soft information. This works as a limiting factor. My findings show that the efficient use of available hard information can improve loan performance and can be used to increase lending without a significant increase in default.

Supplementary Material

Supplementary data are available at *Review of Finance* online.

Data Availability Statement

Data are owned by a third party. The data underlying this article were provided by one of the largest national banks in India under license/by permission. Data will be shared on request to the corresponding author with permission from the bank.

References

Agarwal, S., Amromin, G., Ben-David, I., Chomsisengphet, S., and Evanoff, D. D. (2014): Predatory lending and the subprime crisis, *Journal of Financial Economics* 113, 29–52.

Agarwal, S. and Ben-David, I. (2018): Loan prospecting and the loss of soft information, *Journal of Financial Economics* 129, 608–628.

Agarwal, S. and Hauswald, R. (2010): Distance and private information in lending, *The Review of Financial Studies* 23, 2757–2788.

Agarwal, S. and Wang, F. H. (2009): Perverse incentives at the banks? Evidence from a natural experiment, Working paper 2009–08.

Agrawal, A., Gans, J. S., and Goldfarb, A. (2019): Artificial intelligence: the ambiguous labor market impact of automating prediction, *Journal of Economic Perspectives* 33, 31–50.

Asch, B. J. (1990): Do incentives matter? The case of navy recruiters, *ILR Review* 43, 89-S–106-S.

Athey, S. (2018): The impact of machine learning on economics, *The Economics of Artificial Intelligence: An Agenda. University of Chicago Press,* 507–547.

Athey, S. and Imbens, G. W. (2019): Machine learning methods that economists should know about, *Annual Review of Economics* 11, 685–725.

Ayyagari, M., Demirgüç-Kunt, A., and Maksimovic, V. (2010): Formal versus informal finance: evidence from china, *The Review of Financial Studies* 23, 3048–3097.

Banerjee, A. V. and Duflo, E. (2014): Do firms want to borrow more? Testing credit constraints using a directed lending program, *The Review of Economic Studies* 81, 572–607.

Bartlett, R., Morse, A., Stanton, R., and Wallace, N. (2019): Consumer-Lending Discrimination in the Fintech Era, Technical Report, National Bureau of Economic Research.

Beck, T., Ioannidou, V., and Schäfer, L. (2017): Foreigners vs. natives: bank lending technologies and loan pricing, *Management Science* 64, 3792–3820.

Behr, P., Drexler, A., Gropp, R., and Guettler, A. (2014): Financial incentives and loan officer behavior: multitasking and allocation of effort under an incomplete contract, *Journal of Financial and Quantitative Analysis* 55, 1243–1267.

Berg, T., Burg, V., Gombović, A., and Puri, M. (2019): On the rise of fintechs: credit scoring using digital footprints, *The Review of Financial Studies* 33, 2845–2897.

Berg, T., Puri, M., and Rocholl, J. (2013): Loan Officer Incentives and the Limits of Hard Information, Technical Report, National Bureau of Economic Research.

Bhowal, S., Subramanian, K., and Tantri, P. (2020): Costs of job rotation: Evidence from mandatory loan officer rotation, Management Science.

Bhue, G., Prabhala, N., and Tantri, P. L. (2019): Can small business lending programs disincentivize growth? Evidence from India's priority sector lending program, Working paper 2019.

Bianchi, D., Büchner, M., and Tamoni, A. (2019): Bond risk premia with machine learning, USC-INET Research Paper 19–11.

Buchak, G., Matvos, G., Piskorski, T., and Seru, A. (2018): Fintech, regulatory arbitrage, and the rise of shadow banks, *Journal of Financial Economics* 130, 453–483.

Bursztyn, L., Fiorin, S., Gottlieb, D., and Kanz, M. (2019): Moral incentives in credit card debt repayment: evidence from a field experiment, *Journal of Political Economy* 127, 1641–1683.

Chakraborty, A. and Hu, C. X. (2006): Lending relationships in line-of-credit and nonline-of-credit loans: evidence from collateral use in small business, *Journal of Financial Intermediation* 15, 86–107.

Chen, D. L., Moskowitz, T. J., and Shue, K. (2016): Decision making under the gamblerâs fallacy: evidence from asylum judges, loan officers, and baseball umpires, *The Quarterly Journal of Economics* 131, 1181–1242.

Chen, T. and Guestrin, C. (2016): Xgboost: a scalable tree boosting system, *22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cheng, I. H., Raina, S., and Xiong, W. (2014): Wall street and the housing bubble, *The American Economic Review* 104, 2797–2829.

Cole, S., Giné, X., Tobacman, J., Townsend, R., Topalova, P., and Vickery, J. (2013): Barriers to household risk management: evidence from India, *American Economic Journal: Applied Economics* 5, 104–135.

Cole, S., Kanz, M., and Klapper, L. (2015): Incentivizing calculated risk-taking: evidence from an experiment with commercial bank loan officers, *The Journal of Finance* 70, 537–575.

Cole, S., Sampson, T., and Zia, B. (2011): Prices or knowledge? What drives demand for financial services in emerging markets?, *The Journal of Finance* 66, 1933–1967.

Cortés, K., Duchin, R., and Sosyura, D. (2016): Clouded judgment: the role of sentiment in credit origination, *Journal of Financial Economics* 121, 392–413.

Demirguc-Kunt, A., Klapper, L., and Singer, D. (2017): Financial inclusion and inclusive growth: a review of recent empirical evidence, The World Bank.

Diamond, D. W. (1984): Financial intermediation and delegated monitoring, *The Review of Economic Studies* 51, 393–414.

Dobbie, W., Liberman, A., Paravisini, D., and Pathania, V. (2018): Measuring bias in consumer lending, Technical report, National Bureau of Economic Research.

Drucker, S. and Puri, M. (2008): On loan sales, loan contracting, and lending relationships, *The Review of Financial Studies* 22, 2835–2872.

Du, Q., Yu, F., and Yu, X. (2017): Cultural proximity and the processing of financial information, *Journal of Financial and Quantitative Analysis* 52, 2703–2726.

Fisman, R., Paravisini, D., and Vig, V. (2017): Cultural proximity and loan outcomes, *American Economic Review* 107, 457–492.

Freixas, X., Guesnerie, R., and Tirole, J. (1985): Planning under incomplete information and the ratchet effect, *The Review of Economic Studies* 52, 173–191.

Friedman, J. H. (2001): Greedy function approximation: a gradient boosting machine, *Annals of Statistics* 29, 1189–1232.

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., and Walther, A. (2018): Predictably unequal? The effects of machine learning on credit markets, Working paper 2018.

Fuster, A., Plosser, M., Schnabl, P., and Vickery, J. (2019): The role of technology in mortgage lending, *The Review of Financial Studies* 32, 1854–1899.

Gathergood, J., Mahoney, N., Stewart, N., and Weber, J. (2019): How do individuals repay their debt? The balance-matching heuristic, *American Economic Review* 109, 844–875.

Griffin, J. M. and Maturana, G. (2016): Who facilitated misreporting in securitized loans?, *The Review of Financial Studies* 29, 384–419.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2019): A survey of methods for explaining black box models, *ACM Computing Surveys (CSUR)* 51, 93.

Healy, P. M. (1984): The effects of bonus schemes on accounting decisions, *Journal of Accounting and Economics* 7, 85–107.

Hertzberg, A., Liberti, J. M., and Paravisini, D. (2010): Information and incentives inside the firm: evidence from loan officer rotation, *The Journal of Finance* 65, 795–828.

Holmstrom, B. and Milgrom, P. (1987): Aggregation and linearity in the provision of intertemporal incentives, *Econometrica: Journal of the Econometric Society 55*, 303–328.

Iyer, R., Khwaja, A. I., Luttmer, E. F., and Shue, K. (2015): Screening peers softly: inferring the quality of small borrowers, *Management Science* 62, 1554–1577.

Keys, B. J., Mukherjee, T., Seru, A., and Vig, V. (2010): Did securitization lead to lax screening? Evidence from subprime loans, *The Quarterly Journal of Economics* 125, 307–362.

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

Kirschenmann, K. (2016): Credit rationing in small firm-bank relationships, *Journal of Financial Intermediation* 26, 68–99.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. (2017): Human decisions and machine predictions, *The Quarterly Journal of Economics* 133, 237–293.

Kleinberg, J., Ludwig, J., Mullainathan, S., and Obermeyer, Z. (2015): Prediction policy problems, *American Economic Review* 105, 491–495.

Larkin, I. (2014): The cost of high-powered incentives: employee gaming in enterprise software sales, *Journal of Labor Economics* 32, 199–227.

Liberti, J. M. and Mian, A. R. (2008): Estimating the effect of hierarchies on information use, *The Review of Financial Studies* 22, 4057–4090.

Mishkin, F. S. and Strahan, P. E. (1999): What Will Technology Do to Financial Structure?, Technical Report, National Bureau of Economic Research.

Mullainathan, S. and Obermeyer, Z. (2017): Does machine learning automate moral hazard and error?, *American Economic Review* 107, 476–480.

Mullainathan, S. and Spiess, J. (2017): Machine learning: an applied econometric approach, *Journal of Economic Perspectives* 31, 87–106.

Murlidharan, S. A. A., Nishesh, N., and Tantri, P. (2020): Familiarity breeds contagion, Working paper 2020.

Oyer, P. (1998): Fiscal year ends and nonlinear incentive contracts: the effect on business seasonality, *Quarterly Journal of Economics* 113, 149–185.

Pierce, L. and Snyder, J. A. (2018): The historical slave trade and firm access to finance in Africa, *The Review of Financial Studies* 31, 142–174.

Qian, J., Strahan, P. E., and Yang, Z. (2015): The impact of incentives and communication costs on information production and use: evidence from bank lending, *The Journal of Finance* 70, 1457–1493.

Rajan, U., Seru, A., and Vig, V. (2015): The failure of models that predict failure: distance, incentives, and defaults, *Journal of Financial Economics* 115, 237–260.

Saleiro, P., Kuester, B., Hinkson, L., London, J., Stevens, A., Anisfeld, A., Rodolfa, K. T., and Ghani, R. (2018): Aequitas: a bias and fairness audit toolkit, arXiv preprint arXiv:1811.05577.

Sirignano, J., Sadhwani, A., and Giesecke, K. (2016): Deep learning for mortgage risk, arXiv preprint arXiv:1607.02470.

Skrastins, J. and Vig, V. (2018): How organizational hierarchy affects information production, *The Review of Financial Studies* 32, 564–604.

Solem, J. E. (2012): *Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA.

Spadon, G., de Carvalho, A. C., Rodrigues, J. F., Jr., and Alves, L. G. (2019): Reconstructing commuters network using machine learning and urban indicators, *Scientific Reports* 9, 1–13.

Taninaga, J., Nishiyama, Y., Fujibayashi, K., Gunji, T., Sasabe, N., Iijima, K., and Naito, T. (2019): Prediction of future gastric cancer risk using a machine learning algorithm and comprehensive medical check-up data: a case-control study, *Scientific Reports* 9, 1–9.

Tzioumis, K. and Gee, M. (2013): Nonlinear incentives and mortgage officersâ decisions, *Journal of Financial Economics* 107, 436–453.