Fintech For The Poor: Financial Intermediation Without Discrimination *

Prasanna Tantri

March 9, 2020

*Prasanna Tantri can be reached at prasanna_tantri@isb.edu. 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.

Fintech For The Poor: Financial Intermediation Without Discrimination

Abstract

We ask whether machine learning algorithms improve the efficiency of screening of the loan officers, and thereby help expand access to formal credit. We obtain loan application level data from an Indian bank. To overcome the selective labels problem, we exploit the incentive driven within officer difference in leniency within a calendar month. We find that the ML algorithm is able to lend 26.6% more at loan officers' delinquency rate or achieve 21.3% lower delinquency rate at loan officers' approval rate. Higher efficiency is achieved without compromising on equity.

1 Introduction

A fast growing literature investigates the impact of fintech on credit markets. The extant literature, mostly based on developed world settings, has explored dimensions such as the contributing factors that led to the emergence of fintechs (Buchak, Matvos, Piskorski, and Seru (2018)), their impact on credit market efficiency (Fuster, Plosser, Schnabl, and Vickery (2019), Berg, Burg, Gombović, and Puri (2019), Bianchi, Büchner, and Tamoni (2019)), and equity (Fuster, Goldsmith-Pinkham, Ramadorai, and Walther (2018), Dobbie, Liberman, Paravisini, and Pathania (2018)). These findings cannot be directly applied to emerging economies where (i) lack of access to credit is severe;(ii) the quality of hard information is suspect; (iii) relationship banking is the norm; (iv) social structures are different. Therefore, the question of whether the use of ML algorithms to assist loan officers can improve access to formal credit without increasing discrimination has considerable real world implications.

Given the bad quality of the hard information and high reliance on soft information by the loan officers,[1] there is reason to believe that formal statistical models are unlikely to determine the exact relationship between observable loan characteristics and loan outcomes. This is because borrowers with similar observable characteristics may differ unobservably (Rajan, Seru, and Vig (2015), Bursztyn, Fiorin, Gottlieb, and Kanz (2019)). Therefore, prima facei, it appears that loan decisions, especially in cases where the borrowers belong to low income category and do not posses credible documents, are not amenable to the use of ML techniques that rely on optimal use of hard information. Equity may be harmed if ML algorithms detect complicated patterns that link loan performance to social status (Fuster, Goldsmith-Pinkham, Ramadorai, and Walther (2018)).

On the other hand, loan application processing is essentially a prediction problem where the decision hinges on expectation of timely loan repayment, and hence, is suited for the application of machine learning algorithms (Kleinberg, Ludwig, Mullainathan, and Obermeyer (2015), Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2017), Mullainathan

[1](Agarwal and Hauswald (2010), Skrastins and Vig (2018), Liberti and Mian (2008), Qian, Strahan, and Yang (2015), Drucker and Puri (2008), Fisman, Paravisini, and Vig (2017))

and Spiess (2017)). Moreover, it is possible that loan officers systematically err in using both hard and soft information either because of the imprecise nature of the information, cognitive limitations, personal biases, or due to distortionary incentives (Keys, Mukherjee, Seru, and Vig (2010), Cortés, Duchin, and Sosyura (2016), Berg, Puri, and Rocholl (2013), Griffin and Maturana (2016), Agarwal, Amromin, Ben-David, Chomsisengphet, and Evanoff (2014), Chen, Moskowitz, and Shue (2016)). In addition, it is also possible that the ML algorithm is able to separate the wheat from the chaff, and hence, use only the reliable part of hard information that is strongly associated with loan performance. Therefore, an efficient use of hard information may outperform loan officers. To the extant information issues and loan officer discrimination are higher for borrowers from the lower strata of the society, an efficient use of ML algorithm may actually improve equity.

We use loan application processing by loan officers of a large listed commercial bank in India as the economic setting for our study. The loan applicants are all individuals and belong to the lower strata of the 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. Expectedly, most borrowers lack formal documents such as tax returns which are helpful in assessing their creditworthiness. A formal credit score is also not available in most cases. Given the above, 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 (Tybout (1983), Ma and Smith (1996), Kirschenmann (2016), Cole, Giné, Tobacman, Townsend, Topalova, and Vickery (2013), Cole, Sampson, and Zia (2011)). Existing lending is based on a combination of hard and soft information that loan officers possess about such borrowers.

We obtain information about 15,088 loan applications made by borrowers located in 196 villages/towns of 10 Indian states. Over and above the information disclosed in loan application, we also have data about the eventual loan decision and time series information about loan repayment at a monthly level. We use categorization of a loan as a loan in default as a measure of loan performance. Using the above data, we examine whether the use of a machine learning algorithm to screen loans results in either higher amount of lending keeping the default rate constant or lower defaults at the same amount of lending. Increase in lending without any increase in defaults implies an improvement in access to credit. Reduction in default rate without any decline in lending also boosts access to finance by bringing down the cost of financial intermediation. As argued by Banerjee and Duflo (2010), a combination of low net-worth, low income, and high administrative costs, a large of part of which is fixed, leads to a vicious cycle of low amount of credit or even denial of credit, high interest rates, continuation of credit constraints, and, low income. Therefore, any reduction in cost of intermediation through lower default rate is also likely to improve access to formal finance through the multiplier effect (Banerjee and Duflo (2010)).

We use the gradient boosted decision trees (Friedman (2001)) by organizing the data at a borrower level. The outcome variable takes the value of one if the loan under consideration defaults even once and zero otherwise. Specifically, we use the XGboost algorithm (Chen and Guestrin (2016)). We use borrower and loan application characteristics available at the time of loan application processing as input variables. As in Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2017) (KLLLM, henceforth), we are faced with selective labels problem. The selective labels problem is a one sided problem as we can easily 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. We do not know whether the later loans would have resulted in an NPA or in repayment. In other words, we can only reject approved cases based on machine learning (ML,henceforth) algorithm's recommendation but not approve cases rejected by the loan officers. There is a second problem as well: the rejected and approved applications may differ in ways not observable by the algorithm. This problem is also important given that loan officers are known to possess soft information (Liberti (2017)).

KLLLM who compare the bail or jail decisions made by judges in New York with the decisions based on predictions of a machine learning (ML,henceforth) 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 and different judges select on unobservables similarly, they compare the actual decisions of strict judges with the decision arrived at using a ML algorithm that uses the decision of the lenient judges as the base. They show that if we start from the actual release set of the most lenient judges and jail additional defendants until the release rate equals the release rate of strict judges, the crime rate is lower for the machine determined group when compared to the actual set released by the strict judges.

Using insights from the literature on loan officer incentives, we 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), Agarwal and Ben-David (2018), Cole, Kanz, and Klapper (2015)). We compare the relative leniency within an officer during the two halves of a month. We find that the officers approve 42% (85%) of loan applications in the first (last) 15 days of a month. In terms of the value of the total loan amount applied to total loan amount granted at an aggregate level, the ratio works out to 37% and 77% in the two halves, respectively. We obtain similar results when we look at the absolute loan amount lent. The default rate is lower (higher) in the first (second) half. Therefore, we 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 officers, we use loans lent in the second half of the month as a starting point, as in KLLLM. We then ask whether by rejecting some loan applications made during the second half of the month, which were actually approved and hence whose outcomes are observables, is it possible to either lend as much as at the beginning of the month with a lower default rate or lend more keeping the default rate unchanged?

Although our empirical strategy is inspired by KLLLM, there are two important differences. First, the definition of strict and lenient is within an officer purely based on time of the month. We require the loan applications that enter in the first and second half of a month to be similar both on observable and un-observable fronts. We directly test this assumption. Second, we define our objective function from the point of view of the bank and not from the point of view of loan officers. In other words, we do not seek to test whether the ML based decisions improve loan officer pay-offs, instead we consider whether they improve the bank's pay-offs.

As a basic sanity check, we test and find that the risk scores generated by the ML algorithm are indeed significantly positively associated with the actual NPAs for the set of applications that are actually approved. The AUC of the algorithm is 77%, which is reasonable (Berg, Burg, Gombović, and Puri (2019)). The loan officers' technology of assessing risk seems to differ significantly from that of the machine. The officers grant loans from all parts of the risk distribution created by the machine. In fact, as in KLLLM, the officers seem to be having the maximum trouble in dealing with cases designated as high risk by the machine.

We then move to test whether the ML algorithm outperforms the loan officers. We start with the assumption that the bank's objective is to maximize the Rupee 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 Rupees 80 (100) and the default rate in the first (last) 15 days is 10% (20%). The thought experiment is to test what happens to the default rate of the last 15 days if the riskiest Rupees 20 worth of loans are denied. Of course, the riskiest loans are identified using the risk scores generated by the ML algorithm. Notice that since all the loans have been actually made, we can compute the actual default rate under this hypothetical situation, and compare the same with the actual strict period portfolio. If the default rate after reducing last 15 days' credit to Rupees 80 is less than 10%, then it is possible to deduce that the ML algorithm is doing a better job of fulfilling the bank's objective. The same conclusion can be reached if the ML algorithm is able to lend more than Rupees 80 while keeping the default rate at 10%. We find that the algorithm delivers a 26.6% reduction in default rate while keeping the loan amount lent unchanged. In an alternative scenario, it delivers a 21.3%

increase in lending while keeping the default rate unchanged.

The modified performance of the last 15 days serves as a valid counter-factual for the actual performance of the first 15 days for following reasons: (i) we test and find that the loan application types do not differ, both observably and un-observably, between the two halves of a month and (ii) since same officers are classified as strict and lenient during different times, screening ability, and soft information remain the same between the two halves. To test unobservable differences we train the ML algorithm using lenient period data and test the same on strict period data. We find that the relationship between inputs and output is almost the same in both the samples. Therefore, significant unobservable differences are unlikely to exist between the two groups.

Next, we consider the possibility that the bank takes into account the loan applications denied. It is possible that the bank's objective, is to provide access to credit to the maximum number of borrowers while keeping the default rate low. Conducting the same thought exercise as before, we find that the following the ML algorithm's recommendation leads to 22.4% reduction in defaults without any change in the loan approval rate. The loan approval rate is defined as the ratio between the number of loan applications approved and the total number of applications received. Alternatively, the algorithm delivers a 30.2% increase in approval rate keeping the default rate unchanged. The results continue to hold even when we redefine approval rate in terms of value of loans applied and the value of loans granted.

There could be a concern that the synthetic portfolio created using the ML algorithm achieves lower default rate at the cost of profitability. In other words, it is possible that algorithm rejects loans on which the loan officer charged higher interest rates. In such a case, lower default rate need not necessarily translate into higher profits. We examine and find that the interest rate charged on cases approved by the lenient officer and rejected by the ML algorithm during contraction is not significantly different from the interest rate charged on other loans. Therefore, it is unlikely that the algorithm achieves lower default rate at the expense of profitability.

We then consider the question of equity. Here we ask whether the ML algorithm discriminates against borrowers from the disadvantaged sections of the society. The Indian constitution identifies some specific social groups as historically disadvantaged. We 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, we conduct the contraction procedure by imposing an explicit condition that the proportion of selected borrowers from the disadvantaged sections cannot be lower than the sample average during the strict period. The out-performance of the ML algorithm remains more or less unchanged. Third, we find that the algorithm performs better than the loan officers even after increasing lending to disadvantaged sections by close to 20%. Fourth, the algorithm continues to out-perform even when information about the social category is not used for training. Finally, the algorithm outperforms the officers when the out-sample test is restricted to borrowers belonging to the disadvantaged communities. Given these results, it is reasonable to conclude that the ML algorithm achieves efficiency without compromising on equity.

In the final part of our empirical analysis, we build a decision tool following KLLLM. Even under stringent set of assumptions, we are able to show that it is useful to use the algorithm to assist loan officers in screening. The algorithm only assists in screening and not monitoring (Diamond (1984)). Therefore, the algorithm can only work as a tool in the hands of the loan officer, and not as their replacement.

This paper directly talks to the growing literature that studies the impact of fintech on credit markets (Fuster, Plosser, Schnabl, and Vickery (2019), Berg, Burg, Gombović, and Puri (2019), Bianchi, Büchner, and Tamoni (2019), Fuster, Goldsmith-Pinkham, Ramadorai, and Walther (2018), Dobbie, Liberman, Paravisini, and Pathania (2018)). We study the use of Ml algorithms in an emerging market context where soft information dominates. Using the incentive driven change in loan officer leniency, we are able to compare screening with and without the use of the ML algorithm. We show that Ml algorithms have the potential to improve access to credit without harming equity.

Our findings contribute to the literature that deals with issues related to access to formal finance (Cole, Sampson, and Zia (2011), Cole, Giné, Tobacman, Townsend, Topalova, and Vickery (2013), Pierce and Snyder (2018), Demirguc-Kunt, Klapper, and Singer (2017), Townsend (1994), Ayyagari, Demirgüç-Kunt, and Maksimovic (2010), Bartlett, Morse, Stanton, and Wallace (2019), Sirignano, Sadhwani, and Giesecke (2016), Khandani, Kim, and Lo (2010)). We show that an efficient use of an ML algorithm can potentially bring down defaults and thereby reduce the cost of lending to the poor.

We also contribute to the large literature that examines the use of soft information in lending (Agarwal and Hauswald (2010), Skrastins and Vig (2018), Liberti and Mian (2008), Beck, Ioannidou, and Schäfer (2017), Fisman, Paravisini, and Vig (2017), Iyer, Khwaja, Luttmer, and Shue (2015), Du, Yu, and Yu (2017), Hertzberg, Liberti, and Paravisini (2010), Chakraborty and Hu (2006), Cheng, Raina, and Xiong (2014), Drucker and Puri (2008), 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. We show that an efficient use of hard information though an ML algorithm is likely to outperform the combination of hard and soft information used by the loan officers.

We also contribute to the growing economics and finance literature that employs machine learning techniques to aid human decision making (Kleinberg, Ludwig, Mullainathan, and Obermeyer (2015), Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2017), Mullainathan and Spiess (2017), Gathergood, Mahoney, Stewart, and Weber (2019), Agrawal, Gans, and Goldfarb (2019), Mullainathan and Obermeyer (2017), Bonsall IV, Leone, Miller, and Rennekamp (2017), Athey (2018), Athey and Imbens (2019)). The issue that we study falls into the category of problems where the outcome is generated by a human decision maker and hence, we see outcomes only for approved cases and not for unapproved cases. This selective labels problem works as a hindrance in comparing machine predictions with human decisions especially when humans are likely to use information unobservable to the machine (Lakkaraju, Kleinberg, Leskovec, Ludwig, and Mullainathan (2017), Jung, Concannon, Shroff, Goel, and Goldstein (2017)). Extant studies use varied econometric techniques to impute outcomes for cases where there are no actual outcomes. All these methods assume that selection happens on observables (Lakkaraju and Rudin (2017)). In contrast, we develop a way of dealing with the selective labels problem by exploiting the within officer difference in leniency which is based purely on day of the month.

2 Data And Institutional Details

We obtain loan application level data of a large commercial bank in India. The bank operates through business correspondents in rural ares. Business Correspondents (BCs, henceforth) are retail agents appointed by banks for the purpose of providing basic banking services at locations that are not serviced by bank branches or ATMs. BCs are typically individuals having a permanent base in a village. Most common profiles of BCs include local shopkeepers, retired bankers, ex-servicemen, agents of government small saving schemes, among others. The government intends to solve the last mile delivery problem using BCs. 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 necessary technological infrastructure for BCs to operate. BCs are paid fee on 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. Each aggregator has a specified geographical jurisdiction. We obtain loan application level data from one such BC aggregator who covers ten large states of India for a large listed commercial bank, which has a nationwide presence. The data-set covers about 196 BCs located in as many different towns and villages.

With respect to 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. In spirit, the BC location serves as a quasi bank branch. At times, the loan officers visit the place of work or farm of the loan applicant and also ask for references. Given the above interactions, it is reasonable to infer that the loan officers not only have hard information but also collect soft information about the loan applicants. In addition, they also obtain inputs from the BCs. Generally, the loan officers belong to the local area. The loan officers have complete authority to sanction loans This allows them to use the soft information they have to the fullest (Liberti and Mian (2009), Skrastins and Vig (2018)).

The dataset combines information about loan applications and loan related transactions. We have applicant level information such as occupation, age, gender, individual income, family income, postal address, and other contact details, number of dependents, among others. We also have information such as the amount applied for, the purpose of the loan, the collateral, and others, which are at the loan application level. We do not have information about the name of the loan applicant.

We present the summary information about the data set in Table 1. Out of 15,088 loan applications in the data, 9,741 are accepted by the loan officers. Therefore, we have loan transactions data for 9,741 borrowers. The data set spans a period between November, 2017 and December, 2019. The average loan tenure is 35 months. The number of loan transaction observations per borrower depends on the starting date of the loan. In total, we have an average of 12 monthly loan repayment observations for every loan.

At a loan-month level, we have information about the monthly installment due, the installment amount paid, overdue amount if any, total loan amount, and the outstanding loan amount. A loan having overdue loan amount is considered a loan in default. The average default rate is 29.51%. The average loan amount is Rupees 157,359 which is close to USD 2,000 at the current exchange rate. The average interest rate charged is close to 13.69%, 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. The interest rate is in the ballpark of rate charged by other lenders to small borrowers.[2]

We then compare the loan application characteristics of approved and rejected cases. We find that approved and rejected cases systematically differ with respect to observable

[2] A comparison of interest rates on loans to small borrowers is presented here https://www.bankbazaar.com/personal-loan/sme-loans.html

characteristics such as the value of assets, and income. We 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. Note that the Indian constitution classifies the population into four social groups. The term scheduled caste (SC) refers to people who historically formed the lowest strata of the society. They were considered untouchables. Scheduled tribes (ST) refers to indigenous people living in some specified tribal areas. Other backward classes refers to those who were above scheduled castes in social hierarchy but nevertheless significantly poor. The remaining population is considered as general (Dushkin (1967)). In terms of social groups, 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 between the rejected and accepted pool. In terms of nominee relationship, applications where the nominee is the spouse are more likely to be accepted. This is also reflected in the fact that the average number of dependents is higher (2.57) for accepted cases when compared to rejected cases (2.49). There seems to be a preference for married loan applicants. We 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.

3 The ML Exercise

For the purpose of ML exercise, we organize the data at an individual borrower level and not borrower repayment level. Note that we can only use 9,741 observations relating to loans as we have outcomes for only those cases where the loans have been sanctioned. Therefore, out of 15,088 applications, 5,347 rejected cases are not considered for training. We use the information provided in loan applications as input variables and a dummy variable representing whether the loan under consideration has ever defaulted as the outcome variable. We then apply a supervised machine learning algorithm. Typically, ML algorithms follow a

80:20 rule, where the algorithm is trained on 80% of the data and tested on 20% of the data. Of course the data points are randomly selected. Accordingly, we use 80% of the overall sample of loans for training and the remaining for out of sample testing. Thus, our training sample has 7,793 and the testing sample has 1,948 borrower level observations.

4 Empirical Strategy And Results

It is crucial to note that the objective of our exercise is to test whether decisions based on an efficient use of hard information by the machine out perform decisions based on a combination of hard and soft information used by the loan officers. If yes, the algorithm can be used as an aid in improving access to finance. Therefore, it is not enough if we train a machine learning algorithm and test it out of sample. It is important to compare the outcomes with decisions made by the loan officers. Therefore, it is also important to explicitly describe the broad economic framework and the set of assumptions that are needed. This is where an exercise aimed at using ML techniques in economic decision making differs from a conventional engineering application.

4.1 The Algorithm

As a first step, we use the XGBoost algorithm, which belongs to the gradient boosted decision trees class. As noted before, we can run the supervised learning model only on the sample of actual loans and not on all loan applications as we have actual outcomes for only those cases where an actual loan was made. As in the case of any ML algorithm, the objective is to estimate a functional relationship between inputs and outputs. The inputs in our case are various loan application characteristics presented in a binary form. The outcome variable that is fed into the algorithm as a part of supervised learning is a dummy variable that takes the value of one if the loan under consideration defaults and zero otherwise. The predicted variable can be interpreted as the probability of loan defaulting, according to the algorithm.

The decision trees work by splitting the input variables into various binary classes. For example, the first split might be whether the loan applicant is a farmer or not. The subsequent splits may be based on other variables such as income, age, and others. Taken to extreme, a leaf can be created for every data point. In other words, the algorithm can be trained on every data point in the training sample and hence it can have prediction for every leaf. Such an algorithm will be an example of perfect overfit in sample, and hence, is unlikely to work out of sample (Mullainathan and Spiess (2017)). To avoid over fitting, 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."

A note on gradient boosting (Chen and Guestrin (2016)) is in order. Gradient boosting is a technique of boosting that uses a gradient descent algorithm to minimize the loss when adding new models. In other words, 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.

To make the model robust, we use a five fold cross validation. In other words, a model created using a part of the training data set is tuned on other four parts and the cycle gets repeated until no more improvements are possible. The process leaves us with predicted risk or what can be loosely interpreted as the probability of default for every observation.

4.1.1 Evaluation of Algorithm Accuracy

AUC ROC Curve

Before proceeding to the comparison between the performance of the algorithm and the loan officers, it is important to test the intrinsic quality of predictions. This step is typical of any ML exercise. We use the commonly used metric known as the area under the ROC[3] curve. The ROC curve is plotted by using two ratios. The first (second) ratio is the ratio between 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 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 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. This is because the ratio between positive among positives and positives among negatives should not change with the risk factor if the risk factors are random. We depict the AUC-ROC curve in figure 1. The AUC works out to close to 77% which is a very high number (Berg, Burg, Gombović, and Puri (2019)). Random guessing would have yielded an AUC of 50% as denoted by the straight line in the figure.

Actual Vs Predicted: To understand how well does the model perform, we compare the actual and the predicted. Figure 2 presents the above comparison. We plot the risk scores provided by the ML algorithm in the horizontal axis and the predicted default risk in the vertical axis. We divide the predicted risk into five bins and plot the average actual default rate within each bin. The test is out of sample. As shown in the figure, higher predicted risk translates into higher actual default rate. The result is reassuring as it shows that the ML algorithm works in the right direction. However, the figure does not say anything about whether the ML algorithm does better or worse than the loan officers.

We then test whether loan officer approval rate is in anyway related to the risk scores

[3]ROC stands for receiver operating characteristic.

generated by the machine. If the risk scores generated by the machine and the risk assessment by officers are directionally similar, we expect to find a negative relationship between the machine generated risk score and the loan approval rate. We plot the association between risk scores predicted by the machine and the loan officer approval rate in figure 3. We divide the predicted risk into ten bins and plot the average actual approval rate within each bin. The risk scores are plotted in the horizontal axis and the approval rates in the vertical axis. We do not find a clean negative relationship between the risk scores and the acceptance rate. Interestingly, as in KLLM, the loan officers have tough time in identifying high risk borrowers. The approval rates are high for both low risk and high risk borrowers. From the above result, at the minimum, it is possible to conclude that the risk model that loan officers apply differs significantly from the assessment of the ML program. The loan officers do a reasonable of job of identifying low risk borrowers. However, it appears that the mismatch between the machine and the loan officers is driven mostly by high risk borrowers.

4.2 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. We look at the problem from the point of view of the bank and not from the point of view of the loan officers. Therefore, we compare the loan decisions from the point of view of maximizing the pay off of the bank and abstract away from the pay offs of loan officers. In other words, we 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.

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

4.2.1 The Selective Labels Problem

Note that for comparison purpose, we need to know the actual outcomes under both the systems of loan decision making. Outcome in this context is whether the loan under consideration has defaulted or not. It is possible to use the actual outcomes for cases where loan officers and the ML algorithm agree. 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 actually accepted by the loan officers. Conversely, the machine may recommend acceptance of cases that are actually rejected by the loan officers. The first category of disagreements do 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 (Lakkaraju, Kleinberg, Leskovec, Ludwig, and Mullainathan (2017), Guidotti, Monreale, Ruggieri, Turini, Giannotti, and Pedreschi (2019)).

One possible solution is imputation using the actual performance of borrowers with similar risk score as those who have missing labels. However, such imputation assumes that the loan applicants accepted and rejected by the loan 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 learn 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. Notice that 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. Consider a case actually rejected by the officer but accepted by the machine. Assume that the risk score assigned to the case is x. It is easy to see that assuming that the above case will have the same outcome as an actually accepted case having a risk score of x is an understatement of risk because the machine does not consider the possibility that the rejected applicant has a higher chance of default. It does not factor in the possibility that the borrower could be heavily alcoholic. Therefore, a policy of imputing outcomes based on risk scores of accepted cases is likely to bias the evaluation procedure in favor of the machine, and eventually lead to machine accepted cases under-performing out of sample.

4.2.2 Contraction of Applications

As we have noted before, for cases rejected by the machine but accepted by loan officers, the counter factual (from the point of view of loan officer's decision) can be ascertained easily. This is because the rejected cases do not default. Therefore, we can ascertain the likely outcome by progressively rejecting actual loans whose applications are considered highly risky by the machines. However, there is no way such a procedure in itself will tell us whether the ML algorithm does better or worse job than the loan officers. For illustration, suppose, we reject top 10% riskiest borrowers and reduce defaults by 20%. However, there is no way to know whether contraction of lending by 10% is an acceptable cost to pay for 20% reduction in defaults.

4.2.3 Using Loan Officer Leniency For Comparison

We then look for ways of comparing the decisions of the loan officers with those of the ML algorithm. We 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), Agarwal and Ben-David (2018), Behr, Drexler, Gropp, and Guettler (2014)). Tzioumis and Gee (2013) show that the loans lent towards the end of the month tend to default more as loan officers compromise on quality to increase quantity. Loan officers in our setting also face lending targets and hence there is reason to believe that the phenomenon noted by Tzioumis and Gee (2013) is likely to manifest in our setting as well.

We first test the above hypothesis that loan officers tend to be more lenient towards the end of the month. To this end, we compare the acceptance rate and default rate between the first and second half of the month. We use the entire sample of 15,088 observations for this purpose. We present the results in Table 2. A lenient regime is likely to be characterized by high acceptance rate and also high default rate. We call the first 15 days "strict"period and the last 15 days "lenient"period. We define acceptance rate in three ways. The definition 1 represents the ratio between the number of loans granted and total loan applications. We find that the ratio is 42% during the strict period and 85% during the lenient period. We then define the ratio based on the relationship between the value of loans granted and applied. When we use the second definition, we find that the ratio is 37% during the strict period and 77% during the lenient period. Notice that the ratios do not differ significantly between definitions. In addition, we look at the loan amount sanctioned. We find that the total loan amount sanctioned during the strict period is Rupees 506 million whereas the same during the lenient period is Rupees 993 million.

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

Observable and Unobservable Differences

As noted in the Introduction, the difference in leniency of loan officers between the first and second half of the month is crucial for identification. As shown by the literature on loan officer incentives (Tzioumis and Gee (2013), Agarwal and Ben-David (2018), Cole, Kanz, and Klapper (2015)), the differences between first half and the second half of the month are unlikely to be related to loan application characteristics but driven by loan officer incentives. 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. We first test the observable part here. Table 3 presents a comparison between strict and lenient period loan applications in terns of observable characteristics such as total assets, income, loan amount applied, occupation, composition of social groups, nominee relationship, number of dependents, and borrower and nominee age. We do not observe any significant difference between the two groups. The results presented in Table 3 show that loan applications do not differ observably in two phases.

We then test for unobservable differences. We use the ML algorithm to do so. We 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 makes reasonable prediction for the strict period, then it is likely that the relationship between inputs and outputs is the same in both the periods. It is crucial to note that although KLLLM use this technique, they are unable to distinguish between unobservable differences and screening ability. In their setting they classify judges as strict and lenient. As they acknowledge, the ML test above will go through even if the lenient judges have superior screening ability but move up the distribution of the unobservable variable because they are lenient. Given that the strict and lenient judges are different, it is not possible in their setting to distinguish between screening ability and other unobservable differences. In our case, it is unlikely that the screening ability changes within the month. 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 is able to test for unobservable differences.

We find that the ML algorithm so developed has a AUC score of 74%, which is not only high but also comparable to the AUC score of our main test. This shows that the algorithm trained on lenient loans does a good job predicting outcomes on the sample of strict loans. Further, we plot the above predictions against actuals in figure 4. We plot the risk factors in the horizontal axis and the actual default rates on the vertical axis. As can be seen, the line is close to 45%. This shows that the actuals and the predicted are close. In other words, the relationship between the inputs and outputs is similar during both the periods. Given the absence of difference in screening ability, the above result can be attributed to the lack of unobservable differences.

4.2.4 The Actual Comparison

We present the results in Table 4. Note that, as explained in Section 2, the table is based on the test sample of 1,948 observations. Therefore, the approval rates are not exactly same as the rates presented in Table 2. The loan amount is expectedly lower in this table as we consider only 20% of the cases. However, loan approval rates are comparable to the overall sample, although slightly different. This is expected as the training and test samples are selected randomly.

In row 1, we define leniency by the actual amount of loan lent. We find that the amount lent (default amount) in strict period is Rupees 78 (15.8) million whereas the same for the lenient period is Rupees 153.4 (52.7) million. Column 5 shows that by following the contraction procedure described above the default amount can be brought down to Rupees 11.6 million even after maintaining the approval amount at Rupees 78.36 million. This is 26.6% lower than the actual strict period default amount of Rupees 15.8 million.

Column 6 shows that with a Rupees 58.3 million reduction in approval rate from the Rupees 153.4 million level of the lenient period to Rupees 95.1 million, the machine is able to achieve the default level of the strict period, that is, 15.8 million. Note that the the actual approved amount during the strict period is Rupees 78.36 million. In other words, the ML algorithm is able increase lending rate by close to Rupees 16.74 million and yet maintain the NPA amount of 15.8. Rupees 16.74 million is 21.36% higher than the actual loan amount lent during the strict period.

In row 2, we consider the loan approval rate in terms of number applications accepted over total applications. The strict period approval (default) rate is 42% (19%) whereas the same for the lenient period is 85% (32%). We start with the lenient period loans and ask what happens to default rate if the lenient period approval rate is brought down from 85% to 42% by rejecting the most risky cases as identified by the ML algorithm. Column 5 shows that by following the contraction procedure described above the default rate can be brought down to 15% even after maintaining the approval rate at 42%. This is 21.1% lower than the actual strict period default rate of 19%. Column 6 shows that with a 30 percentage points reduction in approval rate from the 85% level of the lenient period to 55%, the machine is able to achieve the NPA level of the strict period. Note that the the actual approval rate during the strict period is 42%, which is significantly lower than the 55% level, the machine can potentially achieve. In other words, the ML algorithm is able to increase lending rate by close to 30.9% and yet maintain the NPA rate of 19%.

In row 3, we define the loan approval rate in terms of the ratio between the amount of loan granted and applied. The result remains directionally similar. The strict period approval (default) rate is 37% (19%) whereas the same for the lenient period is 77% (32%). Column 5 shows that by following the contraction procedure described above the default rate can be brought down to 13% even after maintaining the approval rate at 37%. This is 31.6% lower than the actual strict period default rate of 19%. Column 6 shows that with a 26 percentage points reduction in approval rate from the 77% level of the lenient period to 51%, the machine is able to achieve the default level of the strict period. Note that the the actual approval rate during the strict period is 37%, which is 40 percentage points lower than the approval rate of the lenient period. In other words, the ML algorithm is able increase lending rate by close to 37.8% and yet maintain the NPA rate of 19%.

The results clearly show that the use of the Ml algorithm improves the overall efficiency of lending and helps increase lending without increase in default rates.

4.3 Possible Concerns

4.3.1 Are the differences significant ?

Prima facie, the differences presented in table 4 appear economically meaningful. We attempt to measure the significance of the difference in a statistical sense. We recognize that coefficients generated by the ML algorithm may not be consistent (KLLLM). Therefore, we create an empirical distribution by bootstrapping. For this purpose, we start with the lenient portfolio and reject applications randomly. In other words, the contraction procedure here is not guided by the ML algorithm. We stop when the approval rate of the lenient period equals the approval rate of the strict period. We then calculate the difference between the actual default rate during the strict period and the default rate achieved by the placebo procedure. We call this difference the placebo difference. We repeat this exercise 10,000 times and plot the placebo differences in figures 5, 6, and 7. We use the three definitions of approval rate in the three figures. The dotted line represents the boundary of 95% of the area. The thick line denotes the actual difference. As can be seen from the figures, the actual difference is way beyond the 95% cut-off. Therefore, it is likely that the difference is significant even in a statistical sense.

4.3.2 Impact On Profits

A skeptic may argue that the contraction procedure achieves lower defaults by leaving out loans on which a loan officer 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 lower default rate in itself is beneficial to the bank.

We cannot calculate the overall profitability of a loan as we do not have information about the eventual recovery in case of defaults. Therefore, we address the above concern in an indirect way. We ask whether the loan officers actually charge higher interest rate on loans rejected by machines but accepted by the officers by estimating the following regression equation:

$$ML\_Rejected_{ijt} = \beta_0 + \beta_j + \beta_t + \beta_1 * Interest\_Rate_{ijt} + \beta \cdot X_{ijt} + \varepsilon_{ijt}, , \tag{1}$$

The date are organized at a loan i level. The dependent variable–$ML\_Rejected_{ijt}$–is a dummy variable that takes the value of one for loans that are rejected by the ML algorithm but accepted by loan officers, and zero otherwise. The explanatory variable $Interest_{ijt}$ is the annualized interest rate charged on a loan i given in a location j during time t. β_j represent location level fixed effects, and β_t represents month X year fixed effects.

The results are presented in Table 5. We conduct this exercise on the test data set. We restrict the sample to loans lent during the lenient period as the contraction exercise is done using the lenient portfolio. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Section 4.2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates.

We ask whether high interest rates explain rejection by the machine. We do not find a strong association between interest rates and the machine's decision to reject. Further, in Table A.1 of the online appendix, we include the entire test sample. Here too, we do not find a strong association between interest rates and rejection by the ML algorithm in the course of contraction. Therefore, it is unlikely that the machine achieves lower level of default by selectively omitting cases on which the loan officer would have charged higher interest rates. Therefore, the portfolio created by the machine is likely to be more profitable as it lends at almost the same interest rate as charged by the officer but, at the same time, achieves lower default rates.

4.4 Impact On Equity

The analysis so far completely ignores equity. It is critical to ask whether the machine is achieving efficiency at the expense of equity. As noted in Section 2, the constitution of India classifies historically disadvantaged communities scheduled castes and scheduled tribes (SCSTs), and provides for affirmative action. We use this classification and ask whether the ML algorithm discriminates against SCSTs ? We conduct several tests.

4.4.1 Are SCSTs More Likely to be Rejected By Machines ?

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

$$ML\_Rejected_{ijt} = \beta_0 + \beta_j + \beta_t + \beta_1 * SC\_ST_{ijt} + \beta \cdot X_{ijt} + \varepsilon_{ijt},, \tag{2}$$

The date are organized at a loan i level. The dependent variable–$ML\_Rejected_{ijt}$–is a dummy variable that takes the value of one for loans that are rejected by the ML algorithm but accepted by loan officers, and zero otherwise. The explanatory variable $SCST_{ijt}$ is a dummy variable that takes the value of one if the borrower under consideration belongs to SC or ST community, and zero otherwise. All other terms have the same meaning as in equation 1.

We report the results in Table 6. We consider the three acceptance rates as defined in Section 4.2. We find that the social status of the borrower does not explain rejection by the machine. Further, we retrain the machine without feeding in information about the social status of the person. We do this to check whether the machine learns about the social status by using other information as in Fuster, Plosser, Schnabl, and Vickery (2019). We repeat the above analysis and present the results in Table A.2 of the online appendix. We do not find any evidence of discrimination. SCSTs are no more likely to be rejected by the machine when compared to other borrowers.[4]

[4]On similar lines, we conduct the main contraction exercise as in Table 4 using an algorithm that does

4.4.2 Are SC-STs Assigned Higher Risk Scores ?

Further, we explicitly test whether the algorithm assigns higher risk scores to SC STs. The test is intended to capture discrimination that might have gone unnoticed due to the nature of the contraction of exercise. In other words, even if SC STs are assigned higher risk score than a comparable non SC ST, they will not be eliminated by the algorithm as long as they are below the cut-off. We estimate regression equation 2 using the risk scores as the dependent variable. We present the results in Table A.4 of the online appendix. We find that the machine is unlikely to assign higher risk score to borrowers belonging to the SC or ST communities.

4.4.3 Contraction With Constraints?

Given the results presented in Section 4.4.1 and 4.4.2, it appears that a contraction exercise which is explicitly constrained to maintain the SC ST proportion should not do significantly worse than the procedure without such constraints. We present the results in Table 7. Here we include a constraint the the proportion of SC-STs selected cannot be lower than the proportion in the actual portfolio of the strict period. In other words, the actual strict period portfolio and the synthetic portfolio do not differ with respect to the proportion of SC-ST borrowers. The results show that the constraint does not make much difference to the performance of the algorithm.

Notice that the default rates presented in rows 1 and 2 of Table 7 are marginally lower than the default rate shown in rows 1 and 2 of Table 4. One may wonder how is that the default rate with constraints is lower than the default rate without any constraints. First of all, it is important to note that the differences are marginal (11.55% in row 1 of Table 4 and 11.1% in row 1 of Table 7). Nonetheless, an explanation is in order here. Note that the ordering of cases for contraction is made based on risk scores assigned by the ML algorithm and not based on default rates. Although default rates and risk scores are linearly related,

not see the borrowers' social category status. We present the results in Table A.3 of the online appendix. The results are directionally similar to the results presented in Table 4.

as shown in figure 2, the ordering is not perfect. It is possible, in some cases, that ex-post, a borrower with lower risk score may have a higher default rate. Therefore, it is possible that the default rate with constraints can be slightly better than the default rate without them when constraints do not have significant impact.

4.4.4 Contraction Considering Only SC-STs

We now examine the performance of the algorithm within the SC-ST sample. We restrict the sample to SC-ST borrowers and perform the contraction exercise as described in Section 4.4.3. If the algorithm performs reasonably well within this sample, the hypothesis that the algorithm is equitable gets strengthened. Such a result would imply that algorithm not only does a reasonable job between SC-STs and others but also within SC-STs.

We preset the results in Table 8. As shown in the table, the algorithm is able to reduce the loan amount in default from Rupees 1.77 million to Rupees 1.43 million without change in volumes. It is also capable of increasing the volume of lending from Rupees 6.93 million to Rupees 8.31 million. We find similar results when we consider the remaining two definitions that look at approval rates instead of amounts. From the above results it is reasonable to conclude that the algorithm works reasonably well within the SC-ST population.

4.4.5 Increasing Lending To SCSTs

Here we ask whether some of the gains in efficiency can be used to enhance equity. In other words, we perform the contraction procedure by increasing the approval rate of SC-STs. Our goal is to arrive at the point at which equity completely absorbs efficiency. At that point, the synthetic portfolio will perform exactly the same as the actual strict portfolio, both in terms of default and loan amount. In this exercise, we are constrained by the possibility that we may exhaust all SC-ST cases available. Recall that we can only use borrowers who actually received a loan during the lenient period. We cannot draw from the rejected application due to the possibility that the rejected borrowers may be systematically different.

We present the results of the contraction exercise in Table 9. Notice that despite maximizing on lending to SC-STs, the algorithm actually does better than the strict period, both in terms of default and the volume of loans. For instance, the algorithm is able to lend Rupees 89.2 million as against Rupees 78.36 million in the strict period while keeping the default rate unchanged. Similarly, the algorithm is able to achieve an amount in default of Rupees 14.2 million as against the strict period's amount in default of Rupees 15.8 million, despite keeping the lending volume unchanged. We find similar results when we look at the other two measures of acceptance rate and default rate. The improvement in outcomes indicate that we hit the limit in terms of maximum possible lending to SC-STs from the pool of loans lent during the lenient period.

4.4.6 Comparison In Terms of Lending to SC-STs

In Table 10, we compare the amount of loan lent to SC-STs and proportion of SCST borrowers in the portfolio of (i) strict officers (ii) the ML algorithm with no constraints (iii) the ML algorithm that seeks to maximize lending to SC-STs. There is not much difference between the actual strict period portfolio and the unconstrained synthetic portfolio. When we compare the first two columns, we find that the SC-ST proportion of the synthetic unconstrained portfolio in terms numbers is 0.2% higher while the same in in terms of value is 0.1% lower. In terms of amount, it is Rupees 0.15 million lower.

Column 3 shows that the actual lending to SC-STs can be significantly increased without any change in default rates. The proportion in terms of number of SC-STs over the total number of borrowers can be increased from 9.3% to 10.4% without any increase in the default rate. If the above proportion is calculated in terms of value, the proportion can go up from 8.5% to 9.7%. Finally, there is scope to increase the absolute amount of lending from Rupees 6.93 million to Rupees 15.89 million, without any increase in default rate.[5]

[5] There is further scope for improvement. However, as shown in Table 9, we hit the limit in terms of available SC-ST loans lent during the lenient period.

4.5 Building A Decision Aid

So far, all our tests attempted to compare the loan decisions of the machine with those of the loan officers. Here, we attempt to build a decision aid for the loan officer, following KLLLM. The attempt is to understand the outcome of a policy where strict period lending is based only on the risk factors calculated by the ML algorithm. Mechanically, the exercise is straight forward. We first organize all loan applications received during the strict period in descending order of risk as identified by the ML algorithm. Next, we reject the riskiest of the applications one by one until we reach a point where the approval rate equals the actual approval rate during the strict period. We repeat this exercise using all three definitions of approval rate.

While implementing the above procedure is straight forward, evaluating its performance is not. The reason is easy to see. There could be loan applications that are actually rejected by the loan officer but accepted based on the above procedure as the risk score assigned to them is not very high. Such applications obviously do not have actual outcomes. Some sort of imputation is required. It is important to clearly specify the assumptions made because the rejected and accepted cases may be un-observably different. Following KLLLM, we make an assumption that upto the point of acceptance rate of lenient period, the relationship between actual default rate and risk factors between strict and the lenient period is the same. Recall that figure 3 actually shows this relationship and the AUC score of an algorithm trained on lenient period and tested on strict period is 74%. Beyond the approval rate of lenient period, we assume that all loans will default.

An example is in order here. Assume that the strict period approval rate is 50% and the lenient period approval rate is 70%. Also assume that using the procedure outlined above 25% cases are actually rejected during the strict period but approved by the machine. Remaining 25% cases have actual outcomes. In this case, we first impute the outcome for 20% (70%-50%) based on the actual outcomes of the loan lent during the lenient period. For this, we use the actual outcomes of the loans that are between 25% and 45% of the lenient period. We make the stringent assumption that the outcomes are missing for the last 25% out of the 50% lent during the strict period and hence impute based on the outcome of cases that between 25% and 50% of the lenient period portfolio. For the last 5% of the cases, we assume that everyone defaults.

We present the results using the above procedure in Table 11. We use the three definitions of approval rates as before. In columns 1 and 2, we present the actual approval rate and the default rate of the strict period. Consider row 1, where we define approval rate as the number of applications accepted to total loan applications. The approval rate and default rate are 42% and 19%, respectively. In column 3, we show that for 27.2% of the cases, we have actual outcomes as these were actually lent during the strict period and the default rate of these loans is 8.2%. We impute outcomes for remaining 14.8% cases based on the procedure described above and find the default rate on the sample to be 24.3%. Column 7 shows that the overall default from the above exercise turns to be 12.4%, which is 34.8% lower than what was actually achieved (19%). We obtain similar results even when we consider the loan amount. We present these results in row 1. Out of the Rupees 78.36 million actually lent, actual outcomes are available for loans worth Rupees 58.4 million and imputed for the remaining Rupees 23.6 million. The imputation procedure leads to a reduction of Rupees 5.99 million in loan defaults, which works to be an economically meaningful–38%. We find similar results even when we define approval rate based on the value of the applications accepted to total value of loan applications. Overall, it appears that the decision aid based on an ML algorithm outperforms the loan officers significantly.

5 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 the human being. Our tests do not answer the question of whether ML does better than humans. The reason is straightforward. The performance of a loan depends on both screening as well as monitoring by the loan officers (Diamond (1984)). The ML tool that we have designed and tested in this study does only the screening part. In fact, our 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 our findings a case for machine outperforming humans in the context of loans can be made only in cases where we are sure that the monitoring effort is zero. Given our data limitations, such a strong assumption looks unrealistic. As well, it is also important to note that our algorithm will fail if the effectiveness of monitoring is linked to screening. However, given that banks regularly practice job rotations (Hertzberg, Liberti, and Paravisini (2010)), it may not always be the case that screening and monitoring are inseparably connected.

6 Conclusion

In this study, we ask whether ML based algorithms can be used to aid loan officers in improving access to formal credit to the poor. Loan application processing by loan officers of a bank, prima facie, 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 machine learning 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 large bank in India, we attempt to answer the above question. Any reduction in cost of financial intermediation is likely to enhance access to formal credit.

In the process, we encounter the selective labels problems as we have actual outcomes only for those cases which were actually selected by the machine. We resolve the selective labels problem by exploiting the fact that, driven by incentives, the loan officers tend to be relatively lenient at the end of a month when compared to the beginning of a month. Using the lenient period portfolio as the base and strict period portfolio as the benchmark, we find that rejecting loan applications which are considered most risky by the ML algorithm but accepted by loan officers improves the performance of the portfolio in terms of default rate, profitability, and overall quantity of lending. We also build a decision aid for the loan officers. Finally, we also verify that the ML algorithm maintains its out performance even after taking into account any equity considerations that the bank may have. In fact, we show that lending to disadvantaged sections can be increased without negatively impacting loan portfolio quality.

These findings, we 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 policy objective of emerging market governments. Given that the quality of hard information itself is suspect, the lenders rely heavily on soft information. This works as a limiting factor. Our findings show that an efficient use of available hard information can improve loan performance and can be used to increase lending without significant increase in default rates.

Figure 1: AUC ROC Curve

This figure depicts the AUC ROC Curve.

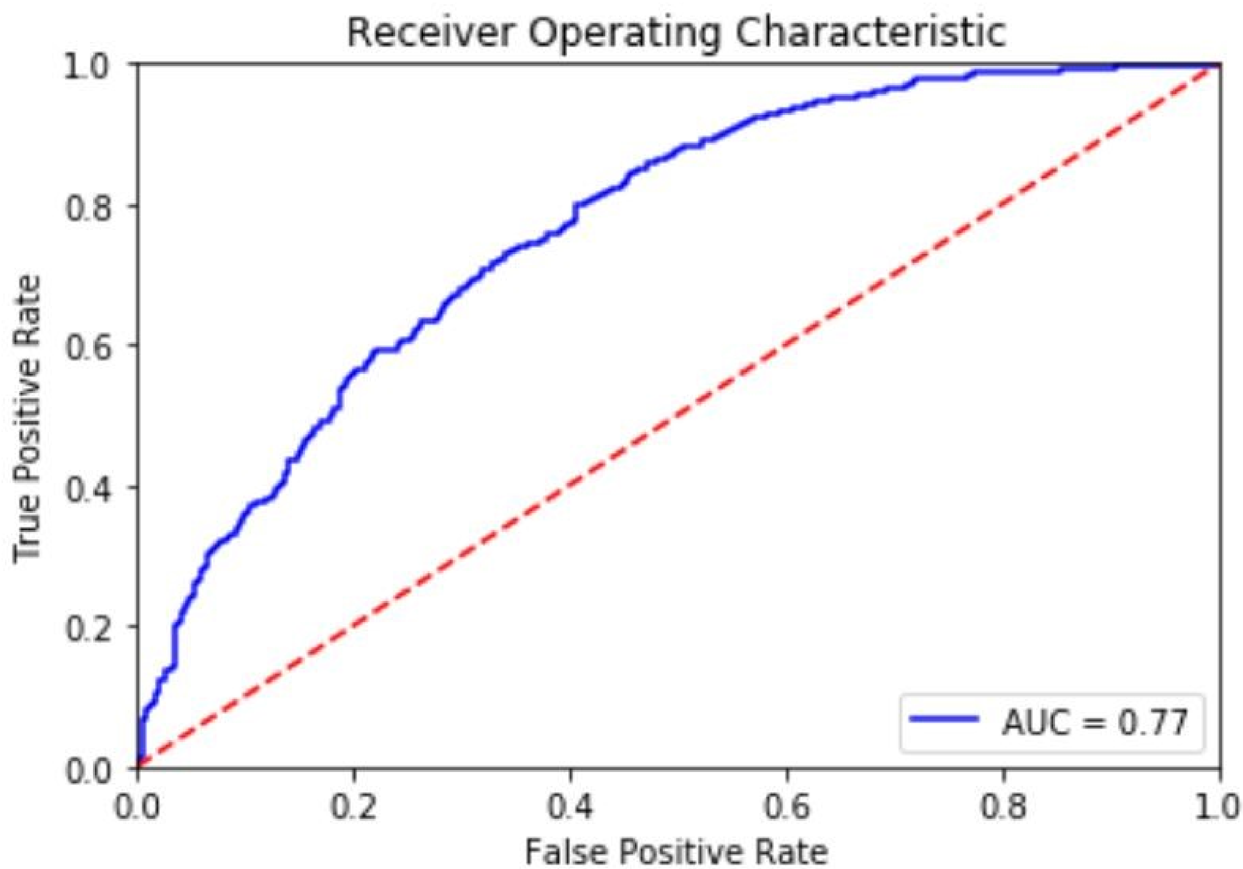

Figure 2: Risk Score And Actual Default

This figure plots predicted risk scores in the horizontal axis and proportion of default in the vertical axis. Risk scores are categorized into five buckets.

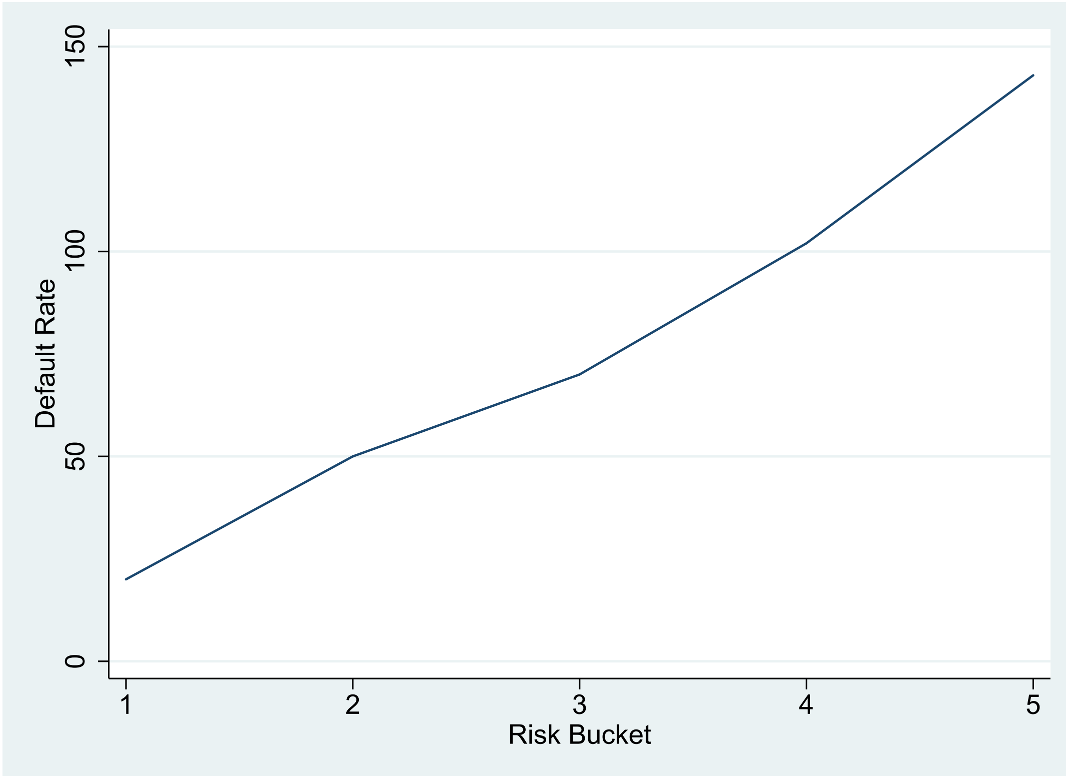

Figure 3: Risk Score And Loan Officers' Selection

This figure plots predicted risk scores in the horizontal axis and the approval rate in the vertical axis. Risk scores are categorized into buckets.

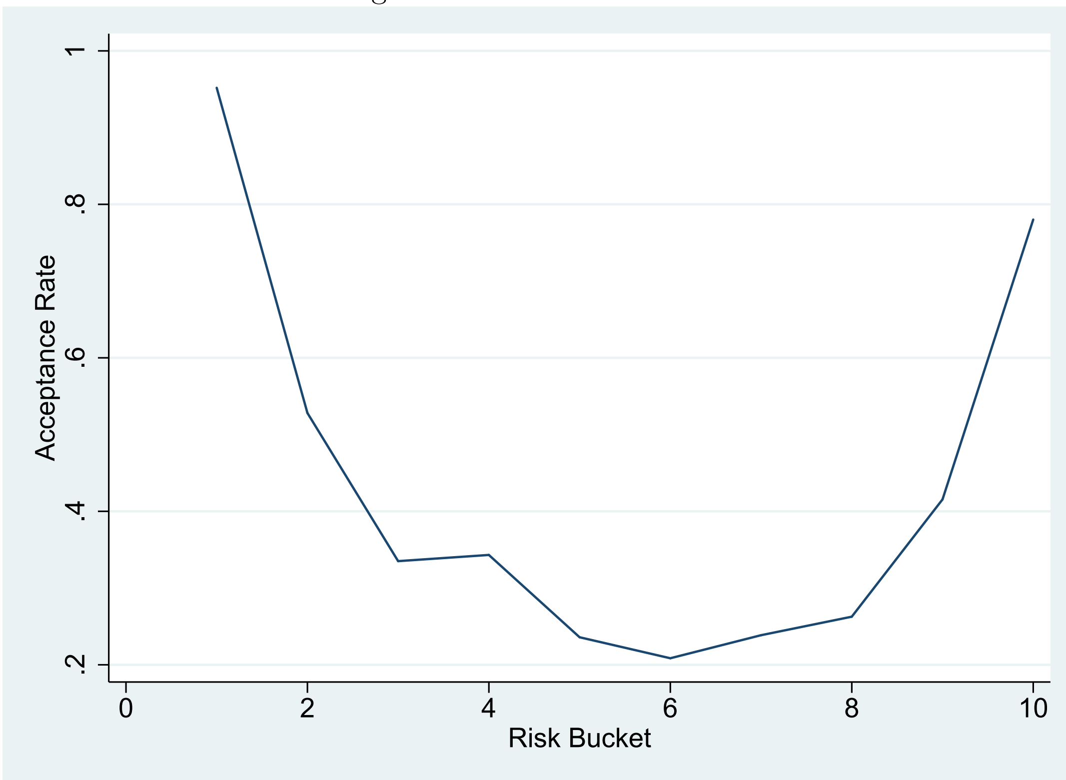

Figure 4: Testing For Unobservables

This figure plots predicted risk scores in the horizontal axis and proportion of default in the vertical axis. Risk scores are categorized into five buckets. The algorithm has been trained on data from the lenient period and tested on the strict period.

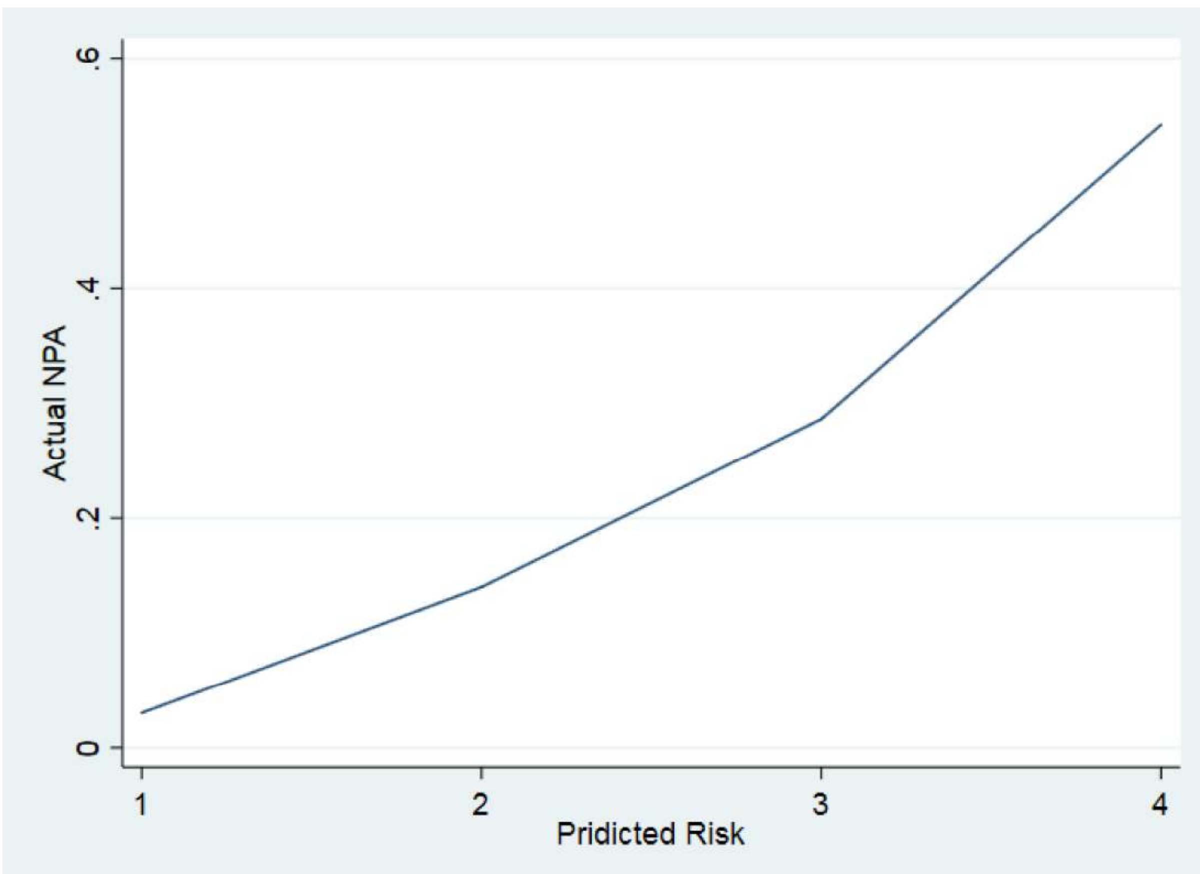

Figure 5: Empirical Distribution - Approval Rate Based on Number of Application

This figure plots the empirical distribution of the difference in default rate between the placebo strict period obtained by randomly rejecting loans from the lenient period and the actual strict period. The bootstrapping procedure is done 1,000 times. Here, the approval rate is defined as the ratio between the number of applications approved and the number of applications received. The dotted line represents 95% of the distribution. The thick line represents the actual difference obtained by following the ML based contraction procedure.

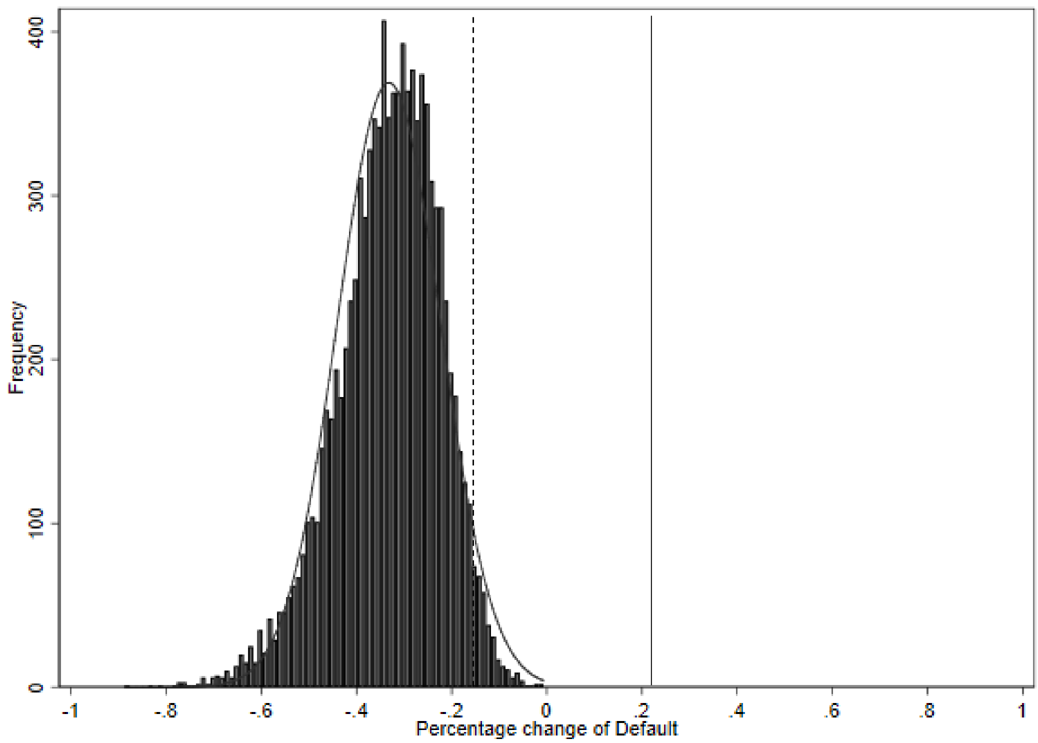

Figure 6: Empirical Distribution - Approval Rate Based on the Rupee Value of Applications

This figure plots the empirical distribution of the difference in default rate between the placebo strict period obtained by randomly rejecting loans from the lenient period and the actual strict period. The bootstrapping procedure is done 1,000 times. Here, the approval rate is defined as the ratio between the value of applications approved and the value of applications received. The dotted line represents 95% of the distribution. The thick line represents the actual difference obtained by following the ML based contraction procedure.

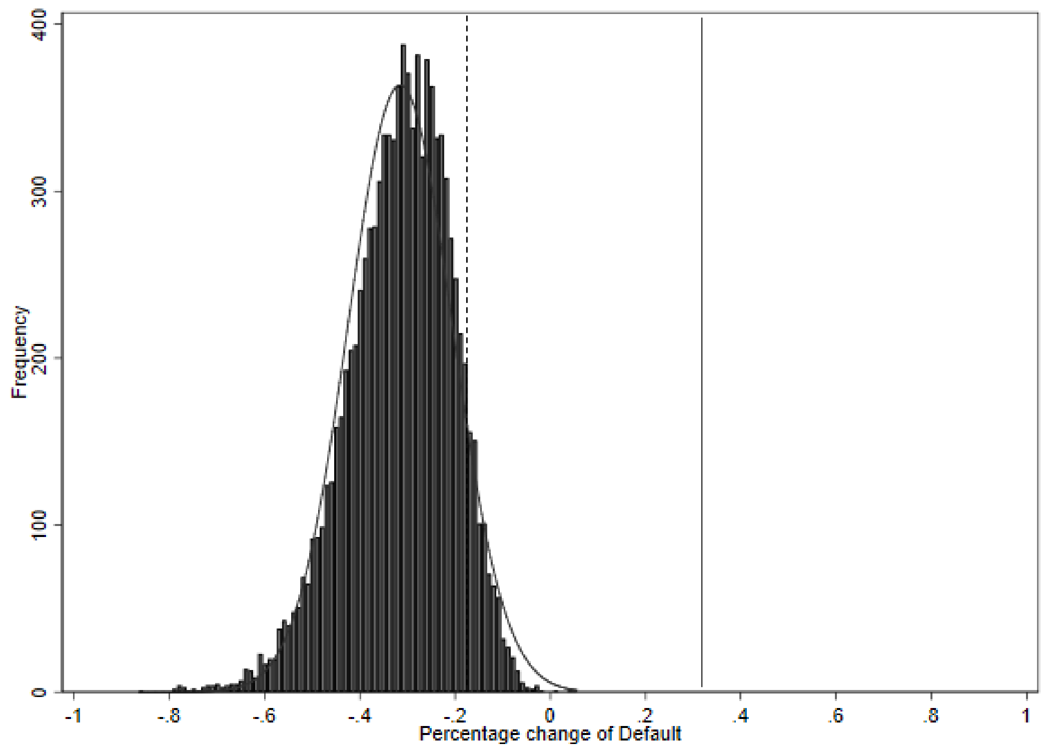

Figure 7: Empirical Distribution - The Actual Amount of Loans Approved
This figure plots the empirical distribution of the difference in default amount between the placebo strict period obtained by randomly rejecting loans from the lenient period and the actual strict period. The bootstrapping procedure is done 1,000 times. Here, the approval rate is defined as the actual amount of loans approved. The dotted line represents 95% of the distribution. The thick line represents the actual difference obtained by following the ML based contraction procedure.

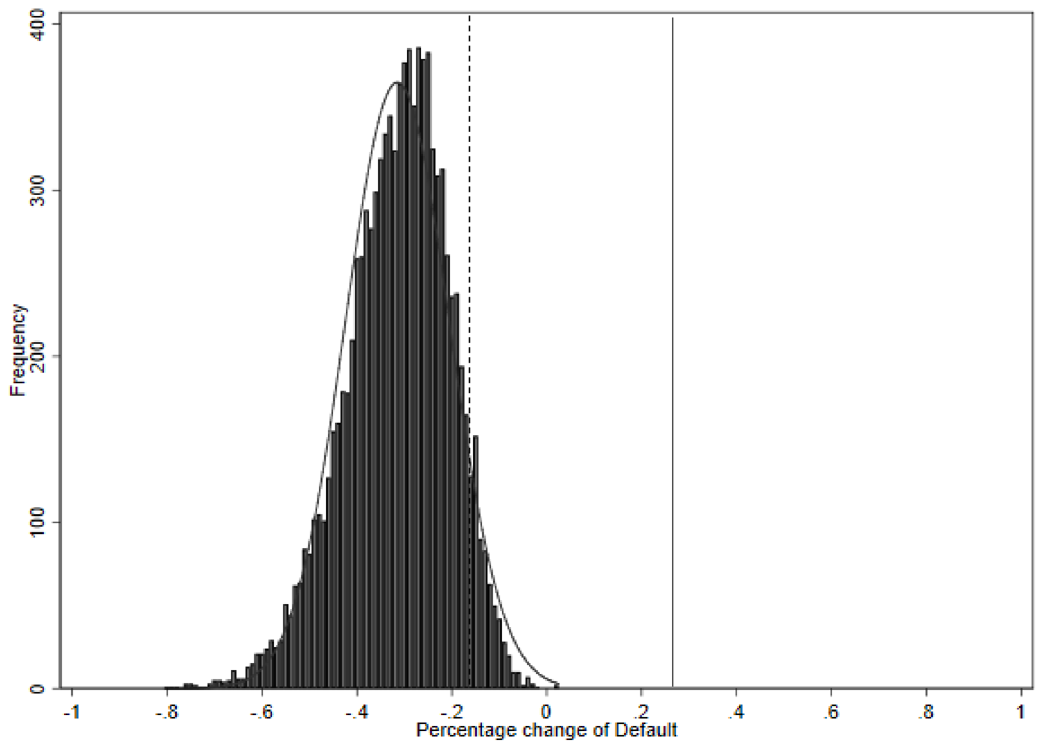

TABLE 1: Sample Construction And Summary

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

	Full Sample	Approved	Rejected	p-value
Sample Size	15,088	9,741	5,347	
Approval Rate	64.56	100.00	0.00	
Loan Information				
States	10			
Cities	196			
Time Period	2017-2019			
Average Loan Tenure	35 months			
Average repayment data covered	12 months			
Outcome Variable				
Default Proportion	29.51			
Average Loan Amount	157,359			
Average Interest Rate	13.69%			
Applicant Characteristic				
Log of Total Assets	11.93	12.07	11.67	0.000
Log of Annual Income	13.19	13.22	13.13	0.000
Occupation				
Business	0.72	0.70	0.74	0.000
Farmers	0.18	0.16	0.22	0.000
Others	0.11	0.14	0.05	0.000
Caste				
General	0.64	0.61	0.69	0.000
Other Backward Class	0.25	0.28	0.19	0.000
Schedule Caste / Schedule Tribe	0.11	0.11	0.12	0.273
Nominee Relationship				
Spouse	0.26	0.31	0.17	0.000
Parents	0.05	0.05	0.04	0.000
Others	0.69	0.63	0.79	0.000
Gender (Female)	0.23	0.23	0.23	0.492
Dependent Members	2.54	2.57	2.49	0.000
Customer Age	35.46	35.51	35.38	0.352
Nominee Age	42.84	41.58	46.70	0.058

TABLE 2: Comparison Between Strict And Lenient Periods

In this table, we compare the strict and lenient periods in terms of acceptance rate and loan performance. Here, we use the full sample of 15,088 borrowers.

	All	Strict	Lenient
Acceptance Rate In Terms of Numbers	0.65	0.42	0.85
Acceptance Rate In Terms of Value	0.59	0.37	0.77
Total Loan Amount Sanctioned (In Rupees Million)	1499	506	993

TABLE 3: Comparison Between Strict And Lenient Regimes

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

Comparison: Strict and Lenient Period				
	All	Stritct	Lenient	P-value
Log of Total Assets	11.93	11.93	11.93	0.888
Log of Annual Income	13.19	13.19	13.18	0.475
Occupation				
Business	0.72	0.71	0.72	0.705
Farmer	0.18	0.18	0.18	0.676
Others	0.11	0.11	0.10	0.285
Caste				
General	0.63	0.63	0.63	0.588
Other Backward Class	0.25	0.25	0.25	0.672
Schedule Caste / Schedule Tribe	0.11	0.11	0.11	0.704
Nominee Relasionship				
Spouse	0.26	0.27	0.26	0.142
Parents	0.05	0.05	0.05	0.763
Others	0.69	0.69	0.69	0.208
Gender (Female)	0.23	0.23	0.23	0.876
Dependent Members	2.54	2.54	2.54	0.939
Customer Age	35.46	35.52	35.40	0.362
Nominee Age	42.84	42.16	43.56	0.549

TABLE 4: COMPARISON BASED ON LOAN APPROVAL RATE AND DEFAULT USING CONTRACTION

In this table, we present the results of the contraction procedure. In columns 1 and 2, we present the loan acceptance rates of strict and lenient periods respectively. We use the three measures of acceptance rates as described in Table 2. In columns 3 and 4, we present default rates under different definitions of acceptance rates and for strict and lenient periods separately. In column 5, we show the default rate that would result by following the ML algorithm and keeping the approval rate unchanged. In column 6, we show the amount of lending achievable by the ML algorithm by maintaining the actual default rate of the strict period.

Table: Contraction

	Volume		Default		Machine	
	Strict	Lenient	Strict	Lenient	Default	Volume
Loan Amount lent (Rupees Million)	78.36	153.40	15.80	52.70	11.60	95.10
Loan Approval Rate	0.42	0.85	0.19	0.32	0.15	0.55
Loan Amount proportion	0.37	0.77	0.19	0.32	0.13	0.51

TABLE 5: DO LOAN OFFICERS CHARGE HIGHER INTEREST RATES FOR RISKY CASES ?

In this table, we 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. We 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 one if the loan under consideration is rejected by the machine and zero otherwise. Interest rate is the rate charged by the officer. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Table 2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates. ***, **, * represents statistical significance at the 1%, 5%, and 10% levels.

	(1)	(2)	(3)	(4)	(5)	(6)
	Acceptance Rate		Loan Proportion		Loan Amount	
Dependent Variable	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected
Interest Rate	6.14	3.97	9.08	7.05	5.90	3.94
	[0.96]	[0.62]	[1.43]	[1.08]	[0.91]	[0.61]
Total Assets		-0.00		-0.00		-0.00
		[-0.22]		[-0.22]		[-0.01]
Annual Income		-0.00		-0.00		-0.00
		[-1.25]		[-1.41]		[-1.33]
Loan Amount Asked		0.00*		0.00*		0.00*
		[1.76]		[1.73]		[1.81]
Borrower Age		-0.00		-0.00		-0.00
		[-0.75]		[-0.67]		[-0.49]
Constant	-0.33	-0.06	-0.72	-0.47	-0.31	-0.08
	[-0.38]	[-0.07]	[-0.83]	[-0.53]	[-0.34]	[-0.09]
Fixed Effects						
Month Year	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,029	1,029	1,029	1,029	1,029	1,029
R-squared	0.34	0.34	0.35	0.35	0.33	0.33

TABLE 6: Do Machines Discriminate ?

In this table, we test whether the machines discriminate based on social status. The data are organized at a loan level. We 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 one if the loan under consideration is rejected by the machine and zero otherwise. SC-ST is a dummy variable that takes the value of one if the borrower belongs to the SC or ST community, and zero otherwise. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Table 2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates. ***, **, * represents statistical significance at the 1%, 5%, and 10% levels.

	(1)	(2)	(3)	(4)	(5)	(6)
	Acceptance Rate		Loan Proportion		LoanAmount	
Dependent Variable	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected
SC ST	-0.06	-0.06	0.06	0.06	0.06	0.06
	[-1.08]	[-1.09]	[1.09]	[1.10]	[1.08]	[1.09]
Total Assets		0.00		-0.00		-0.00
		[0.26]		[-0.27]		[-0.05]
Annual Income		0.00		-0.00		-0.00
		[1.04]		[-1.16]		[-1.12]
Loan Amount Asked		-0.00*		0.00**		0.00**
		[-2.00]		[2.05]		[2.05]
Borrower Age		0.00		-0.00		-0.00
		[0.81]		[-0.77]		[-0.54]
Constant	0.50***	0.53***	0.52***	0.48***	0.50***	0.45***
	[83.10]	[10.84]	[89.51]	[9.76]	[82.25]	[9.32]
Fixed Effects						
Month Year	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,029	1,029	1,029	1,029	1,029	1,029
R-squared	0.34	0.34	0.35	0.35	0.33	0.33

TABLE 7: CONTRACTION MAINTAINING SC-ST PROPORTION

In this table, we present the results of the contraction procedure. We impose a condition that the proportion of SC-STs should be maintained. In columns 1 and 2, we present the loan acceptance rates of strict and lenient periods respectively. We use the three measures of acceptance rates as described in Table 2. In columns 3 and 4, we present default rates under different definitions of acceptance rates and for strict and lenient periods separately. In column 5, we show the default rate that would result by following the ML algorithm and keeping the approval rate unchanged. In column 6, we show the amount of lending achievable by the ML algorithm by maintaining the actual default rate of the strict period.

	Volume		Default		Machine	
	Strict	Lenient	Strict	Lenient	Default	Volume
Loan Amount lent (Rupee Million)	78.36	153.40	15.80	52.70	11.1	95.10
Loan Approval Rate	0.42	0.85	0.19	0.32	0.14	0.56
Loan Amount proportion	0.37	0.77	0.19	0.32	0.16	0.48

TABLE 8: Contraction Considering Only SC-STs

In this table, we present the results of the contraction procedure. We restrict the sample to loans lent to SCs and STs. In columns 1 and 2, we present the loan acceptance rates of strict and lenient periods respectively. We use the three measures of acceptance rates as described in Table 2. In columns 3 and 4, we present default rates under different definitions of acceptance rates and for strict and lenient periods separately. In column 5, we show the default rate that would result by following the ML algorithm and keeping the approval rate unchanged. In column 6, we show the amount of lending achievable by the ML algorithm by maintaining the actual default rate of the strict period.

	Volume		Default		Machine	
	Strict	Lenient	Strict	Lenient	Default	Volume
Loan Amount lent (Rupees Million)	6.93	15.80	1.77	6.51	1.43	8.31
Loan Approval Rate	0.37	0.90	0.27	0.40	0.18	0.60
Loan Amount proportion	0.32	0.81	0.26	0.41	0.20	0.49

TABLE 9: CONTRACTION- MAXIMIZE LENDING TO SC-STS

In this table, we present the results of the contraction procedure. Here, we try to maximize lending to SCSTs without doing any worse than than strict portfolio in terms of defaults and lending. In columns 1 and 2, we present the loan acceptance rates of strict and lenient periods respectively. We use the three measures of acceptance rates as described in Table 2. In columns 3 and 4, we present default rates under different definitions of acceptance rates and for strict and lenient periods separately. In column 5, we show the default rate that would result by following the ML algorithm and keeping the approval rate unchanged. In column 6, we show the amount of lending achievable by the ML algorithm by maintaining the actual default rate of the strict period.

Table: Contraction with maximisng SC_ST

	Volume		Default		Machine	
	Strict	Lenient	Strict	Lenient	Default	Volume
Loan Amount lent (Rupees Million)	78.36	153.4	15.8	52.7	14.2	89.2
Loan Approval Rate	0.42	0.85	0.19	0.32	0.18	0.48
Loan Amount proportion	0.37	0.77	0.19	0.32	0.18	0.46

TABLE 10: Comparison Based On Lending To SC-STs

In this table, we compare the actual strict period, unconstrained contraction, and contraction that aims to maximize lending to SC-STs in terms of lending to SC-STs. In row 1 (2), we consider loan acceptance rate in terms of numbers (value) of loans. In row three, we consider the loan amount.

Method	**Actual Strict Period**	**Simple Contraction**	**Maximizing SCST**
Acceptance Rate	0.093	0.095	0.104
Loan Proportion	0.085	0.084	0.097
Loan Amount	6.93	6.78	15.89

TABLE 11: DECISION AID

In this table, we present the results from a decision aid that is unconstrained. Rows are organized as in Table 4. In columns 1 and 2, we present the actual approval rates and default rates of strict periods. In columns 3 and 4, we present the proportion and default rates for those borrowers who are selected by both the officer and machine. In columns 5 and 6, we present the proportion and default rates for those borrowers who are not selected by the officers but selected by the machine. In columns 7 and 8, we present the overall numbers for the machine by adding up the two categories.

Method	Strict Agent		Actual Identfied		Imputed		Machine	
	Volume	Default	Volume	Default	Volume	Default	Volume	Default
Loan Amount lent (Rupee Million)	78.36	15.80	58.400	5.180	23.600	4.630	78.36	9.810
Loan Approval Rate	0.42	0.19	0.272	0.082	0.148	0.243	0.42	0.124
Loan Amount proportion	0.37	0.19	0.251	0.088	0.120	0.269	0.37	0.147

References

Agarwal, S., G. Amromin, I. Ben-David, S. Chomsisengphet, and D. D. Evanoff (2014): "Predatory lending and the subprime crisis," *Journal of Financial Economics*, 113(1), 29–52.

Agarwal, S., and I. Ben-David (2018): "Loan prospecting and the loss of soft information," *Journal of Financial Economics*, 129(3), 608–628.

Agarwal, S., and R. Hauswald (2010): "Distance and private information in lending," *The Review of Financial Studies*, 23(7), 2757–2788.

Agrawal, A., J. S. Gans, and A. Goldfarb (2019): "Artificial intelligence: the ambiguous labor market impact of automating prediction," *Journal of Economic Perspectives*, 33(2), 31–50.

Athey, S. (2018): "The impact of machine learning on economics," in *The economics of artificial intelligence: An agenda*. University of Chicago Press.

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

Ayyagari, M., A. Demirgüç-Kunt, and V. Maksimovic (2010): "Formal versus informal finance: Evidence from China," *The Review of Financial Studies*, 23(8), 3048–3097.

Banerjee, A. V., and E. Duflo (2010): "Giving credit where it is due," *Journal of Economic Perspectives*, 24(3), 61–80.

Bartlett, R., A. Morse, R. Stanton, and N. Wallace (2019): "Consumer-lending discrimination in the FinTech era," Discussion paper, National Bureau of Economic Research.

Beck, T., V. Ioannidou, and L. Schäfer (2017): "Foreigners vs. natives: Bank lending technologies and loan pricing," *Management Science*, 64(8), 3792–3820.

Behr, P., A. Drexler, R. Gropp, and A. Guettler (2014): "Financial incentives and loan officer behavior: Multitasking and allocation of effort under an incomplete contract," *Journal of Financial and Quantitative Analysis*, pp. 1–51.

Berg, T., V. Burg, A. Gombović, and M. Puri (2019): "On the Rise of FinTechs: Credit Scoring Using Digital Footprints," *The Review of Financial Studies*.

Berg, T., M. Puri, and J. Rocholl (2013): "Loan officer incentives and the limits of hard information," Discussion paper, National Bureau of Economic Research.

Bhowal, S., K. Subramanian, and P. Tantri (2014): "Soft Information And The Cost Of Job Rotation: Evidence From Loan Officer Rotation," .

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

Bonsall IV, S. B., A. J. Leone, B. P. Miller, and K. Rennekamp (2017): "A plain English measure of financial reporting readability," *Journal of Accounting and Economics*, 63(2-3), 329–357.

Buchak, G., G. Matvos, T. Piskorski, and A. Seru (2018): "Fintech, regulatory arbitrage, and the rise of shadow banks," *Journal of Financial Economics*, 130(3), 453–483.

Bursztyn, L., S. Fiorin, D. Gottlieb, and M. Kanz (2019): "Moral incentives in credit card debt repayment: Evidence from a field experiment," *Journal of Political Economy*, 127(4), 000–000.

Chakraborty, A., and C. X. Hu (2006): "Lending relationships in line-of-credit and nonline-of-credit loans: Evidence from collateral use in small business," *Journal of Financial Intermediation*, 15(1), 86–107.

Chen, D. L., T. J. Moskowitz, and K. Shue (2016): "Decision making under the gamblers fallacy: Evidence from asylum judges, loan officers, and baseball umpires," *The Quarterly Journal of Economics*, 131(3), 1181–1242.

Chen, T., and C. Guestrin (2016): "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Cheng, I.-H., S. Raina, and W. Xiong (2014): "Wall street and the housing bubble," *The American Economic Review*, 104(9), 2797–2829.

Cole, S., X. Giné, J. Tobacman, R. Townsend, P. Topalova, and J. Vickery (2013): "Barriers to household risk management: evidence from India," *American economic journal. Applied economics*, 5(1), 104.

Cole, S., M. Kanz, and L. Klapper (2015): "Incentivizing Calculated Risk-Taking: Evidence from an Experiment with Commercial Bank Loan Officers," *The Journal of Finance*, 2(70), 537–575.

Cole, S., T. Sampson, and B. Zia (2011): "Prices or knowledge? What drives demand for financial services in emerging markets?," *The journal of finance*, 66(6), 1933–1967.

Cortés, K., R. Duchin, and D. Sosyura (2016): "Clouded judgment: The role of sentiment in credit origination," *Journal of Financial Economics*, 121(2), 392–413.

Demirguc-Kunt, A., L. Klapper, and D. Singer (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(3), 393–414.

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

Drucker, S., and M. Puri (2008): "On loan sales, loan contracting, and lending relationships," *The Review of Financial Studies*, 22(7), 2835–2872.

Du, Q., F. Yu, and X. Yu (2017): "Cultural proximity and the processing of financial information," *Journal of Financial and Quantitative Analysis*, 52(6), 2703–2726.

Dushkin, L. (1967): "Scheduled caste policy in india: History, problems, prospects," *Asian Survey*, pp. 626–636.

Fisman, R., D. Paravisini, and V. Vig (2017): "Cultural proximity and loan outcomes," *American Economic Review*, 107(2), 457–92.

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

Fuster, A., P. Goldsmith-Pinkham, T. Ramadorai, and A. Walther (2018): "Predictably unequal? the effects of machine learning on credit markets," *The Effects of Machine Learning on Credit Markets (November 6, 2018)*.

Fuster, A., M. Plosser, P. Schnabl, and J. Vickery (2019): "The role of technology in mortgage lending," *The Review of Financial Studies*, 32(5), 1854–1899.

Gathergood, J., N. Mahoney, N. Stewart, and J. Weber (2019): "How Do Individuals Repay Their Debt? The Balance-Matching Heuristic," *American Economic Review*, 109(3), 844–75.

Griffin, J. M., and G. Maturana (2016): "Who facilitated misreporting in securitized loans?," *The Review of Financial Studies*, 29(2), 384–419.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi (2019): "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, 51(5), 93.

Hertzberg, A., J. M. Liberti, and D. Paravisini (2010): "Information and incentives inside the firm: Evidence from loan officer rotation," *The Journal of Finance*, 65(3), 795–828.

Iyer, R., A. I. Khwaja, E. F. Luttmer, and K. Shue (2015): "Screening peers softly: Inferring the quality of small borrowers," *Management Science*, 62(6), 1554–1577.

Jung, J., C. Concannon, R. Shroff, S. Goel, and D. G. Goldstein (2017): "Simple rules for complex decisions," *Available at SSRN 2919024*.

Keys, B. J., T. Mukherjee, A. Seru, and V. Vig (2010): "Did Securitization Lead to Lax Screening? Evidence from Subprime Loans," *The Quarterly Journal of Economics*, 125(1), 307–362.

Khandani, A. E., A. J. Kim, and A. W. Lo (2010): "Consumer credit-risk models via machine-learning algorithms," *Journal of Banking & Finance*, 34(11), 2767–2787.

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

Kleinberg, J., H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan (2017): "Human decisions and machine predictions," *The quarterly journal of economics*, 133(1), 237–293.

Kleinberg, J., J. Ludwig, S. Mullainathan, and Z. Obermeyer (2015): "Prediction policy problems," *American Economic Review*, 105(5), 491–95.

Lakkaraju, H., J. Kleinberg, J. Leskovec, J. Ludwig, and S. Mullainathan (2017): "The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 275–284. ACM.

Lakkaraju, H., and C. Rudin (2017): "Learning cost-effective and interpretable treatment regimes," in *Artificial Intelligence and Statistics*, pp. 166–175.

Liberti, J. M. (2017): "Initiative, incentives, and soft information," *Management Science*, 64(8), 3714–3734.

Liberti, J. M., and A. R. Mian (2008): "Estimating the effect of hierarchies on information use," *The Review of Financial Studies*, 22(10), 4057–4090.

——— (2009): "Estimating the effect of hierarchies on information use," *Review of financial studies*, 22(10), 4057–4090.

Ma, C.-H., and B. D. Smith (1996): "Credit market imperfections and economic development: Theory and evidence," *Journal of Development Economics*, 48(2), 351–387.

Mullainathan, S., and Z. Obermeyer (2017): "Does machine learning automate moral hazard and error?," *American Economic Review*, 107(5), 476–80.

Mullainathan, S., and J. Spiess (2017): "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, 31(2), 87–106.

Pierce, L., and J. A. Snyder (2018): "The historical slave trade and firm access to finance in Africa," *The Review of Financial Studies*, 31(1), 142–174.

Qian, J., P. E. Strahan, and Z. Yang (2015): "The impact of incentives and communication costs on information production and use: Evidence from bank lending," *The Journal of Finance*, 70(4), 1457–1493.

Rajan, U., A. Seru, and V. Vig (2015): "The failure of models that predict failure: Distance, incentives, and defaults," *Journal of financial economics*, 115(2), 237–260.

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

Skrastins, J., and V. Vig (2018): "How organizational hierarchy affects information production," *The Review of Financial Studies*, 32(2), 564–604.

Tantri, P. (2016): "Contagious Effects Of A Political Intervention In Debt Contracts: Evidence Using Loan Level Data," .

Townsend, R. M. (1994): "Risk and insurance in village India," *Econometrica: Journal of the Econometric Society*, pp. 539–591.

Tybout, J. R. (1983): "Credit rationing and investment behavior in a developing country," *The Review of Economics and Statistics*, pp. 598–607.

Tzioumis, K., and M. Gee (2013): "Nonlinear incentives and mortgage officers decisions," *Journal of Financial Economics*, 107(2), 436–453.

Appendix

TABLE A.1: Do Loan Officers Charge Higher Interest Rates For Risky Cases?- Full Sample

In this table, we 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. We use the test sample. The dependent variable is a dummy variable that takes the value of one if the loan under consideration is rejected by the machine and zero otherwise. Interest rate is the rate charged by the officer. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Table 2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates. ***, **, * represents statistical significance at the 1%, 5%, and 10% levels.

	(1)	(2)	(3)	(4)	(5)	(6)
	Acceptance Rate		Loan Proportion		Loan Amount	
Dependent Variable	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected
Interest Rate	8.07	6.93	8.07	6.93	8.07	6.93
	[1.52]	[1.29]	[1.52]	[1.29]	[1.52]	[1.29]
Total Assets		-0.00		-0.00		-0.00
		[-1.43]		[-1.43]		[-1.43]
Annual Income		0.00		0.00		0.00
		[0.27]		[0.27]		[0.27]
Loan Amount Asked		0.00		0.00		0.00
		[0.83]		[0.83]		[0.83]
Borrower Age		-0.00		-0.00		-0.00
		[-0.97]		[-0.97]		[-0.97]
Constant	-0.77	-0.58	-0.77	-0.58	-0.77	-0.58
	[-1.05]	[-0.78]	[-1.05]	[-0.78]	[-1.05]	[-0.78]
Fixed Effects						
Month Year	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,547	1,547	1,547	1,547	1,547	1,547
R-squared	0.16	0.17	0.16	0.17	0.16	0.17

TABLE A.2: Do Machines Discriminate ? Training Without Using Social Category Information

In this table, we test whether the machines discriminate based on social status. The data are organized at a loan level. The ML algorithm does not see the social category information during training. We 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 one if the loan under consideration is rejected by the machine and zero otherwise. SC-ST is a dummy variable that takes the value of one if the borrower belongs to the SC or ST community, and zero otherwise. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Table 2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates. ***, **, * represents statistical significance at the 1%, 5%, and 10% levels.

	(1)	(2)	(3)	(4)	(5)	(6)
	Acceptance Rate		Loan Proportion		LoanAmount	
Dependent Variable	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected	ML Rejected
SC ST	-0.04	-0.04	0.04	0.04	0.04	0.04
	[-0.93]	[-0.92]	[1.37]	[1.36]	[1.01]	[1.01]
Total Assets		0.00		-0.00		-0.00
		[0.12]		[-0.39]		[-0.41]
Annual Income		-0.00		0.00		0.00
		[-0.71]		[1.27]		[1.23]
Loan Amount Asked		0.00		-0.00		-0.00
		[0.66]		[-0.97]		[-1.14]
Borrower Age		0.00		-0.00		-0.00
		[1.17]		[-0.98]		[-1.04]
Constant	0.49***	0.42***	0.53***	0.59***	0.52***	0.59***
	[125.17]	[7.75]	[162.75]	[10.56]	[140.43]	[10.90]
Fixed Effects						
Month Year	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Observations	1,019	1,019	1,019	1,019	1,019	1,019
R-squared	0.36	0.36	0.36	0.36	0.36	0.36

TABLE A.3: CONTRACTION WITHOUT CONSIDERING SOCIAL STATUS

In this table, we present the results of the contraction procedure. We do not consider the social status while training the algorithm. In columns 1 and 2, we present the loan acceptance rates of strict and lenient periods respectively. We use the three measures of acceptance rates as described in Table 2. In columns 3 and 4, we present default rates under different definitions of acceptance rates and for strict and lenient periods separately. In column 5, we show the default rate that would result by following the ML algorithm and keeping the approval rate unchanged. In column 6, we show the lending amount achievable by the ML algorithm by maintaining the actual default rate of the strict period.

	Volume		Default		Machine	
	Strict	Lenient	Strict	Lenient	Default	Volume
Loan Amount lent (Rupees Million)	78.36	153.40	19.00	50.30	10.60	102.00
Loan Approval Rate	0.42	0.85	0.24	0.31	0.14	0.68
Loan Amount proportion	0.37	0.77	0.24	0.31	0.13	0.63

TABLE A.4: SOCIAL STATUS AND RISK SCORES USING THE FULL SAMPLE

In this table, we test whether the machines discriminate based on social status by assigning higher risk scores to SCs and STs. The data are organized at a loan level. We use the test sample here. The dependent variable is the risk score assigned by the algorithm to a loan. SC ST is a dummy variable that takes the value of one if the borrower belongs to the SC or ST community, and zero otherwise. We include borrower level characteristics such as the value of assets, annual income, loan amount applied for, and age as control variables. We include fixed effects at the location level and year X month level. We use the three definitions of acceptance rates, as defined in Table 2, in different columns. The standard errors are clustered at the location level, and adjusted t-statistics are reported in parentheses below the regression estimates. ***, **, * represents statistical significance at the 1%, 5%, and 10% levels.

	(1)	(2)
	With Caste	
Dependent Variable	Risk Score	Risk Score
SC ST	-0.01	-0.01
	[-0.38]	[-0.32]
Total Assets		0.00
		[0.19]
Annual Income		-0.00*
		[-1.89]
Loan Amount Asked		0.00***
		[3.64]
Borrower Age		-0.00
		[-0.96]
Constant	0.49***	0.47***
	[226.34]	[20.22]
Fixed Effects		
Month Year	Yes	Yes
City	Yes	Yes
Observations	1,585	1,585
R-squared	0.42	0.43