Working Papers

FEDERAL RESERVE BANK OF BOSTON

No. 19-16

How Magic a Bullet Is Machine Learning for Credit Analysis? An Exploration with FinTech Lending Data

J. Christina Wang and Charles B. Perkins

Abstract:
FinTech online lending to consumers has grown rapidly in the post-crisis era. As argued by its advocates, one key advantage of FinTech lending is that lenders can predict loan outcomes more accurately by employing complex analytical tools, such as machine learning (ML) methods. This study applies ML methods, in particular random forests and stochastic gradient boosting, to loan-level data from the largest FinTech lender of personal loans to assess the extent to which those methods can produce more accurate out-of-sample predictions of default on future loans relative to standard regression models. To explain loan outcomes, this analysis accounts for the economic conditions faced by a borrower after origination, which are typically absent from other ML studies of default. For the given data, the ML methods indeed improve prediction accuracy, but more so over the near horizon than beyond a year. This study then shows that having more data up to, but not beyond, a certain quantity enhances the predictive accuracy of the ML methods relative to that of parametric models. The likely explanation is that there has been data or model drift over time, so that methods that fit more complex models with more data can in fact suffer greater out-of-sample misses. Prediction accuracy rises, but only marginally, with additional standard credit variables beyond the core set, suggesting that unconventional data need to be sufficiently informative as a whole to help consumers with little or no credit history. This study further explores whether the greater functional flexibility of ML methods yields unequal benefit to consumers with different attributes or who reside in locales with varying economic conditions. It finds that the ML methods produce more favorable ratings for different groups of consumers, although those already deemed less risky seem to benefit more on balance.

JEL Classifications: G23, C52, C53, C55
Keywords: FinTech/marketplace lending, supervised machine learning, default prediction

J. Christina Wang is a senior economist and policy advisor in the research department of the Federal Reserve Bank of Boston, and Charles B. Perkins was a research associate in the research department. Their e-mail addresses are christina.wang@bos.frb.org and chbperkins@gmail.com, respectively. The authors thank Joe Peek for helpful discussions and Tzuo Hann Law of Boston College and Allen Sirolly for sharing the loan data. They also thank Lauren Mostrom and Melanie Qing for research assistance, and the Boston Fed's Economic Research Information Center, especially Mike Corbett, for assistance. This paper presents preliminary analysis and results intended to stimulate discussion and critical comment. The views expressed herein are those of the authors and do not indicate concurrence by the Federal Reserve Bank of Boston, the principals of the Board of Governors, or the Federal Reserve System. This paper, which may be revised, is available on the website of the Federal Reserve Bank of Boston at https://www.bostonfed.org/publications/research-department-working-paper.aspx.

This version: October 14, 2019 **https://doi.org/10.29412/res.wp.2019.16**

I. Introduction

Since the mid-2000s, a new group of lenders, powered by digital technology and operating largely through the internet, has entered the market for retail consumer lending. This new form of consumer credit is variably referred to as FinTech, digital, online, or marketplace lending. In general, FinTech lenders develop systems that enable them to accept and quickly process loan applications online, as well as rapidly deploy investor funding to borrowers. These lenders enjoyed rapid growth in market share after the financial crisis and thus have gained increased attention in recent years.

Advocates of FinTech lending tout the superiority of these lenders' technology for credit assessment: They employ more data, including unconventional data (beyond those in individuals' credit files), and analyze these data with more advanced methods (such as machine learning models) to arrive at a more accurate score of each applicant's credit risk within hours, if not minutes. All this has the potential to make credit more affordable and accessible to consumers as well as small businesses. On the other hand, some have expressed concerns that these more complex models are "black boxes" that may render lenders' decision rules too opaque. Such opacity gives rise to the risk of some protected consumer groups facing implicit discrimination. Moreover, a lender may incur the risk of violating fair lending rules if it is unable to satisfactorily explain to a consumer why she was denied credit.

To help gain a better understanding of these issues, this study evaluates four aspects of some popular classes of machine learning (ML) methods. First, it assesses the extent to which such methods improve the accuracy of default predictions for a given set of data. The general intuition concerning ML methods' potential for producing superior predictions is that they are able to fit more flexibly, and thus more accurately, the unknown functions that characterize the relationship between a loan's outcome and its characteristics. Second, this study examines which covariates (also referred to as features or input variables) are influential across different models and offers an intuitive presentation of their relationship with the loan outcome according to the ML models. This study next explores the extent to which there is “synergy” between methods and the amount of data. Is the accuracy gain from using ML models greater with a larger set of covariates or with more observations, or both? Fourth, we examine if the use of ML methods tends to produce more accurate or more favorable ratings of borrowers who have specific combinations of observed characteristics or are from locales with better or worse economic conditions.

Specifically, to gauge the relative prediction performance of ML methods, we first estimate the commonly employed logistic regression as the benchmark model for default predictions; we do so using loan-level data from the largest FinTech lender of unsecured personal loans. We then compare the logistic predictions with those from ML models, primarily random forests and tree-based gradient boosting. Both are ensemble methods built on tree-based models, which excel at uncovering complex, multilevel interactive effects. By aggregating the output of many trees, these models can arrive at a better ultimate prediction, but at the cost of losing the intuitive interpretability of individual trees. These methods have been found to perform well in classification problems (see, for example, Lessmann et al. 2015), even within the ML universe.[1]

Unlike most other ML studies of loan default, which consider only the borrower- and loan-specific ex ante indicators as predictors, our model also accounts for the ex post economic condition faced by the borrower, approximated mainly by the local unemployment rate. This is motivated by an economic model of unsecured consumer credit, which recognizes that a loan’s realized outcome depends on not only the borrower’s personal risk attributes, which are pre-determined, but also her economic circumstances ex post. In particular, a borrower becomes much more likely to default on a loan if she loses her job, and the job loss can be due to shocks entirely exogenous to the borrower. This means that borrowers with the same observed risk indicators can have rather different default outcomes if they face different economic conditions, over a

[1] Friedman, Hastie, and Tibshirani (2000), for example, show the equivalence between stepwise additive models and boosting, and explain its superior performance in classification problems accordingly.

business cycle or from residing in different geographic locales. This consideration is perhaps particularly relevant for our sample period, which includes the years during and immediately after the Great Recession, as well as the later years when the economy resumed normal expansion. We find that accounting for the ex post local unemployment rate improves not only the in-sample but also (albeit to a lesser degree) the out-of-sample fit of default predictions.

Unlike econometric analysis, the goal of which is to identify causal relationships, ML methods aim to maximize the accuracy of predictions, especially those that are out-of-sample. To this end, ML methods rely primarily on cross validation (CV) to minimize overfitting. Most ML studies maintain the assumption of a constant data distribution across samples. However, this assumption may well be violated in consumer credit data over a business cycle, as economic conditions evolve systematically, leading to changes in either the distribution of covariates or their relationship with the default outcome, or both. Such "instability," or drift, is especially likely to cause biases in out-of-sample predictions if models are estimated, or "trained," using data over only part of a cycle. This describes the situation with FinTech consumer loans, which gained scale only after the last recession. Moreover, the resulting bias may be much more acute for ML methods because they are nonparametric and thus cannot extrapolate. We partition out-of-training-sample data along multiple dimensions to gauge how much data and model drift over the sample period affects the accuracy of out-of-sample predictions. For instance, given models trained on loans originated in a given month, we compare their prediction accuracy on loans originated in the same month but put aside for testing versus loans originated in other months, which are subject to changes in data distribution. We find some evidence that the ML models tend to yield worse relative prediction fit on loans originated outside of the period used to train the model. We further examine the extent to which the change in economic conditions helps explain the variation in out-of-sample fit.

To gain intuition about the ML methods, we examine the importance of each input variable in each ML model, which is defined by an input's overall contribution (inclusive of its interactions with other covariates and among all the trees trained for each model) to lowering the loss function. In general, however, there may not exist even (positive) rank correlation between these ML models' feature importance measure and the relative absolute magnitude of the logistic coefficient except under special conditions, because tree-based models permit high-order interactive effects among covariates. As a result, the marginal effect of a single input or a pair of input variables can be highly nonlinear.

These tree-based ML methods also carry out variable selection implicitly, although it is obscured by averaging across many trees. To examine this aspect of the models specifically, we compare their feature importance rankings with the variable selection more intuitively produced by the least absolute shrinkage and selection operator (LASSO). In general, we find similarly high ranking for a handful of covariates (including the economic conditions faced by a borrower after the loan has been originated) according to both tree-based importance measures and the logistic coefficients, suggesting a robust first-order relationship between them and the default outcome.

The next topic to explore is the extent to which the accuracy gain from using ML models is greater when there are more data, in terms of both the number of observations and the number of available variables. We find that the prediction performance of ML methods improves in absolute terms with larger data sets, but not necessarily relative to that of the logistic model beyond certain levels of sample size. By comparison, there is a more consistent pattern of the ML methods gaining more when supplied with larger numbers of features. With the addition of even moderately informative features, there is larger, albeit modest, relative accuracy gain from the ML methods. It is thus plausible that the combination of ML methods and a large number of additional variables that are jointly informative can lead to more accurate predictions, although perhaps meaningfully, only for consumers who currently have little or no credit history.

Finally, taking advantage of the intrinsic ability of these tree-based ML methods to fit heterogeneous relationships, this study examines whether they produce more accurate or favorable ratings for borrowers with specific observed characteristics. For example, do ML methods improve the rating accuracy more for consumers with risk scores near the bottom of the prime range than for those with higher scores? We may expect ML methods to rate some borrowers more accurately than others because the default relationship may be more complex for borrowers with certain characteristics, and they thus can benefit more, as the ML models can fit a different and more flexible function in that subspace of the data.[2] It must be noted that there is no demographic information, such as race or gender, in this FinTech loan data set. This is conducive to fair lending, in that the lender in principle cannot discriminate along these dimensions. Hence, our analysis is meant to explore whether certain demographic groups may indirectly benefit more or less from the ML methods, to the extent that their protected characteristics are correlated with some of the observed risk indicators, such as risk scores and income.

Relatedly, by the same logic, we explore whether ML methods tend to favor borrowers from locales with particular economic conditions. For instance, are borrowers from places with high, versus low, per capita income or high, versus low, unemployment more likely to be granted better risk grades or rated more accurately by the ML methods than by the logistic regression?

The remainder of this study is organized as follows. Section II reviews a simple model of consumer credit, which makes clear how the outcome of a loan also depends on the ex post economic shocks that impact the borrower. Section III explains the ML methods used in this study, as well as their implementation in the data. Section IV discusses the issues that are relevant for the empirical specifications, including the statistics to use for comparing performance across models and the timing of data for computing in-sample versus out-of-sample test statistics. Section V presents the data and the estimation results using the array of models. Section VI concludes.

II. A Simple Model of Fixed-rate Consumer Installment Loans

This section presents a simple model of a consumer's optimal repayment decision on a fixed-rate personal installment loan. Its main objective is to demonstrate that

[2] Fuster et al. (2018) find that black and Hispanic mortgage borrowers seem to benefit less from the ML methods' greater flexibility.

repayment outcome depends on not only the borrower's ex ante attributes but also the ex post income shocks she experiences.[3] Therefore, if a model of realized default is fitted on only each borrower's ex ante type, it will suffer from an omitted variable bias. Furthermore, a credit score estimated using such a model will have a higher error rate if it is used to evaluate future loan applicants, and it may be biased systematically over a business cycle as macroeconomic conditions change.

A consumer *i* may need to borrow upon experiencing a temporary expenditure or income shock. If *i* succeeds in obtaining a loan of size B_{it} at the start of period *t*, she can choose to repay out of a net income of $Y_{i,t+1}$ at the end of period *t*.[4] We assume that $Y_{i,t+1} = \omega_{i,t+1}\Psi_i Y_{t+1}$. Y_{t+1} is the average per capita income determined by the overall local economic conditions at $t + 1$.[5] Ψ_i measures consumer *i*'s relative permanent income. A continuum of Ψ_i is assumed to be independent and identically distributed (i.i.d.) across consumers, as well as over time, with E(Ψ) = 1. Each consumer is also subject to idiosyncratic income shocks, denoted by $\omega_{i,t+1}$. The ω's are assumed to be i.i.d. random draws across individuals following a time-invariant differentiable cumulative distribution (c.d.f.) $G(\omega)$ over a non-negative support with E(ω) = 1. Moreover, $\omega_{i,t+1}$ is independent of the consumer's type.

Each consumer's ex ante default risk type is indexed by θ_i, which can be regarded as measuring her disutility from defaulting on debt: Should *i* default, she would be able to consume only $(1 - \theta_i)Y_{i,t+1}$. A higher θ_i thus means a lower willingness to default.[6] θ_i can

[3] It is adapted from Montoriol-Garriga and Wang (2011), who developed a credit model for small business loans. See that study for detailed derivations of the equilibrium pricing of loans.

[4] The *t*+1 time subscript is used to signal that a variable's value is unknown at the start of time *t* and will be realized at the end of the period. Note that net income, $Y_{i,t+1}$, not only depends on *i*'s gross income but is also net of her other obligations, such as taxes, mortgage, or rent, and revolving credit card debt. The higher a consumer's existing obligations (as reflected in a high debt-service-to-income ratio), the less net income she will have to repay the loan. We directly model the net income in order to focus on analyzing the loan contract terms.

[5] Y_{t+1} can be thought of as a composite term comprising a national and a regional component.

[6] Athreya, Tam, and Young (2012) and others have shown that lenders' ability to discern borrowers' nonpecuniary default cost is important in accounting for consumer debt dynamics.

be mapped into a credit grade, which incorporates information from a credit score, such as the FICO score, along with any additional quality signals uncovered by the lender through further screening. To obtain any such signals, a lender has to incur a cost (such as the fee paid to purchase credit bureau data and the operating cost of collecting applicants' social media data). We assume θ_i to be i.i.d. across a continuum of applicants and over time with a c.d.f. $J(\theta_i)$, $\theta \in [0, 1]$, and $E(\theta_i) \in (0, 1)$.

We denote the contractual interest rate (also referred to as the yield to maturity) on the loan to borrower *i* as $\hat{Z}_{i,t+1}$.[7] At the end of time *t*, *i* can consume $Y_{i,t+1} - B_{it}\hat{Z}_{i,t+1}$ if she repays the loan. If she defaults, she consumes $(1 - \theta_i)Y_{i,t+1}$. So she will default only if $Y_{i,t+1} - B_{it}\hat{Z}_{i,t+1} < (1-\theta_i)Y_{i,t+1}$. Rearranging terms yields $\theta_i Y_{i,t+1} < B_{it}\hat{Z}_{i,t+1}$. Plugging in the expression for $Y_{i,t+1}$ and normalizing by Y_t, we can express the condition of default as

$$\left(\theta_i b_{it}^{-1}\right)\omega_{i,t+1}R_{t+1} < \hat{Z}_{i,t+1}, \tag{1}$$

where $b_{it} := B_{it} / \Psi_i Y_t$ is *i*'s loan-to-income ratio (LTI), and $R_{t+1} := Y_{t+1}/Y_t$ is the growth rate of local per capita income. All the elements of b_{it} are known at time *t*. There are two sources of uncertainty, one aggregate (income growth R_{t+1}) and one idiosyncratic ($\omega_{i,t+1}$). Equation (1) shows that, in terms of the repayment prospect, a higher LTI is equivalent to a lower credit grade θ_i. Default thresholds are formally defined below.

Definition: For given θ_i and b_{it}, there is a 1-to-1 mapping between the interest rate $\hat{Z}_{i,t+1}$ and a threshold value, known at t, for the individual income growth rate $\hat{R}_{i,t+1} = \omega_{i,t+1}R_{t+1}$, below which loan i will be in default. If the aggregate growth rate R_{t+1} is also given, an analogous threshold can be defined for the idiosyncratic income shock $\omega_{i,t+1}$, denoted $\hat{\omega}_{i,t+1}$:

$$R_{i,t+1} := \hat{Z}_{i,t+1} / \left(\theta_i b_{it}^{-1}\right) \text{ and } \hat{\omega}_{i,t+1} := \hat{Z}_{i,t+1} / \left(\theta_i b_{it}^{-1} R_{t+1}\right). \tag{2}$$

[7] Even though $\hat{Z}_{i,t+1}$ is contracted and known at the beginning of *t*, we keep the (*t*+1) subscript to signify that whether it can be collected depends on the realization of $\omega_{i,t+1}$ and R_{t+1}.

Note that $G(\hat{\omega}_{i,t+1})$ is the probability of default (PD) of borrower *i* for a given aggregate growth *R*$_{t+1}$, while $\mathrm{E}_{R_{t+1}}\left[G(\hat{\omega}_{i,t+1})\right]$ is *i*'s unconditional PD with the expectation E$_R$[.] taken over the distribution of *R*$_{t+1}$. PD rises in the loan rate charged $\hat{Z}_{i,t+1}$, all else being equal, because the chance that the net income will be sufficient to cover the loan payment decreases as the loan rate charged increases. Consistent with intuition, (2) also shows that, for any given $\hat{Z}_{i,t+1}$, a higher θ_i lowers $\hat{\omega}_{i,t+1}$ and, hence, *i*'s PD. A good state of the economy (that is, a higher *R*$_{t+1}$, which tends to coincide with low unemployment) lowers the PD for all borrowers.

The consumer *i* will choose to accept a loan offer if her expected present value of utility with borrowing is no lower than that without:

$$u\left(Y_{it}+B_{it}\right)+\beta \mathrm{E}_t\left\{\max\left[u\left(Y_{i,t+1}-B_{it}\hat{Z}_{i,t+1}\right),u\left(\left(1-\theta_i\right)Y_{i,t+1}\right)\right]\right\}\geq u\left(Y_{it}\right)+\beta \mathrm{E}_t\left[u\left(Y_{i,t+1}\right)\right]. \quad (3)$$

β is the consumer's discount factor. The lower the current income (Y_{it}) is relative to the future income ($Y_{i,t+1}$), and the lower the type (θ_i), the more likely *i* will accept a loan offer with given interest rate $\hat{Z}_{i,t+1}$, as her reservation interest rate is higher.[8]

III. Fundamentals of The Machine Learning Methods

This section summarizes briefly the fundamental structure of the ML methods applied in this study. Appendix I provides more detail on each method. All of these methods are supervised ML, in that the data used to train the models come with input-output pairs and the targeted output value is already labeled.[9] Our goal is to explore a range of models that predict the probability of credit default to understand not only the

[8] The larger the current income shortfall relative to the expected future income, which is more prevalent during recessions, the greater the desire to borrow and smooth consumption. See, for example, Athreya, Tam, Young (2009) for a more detailed derivation of the consumer's optimization problem.

[9] This contrasts with unsupervised learning, where data are not classified. The unsupervised algorithms aim to group data or estimate the probability density of inputs.

empirical gain that can be achieved with ML methods but also the intuition of the underlying mechanisms. Thus, we mainly consider two classes of methods built on trees: random forests and stochastic gradient boosting. Among other advantages, tree-based models can be used to carry out covariate selection; that is, to choose the features (or predictors in the ML literature) that truly help predict the outcome, once adequate regularization is imposed to curtail overfitting. They thus can handle high-dimension problems; that is, the number of covariates is large relative to (or even exceeds) the number of data points. This is a function routinely performed by L1-regularized regressions, prominently the linear least absolute shrinkage and selection operator (LASSO). Therefore, we also employ LASSO to gain further intuition about the contribution of individual covariates. We do not study in detail the other broad categories, chiefly support vector machine and neural networks, because the former does not produce probabilities, while the latter lacks interpretability. We estimate a standard logistic regression to serve as a benchmark for comparison.[10] We then use the same loan-level data to estimate, or "train," the ML models.

III.1 Least Absolute Shrinkage and Selection Operator (LASSO) and Ridge Regression

LASSO (Tibshirani 1996) is a popular regression method that provides covariate selection and estimates their coefficients through L1-regularization:

$$\beta = \arg\min_{\beta} L(\beta; y, X) + \lambda\left|\beta\right|. \tag{4}$$

$L(.)$ is the loss function, β the vector of model coefficients, and λ the penalty parameter. In effect, LASSO reduces the magnitude of each coefficient relative to its counterpart from an ordinary least squares (OLS) regression by a fixed amount (depending on λ) while

[10] Archived web pages, such as that posted on July 21, 2009, show that Prosper's (the second-largest online lender of personal loans) early risk-scoring model in fact used to be logistic: https://web.archive.org/web/20090721215108/http://www.prosper.com:80/help/topics/general-prosper_score.aspx

truncating those below a threshold (also dependent on λ) strictly at zero.[11] LASSO thus tends to reduce overfitting and improve the prediction accuracy while making the final model easier to interpret. (See Appendix I for a more detailed exposition of LASSO and the other models summarized below.)

Ridge regressions differ from LASSO only in the form of the regularization: Ridge uses L2 penalty, meaning the term $\lambda|\beta|$ in equation (4) becomes $\lambda\beta^2$. This change affords ridge regressions the advantage of having a closed-form solution. The geometry of the L1 versus the L2 penalty means that LASSO is more suitable for problems with a sparse structure (that is, moderate to large effects from just a small set of covariates), whereas ridge is more suitable for cases in which a large number of covariates also exert some small effect (Tibshirani 1996). If we find that the LASSO predictions tend to be more accurate than the ridge predictions, we can infer that default is largely influenced by just a handful of covariates. Otherwise, default likely depends on many factors.

III.2 Classification and Regression Trees

Tree-based models are also referred to as "recursive partition," signifying that this method repeatedly partitions the sample space, ultimately dividing it into a set of exhaustive and mutually exclusive nodes (also known as leaves). At each step of the dominant two-way-split method, the sample is split into two parts, with the split point chosen to yield the largest decline in the loss function. Consider classification with $y = \{0, 1\}$ and the Gini index as the loss function; denote the two subspaces after a split as $R_1(j, s) = \{X \mid X_j \leq s\}$ and $R_2(j, s) = \{X \mid X_j > s\}$, and the corresponding number of observations N_1 and N_2, with $N_k = \#(x \in R_k(j, s))$, $k = 1, 2$; then choose each split at value s of feature X_j, so that

$$\min_{j,s}\left\{\min_{\hat{p}_1} N_1 \sum_{x_i \in R_1(j,s)} \left[\hat{p}_1\left(1-\hat{p}_1\right)\right] + \min_{\hat{p}_2} N_2 \sum_{x_i \in R_2(j,s)} \left[\hat{p}_2\left(1-\hat{p}_2\right)\right]\right\}, \tag{5}$$

[11] When the covariates are not orthonormal, this is only an approximate description of the relationship between LASSO and OLS coefficients.

where the predicted probability of $y = 1$, $\hat{p}$, is calculated as the mean in each subspace: $\hat{p}_k = \sum_{x_i \in R_k(j,s)} I(y_i = 1) \Big/ N_k$, $k = 1, 2$, and each is weighted by the number of observations in that subspace. Let Π denote the partitioning of the feature space $\mathbb{X}$, n(Π) the number of terminal leaves, and $\boldsymbol{\ell}$(x; Π) the leaf $\boldsymbol{\ell}$ ∈ Π such that x ∈ $\boldsymbol{\ell}$:

$$\Pi = \{ \boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots, \boldsymbol{\ell}_{n(\Pi)} \}, \ \bigcup_{j=1}^{n(\Pi)} \boldsymbol{\ell}_j = \mathbb{X} \text{ and } \hat{p}(x;\Pi) = \frac{\sum_{X_i \in \boldsymbol{\ell}(x;\Pi)} I(Y_i = 1)}{\#\left(X_i \in \boldsymbol{\ell}(x;\Pi)\right)}. \tag{6}$$

This in effect achieves nonparametric estimates of potentially highly complex heterogeneous models for subjects with different attributes. To gain intuition, consider this mapping between trees and standard regressions: If one knew the variables and the values at which they are used to partition a sample, one could define dummy variables according to these split points and replicate a tree model with a linear regression using these dummy variables and their interactions corresponding to the tree structure. Tree models are a systematic way to find the relevant split points, and they effectively carry out variable selection in the process. They are also robust to missing data and outliers. The most widely used algorithm is the classification and regression trees (CART) developed by Breiman et al. (1984), although it suffers from intrinsic biases (Strobl et al. 2007a).

III.3 Random Forests

The random forest method grows many trees on bootstrapped random samples that also randomize over the features in order to reduce the correlation across trees. The final prediction is derived as an average of these trees (see Breiman 2001). Random forests are found to be effective at curtailing overfitting while reducing variance, as well as at decreasing the risk of important features being masked due to high correlation with other covariates, in large part owing to the feature randomization and averaging. The use of random subsets of features also makes the random forest method especially suitable for applications with high-dimension covariates, such as genetic data, which often include more covariates than observations. The random forest method is typically easier to implement than boosting, because it requires fewer hyperparameters to be tuned and often its prediction error rate is competitive with that of boosting.[12]

III.4 Stochastic Gradient Boosting

Boosting, in particular, gradient boosting, can be regarded as a gradient descent approach to fitting unknown functions (Friedman 2001, for example). Friedman, Hastie, and Tibshirani (2000) point out the connection between boosting and additive basis expansion in estimation, as boosting repeatedly applies a simple model (often called a base learner), and each successive step focuses on the misses from past steps. Gradient boosting fits a model at each step that achieves the largest gradient descent with regard to the residual from the last step. The weighted average of predictions produced by these base learners converges to a prediction that is often much more accurate than that which any single model alone can achieve. Boosting models that use classification trees as base learners (referred to as boosted trees) have been found to perform well in classification problems (see, for example, Lessmann et al. 2015 for evidence). We therefore apply boosted tree models in our study, with CART as the base learners.

Hastie, Tibshirani, and Friedman (2008) further point out that boosting effects a path of L1-regularized estimation. Rosset, Zhu, and Hastie (2004) prove that boosting is generally equivalent to a L1-regularized path to a maximum margin classifier. In this sense, it is structurally similar to kernel support vector machines (SVM), since both are methods for regularized optimization with high-dimension features. In our experiments, however, we find that SVM performs noticeably worse than random forests and boosted trees (results available upon request), which is similar to the findings in the benchmarking study by Lessmann et al. (2015). Moreover, SVM does not directly produce estimates of default probability. We therefore choose to omit SVM.

The boosting and random forest methods applied in this study both are ensembles

[12] Hastie, Tibshirani, and Friedman (2008) show several examples in which, as the number of trees grows, the error rate of random forests converges more quickly to its asymptotic level than does the error rate of tree-based boosting, which continues to improve, albeit rather slowly, when there are large numbers of trees.

of trees, but they differ from each other in an important way. The successive tree learners in boosting evolve to fit the misses of previous steps, and the final prediction is a weighted average. Boosting thus in principle reduces both bias and variance. By comparison, all the trees in a random forest are grown the same way (on different random subsamples) and identically distributed. The random forest method therefore is good at reducing variance (through averaging), but without reducing bias (see Friedman, Hastie, and Tibshirani 2000, for example). As will be shown, the boosted tree method also tends to surpass the other methods in default prediction in our data.

III.5 Implementation of Machine Learning Methods

This subsection briefly outlines the implementation of the ML methods, mainly in the programming language Python. These models all require one or more hyperparameters (such as the degree of regularization in LASSO), and they are tuned using the standard practice of cross validation. We specifically apply fivefold CV: set aside a random subset (one-fifth) of the training data as the validation sample and train the model on the remaining (four-fifths) of the data for given hyperparameter values; then estimate the predicted values on the validation sample; last, average the prediction errors across the five random validation samples. The average loss or prediction error (measured by statistics such as mean squared error [MSE]) of a trained model $\hat{y}=\hat{f}(x)$ (with y denoting the outcome and x the covariates) given the hyperparameter value α is calculated as

$$\mathrm{L}^{CV}\left(\hat{f},\alpha\right)=\frac{1}{N}\sum_{i=1}^{N}L\left(y,\hat{f}^{-\kappa(i)}(x_i,\alpha)\right),\ \kappa\colon \{1,\ldots,\mathrm{N}\}\mapsto\{1,\ldots,K\},\ K=5, \tag{7}$$

where N and K are the number of observations and CV folds, respectively. $\kappa(.)$ denotes the indexing function randomly mapping observations to a CV fold, and $\hat{f}^{-\kappa(i)}$ denotes the model fitted with the κ-th fold (containing i) removed. The optimal hyperparameter value is the α that minimizes the average loss $\mathrm{L}^{CV}\left(\hat{f},\alpha\right)$. Given this hyperparameter, a model's prediction performance is then calculated on a separate test data set.

Much of the discretion in implementation concerns the tuning of the hyperparameters of each model to prevent overfitting and achieve the best fit within the estimation sample. In general, we tune over the standard range of each hyperparameter provided in each software package. For a few parameters (such as the number of trees in a random forest) that have recommended values from the literature, we find that, after experimenting with a number of alternatives, these values generally can be adopted. Appendix I details the software packages used to estimate each ML model, the tuning procedure, and the final choice of the hyperparameters.

Implementing the ML methods also involves several other issues not relevant for estimating regressions. These are discussed briefly here and detailed in Appendix I. The first is how to discretize the continuous covariates in order to minimize the risk that CART's inherent bias (as discussed above) may lead the tree-based models to bias up the importance ranking of these continuous features. Note that such discretization is needed only to measure the relative importance of each feature more accurately. It is unnecessary for just making predictions, although it may improve performance. In fact, we verify that the predictions are little changed with the discretized feature values.

Second, we explore if the prediction performance can be improved by reducing the imbalance of the two classes of cases in the data; that is, more than 85 percent of loans were paid off instead of defaulted on. Many studies have demonstrated that imbalanced data can bias model estimates, and a number of remedies have been developed (see the review by Guo et al. 2017). Given that our sample size becomes reasonably large from 2012 onward, we adopt the method of random undersampling: To form a re-balanced sample, we retain all the defaulted loans in each month, along with an approximately equal number of paid-off loans randomly drawn from that same month.

Third, we examine the extent to which it is necessary to calibrate the default probabilities predicted by the ML methods, since tree-based models, while generating accurate ranking, sometimes have been found to produce biased estimates of probability. We find that our boosting model generally yields sensible probability estimates, likely because its objective function is also the binomial likelihood. By comparison, random forest fitted values display signs of underfitting (that is, being further away from 0 and 1), but are not noticeably improved with calibration.

IV. Model Specifications, Intuition, and Performance Comparison

IV.1 Model Estimation: Timing of Estimation and Out-of-Sample Predictions

As noted, the emphasis of ML methods is to optimize predictions on data outside of those used to train the model. It is thus natural to focus on the out-of-sample prediction accuracy to test the superiority of ML methods. An ensemble of trees can make more accurate predictions, because trees allow for complex heterogeneous relationships in different parts of the data space. We therefore also use the ML models to study how the marginal effect of each important covariate varies over the support of those covariates. For example, we examine how a given increase in the FICO score affects the default probability differently depending on the loan's other features.

Just as importantly, we explore the extent to which it is valid to assume (routinely in ML studies) that the training and the test samples are identically distributed. A violation of this assumption can degrade a model's out-of-sample predictions. If we use superscripts *tr* and *te* to denote the training and the test sample, respectively, then two conditions are needed to ensure a consistent estimate of the mean outcome: (1) the covariate distribution (denoted $p(x)$) remains constant, that is $p(x^{tr}) = p(x^{te})$; and (2) the conditional distribution of the outcome given the covariates (denoted $p(y|x)$) remains constant, that is $p(y^{tr}|x^{tr}) = p(y^{te}|x^{te})$. Denote the trained model along with its parameters as $\hat{f}^{tr}(.)$ and $\hat{\theta}^{tr}$, respectively, where $\hat{\theta}^{tr} = n^{-1} \underset{\theta}{\arg\min} \sum_{i=1}^{n} \mathbf{L}\left[x_i^{tr}, y_i^{tr}, \hat{f}^{tr}\left(x_i^{tr}, \theta\right)\right]$, and $\mathbf{L}(.)$ is the loss function. The fitted value of y^{tr} conditional on x^{tr} is $\hat{y}^{tr} = \hat{f}^{tr}(.)$, with $\hat{f}^{tr}(.)$ being the sample counterpart to $p(y^{tr}|x^{tr})$ in the population. Then the loss metric in the test sample has the following relationship with the training sample:

$$\mathbf{E}_{x^{te}}\left[\mathbf{L}\left(x^{te}, y^{te}, f^{te}\left(x^{te}, \theta^{te}\right)\right)\right] = \int_{x^{te}} \mathbf{L}\left(x^{te}, y^{te}, f^{te}(.)\right) p(x^{te}) dx$$

$$= \int_{x^{te}} \mathbf{L}\left(x^{te}, y^{te}, \frac{f^{te}\left(x^{te},\theta^{te}\right)}{\hat{f}^{tr}\left(x^{te},\hat{\theta}^{tr}\right)} \hat{f}^{tr}\left(x^{te},\hat{\theta}^{tr}\right) \right) \frac{p(x^{te})}{p(x^{tr})} p(x^{tr}) dx \cdot \tag{8}$$

Clearly, both the covariate distribution and the conditional relationship need to remain the same in order to obtain the correct estimate of the true test sample error.

It is likely that either or both of these assumptions do not hold in this FinTech loan-level data set.[13] One notable example of time-varying $p(x)$ is that the unemployment rate has fallen steadily since the end of the Great Recession, so a tree model trained on data in late 2009 is "blind" to the lower values of this covariate in later years and thus may over-predict default rates, all else being equal. There are potentially multiple causes for time-varying $p(y|x)$; two likely important ones are that the composition of applicants to this new type of lender may change over time as it gains recognition, and the lender may alter its screening standards in response to changing investor risk tolerance. It is important to understand how such data instability may affect the output of ML methods and make appropriate adjustments. Otherwise, these models may break down in the next downturn, and, since they are being applied more widely, that can lead to severe losses.

We therefore design our estimation, including how the covariates are measured, and out-of-sample testing to facilitate exploring the extent to which such potential data or model drift, or both, may affect the relative performance of ML methods. This is a perspective that previous studies do not highlight. We adopt the method of frequent updating of the model, instead of formal testing, to gauge the impact of data instability, because asymptotic sampling distribution needed for inference has not been established for any ML methods other than random forests under strong conditions.[14] Specifically, since the origination time is known by month, for our baseline results, we train a separate

[13] Although some ML studies refer to these as data nonstationarity, we avoid this terminology in this study since $p(x)$ or $p(y|x)$,or both, vary over our sample period only because they change over a business cycle, but this data set covers mostly just the expansion phase of one cycle. In the standard definition, for a series to be nonstationary, its moments need to be non-constant as time asymptotes to infinity (that is, $t \to \infty$). The credit data may well be stationary if we had observations over a full business cycle or longer period.

[14] And it would not be fruitful to apply trend-break tests (such as in Bai and Perron 1998) because of the limited time-series dimension of the data, even if the statistics for inference existed.

model on loans originated in each month (t), which we refer to as cohort t in the subsequent analysis. We then use this model to predict the outcome of loans originated in later months (cohorts $t + h$, $h > 0$). To test robustness of the baseline findings, we also re-estimate the statistics on expanded training samples composed of varying numbers of cohorts. We find reasonably consistent evidence for data and model drift. To maximize the number of months (h) over which to test the models, we study only three-year loans here, but the method is equally applicable to five-year loans.[15]

Since it is practically impossible to estimate the degree of freedom and in turn use formal inference with complex models such as trees, we use a fivefold CV to estimate nonparametrically the mean and standard error of the extra-sample prediction errors.[16] That is, when estimating the models for month t, we randomly divide the data into five equal subsets, train each model on four-fifths of the data, and then compute test statistics (such as MSE) on the remaining one-fifth of the data. The test statistic across the five same-month test samples is used to approximate the distribution of the true out-of-sample estimate. Hastie, Tibshirani, and Friedman (2008) note that the average CV error is often an unbiased estimate of the true out-of-sample mean error, but can underestimate the true error rate by 10 percent for complex models such as trees.[17] Just as importantly for our purpose, these statistics computed on the same-month test data, which are drawn from the same sample as the training data, should be free of any data instability and measure only the sampling variance. If these criteria are met, the test statistics can be used to assess whether the performance statistics computed on loans originated in later months are significantly different, which presumably would be due to some form of data or model instability.

We use publicly available LendingClub data covering loans originated from November 2008 through September 2015, because our data run through November 2018

[15] LendingClub, the largest online lender of unsecured personal loans, offers two maturities on these loans: three and five years. The majority of its loans are for three years.

[16] Fivefold or tenfold CVs are recommended by Kohavi (1995), among others.

[17] This is because the validation sets affect the search for best trees, and the standard error can be large.

and we need at least 36 months after each origination month to measure the outcome on all the loans within that cohort.[18]

IV.2 Empirical Specifications: Choices and Transformation of Covariates

As the benchmark for comparison, we estimate the following logit model of the default outcome of each loan:

$$\Pr\left(O_{izt}=1\right)=e^{\mathbf{W}_{izt}}\Big/\left(1+e^{\mathbf{W}_{izt}}\right), \text{ where } \mathbf{W}_{izt}\equiv g_{it}+\beta X_{zt}+\left(\delta_z+\varepsilon_{izt}\right). \qquad (9)$$

O_{izt} = 1 when a loan *i* from zip code area *z* in cohort *t* was in default later; it equals 0 otherwise.[19] g_{it} denotes the set of person- and loan-specific characteristics, such as the FICO score, DTI, number of credit inquiries within the last six months, loan amount, and loan purpose. These are meant to capture the loan's ex ante credit risk. X_{zt} measures the local economic conditions both ex ante and ex post. The ex ante indicators include lagged unemployment rates and a number of variables measuring the state of the banking market for locale *z*, such as the average ratio of tier-one capital and the ratio of nonperforming commercial and residential real estate loans (over tier-one capital). Ex post economic conditions are represented by the average unemployment rate and house price growth rate over the life of a loan. δ_z denotes the zip-code-specific fixed effects, which encompass persistent unobserved characteristics of each locale. ε_{izt} is the residual.

To study how ML methods' relative performance vis-à-vis the logistic model may depend on the number and type of covariates, we consider three sets of covariates that correspond to different practical cases. First, as our baseline specification, we use all the available covariates, comprising individual-specific (g_{it}) and locale-specific (X_{zt})

[18] LendingClub started making loans in July 2007, but the volume was extremely small until early 2008. We choose November 2008 because LC changed its credit policy so that no applicants with a FICO score below 660 have been approved since then.

[19] For future reference, loans originated in the same month are referred to as a cohort. So, for example, loans originated in month-*t* are referred to as cohort-*t*.

variables, as well as a full set of zip-code-specific fixed effects. See Appendix II for the list of all the available g_{it} and X_{zt} variables. This should in principle give ML methods the best opportunity to shine, since they supposedly have the ability to pick the relevant features for making predictions. To minimize the known bias of CART models when they are faced with a mix of discrete and continuous variables, we discretize the features with continuous values and conduct robustness tests that compare the predictions by using the original continuous input versus different mappings into discrete values; see Appendix A2.4 for details. On the whole, we find the bias to be small.

We then compare this baseline of ML models' relative performance with an alternative case, where we consider only a smaller but presumably most informative set of covariates that were explicitly referenced in LendingClub's early grade model.[20] Comparing these two cases gauges how much greater prediction accuracy can be gained through the ML models by making use of the large number of additional covariates that may be informative, albeit perhaps only moderately.

We later consider two additional sets of input variables, one of which uses only ex ante variables in order to quantify how much ex post economic conditions help explain loan outcome, while the other mimics the data availability probably relevant for consumers with little credit history. They are detailed in Section V.4.

IV.3 Model Intuition: Feature Importance and Partial Dependence

The emphasis of ML methods on predictions also means that individual coefficients are not as meaningful. In particular, if some variables are correlated, then

[20] The typical risk scoring rules that LendingClub posted from July 11, 2011, through October 31, 2012, can be found on the Internet Archive. These two dates are pinned down by the first and the last dated pages containing the explicit grading rules, which remained virtually unchanged during this period. The Internet Archive crawler, however, samples any given webpage only periodically, so these rules likely were active over a longer period. For an example, see the exact rules posted on October 20, 2012, at https://web.archive.org/web/20121020205505/http://www.lendingclub.com/public/how-we-set-interest-rates.action

coefficients may vary noticeably from one training data set to the next, whereas the predictions can still be similar. In regression trees with multilevel interactive effects and tree-based ensemble methods such as random forests and gradient boosting, it is nearly impossible to directly quantify individual coefficients. Nonetheless, to gain intuition, we make use of the feature importance measures provided by each method.

In random forests, a feature is deemed more important if it reduces the average loss (often the Gini impurity index) of all the trees in a forest (see Pedregosa et al. 2011 and Breiman 2001).[21] This is the default importance measure computed by the software. However, as noted above and explained in Appendix A1.1, random forests suffer from the inherent bias in the underlying CART algorithm and bootstrapping with replacement, and this can result in a biased importance value. As will be explained in greater detail below, we discretize covariates with continuous values to reduce the bias, and robustness checks indicate this mostly eliminates the small bias.[22]

In boosting models, a feature is more important if it leads to a greater cumulative gain in the objective function from all the constituent base learners. In the algorithm we use (that is, XGBoost), the objective is essentially to maximize a binomial log likelihood function, so a feature's importance is the average increase in this objective accounted for by this feature. For both ML methods, the feature importance is measured on a relative scale, so we normalize the sum to 100.

We compare covariate importance across three models estimated with all the available features: the LASSO, the random forests, and the boosted trees. All three models have built-in algorithms to handle high-dimension covariates. For the logistic model, however, we find that using only the subset of covariates that are assigned non-zero LASSO coefficients tends to yield more robust estimates. We restrict the feature space, because the logistic model tends not to produce sensible estimates when fed a large

[21] The Python sklearn package calculates the tree level importance as the "(normalized) total reduction of the criterion brought by that feature."

[22] We also tried another partial correction: the importance measure computed using out-of-bag permutations (see Parr et al. 2018). But we chose not to use it for baseline results, because the computation is too time consuming.

number of covariates (some of which may be unimportant, such as those not selected by LASSO) relative to the size of the data set. This problem is particularly relevant when using pre-2012 data due to the much smaller number of loans.

The influence of different covariates on default can also be illustrated with partial dependence plots, which depict a subset of inputs' marginal effect after we account for the (average) effect of the other inputs. If the inputs (X_1, ..., X_p) are partitioned into two sets, X_S and X_C, with $S \subset \{1, ..., p\}$ and $S \cup C = \{1, ..., p\}$, then the partial dependence of $f(X)$ on X_S is

$$f_S(X_S) = \mathrm{E}_{X_C}\left[f\left(X_S, X_C\right)\right].$$

$f_S(X_S)$ is analogous to the average marginal effect of inputs X_S in models such as the logit. In tree-based models, for which interactive effects among inputs are an integral part, $f_S(X_S)$ is inclusive of the average interactive effect between X_S and X_C. Thus, $f_S(X_S)$ is a measure of X_S's partial effect alone only if X_S and X_C have little or no interactions; for example, they are purely additive or multiplicative. This is rarely feasible, in part because each such plot can depict clearly at most a 3-D relationship, and thus is best at revealing the joint effect of just two inputs. Given the limitations, it is natural to focus on the covariates identified as important and more likely to have interactive effects among them.

IV.4 Metrics for Model Comparison

We consider two metrics to measure and compare the performance across models estimated for and tested on each monthly cohort of loans. The first statistic is the mean squared error (MSE).[23] The second metric is the area under the receiver operating characteristic curve (AUC). In the case of binary outcome variables, the MSE and the AUC emphasize different aspects of the model predictions (typically the predicted score of the chosen outcome). The MSE penalizes a predicted value for being far from the true value but does not explicitly account for ranking, whereas the AUC penalizes the wrong

[23] It is also known as the Brier Score of probability for Bernoulli outcomes; see Brier (1950).

ordering but puts no weight on the distance from the true value.[24]

The MSE is widely used in statistics to measure prediction accuracy. Let y_i be the actual value of the binary outcome {0,1}, and let f_i be the probability score of $y_i = 1$ predicted by some model, and thus $f_i \in [0,1]$. Then $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - f_i)^2$.

The area under a receiver operating characteristic (ROC) curve is the loci of the probability of true positive versus false positive for a given model.[25] The AUC, the second metric, thus ranges from 0.5 (corresponding to a useless model that generates completely random ranking) to 1.0 (corresponding to a perfect classifier).[26] The Gini index is related to the AUC: Gini + 1 = 2 * AUC (Hand and Till 2001). The AUC can be interpreted in several ways; for example, it measures the probability of a randomly drawn observation with y = 1 ranked higher than a random observation with y = 0.[27] This makes the AUC useful for assigning loan risk grades at the time of application.

The literature notes several problems with the AUC as a measure of classifier performance. One problem is that the AUC may not be able to rank two models when the two corresponding ROC curves cross each other (at the probability threshold z), so that one model produces more precise ranking up to the probability z whereas the other model does better above z. Perhaps more problematic is the finding by Hand (2009) that the implicit misclassification cost distribution underlying the AUC is related to the predicted score distribution produced by each model, so that the AUC fails the criterion of an apples-to-apples comparison across models.[28] Nevertheless, the AUC often performs well despite its theoretical limitations (as noted in Fawcett 2006, for example).

[24] A wrong ordering arises if a data point with a realized outcome of 1 is ranked lower than a data point with an outcome of 0. Which outcome is labeled 1 versus 0 is immaterial.

[25] The ROC curve can be used to decompose the MSE into components that measure the degree to which each segment of the curve is distinct from the other segments in the share of $y_i = 1$ (resolution, the higher the better) versus the dispersion across observations belonging to that segment (reliability, the smaller the better); see Murphy (1986) and Pelánek (2015), for example.

[26] The AUC could fall below 0.5 on a test sample given a badly performing model, but then the predictions could simply be reversed.

[27] See, for example, Hand (2009) for an exposition of the multiple interpretations of the AUC.

[28] Hand and Anagnostopoulos (2014) propose an alternative, the H-measure, which solves this

In fact, the AUC is the only metric that can be used to compare the performance of our models with that of the risk grades assigned by LendingClub (LC), because LC provides no ex ante default probability associated with the grades. Instead, we can construct an ROC curve based on LC risk grades for each loan cohort, and then compare its AUC with those of our models computed on the same-month test samples.

V. Empirical Evidence of Machine Learning's Advantage

V.1 Data

We use data from four categories primarily. Foremost is the LendingClub data of individual applications and loans. The available data, organized by month, covers the period from June 2007 through September 2018. However, virtually all the existing analyses use data only from November 2008 on, because the volume of lending in 2007 was too small, and the share of applicants with a FICO score below 660 fell to 0 percent that month (from about 20 percent in 2007) and has remained at 0 percent ever since. Moreover, we focus exclusively on the three-year loans, because they are more conducive to our analysis: There are many more three-year loans than five-year loans, and the outcome of a three-year loan is known two years sooner. See Appendix III for further details about the LC data. In the interest of space, summary statistics of the LC data are omitted; they can be found in Jagtiani and Lemieux (2018).

This data set is augmented with real economic indicators for each individual's local geographic area, which is defined at the three-digit zip code level. The raw data by geography are provided at the county level: monthly unemployment rate, quarterly house price index, annual income per capita, poverty rate, education attainment, annual population by age group, and fraction of population with various types of debt outstanding by age group. See Appendix III for further details of these data. To map the

problem by applying the same misclassification cost distribution across models. It is, however, not widely used.

data by county to data by five-digit zip code, we use the mapping file compiled by the University of Missouri Census Data Center. The data are then aggregated to the three-digit zip code level using the population or labor force (for the unemployment rate) in each five-digit zip code area as weight. More specifically, the value of a variable in the three-digit zip code area, denoted x, is calculated as

$$X_{i,t}^{zip3} = \sum_{j \in i\_zip3} w_j X_{j,t}^{zip5} \text{ , with } w_j = \sum_{j \in i\_zip3} \left(pop_j^{zip5} \Big/ \sum_j pop_j^{zip5} \right) \text{, so that } \sum_{j \in i\_zip3} w_j = 1 \,. \quad (10)$$

Because they are the weighted averages of the values in the constituent five-digit zip code areas, the three-digit zip-code-level data understate the cross-area heterogeneity and thus may underestimate the significance of the impact of local economic conditions.

An additional set of variables at the three-digit zip code level measure the borrowing and credit risk of consumers, which is constructed by the Federal Reserve Bank of New York using data from the Consumer Credit Panel.[29] Among these are variables used in the modeling of commercial credit scores (such as the FICO score), including the utilization rate of revolving credit, the number of credit inquiries over the past 12 months, and the number and balance of accounts 90-plus days past due. Total debt payment is combined with per capita income to measure debt burden.[30]

The fourth set of data pertain to measures of local banking market conditions, in terms of the degree of competition and bank health. These indicators are meant to capture factors that affect credit supply by banks; a tighter supply of bank loans tends to encourage demand for new, alternative sources of credit, and vice versa. The first indicator is the importance of the four largest banks (Bank of America, Citigroup, JPMorgan Chase, and Wells Fargo), which is measured using these banks' share in deposits held in bank branches within a local area as a proxy for their overall activity in

[29] This is a nationally representative random 5 percent sample from anonymized individual consumer credit accounts maintained by Equifax. For details, see Lee and van der Klaauw (2010).

[30] All the calculations of account balance adjust for joint account ownership to avoid double counting. For more details on data construction, see Appendix III.

that locale. This is referred to as the Big-4 share in the analysis that follows. These banks curtailed lending notably for several years after the crisis (see, for example, Chen, Hanson, and Stein 2017), in part due to enhanced capital requirements and regulatory oversight. Also, Bank of America and Citigroup saw their capital severely impaired by mortgage-related losses. Next is a common measure of competition in the local banking market, computed as the Herfindahl–Hirschman Index (HHI) of deposits in bank branches within a zip code area. Three variables are used to measure potential capital constraints on banks operating in the local area. The first two are nonperforming residential and commercial real estate (RRE and CRE) loans, respectively, as a share of tier-one capital; the third is the tier-one capital ratio itself as a comprehensive measure of capital adequacy.[31] By comparison, a high loan-delinquency ratio can curb lending because it signals future capital pressure. The zip-code-level values are weighted averages computed from county-level data using a procedure similar to the one used for the real economic variables described above, with the weight for each bank equal to its share in total deposits in the county. See Appendix III for details of the derivation.

V.2 Do Machine Learning Methods Improve Prediction Accuracy in General?

In this section, we report model output from the benchmark logit regressions and from the two ML methods, and compare the accuracy of their default predictions, in terms of both the CV and the true out-of-sample AUC and MSE.

Figures 1a and 1b use boxplots to summarize the CV and multiple out-of-sample AUC and MSE, respectively, of all five models we estimate: logistic, random forest, boosted tree, LASSO, and ridge. Every model is fitted with all the available features (see Appendix II), but recall that both tree-based methods and LASSO in effect select only the relevant features for predicting default. Three distributions are depicted for each model:

[31] Nonperforming loans are defined as loans that are 30-plus days past due or nonaccrual to capture the forward-looking effect. Coefficient significance remains the same if nonperforming is defined as 90-plus days past due or nonaccrual, which is the measure used in most banking studies.

The range of the performance metric for the median fold of the training-month CV samples, the 12-months-ahead loan cohort, and the 36-months-ahead loan cohort are plotted in the first, second and third block, respectively.[32]

Several basic patterns emerge. First and foremost, boosted trees perform the best in terms of the (highest) AUC, regardless of the horizon of the sample. Random forests are typically ranked second, except when they are surpassed by the ridge model on the 36 months ahead test sample. This finding largely agrees with those reported in other studies (such as Lessmann et al. 2015), most of which, however, consider only the CV samples. In our case, this relative relationship holds for both the CV samples and nearly all the test samples of future loan cohorts. LASSO and ridge, although inferior to the two ML models, still perform largely better than logistic, on average.

The relative ranking turns out to vary over the sample years. As Table 1 shows, when the sample is divided at 2012:M1, LASSO fails only in the first half (Panel A), but afterward it becomes nearly competitive with the two ML models in the CV and near-future test samples. Before 2012:M1, LASSO in fact often predicts no better than mere random predictions, which would yield an AUC of 0.5. This is due to the number of loans made in each month before 2012:M1 being so small that LASSO sets non-zero coefficients on too few features.[33] This dynamic is the result of exponential growth of FinTech personal lending as a new entrant to this segment of the credit market starting in 2008, especially after the financial crisis. For the same reason, the variance of the AUC is also much larger, on average, in the earlier subsample.

[32] Recall that the CV samples are all within the same month as the training data (from cohort t) and thus should gauge a model's fit on identically distributed extra-sample data, while the 12- and 36-months-ahead loan cohorts are meant to gauge the degree of potential data or model drift in the near, versus far, future. What is labeled as 36 months ahead in fact refers to loans originated t+37 months after to ensure that all cohort-t loans (including those receiving funds in t+1) have matured. Given the origination date only by month, this is the first month when all cohort-t data can be used to train models if decisions had to be made in real time.

[33] The number of loans grew steadily from 209 in 2008:M11 to a little more than 2,200 in 2011:M12. Afterward, it continued to grow, reaching nearly 25,000 per month, on average, in 2014:Q4.

By comparison, the two ML models in fact deliver greater gains in AUC compared with the logistic model before 2012:M1, confirming their advantage in fitting small data sets with a relatively large number of covariates. This can be seen more directly in Figure 2, which plots the AUC of the two ML models vis-à-vis the logistic model for the subsample periods through 2012:M12 and afterward, respectively.[34] The ML models show a larger difference in median AUC (and mean AUC, see Table 1) compared with the logistic before 2012:M12 than afterward. At the same time, as would be expected, the smaller sample sizes through 2012:M12 result in greater variability (that is, larger time-series standard errors) of the AUC than in the later years. This pattern can be seen in greater detail in Figure A1 in Appendix IV, which plots the time series of the AUC of the CV samples, along with 3, 12, and 36 months out loan cohorts for the two ML models.

Figure 2 also reveals another pattern of the relative performance of random forests and boosted trees: Their superiority over the logistic model (especially in terms of the AUC) is more pronounced in the CV samples than in tests on future loan cohorts. This suggests that caution may be especially needed in working with these ML models: Their predictions can deteriorate more than those of an otherwise inferior parametric model would when the data drift and the existing nonparametric mappings do not apply to the expanded support.[35] The universal deterioration in prediction accuracy, on the other hand, indicates that the relationship between default and the covariates changes over time. Therefore, any model would likely need to be updated periodically in response to changes, for example, in macroeconomic conditions. However, we must also be mindful of the potential of exacerbating the procyclicality of credit supply if the lender revises down the mean default rate (and thus the average credit grade) following a boom and

[34] The next section will conduct more formal analysis of the relationship between sample size and the ML models' relative performance.

[35] In tree models, the mean response within each leaf of the training sample is used to predict the response on test data. As shown in Section III.2, the range of covariate values (support) within some terminal nodes are unbounded, thus in theory they cover data whose support exceeds that of the training sample, but the mean response value within each such leaf may well be a biased estimate of the mean response for feature values outside of the existing support.

revises it up following a downturn.[36] The effect would be akin to the procyclicality due to the risk-based bank capital requirement (for a summary, see Kashyap and Stein 2004).

Given the relatively small number of loans per month in the early years, a model trained on multiple months is likely to produce better out-of-sample predictions than a model trained on only one month of data in those years. Figure 3 displays the range of changes in the AUC for random forests and boosted trees (Panels a and b, respectively) when the training sample is expanded from loans made in a single month to loans made in as many as six months in the period before 2012:M1.[37] There is no clear gain within the CV samples, but there is a clear gain in most of the test samples of future loan cohorts, even the 36-months-ahead cohort.

In terms of the MSE, both ML models are roughly on par with the LASSO and ridge models. All four models in fact have a similarly smaller MSE than that of the logistic model (Figure 1 and Table 1), suggesting that this pattern of relative performance is likely driven by regularization, which is known to prevent overfitting and thus produce better out-of-training-sample predictions. All but the logistic model are regularized in some form.[38] The cross-model pattern differs from that for the AUC.

The AUC depends on only the default risk ranking, while the MSE measures the absolute deviation of the predicted probability. The different patterns regarding the AUC and the MSE reported above indicate that the ML methods improve the accuracy of default predictions more in terms of the ranking than of the magnitude of the probability. One potential implication of this result is that, relative to parametric methods such as the logistic model, these ML methods enable entities that provide ordinal ratings of

[36] The relationship between default and observables can change if the lender alters its approval or rating model over time for other reasons, such as to cater to investor demand. Rajan, Seru, and Vig (2015), for example, document such a dynamic in the case of subprime mortgage loans.

[37] All the models other than the baseline specifications, including those using more than one month of training data, are trained at quarterly frequency to economize on computation. These plots consider only the model producing the highest AUC among the CV samples.

[38] Python's LogisticRegression routine in fact permits both L1 and L2 regularization, but we have to disable it in effect, because the regression would assign (essentially) zero coefficients to most features with any nontrivial degree of regularization.

prospective borrowers to achieve noticeably more accurate ratings, but only somewhat more accurate estimates of the exact probability of default. Since the latter is what matters for the pricing of (consumer) credit products, investors in such credit instruments should be mindful of this limitation if they apply these ML methods to guide investment decisions.

V.2.1 Comparison of Prediction Performance with LendingClub's Credit Grades

This subsection examines how these ML models' prediction accuracy compares with that of the credit grades posted by LendingClub (LC). This comparison is conditioned on the publicly available data used here, which are almost certainly a strict subset of the variables available to the lender.[39] Recall that we can make the comparison based only on the AUC metric, because the credit grades have no probabilities associated with them. Since LC can use only past data in assigning grades to loans originated in month *t*, we estimate a version of models that use only data available in real time up to *t* to achieve a fair comparison.[40] Since our models are trained using actual binary outcome (of default or not) to estimate the default probability, the most comparable approximation to a lender's real-time decision is to test the prediction accuracy of a model that is trained on loans originated at least three years ago, so that all the outcome is known.

Figure 4 depicts the results of this exercise. It is clear that the lender's risk grades consistently have a higher AUC (the thick gray line) than do the ML models trained using

[39] LendingClub's data advantage almost certainly grows over time, as it accumulates repeat borrowers and more general information on borrower behavior specific to FinTech personal loans. Little of this information is available to outsiders.

[40] For parametric models, one could use forecasts of the covariates measuring ex post economic conditions over a loan life in training a model. However, as Stahlecker and Trenkler (1993) show, using a proxy variable often does not improve prediction accuracy. This is what we find when inserting forecasts of the unemployment rate in the logistic model (available upon request). Note that all the information in the forecasts comes from the lagged variables; it is just combined specifically by the functional form used in the estimation. Thus, for nonparametric tree-based models that allow the lagged variables to explain the outcome in a fully flexible way, there is no gain to also including the forecasts.

ex ante covariates only (the thin solid lines).[41] This is likely due more to the better data available to the lender, which has not only additional predictors, but also more timely updates on loan repayment (such as being 30 or 60 days late). This should enable the lender to use more recent loan cohorts to train the prediction models. As the earlier results show, models' out-of-sample performance deteriorates almost monotonically in the time elapsed. In particular, boosted trees' AUC can compete with risk grades' AUC if data from as recent as 12 months earlier can be used to train the model. It is interesting to note that the AUC of all the predictions shows a gradual but steady rise since late 2012, likely reflecting the growing amount of data. It is possible that part of the lender's accuracy advantage also stems from using methods such as neural networks, which often achieve better prediction accuracy but fall short on interpretability, although the marginal gain relative to boosted trees is likely to be limited (such as the case in the study by Lessmann et al. 2015).

For comparison, the AUC of the models trained on the same loan cohorts but inclusive of the ex post change in the unemployment rate and house price growth over a loan's life (which is the baseline specification) are depicted with dashed lines. It is not surprising that, most of the time, the model with ex post covariates yields an AUC that is higher than the AUC of the model without, and occasionally even higher than the AUC of the lender's risk grades. This indicates the importance of ex post economic conditions in influencing loan outcomes. On the other hand, in a few months (including those in early 2015), the model with only ex ante covariates performs better. This is a sign of either data drift or model drift, or both. For example, as unemployment rates across most locales drift steadily lower over the sample years, a 1 percentage point higher unemployment rate relative to the US average may imply a different impact on the default rate. This is explored further below.

[41] Recall these models are trained quarterly, while the AUC of the lender's grades is monthly.

V.3 Model Intuition: Feature Importance

To enhance the ML models' transparency, we examine measures of how much each covariate contributes to explaining the default outcome. For regression models, one simply needs to inspect the point estimate and significance of coefficients. No exact counterpart exists for the ML models, in part because trees (beyond stumps) permit multilevel interactions. For the tree models, the importance of a feature is measured by how much it improves the objective function (such as lowering the Gini impurity index) at all the nodes where it is used to split the data. The importance scores are usually reported relatively, and we choose to normalize the sum to 100.

As a baseline for comparison with feature importance in the ML models, Figure 5 depicts the ordinary least squares (OLS) coefficients and t statistics on the features selected by LASSO.[42] Recall that each coefficient in linear models of binary dependent variables roughly equals the average partial or marginal effect (AME) of a unit increment (Wooldridge 2010). Outside of the zip code dummy variables, only about 25 or so covariates are ever assigned non-zero coefficients across the models trained using single-month data, and the top 14 are plotted in Panel a. The signs of the coefficients conform to the usual intuition, while their magnitude varies somewhat from month to month.[43] For example, the more unemployment rises in a borrower's zip code area, the higher her chance of default; the higher her credit score, the lower her chance of default. Renters are more likely to default, and so are those who borrowed for small-business purposes (relative to those who borrowed to pay off credit card debt).

Among the covariates, the average unemployment rate (UR) change over a loan's

[42] Belloni and Chernozhukov (2013) show that the OLS using the variables selected by LASSO in high-dimensional sparse models yields consistent and less biased estimates in-sample. We report the post-LASSO OLS coefficients only for importance analysis because their magnitude is more intuitive, whereas predictions must still be based on the raw LASSO coefficients. Covariates enter OLS in their native scale but must be standardized in LASSO. Each LASSO coefficient then equals a given reduction (determined by the degree of regularization) on its OLS counterpart.

[43] The range of coefficient size and significance since 2012:M1 exhibits a distribution similar to the one in Figure 5, whereas fewer inputs are significant in the earlier months (for example, DTI is not chosen in many months) due to the much smaller sample size.

life turns out to be the most significant determinant, and its magnitude is nontrivial: A 1 percentage point rise in the UR raises the default probability a median of about 8 percent. In terms of significance, the UR is followed by the FICO score, (log) annual income, number of credit inquiries over the last six months, DTI, the average house price growth over the loan life, and being a renter. These covariates can be classified as important if we apply the criterion that the (absolute) median t-statistic of a covariate exceeds the 5 percent critical value.[44]

As a more standard baseline to compare with the ML models, Figure 6 plots the point estimates and z statistics of the logistic coefficients from the logistic regression estimated using monthly data since 2012:M1.[45] The sign and relative magnitude of each coefficient coincides fully with those of its LASSO counterpart. In particular, the average UR change over a loan's life is again the most significant, and its magnitude is also large. On the other hand, if we apply a threshold of the 5 percent critical value to the median z statistic, then only the average UR change, the FICO score, and (log) annual income are significant.[46]

Figures 7 and 8 depict the feature importance scores of the random forest and boosted tree models. The box for each feature delineates the range of importance scores across models trained on monthly data. In terms of the relative importance, the two ML models agree on the top-four most important features, even though their exact formulas for the importance metric differ: the UR change and house price growth over a loan's life, the FICO score, and annual income. These inputs are also deemed significant in the above regression models. One systematic difference between the two ML methods is a greater cross-feature dispersion in their importance from the random forest model than from the

[44] In contrast, borrowing for small-business purposes is associated with substantially higher default probability. But the probability is significantly higher in only a little over a quarter of the training months in the full sample, albeit more than half of the monthly samples since 2012:M1.
[45] The logistic model using the pre-2012:M1 samples yields more extreme point estimates but much lower significance, because it cannot cope with small data sets with many covariates.
[46] By comparison, in the post-LASSO OLS estimates since 2012:M1, DTI, number of inquiries over the last six months, and the renter indicator are also important (not shown for brevity).

boosted tree model. Specifically, all the variables other than those four above are, by comparison, much less important in random forests, with inquiries over the last six months and thenutilization rate on revolving lines slightly ahead of the pack (Figure 7). In contrast, the covariates are more similarly important in the specific implementation of the boosted tree model we use (Figure 8), likely due to the additional regularization to curb the concentration of too many splits of the data along just one feature (see Appendix I for more detail). Another possible reason is the different loss functions used by the two methods, with the function for boosting more concave than the one for random forests.

In short, to a fair degree, a common core set of covariates are regarded as most important—in terms of improving the precision of default predictions—across all the methods. This indicates that a core set of covariates accounts for the bulk of the variation in default, and much of this influence can be captured by the direct first-order effect of these covariates. It in turn suggests that a best-of-both-worlds approach to modeling credit default behavior is feasible: Utilize ML methods to better capture the higher-order terms that affect loan outcome, as they can fit unknown functions more flexibly, while at the same time mitigate the opacity of ML methods through comparisons with the less flexible but more intuitive regression models.

V.3.1 Partial Dependence and Interactive Effects

To augment our understanding of the ML models beyond the importance scores reported above, we examine the sign of the inputs' impact on loan outcome using partial dependence plots. As noted, we focus on covariates identified above as important.

Figure 9 depicts the joint marginal effect of the unemployment rate change over a loan's life and the FICO score from a boosted tree model trained on loans originated in 2012:M7. A larger UR increase (or a smaller decrease) and a lower FICO score are each associated with a higher default probability. The pattern further indicates an interactive effect between (local) labor market conditions and a borrower's credit score: An increase in the UR is associated with a larger increase in default probability for those with lower scores. For those with FICO scores above 760 or so, the risk of default barely rises with the

UR. In contrast, for those with scores below 700, a larger UR increase raises the default risk more substantially.[47] Qualitatively similar patterns of the joint effect of UR change over a loan's life and the FICO score on the risk of default are also observed in boosted tree models trained on loans originated in other months. This is consistent with the perception that consumers with low credit scores are more prone to income declines or job losses during downturns. In fact, their low scores may well be partly due to the greater income or employment volatility.

Another fairly robust interactive joint effect on default risk is between the level of income and the FICO score. Figure 10 illustrates the relationship with estimates also from boosted trees using loans from 2012:M7. First note that there are consumers with income in the bottom decile or so but high FICO scores, and vice versa. For borrowers with annual income in or near the top decile, the default risk is mostly flat with respect to the credit score, until the score falls below 700, where the predicted default rate becomes noticeably higher, reaching to nearly 30 percent. In contrast, for those with income in the bottom two deciles, the default risk rises more or less monotonically as the credit score falls. Likewise, the increase in default risk due to lower income is steeper for those with scores below 700. Qualitatively, the same pattern is also observed for loans made in other months. This is consistent with high-income workers tending to have more stable employment. Another plausible explanation is that they also have more savings, and thus are able to smooth expenditures, including debt repayment.

We find a few additional, albeit weaker or non-monotonic, interactive effects. For example, it is mostly just among those with low FICO scores that increasing loan size and house price declines raise the default risk. Low utilization rates on revolving lines are in fact associated with the average or even higher default risk, but only among those with FICO scores below 700. In sum, our exercises have shown that the tree-based ML methods can be used to uncover interactive effects among covariates, which in turn can help refine

[47] Although one of the highest default risks is found among some borrowers who have a score near 770. It is an example of boosted trees' ability to fit highly nonlinear functions.

quantitative models of consumer credit.

V.4 The More Data the Better for Machine Learning Methods?

In this section, we conduct some preliminary analysis of the extent to which having more data, in terms of both the number of observations and the number or type of input features, affects the prediction performance of ML methods, especially relative to those of the typical regression models.

First, we examine how much the ML models benefit from having a larger number or different varieties of input variables. We consider three alternative sets of inputs to compare with our baseline specification. The first is ex ante information only, which is a natural setup for making loan decisions in real time. The second alternative includes only a small set of (eight) individual-specific indicators, which the industry found to be predictive of default (and mostly confirmed by the model output reported earlier), and thus was adopted by the lender in its early risk-grading algorithms.[48] The third set also contains just a small number of individual-level indicators, but these are likely not as informative, because they are restricted to ones that are also plausibly available for consumers with little credit history, often referred to as the thin-credit-file cases. These inputs include number of inquiries over the last six months, length of credit history, months since the last inquiry, total current balance and total high credit limit, and requested loan amount. They are then augmented with all the local economic variables included in the baseline specification. This last alternative is meant as a rudimentary exploration of how well ML models can rate consumers with thin credit files.

Figure 11 compares the AUC of each of the two ML models relative to the logistic's AUC under the baseline specification reported earlier (which contains the full set of borrower indicators publicly available from the lender, along with the local economic variables) versus the above three alternative specifications. This comparison can be made

[48] These include the FICO score, number of inquiries over the last six months, DTI, utilization rate of revolving credit lines, length of credit history, number of total and currently open credit accounts, and requested loan amount. See Footnote 20 for more details.

for any test samples, so we consider two "polar" cases: One is across the CV test samples, which are drawn from loans made in the same month as the training data, while the other is the "far-future" test sample, which consists of loans made 25 to 36 months after the training cohort. The pattern is broadly similar across the two ML models.[49]

First, with the exception of the specification (LC Early Vars) that uses only the limited subset of individual risk indicators, the ML models (especially boosted trees) mostly yield an AUC higher than that of the logistic. For any given set of inputs, the ML models' relative AUC is always better for the CV samples than for the far-future samples. This is hardly surprising, to the extent there is data or relationship drift over time. In terms of the degree of AUC deterioration due to the time elapsed, the models with ex post variables (Baseline and Thin Credit models, specifically) seem to suffer the most, suggesting that the distribution of ex post economic conditions or their influence on default (possibly in part through their interaction with individual creditworthiness) has changed the most as compared to the borrower-specific ex ante indicators. At far horizons, the model without ex post covariates performs basically as well as the one with them (that is, Only Ex Ante Vars versus Baseline).

When we compare across the input sets with more or less borrower-level indicators (that is, Baseline versus Thin Credit), the absence of even the most informative person-specific risk indicators (such as the FICO score or the DTI) lowers the relative AUC by only a modest amount, on average.[50] The resulting relative AUC difference is especially small for the median CV subsample, suggesting that the inclusion of economic variables can make up for the lack of person-specific risk gauges to a large extent, but only if the conditions in the test and training samples are similar. By comparison, considering the most informative personal risk measures but none of the local economic variables (specifically, LC Early Vars versus Thin Credit) lowers the relative AUC by more than 0.05, even at far horizons, indicating the important influence of economic conditions on

[49] The pattern is also similar for the relative MSE, displayed in Figure A2 in Appendix IV.

[50] Recall that both of these specifications include all local economic indicators. The relative AUC loss is, on average, close to 0.05 from boosted trees, versus 0.02 to 0.03 from random forests.

the outcome of unsecured personal loans. In fact, just including the ex ante local economic conditions (that is, Only Ex Ante Vars versus LC Early Vars) raises the relative AUC by even more (close to 0.10 from boosted trees) at far horizons.

We next explore how increasing the sample size affects the relative prediction accuracy of the two ML methods vis-à-vis the logistic regression. For this comparison, we vary the training sample's size by random draws, over the range of 100 to 10,000 observations, from a given subset of data. We choose a few subsample periods so that each contains about 10,000 observations in total. Predictions are then computed on loans originated 25 to 36 months after the cohort used to train the models. Figure 12 reports the AUC estimates from two training sample periods (2010:M6 to 2011:M6 and 2011:M7 to 2012:M1), which are representative of the basic pattern, despite the notable variations due to the specific model and sample period. From 100 to 500 observations, the relative AUC of each ML model rises noticeably. Beyond 5,000 observations, the relative AUC of each in fact falls, albeit only slightly.[51] Each model's relative AUC peaks somewhere in between and generally hovers within a somewhat narrow range. The overall pattern is similar for the relative MSE (Figure A3 in Appendix IV), which is nearly identical for boosted trees and random forests. The ML models' advantage peaks at about either 500 or 1,000 observations, depending on the sample period.

That "diminishing returns" in sample size seem to exist for the ML methods is not surprising: Their greater flexibility partly compensates for the lack of data, so their advantage over parametric models is more visible with moderately small datasets. Note that this is solely about the relative, not the absolute, performance, as all models' prediction accuracy improves with more data, albeit at a decreasing rate. There is corroborating evidence in Figure 3, as the average level of out-of-sample AUC keeps rising for both boosted trees and random forests as more months of data are added to the training sample (up to 20,000 observations). Nevertheless, with sufficiently abundant

[51] Note that the level of every model's AUC by and large continues to rise even with training data of 50,000 observations.

data, parametric models may in fact be preferred if they approximately capture the relationship between the covariates and default, since they can produce more accurate out-of-sample predictions when subject to data drift.

Furthermore, there is a small gain in the ML models' relative prediction accuracy from expanding beyond the small set of the most important covariates used in LendingClub's early risk rating algorithm. This likely implies that the unconventional data FinTech lenders utilize can further improve the prediction accuracy, but such data need to be more informative than the marginal credit indicators that are already contained in consumers' credit files in order to bring about material improvement. Hence, unconventional data (also referred to as alternative data) likely add nontrivial value only with regard to individuals with little or no credit records, for whom these new data (such as payment history on cell phone bills and utility bills) substitute for the missing key credit indicators. Any gains in prediction accuracy from using alternative data must be balanced against potential risks to consumer protection, data privacy, and security. For example, full disclosure of the use of such data should be mandatory, and consumer consent should be required whenever feasible. Analysis is needed to minimize the risk of certain groups being disadvantaged. These issues are beyond the scope of this study (see Wang 2018 for more in-depth discussions).

By comparison, it seems much more useful to consider local economic indicators, even just the ex ante conditions, in modeling default on these FinTech personal loans. Note that these variables are not meant to stereotype borrowers. Instead, as illustrated in the simple economic model, when granular data on each borrower's income process are not available, local economic conditions (proxied by a factor such as the unemployment rate) function solely as an approximation for an individual borrower's income risk, and hence their ability to repay, as evidenced by the fact that the average unemployment rate during a loan's life is invariably the most important covariate. Whenever available, the individual income measures should be used instead. To the extent that rules prohibit considering any neighborhood characteristics in rating a consumer's credit risk, finding individual-level indicators for income risk likely yields the highest payoff for a lender.

V.5 Do Machine Learning Methods Help Some Borrowers More than Others?

Many have expressed concerns that the complex and often opaque ML methods may lead to unintended discrimination against certain groups of consumers, especially those with protected attributes. Here, we explore if, in these online lending data, the ML methods predict more accurate or more favorable credit grades than standard regressions do for consumers with certain observed attributes. The intuition that relative model output may be unequal comes from the function relating covariates to loan outcome possibly differing across consumers and the ML methods' greater flexibility potentially enabling them to better fit the more complex functions for certain groups of consumers. It must be noted that LendingClub's data contain no information on loan applicants' gender and race, which can be regarded as a plus for the goal of fair lending. Hence, our analysis is meant only to make indirect inferences about whether ML methods may benefit certain groups of consumers more if their demographic attributes are correlated with observed credit indicators such as DTI or credit scores, which are available in the LC data.

We mainly explore whether the ML models tend to predict default more accurately (that is, a lower MSE) or more favorably (that is, a lower predicted than actual default rate) *relative* to the logistic model for individuals with good or blemished credit records. Specifically, these risk indicators are examined: the lender's risk score (which is arguably the sufficient statistic of a loan's credit risk and is never used as a model input), the FICO score, DTI, (log) annual income, and loan amount. Yet another dimension along which unintended unequal impact of ML methods on different populations may manifest is across different geographic areas. We thus also explore if the ML methods predict more accurate or more favorable default risk across locales along the following dimensions: total population (which proxies for urban centers versus rural areas), share of population below the poverty line, the local unemployment rate's deviation from the US average, house price growth, and share of residents with utilization rates of credit card limits above 85 percent.

Linear regressions are used as a parsimonious way to summarize the first-order statistical relationship between borrower attributes and local economic conditions and the

MSE and the error:

$$\Delta_{i,t,s} = \alpha_{t,s} + \alpha_{z} + \beta y_{it} + \gamma \mathbf{X}_{it} + \varepsilon_{i,t,s}, \qquad (11)$$

where $\Delta_{i,t,s}$ denotes MSE or error (defined as actual outcome minus predicted probability) on loan *i* originated at time *t* of the prediction made with a model trained on data in month *s*, y_{it} denotes the attribute of interest, and $\mathbf{X}_{it}$ the vector of controls.[52] We use *s* = five months of data, every six months from 2012:M1 to 2014:M1, to train models for this exercise. Each model is then tested on loans 1 to 24 months out, presuming this is the relevant range of out-of-sample horizons. Since β is merely a convenient summary of the correlation between $\Delta_{i,t,s}$ and y_{it}, each y_{it} enters individually when it is borrower-specific. But when y_{it} is a zip-code-level variable, the individual attributes are controlled for so that β measures the marginal effect of local conditions. A full set of fixed effects is included for every training and test sample pair ($\alpha_{t,s}$) and every zip code (α_{z}), by which standard errors are also clustered to account for the within correlation of $\Delta_{i,t,s}$.

Table 2 shows the estimates for boosted trees' relative MSE. The MSE is smaller for higher risk grades, but the difference is insignificant. Likewise, all the other linear relationships are insignificant, with the exception of house price growth—the higher this rate, the lower the MSE. When all the attributes are considered jointly (the last column), the coefficient on house price growth retains its significance while the MSE now rises in the FICO score (meaning it becomes more accurate for borrowers with lower scores). The pattern for the random forests' relative MSE is similar, with most linear coefficients being insignificant. Thus, it appears these two ML methods do not systematically generate more accurate predictions based on the MSE for subgroups of borrowers.

In contrast, there is some evidence that the raw prediction errors (*relative* to the logistic model's estimates) vary systematically across consumers with different attributes. Table 3a reports the pattern for boosted trees' relative prediction errors. It appears that predicted default probability is slightly lower (hence more positive errors) for borrowers

[52] The error is measured in percent, while the MSE is also scaled up 100 times, to avoid an excessive number of leading zeros in the coefficients.

with better credit grades, by 2 to 4 percentage points for the top two grades, and statistically significant. A similar result is found for the FICO score (with the probability 6 percentage points lower for every 100 points difference in the score) and log annual income, as well as DTI (hence the negative coefficient). With regard to local-area characteristics, lower predicted probability is found for locales with a larger population and faster house price growth but also a higher unemployment rate and, rather slightly, for locales with a larger share of high utilization of credit card limits (after we control for individual attributes). Except for the lender risk grade and credit card utilization, the other attributes retain their coefficients qualitatively when all are considered jointly, while the poverty share becomes negatively correlated with the prediction error. On the whole, boosted trees tend to predict relatively more favorable default probabilities for individuals already deemed more creditworthy and for locales with better economic circumstances.

By comparison, random forests' prediction errors (relative to the logistic models') exhibit less of this one-sided bias. In particular, as shown in Table 3b, the predicted probability is, on average, higher (thus a more negative error) for consumers with better risk grades and locales with lower poverty shares, and it is uncorrelated with annual income or DTI (or positively correlated with the latter). Even for coefficients with the same sign as their boosted tree counterpart, the magnitude is smaller. Overall, random forests are more likely to predict lower default probabilities for borrowers deemed more risky. The main reason for these different patterns between the two ML methods is likely random forests' tendency to predict odds that are closer to the average, which translates into higher (lower) predicted odds for less (more) risky borrowers relative to the logistic estimate, due to the built-in reliance on averaging to reduce variance. This indicates that understanding the full implication of the mechanics of a method for the predicted values is important, as is the need to avoid other potential biases, such as the implicit bias of the existing institutions that is embedded in the input data.

Not inconsistent with this explanation of the pattern of the relative error is our finding that there are few monotonic relationships between borrower attributes and the

ML models' AUC relative to the logistic AUC. The AUC is computed by risk grade and by decile of each of the other attributes' value. Table 4 reports the OLS coefficient of the relative AUC of boosted trees on the attribute decile indicator. Most coefficients are insignificant. Perhaps more important, the boosted tree model's relative AUC shows a (insignificant) monotonic relationship only with the decile ranking of log loan amount (higher AUC for larger loans), log local population (higher AUC for less populated locales), and share with a college degree (more accurate for places with a lower share).[53] Essentially the same outcome is also found for random forests' relative AUC (not shown in the interest of space). Thus, the general pattern based on relative AUC is that these ML methods do not appear to consistently rate certain subgroups of borrowers less accurately, certainly not those who are currently deemed more risky. Since the ML methods are more likely to be used for the AUC of their predictions, which is their comparative advantage, as shown earlier, these AUC-based non-results are arguably more relevant for gauging the likelihood that the ML methods may unintentionally lead to unequal treatment.

All findings considered, the preliminary evidence provides rather limited support for the concern that these ML methods disadvantage certain groups of consumers. It should, however, be noted that this finding is conditional on the given data. Further analysis is needed to assess whether bias is embedded in the data used as model inputs, and if so, what the consequence of such implicit bias is.

VI. Conclusion

FinTech lending to consumers has experienced tremendous growth over the past decade. Since its inception about the time of the Great Recession, advocates have asserted that the superior technology of FinTech lenders, including both more data and more advanced quantitative methods, enable them to excel over the traditional lenders. This study carries out a preliminary assessment of this claim by examining the extent to which

[53] When we use a raw index of deciles, even fewer coefficients are significant due to the added restriction of a linear relationship with the integer decile index.

two classes of machine learning (ML) methods commonly applied to classification problems improve the accuracy of out-of-sample predictions of default. It further investigates whether having more data, in terms of both more observations and more predictors, help the ML methods more than the regular regression models. Moreover, it explores if the ML methods help certain subgroups of borrowers more, in terms of either observed attributes or geographic location.

We find that the two ML methods, random forests and boosted trees, indeed improve the accuracy of default predictions in the largest FinTech personal loan lender's public dataset. In particular, the ML methods are superior to the benchmark logistic model more so in their ability as classifiers (that is, to separate defaulted loans from the rest of the loans through ordinal ranking) than in terms of the accuracy of their numerical predictions of the probability of default. More important, ML models improve default predictions much more for in-sample estimates than out-of-sample estimates. We find that an important reason for this pattern is data drift and model drift—the changing distribution of the covariates over time and their relationship with the default outcome. In fact, at a far enough horizon, data and model drift cause the ML models' out-of-sample predictions to underperform that of the logistic model. These findings indicate caution is needed in applying ML methods, especially over a business cycle.

In terms of the most influential covariates, all the empirical models consistently find ex post economic conditions faced by borrowers to be among the most important in explaining the default outcome, as implied by the economic model. In addition, across the parametric and the ML models, a similar handful of covariates (including the ex post economic indicators, the risk score, the number of recent inquiries, and the debt-to-income ratio) are found to be consistent determinants of default.

We find a hump-shaped relationship between sample size and the ML models' relative performance. More observations help all the models predict more accurately. In many subsample periods, however, the ML methods' performance relative to that of the logistic model peaks at about one or a few thousand observations and diminishes beyond that. Adding some even moderately informative borrower-specific covariates improves the ML methods' relative prediction performance mildly. This suggests that unconventional data can help, but are most likely to bring about materially more accurate credit ratings only for consumers with little or no credit history, as such data substitute for the absence of the more informative credit variables. Moreover, the utility of such data should be balanced against potential risk to consumer protection, data privacy, and security. On the other hand, accounting for local economic conditions improves the prediction accuracy more noticeably for all loans.

We find little statistically significant evidence that the ML methods generate more accurate predictions of default than the logistic model does for subgroups of borrowers depending on their risk attributes, income, or where they live. This is especially true in terms of the area under the receiver operating characteristic curve (AUC), the metric in which the ML methods are found to be more accurate. By comparison, we find suggestive evidence that the ML methods tend to predict slightly lower relative estimates of default probability for borrowers with better values for the typical ex ante indicators of default risk, such as a lower credit score or a higher debt-to-income ratio, although there is a notable difference between the estimates by boosted trees and those by random forests in some cases. On the other hand, the default risk for borrowers living in locales with more difficult economic conditions is rated better in a few cases. On the whole, it appears that these ML methods are unlikely to cause a notable disadvantage for borrowers with prime or near-prime credit scores. However, more work is needed to assess whether borrowers with weaker credit indicators may be more disadvantaged by the application of ML methods.

While our estimates detect only limited gain from the use of these ML methods in terms of accuracy of default predictions, this finding may be specific to our data, as only a set of standard credit variables are available. Having data on additional credit-relevant covariates that are available to the lender, such as the payment history on past FinTech loans or more timely loan repayment information, may enable these ML models to improve the prediction accuracy more substantially.

References

Athreya, Kartik, Xuan S. Tam, and Eric R. Young. 2009. “Unsecured Credit Markets are Not Insurance Markets.” *Journal of Monetary Economics* 56(1): 83–103.

Athreya, Kartik, Xuan S. Tam, and Eric R. Young. 2012. “A Quantitative Theory of Informat ion and Unsecured Credit.” *American Economic Journal: Macroeconomics* 4(3): 153–183.

Athey, Susan, and Guido Imbens. 2016. “Recursive Partitioning for Heterogeneous Causal Effects.” *Proceedings of the National Academy of Sciences* 113(27): 7353–7360.

Bai, Jushan, and Pierre Perron. 1998. “Estimating and Testing Linear Models with Multiple Structural Changes.” *Econometrica* 66(1): 47–78.

Belloni, Alexandre, and Victor Chernozhukov. 2013. “Least Squares after Model Selection in High-Dimensional Sparse Models.” *Bernoulli* 19(2): 521–547.

Breiman, Leo, Jerome Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. New York: Chapman and Hall.

Breiman, Leo. 1996. “Bagging Predictors.” *Machine Learning* 26(2): 123–140.

Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45(1): 5–32.

Brier, Glenn W. 1950. “Verification of Forecasts Expressed in Terms of Probability.” *Monthly Weather Review* 70(1): 1–3.

Chen, Brian S., Samuel G. Hanson, and Jeremy C. Stein. 2017. “The Decline of Big-Bank Lending to Small Business: Dynamic Impacts on Local Credit and Labor Markets.” NBER Working Paper No. 23843.

Chen, Tianqi, and Carlos Guestrin. 2016. “XGBoost: A Scalable Tree Boosting System.” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discover and Data Mining* 785–794.

Efron, Bradley, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. 2004. “Least Angle Regression.” *Annals of Statistics* 32(2): 407–499.

Fawcett, Tom. 2006. “An Introduction to ROC Analysis.” *Pattern Recognition Letters* 27(8): 861–874.

Friedman, Jerome H., Trevor Hastie, and Robert Tibshirani. 2000. “Additive Logistic Regression: A Statistical View of Boosting.” *The Annals of Statistics* 28(2): 337–407.

Friedman, Jerome H. 2001. “1999 Reitz Lecture: Greedy Function Approximation: A Gradient Boosting Machine.” *The Annals of Statistics* 29(5): 1189–1232.

Friedman, Jerome H. 2002. “Stochastic Gradient Boosting.” *Computational Statistics & Data Analysis* 38(4): 367–378.

Fuster, Andreas, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther. 2018 "Predictably Unequal? The Effects of Machine Learning on Credit Markets." Working paper. https://ssrn.com/abstract=3072038.

Guo, Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. "Learning from Class-imbalanced Data: Review of Methods and Applications." *Expert Systems with Applications* 73(1): 220–239.

Hand, David J., and Robert J. Till. 2001. "A Simple Generalisation of the Area under the ROC Curve for Multiple Class Classification Problems." *Machine Learning* 45(2): 171–186.

Hand, David J. 2009. "Measuring Classifier Performance: A Coherent Alternative to the Area under the ROC Curve." *Machine Learning* 77(1): 103–123.

Hand, David J., and Christoforos Anagnostopoulos. 2014. "A Better Beta for the *H* Measure of Classification Performance." *Pattern Recognition Letters* 40(1): 41–46.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2008. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics, Second Edition. Berlin, Germany: Springer.

Hothorn, Torsten, Kurt Hornik, and Achim Zeileis. 2006. "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics* 15(3): 651–674.

Jagtiani, Julapa, and Catharine Lemieux. 2018. "The Roles of Alternative Data and Machine Learning in Fintech Lending: Evidence from the LendingClub Consumer Platform." Federal Reserve Bank of Philadelphia Working Paper No. 18-15.

Kashyap, Anil K., and Jeremy C. Stein. 2004. "Cyclical Implications of the Basel-II Capital Standards." *Federal Reserve Bank of Chicago Economic Perspective* First Quarter: 18–31.

Kohavi, R. 1995. "A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection." *The International Joint Conference on Artificial Intelligence* 1137–1143.

Lee, Donghong, and Wilbert van der Klaauw. 2010. "An Introduction to the FRBNY Consumer Credit Panel." Federal Reserve Bank of New York Staff Report No. 479.

Lessmann, Stefan, Bart Baesens, Hsin-Vonn Seow, and Lyn C. Thomas. 2015. "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring: An Update of Research." *European Journal of Operational Research* 247(1): 124–126.

Lin, Yi, and Yongho Jeon. 2006. "Random Forests and Adaptive Nearest Neighbors." *Journal of the American Statistical Association* 101(474): 578–590.

Montoriol-Garriga, Judit, and J. Christina Wang. 2011. "The Great Recession and Bank Lending to Small Businesses." Federal Reserve Bank of Boston Research Department Working Papers No. 11-16.

Murphy, Allan H. 1986. "A New Decomposition of the Brier Score: Formulation and Interpretation." *Monthly Weather Review* 114(12): 2671–2673.

Oshiro, Thais Mayumi, Pedro Santoro Perez, and José Augosto Baranauskas. 2012. "How Many Trees in a Random Forest?" International Workshop on Machine Learning and Data Mining in Pattern Recognition, MLDM 2012: *Machine Learning and Data Mining in Pattern Recognition* 154–168.

Parr, Terence, Kerem Turgutlu, Christopher Csiszar, and Jeremy Howard. 2018 "Beware Default Random Forest Importances." Accessible at https://explained.ai/rf-importance/. Accessed on April 26, 2019.

Pedregosa, Fabian, et al. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12(1): 2825–2830

Pelánek, Radek. 2015. "Metrics for Evaluation of Student Models." *Journal of Educational Data Mining* 7(2): 1–19.

Rajan, Uday, Amit Seru, and Vikrant Vig. 2015. "The Failure of Models That Predict Failure: Distance, Incentives, and Defaults." *Journal of Financial Economics* 115(2): 237–260.

Stahlecker, Peter, and Götz Trenkler. 1993. "Some Further Results on the Use of Proxy Variables in Prediction." *Review of Economics and Statistics* 75(4): 707–11.

Strobl, Carolin, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. 2007a. "Bias in Random Forest Variable Importance Measures: Illustrations, Sources, and a Solution." *BMC Bioinformatics* January 25, 2007: 8–25.

Strobl, Carolin, Anne-Laure Boulesteix, and Thomas Augustin. 2007b. "Unbiased Split Selection for Classification Trees Based on the Gini Index." *Computational Statistics & Data Analysis* 52(1): 483–501.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1): 267–288.

Wang, J. Christina. 2018. "Technology, the Nature of Information, and FinTech Marketplace Lending." Federal Reserve Bank of Boston Research Department Current Policy Perspectives No. 18-3. Available at https://www.bostonfed.org/-/media/Documents/Workingpapers/PDF/2018/cpp1803.pdf.

Wager, Stefan, and Susan Athey. 2018. "Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests." *Journal of the American Statistical Association* 113(523): 1228–1242.

Wooldridge, Jeffrey M. 2010. *Econometric Analysis of Cross Section and Panel Data.* Cambridge, MA: MIT Press.

Figure 1a. Area Under ROC Curve (AUC) of Cross-Validation and Future Test Samples

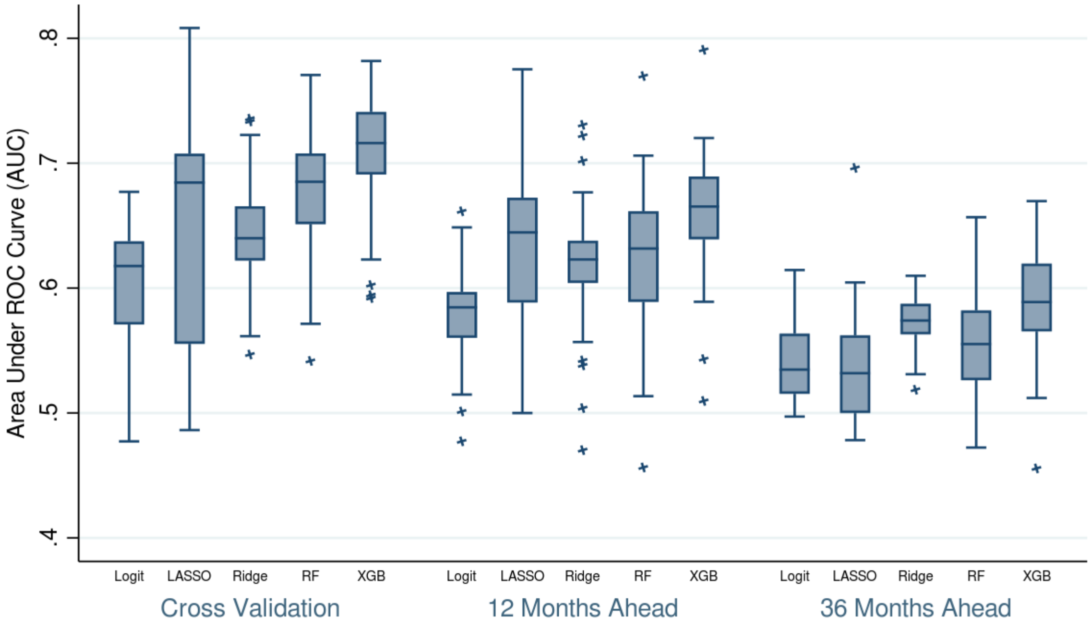
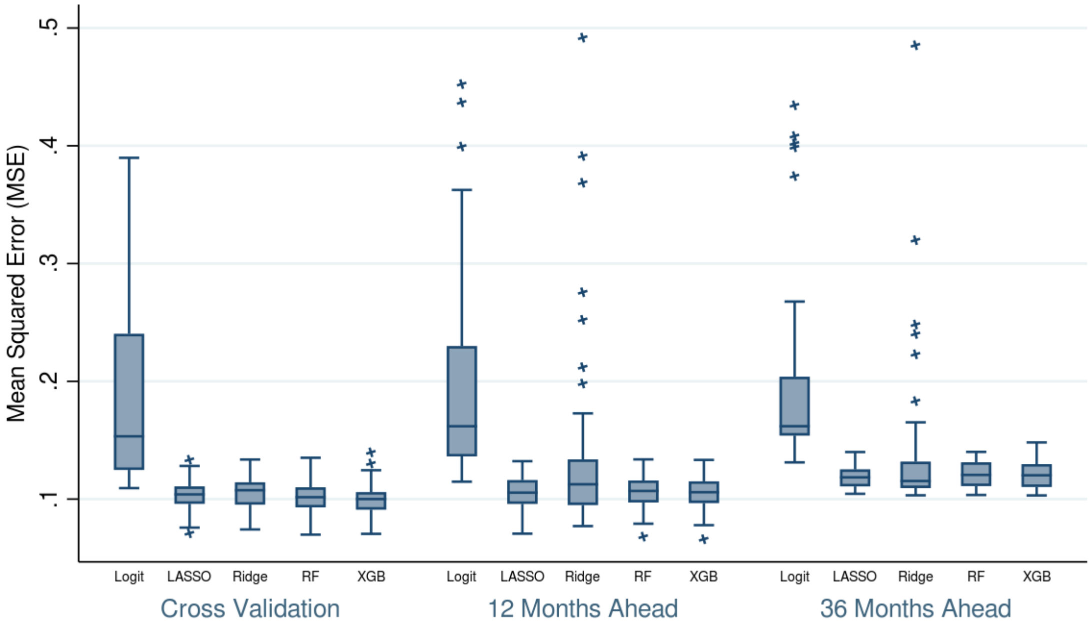


Figure 1b. Mean Squared Errors (MSE) of Cross-Validation and Future Test Samples



Notes: Panels a and b depict the range of the AUC and the MSE, respectively, of predicted probability of default by the five models. Cross Validation: the median fold of the fivefold cross validation test subsamples. 12 Months: 12-month ahead loan cohort as test sample. 36 Months: 37-month ahead test sample, to ensure no overlap with the cohort of 36-month loans used to train the model. RF: random forest. XGB: boosted trees.

Figure 2a. AUC of Random Forest and Boosted Trees Relative to Logistic Model

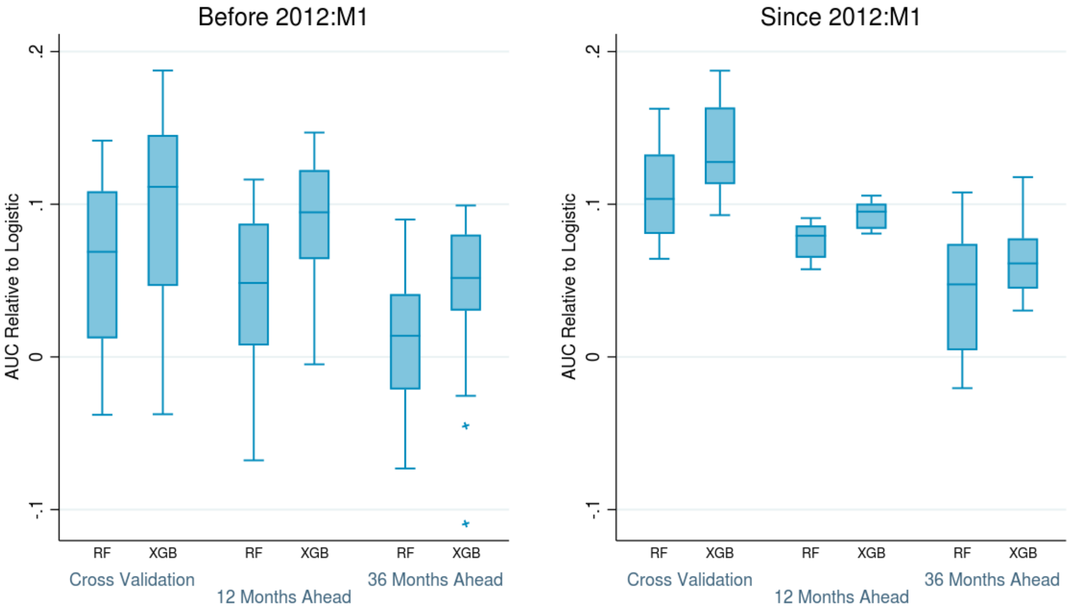
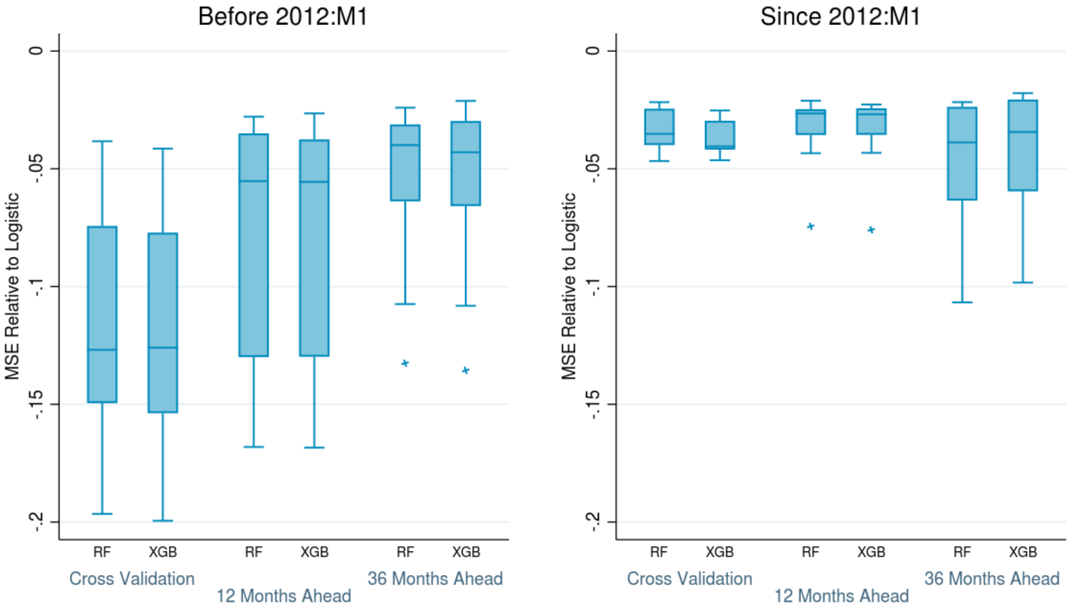


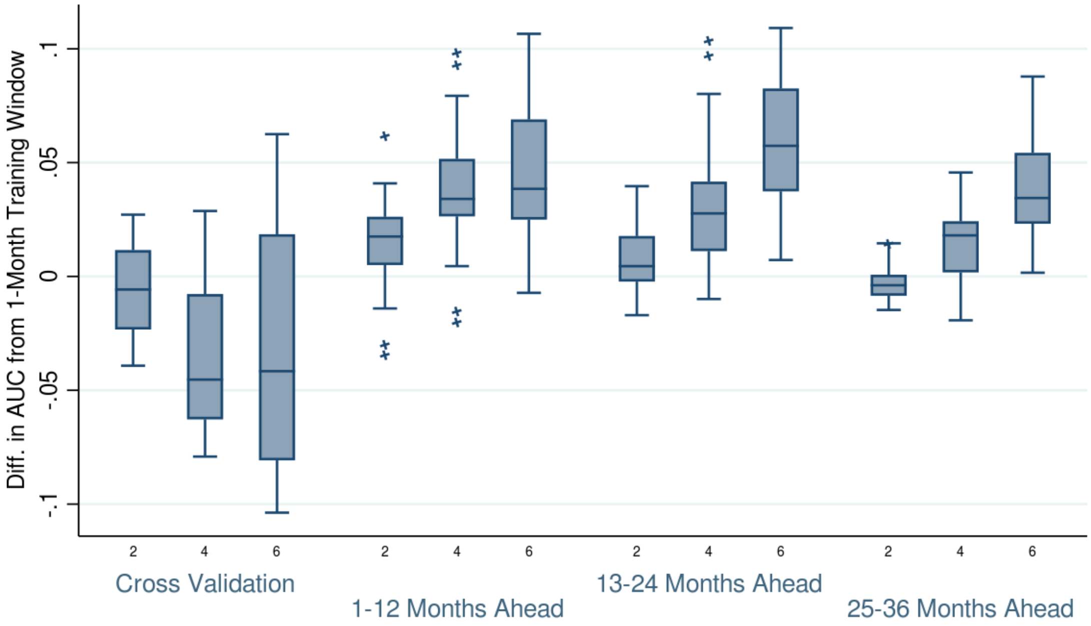
Figure 2b. MSE of Random Forest and Boosted Trees Relative to Logistic Model



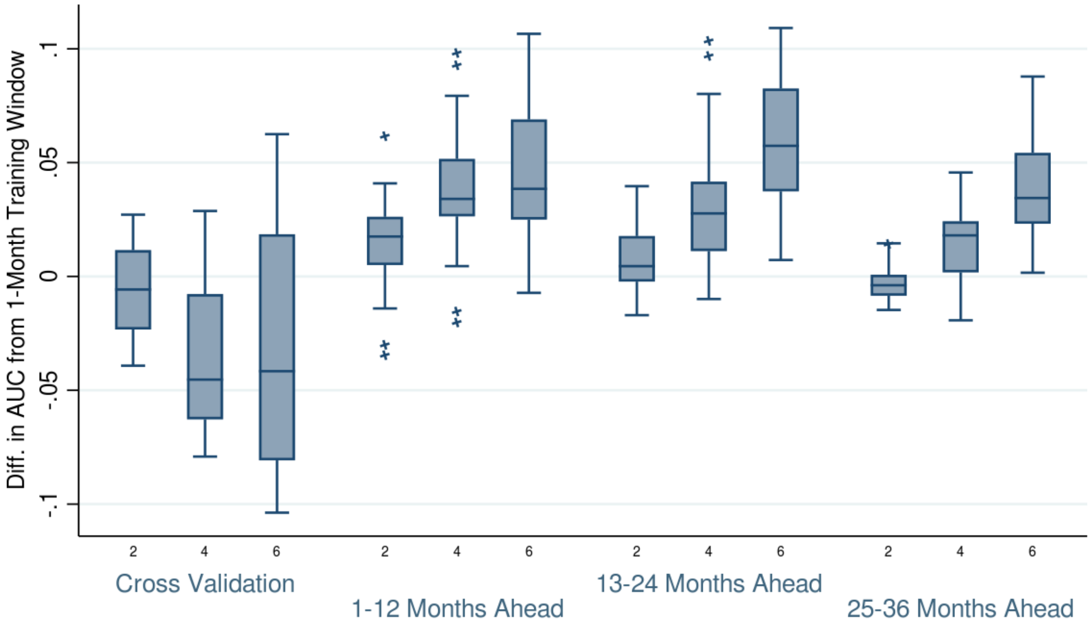
Notes: Panels a and b depict the AUC and the MSE of random forests and boosted trees relative to that of the logistic model, respectively. The statistics are reported separately for the sample period before 2012:M1 and afterward.

Figure 3. Changes in AUC from Training Models Using Multiple Months of Loans (for Training Samples through 2012:M1)

Panel a. Random Forest Model

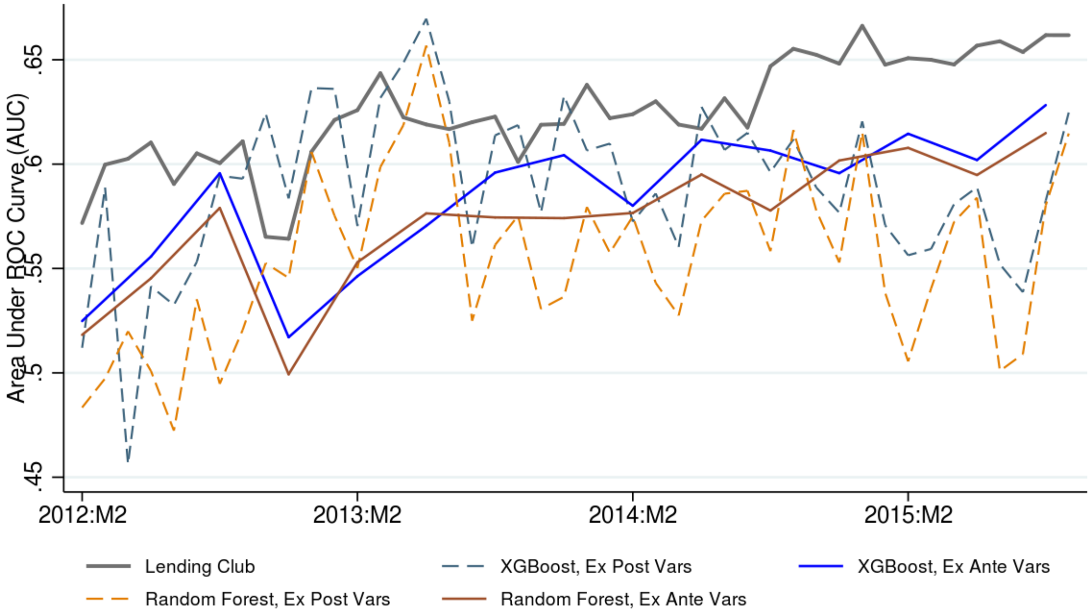


Panel b. Boosted Tree Model



Notes: This figure depicts the AUC of predicted default probability from the baseline specification of the two ML models, estimated using 2, 4 or 6 months of loans among those originated up to and inclusive of 2012:M1. Out-of-sample predictions are produced for loan cohorts 1 to 12 months, 13 to 24 months, and 25 to 36 months after the last month in the training sample.

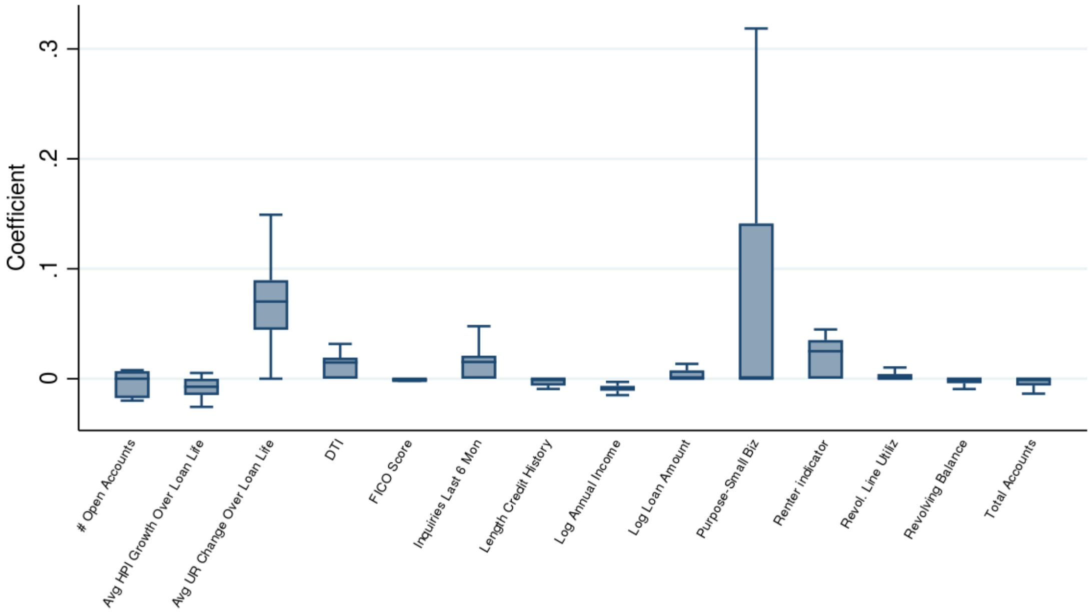
Figure 4. AUC Comparison: LendingClub Credit Grades versus the ML Models



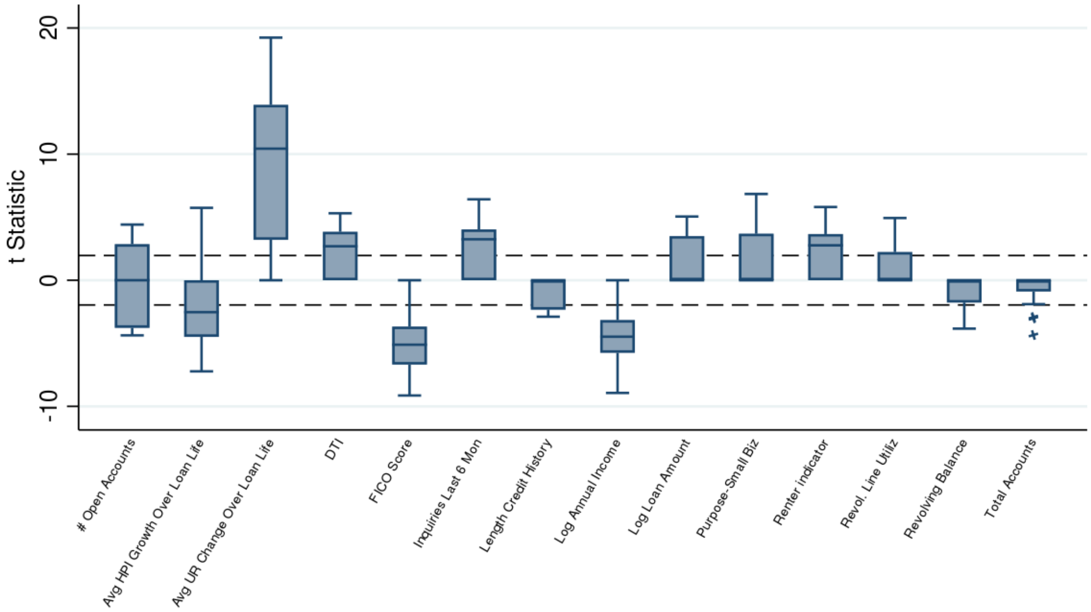
Notes: This figure compares the AUC implied by LendingClub's credit grades and our random forest and boosted tree models, all trained using loans originated 37 months before the date on the horizontal axis. Solid lines for our models: trained without ex post change in unemployment rate and house price over loan life. Dashed lines: models inclusive of these covariates.

Figure 5. Post-LASSO Ordinary Least Squares Coefficients and Significance: Full Sample

Panel a. Coefficients

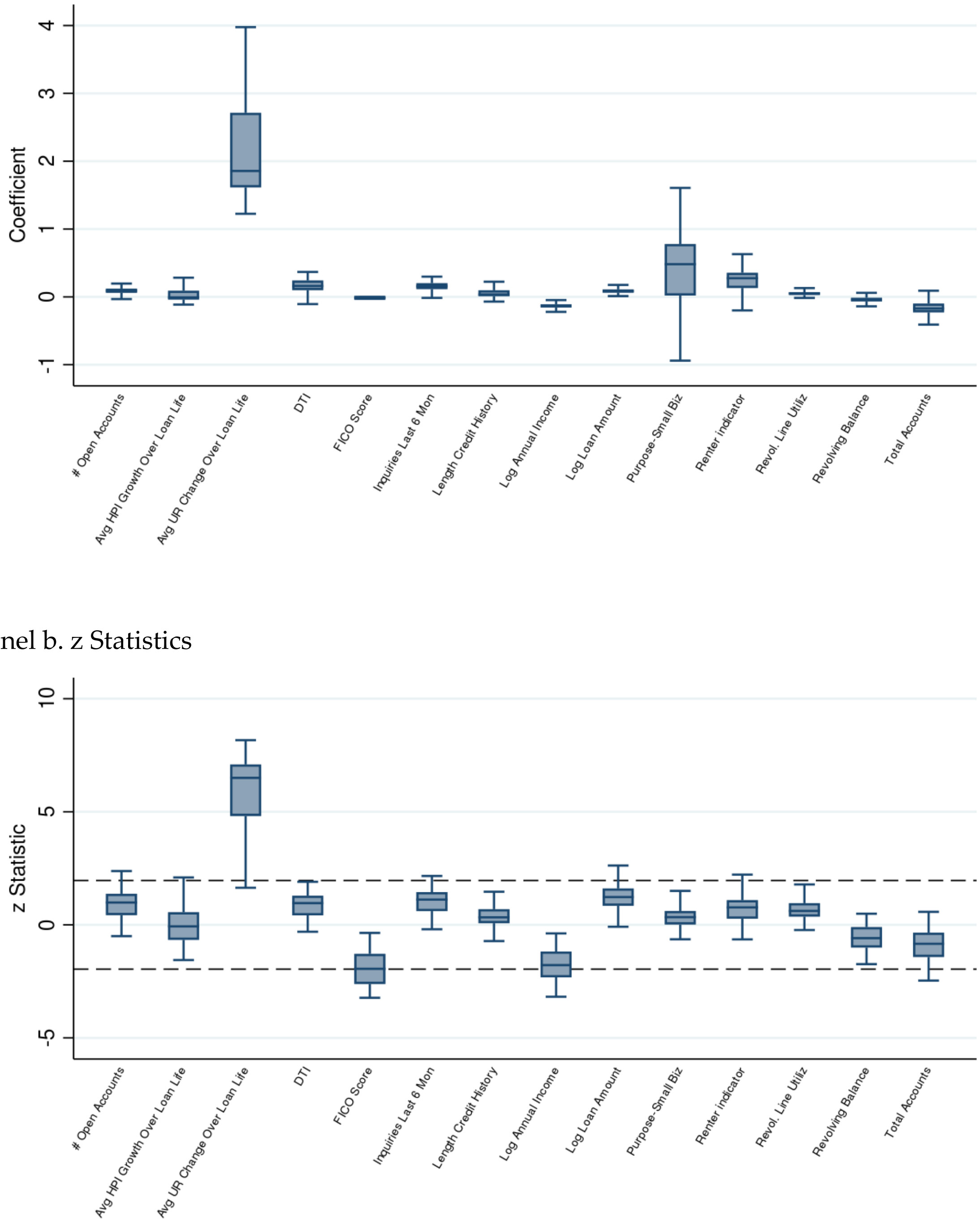


Panel b. t Statistics



Notes: Panels a and b depict the OLS coefficients and t statistics, respectively, using only those covariates that are chosen by the LASSO with loan data by month over the full sample (2009:M1 through 2014:M2). Values outside the 25 or 75 percentiles are omitted in Panel a to avoid extreme coefficients, especially on small-business loans.

Figure 6. Covariate Coefficients and Significance of the Logistic Model: Since 2012:M1



Panel a. Coefficients

Panel b. z Statistics

Notes: Panels a and b depict the logistic coefficients and z statistics, respectively, using all the covariates on loan data by month since 2012:M1. Values outside the 25 or 75 percentiles are omitted in both panels to avoid extreme values.

Figure 7. Feature Importance from the Random Forest Models: Full Sample

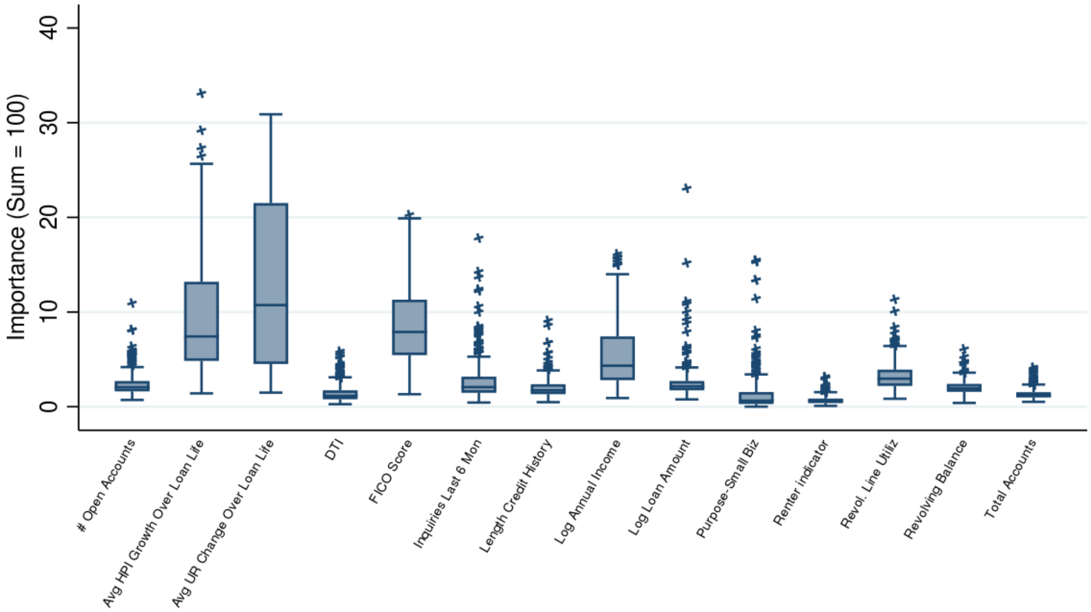
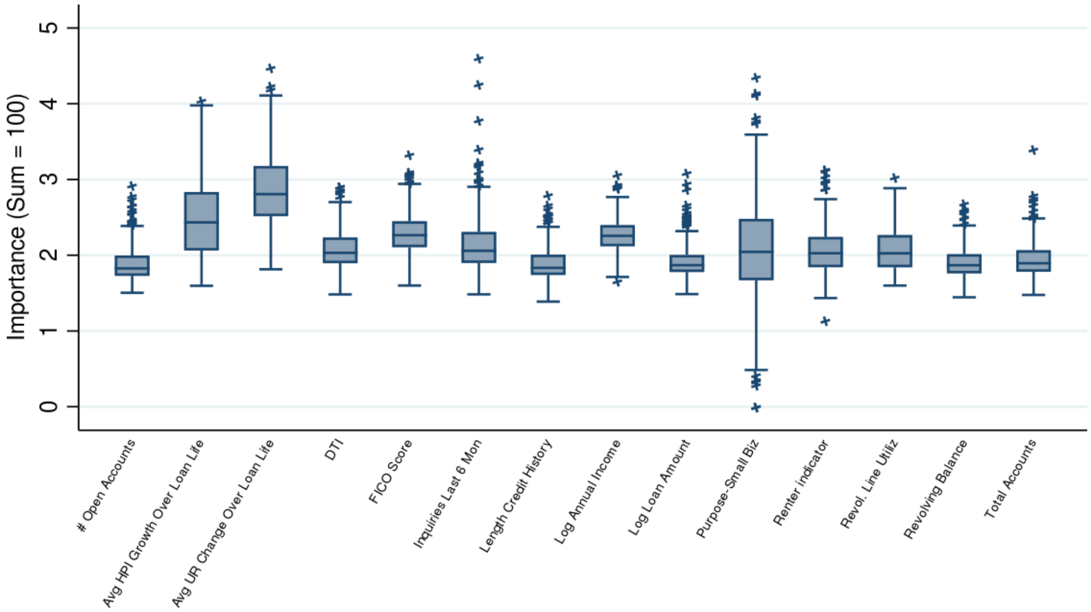
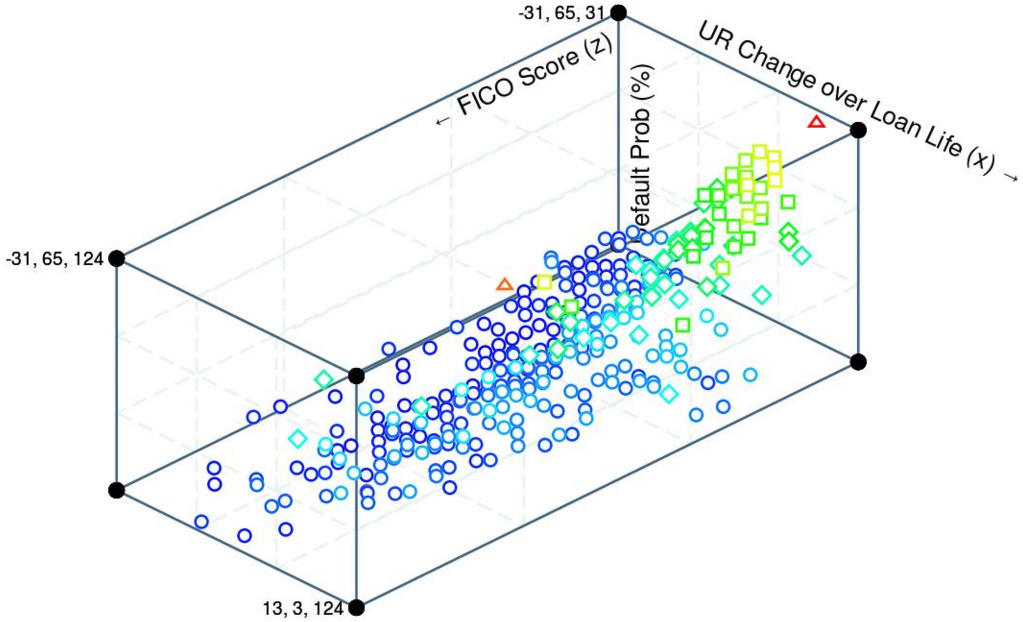


Figure 8. Feature Importance from the Boosted Tree Models: Full Sample



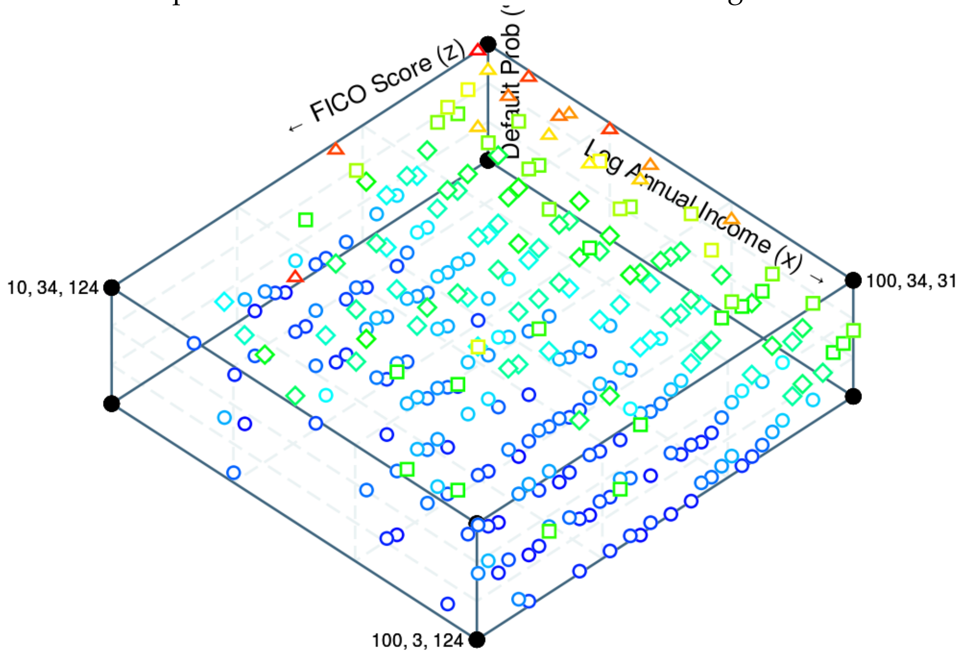
Notes: Panels a and b depict the OLS coefficients and t statistics, respectively, on the covariates that are chosen by the LASSO using loan data by month over the full sample (2009:M1 through 2014:M2). Values outside the 25 or 75 percentiles are omitted in Panel a to avoid extreme ests coefficients, especially on small-business loans.

Figure 9. Partial Dependence of Default on FICO Score and Unemployment Rate Change over a Loan's Life



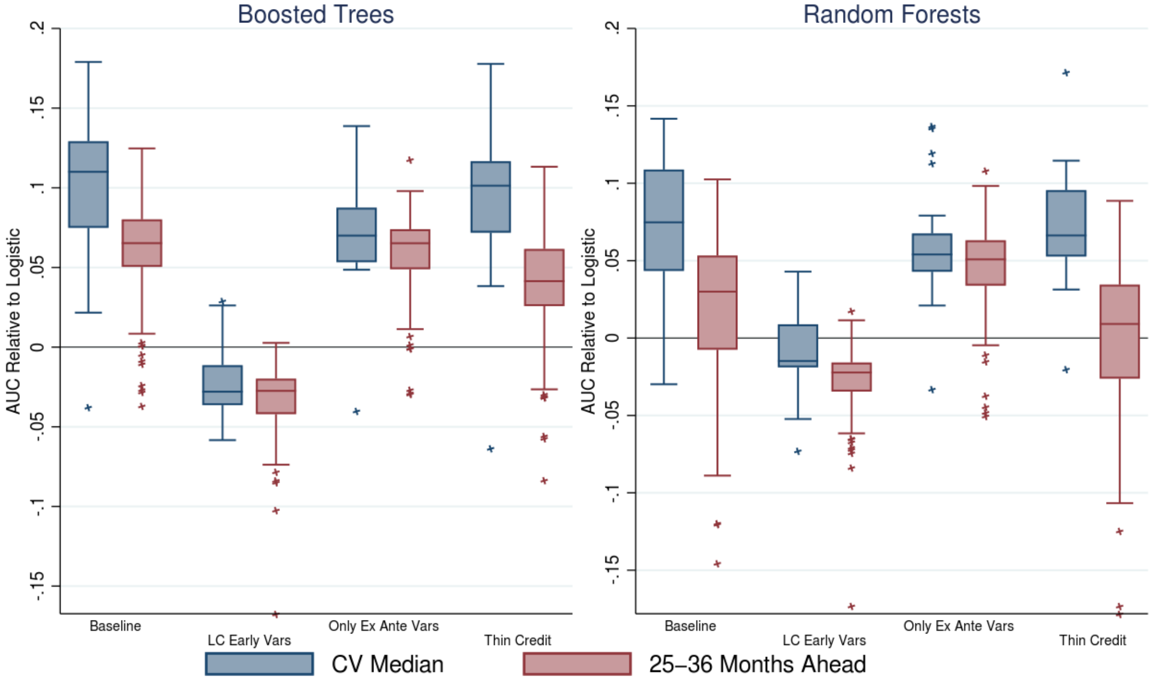
Notes: The two inputs, unemployment rate (UR) change over loan life and the FICO score, are plotted on the x and z axes, respectively, with the arrows indicating the direction of increasing values. Both inputs are rescaled to optimize the proportion: x = UR change over loan life * 10 (–31 means –3.1 percent, for example), z = (the FICO score – 600)/2 (31 means a FICO of 662, for example). The default probability is predicted using a boosted tree model trained on loans made in 2012:M7. It is measured in percent and depicted on the vertical y axis, rising from bottom to top. The three coordinates are reported in the order of the x, y, and z axes, respectively. The marker colors from lower to higher default probabilities are blue (dark to light) circles, cyan to green diamonds, green squares, yellow to red triangles.

Figure 10. Partial Dependence of Default on FICO Score and Log Annual Income



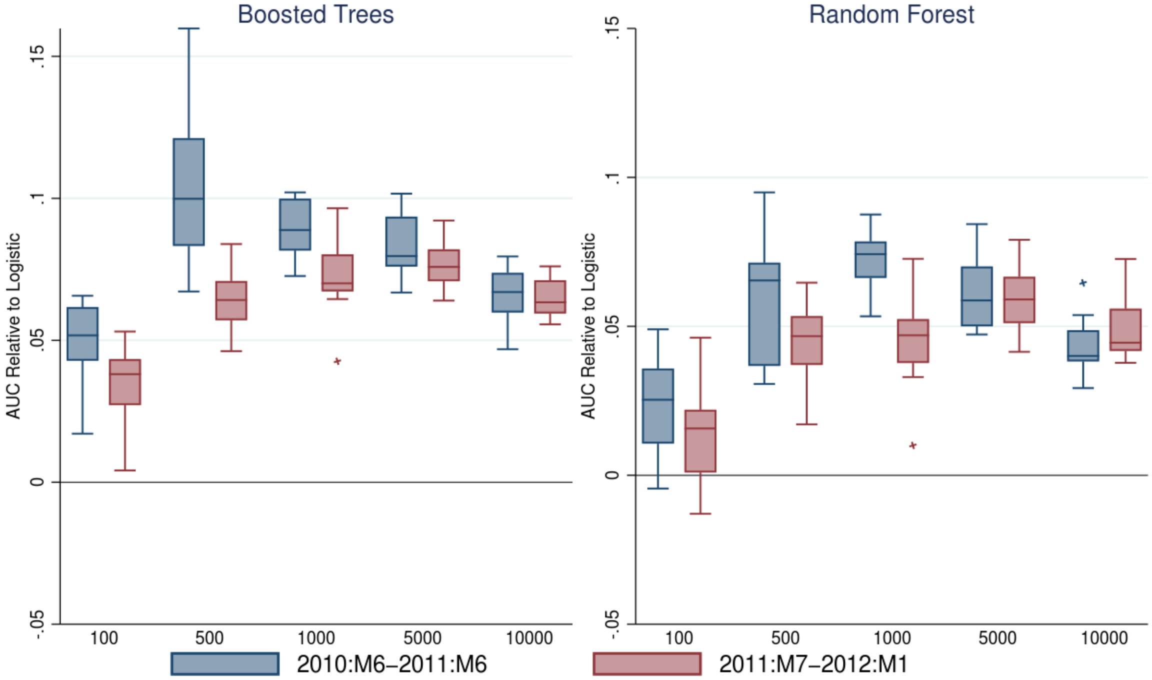
Notes: The two inputs, percentiles of log annual income and the FICO score, are plotted on the x and z axes, respectively, with the arrows indicating the direction of increasing values. Both inputs are rescaled to optimize the proportion: x = percentiles of log annual income (divided into 10 bins), z = (the FICO score – 600)/2 (31 means a FICO of 662, for example). The default probability is predicted using a boosted tree model trained on loans made in 2012:M7. It is measured in percent and depicted on the vertical y axis, rising from bottom to top. The three coordinates are reported in the order of the x, y, and z axes, respectively. The marker colors from lower to higher default probabilities are blue (dark to light) circles, cyan to green diamonds, green squares, yellow to red triangles.

Figure 11. Effect of Different Scopes of Covariates on ML Models' Relative AUC



Notes: This plot compares the boosted trees' and random forests' AUC relative to the logistic's from models trained using four different sets of covariates. "LC Early Vars" contains only the subset of inputs used in the lender's risk grade model posted from mid-2011 to late 2012 (see Footnote 20 for details). ”Only Ex Ante Vars” excludes average unemployment rate change and house price growth over a loan's life. "Thin Credit" includes all the local economic indicators as in the Baseline model, but only a restricted small subset of borrower-specific credit indicator. ”CV Median”: the median of the fivefold cross validation test samples. "25–36 Months Ahead": test samples consisting of loans originated 25 to 36 months after the cohort used to train the model.

Figure 12. Effect of Different Sample Size on ML Models' Relative AUC



Notes: This plot compares the boosted trees' and random forests' AUC relative to the logistic's from models trained using increasing numbers of observations drawn randomly from the same sample period. The AUC is computed on the test samples consisting of loans originated 25 to 36 months after the cohort used to train the model. Two periods are considered for the training data because they are about the same size in total (10,955 versus 10,308 observations) 2010:M6–2011:M6 versus 2011:M7–2012:M1, respectively. "10000" thus denotes the full subsample.

Table 1. AUC of Model Predictions at Different Out-of-sample Horizons

Panel A. Training Data 2008:M11 to 2011:M11

Model		**CV**	**3 Months Ahead**	**12 Months Ahead**	**36 Months Ahead**
Logistic		0.589	0.572	0.572	0.548
	SD	(0.052)	(0.037)	(0.043)	(0.030)
	Median	0.600	0.573	0.578	0.539
	IQR	[0.081]	[0.046]	[0.063]	[0.041]
Random Forest		0.659	0.643	0.616	0.557
	SD	(0.050)	(0.057)	(0.064)	(0.042)
	Median	0.660	0.660	0.602	0.555
	IQR	[0.063]	[0.088]	[0.085]	[0.054]
XGBoost		0.694	0.680	0.661	0.593
	SD	(0.048)	(0.050)	(0.052)	(0.041)
	Median	0.700	0.690	0.663	0.595
	IQR	[0.056]	[0.043]	[0.060]	[0.051]
LASSO		0.601	0.615	0.604	0.540
	SD	(0.096)	(0.100)	(0.086)	(0.044)
	Median	0.582	0.638	0.622	0.533
	IQR	[0.194]	[0.195]	[0.168]	[0.064]
Ridge		0.647	0.636	0.621	0.574
	SD	(0.046)	(0.051)	(0.055)	(0.019)
	Median	0.642	0.647	0.628	0.577
	IQR	[0.064]	[0.048]	[0.057]	[0.025]

Panel B. Training Data 2012:M1 to 2014:M2

Model		CV	3 Months Ahead	12 Months Ahead	36 Months Ahead
Logistic		0.617	0.614	0.598	0.509
	SD	(0.051)	(0.021)	(0.014)	(0.009)
	Median	0.626	0.616	0.597	0.506
	IQR	[0.083]	[0.041]	[0.023]	[0.008]
Random Forest		0.723	0.682	0.674	0.551
	SD	(0.022)	(0.026)	(0.015)	(0.043)
	Median	0.722	0.689	0.675	0.556
	IQR	[0.038]	[0.038]	[0.024]	[0.075]
XGBoost		0.749	0.713	0.691	0.573
	SD	(0.020)	(0.024)	(0.014)	(0.027)
	Median	0.748	0.713	0.695	0.570
	IQR	[0.035]	[0.042]	[0.015]	[0.032]
LASSO		0.728	0.717	0.685	0.526
	SD	(0.018)	(0.014)	(0.028)	(0.030)
	Median	0.733	0.712	0.697	0.530
	IQR	[0.023]	[0.025]	[0.017]	[0.044]
Ridge		0.647	0.643	0.623	0.565
	SD	(0.030)	(0.015)	(0.022)	(0.014)
	Median	0.642	0.644	0.625	0.566
	IQR	[0.049]	[0.022]	[0.024]	[0.012]

Notes: The two panels of this table report the prediction AUC of all five models. "CV" is the left-out test subsample used in the fivefold cross validation. The other three columns denote test samples of loans made 3, 12, and 36 months after the training loan cohort.

Table 2. Correlation of Borrower Characteristics with Boosted Tree Model's Relative MSE

Risk Grade A	-1.270										-1.718
	(1.661)										(1.535)
Risk Grade B	-1.219										-1.421
	(1.343)										(1.306)
Risk Grade C	-0.968										-1.074
	(1.050)										(1.044)
Risk Grade D	-1.020										-1.092
	(0.897)										(0.901)
Risk Grade E	-0.739										-0.772
	(0.689)										(0.692)
FICO Score		0.00253				0.00198	0.00207	0.00199	0.00199	0.00198	0.00637
		(0.00759)				(0.00691)	(0.00691)	(0.00691)	(0.00690)	(0.00691)	(0.00341)
Debt-to-Income Ratio			-0.00211			0.00323	0.00334	0.00319	0.00321	0.00324	3.83e-05
			(0.0228)			(0.0170)	(0.0169)	(0.0170)	(0.0170)	(0.0170)	(0.0137)
Log of Applicant Income				0.284		0.321	0.321	0.319	0.321	0.320	0.379
				(0.347)		(0.297)	(0.298)	(0.297)	(0.297)	(0.298)	(0.244)
Log of Loan Amount					0.0586	-0.0734	-0.0736	-0.0734	-0.0733	-0.0736	-0.0457
					(0.0738)	(0.118)	(0.118)	(0.118)	(0.118)	(0.118)	(0.145)
Log of 3-digit Zip Code Population						-0.0400					-0.667
						(1.373)					(0.863)
Unemploy. Rate Difference from US Rate							1.683**				1.682**
							(0.428)				(0.442)
HPI Growth Rate (t-1)								-0.0596*			-0.0634*
								(0.0290)			(0.0293)
Poverty Share (%)									0.529*		0.467
									(0.262)		(0.254)
Share with Card Utilization >= 85%										0.194	0.0479
										(0.355)	(0.340)
Observations	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105
R-squared	0.344	0.344	0.344	0.344	0.344	0.344	0.345	0.345	0.345	0.344	0.345
# of Training-Test Month Clusters	116	116	116	116	116	116	116	116	116	116	116

Notes: This table summarizes the correlation between the boosted tree model's MSE relative to the logistic model's and borrower characteristics using the OLS coefficients. All the regressions include fixed effects of each training-test sample pair and zip codes, and they cluster standard errors along these two dimensions. ** and * denote significance at the 1 percent and 5 percent critical levels, respectively.

Table 3a. Correlation of Borrower Characteristics with Boosted Tree Model's Relative Prediction Error

Risk Grade A	4.196**										0.765
	(1.077)										(0.752)
Risk Grade B	2.459**										1.011
	(0.812)										(0.692)
Risk Grade C	0.976										0.250
	(0.651)										(0.609)
Risk Grade D	0.0492										-0.321
	(0.585)										(0.571)
Risk Grade E	-0.111										-0.340
	(0.469)										(0.469)
FICO Score		0.0639**				0.0588**	0.0587**	0.0588**	0.0587**	0.0588**	0.0543**
		(0.00880)				(0.00798)	(0.00800)	(0.00798)	(0.00798)	(0.00798)	(0.00677)
Debt-to-Income Ratio			-0.0867**			-0.0542**	-0.0543**	-0.0542**	-0.0542**	-0.0543**	-0.0495**
			(0.0204)			(0.0142)	(0.0142)	(0.0142)	(0.0142)	(0.0143)	(0.0132)
Log of Applicant Income				2.137**		1.266**	1.268**	1.269**	1.267**	1.269**	1.197**
				(0.392)		(0.317)	(0.317)	(0.317)	(0.317)	(0.317)	(0.296)
Log of Loan Amount					1.162**	0.325**	0.325**	0.325**	0.325**	0.326**	0.283**
					(0.141)	(0.0886)	(0.0886)	(0.0887)	(0.0886)	(0.0886)	(0.0956)
Log of 3-digit Zip Code Population						0.932					1.114
						(1.722)					(1.625)
Unemploy. Rate Difference from US Rate							-0.637				-0.601
							(0.615)				(0.641)
HPI Growth Rate (t-1)								0.0232			0.0297
								(0.0468)			(0.0460)
Poverty Share (%)									-0.773*		-0.734
									(0.390)		(0.387)
Share with Card Utilization >= 85%										-0.484	-0.397
										(0.498)	(0.485)
Observations	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105
R-squared	0.599	0.601	0.598	0.598	0.598	0.602	0.602	0.602	0.602	0.602	0.602
# of Training-Test Month Clusters	116	116	116	116	116	116	116	116	116	116	116

Notes: This table summarizes the correlation between the boosted tree model's MSE relative to the logistic model's and borrower characteristics using the OLS coefficients. All the regressions include fixed effects of each training-test sample pair and zip codes, and they cluster standard errors along both of these two dimensions. ** and * denote significance at the 1 percent and 5 percent critical levels, respectively. .

Table 3b. Correlation of Borrower Characteristics with Random Forest Model's Relative Prediction Error

Risk Grade A	-1.538										-3.455**
	(1.050)										(0.726)
Risk Grade B	-1.728*										-2.603**
	(0.783)										(0.662)
Risk Grade C	-1.839**										-2.375**
	(0.631)										(0.585)
Risk Grade D	-1.646**										-2.004**
	(0.571)										(0.553)
Risk Grade E	-0.984*										-1.280**
	(0.460)										(0.455)
FICO Score		0.0266**				0.0256**	0.0256**	0.0256**	0.0256**	0.0256**	0.0342**
		(0.00853)				(0.00772)	(0.00774)	(0.00772)	(0.00772)	(0.00772)	(0.00651)
Debt-to-Income Ratio			0.0622**			0.0587**	0.0586**	0.0587**	0.0587**	0.0587**	0.0525**
			(0.0193)			(0.0131)	(0.0130)	(0.0131)	(0.0131)	(0.0131)	(0.0120)
Log of Applicant Income				0.0928		-0.590	-0.588	-0.587	-0.589	-0.587	-0.473
				(0.397)		(0.331)	(0.331)	(0.331)	(0.331)	(0.331)	(0.313)
Log of Loan Amount					1.094**	1.148**	1.148**	1.148**	1.148**	1.148**	1.201**
					(0.134)	(0.0791)	(0.0790)	(0.0791)	(0.0791)	(0.0790)	(0.0863)
Log of 3-digit Zip Code Population						1.218					1.570
						(1.865)					(1.664)
Unemploy. Rate Difference from US Rate							-1.002				-1.005
							(0.639)				(0.663)
HPI Growth Rate (t-1)								0.0377			0.0422
								(0.0459)			(0.0452)
Poverty Share (%)									-0.617		-0.568
									(0.385)		(0.382)
Share with Card Utilization >= 85%										-0.359	-0.254
										(0.491)	(0.479)
Observations	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105	1,351,105
R-squared	0.605	0.605	0.605	0.605	0.605	0.606	0.606	0.606	0.606	0.606	0.607
# of Training-Test Month Clusters	116	116	116	116	116	116	116	116	116	116	116

Notes: This table summarizes the correlation between the random forest model's prediction errors relative to the logistic model's and borrower characteristics using the OLS coefficients. All the regressions include fixed effects of each training-test sample pair and zip codes, and they cluster standard errors along both these two dimensions. ** and * denote significance at the 1 percent and 5 percent critical levels, respectively.

Table 4. Boosted Tree Model's Relative AUC by Indicators of Borrower Characteristic Deciles

				Log of applicant	Log of loan	Log of 3-digit	UR Diff. from	HPI growth	Poverty	Share with a	Share of card
		FICO	DTI	income	amount	zip code pop.	US Rate	rate (t-1)	share (%)	college degree	utilization >= 85%
Risk Grade A	-0.116										
	(0.890)										
Risk Grade B	-1.354										
	(0.717)										
Risk Grade C	-1.017										
	(0.477)										
Risk Grade D	-0.993										
	(0.422)										
Risk Grade E	-0.456										
	(0.655)										
Decile 1		-0.497	-0.579	-0.996	-1.647*	2.691	1.691	3.672*	1.209	-0.0930	1.283
		(0.465)	(0.713)	(0.577)	(0.511)	(0.983)	(0.977)	(0.900)	(0.781)	(0.191)	(1.003)
Decile 2		-0.746	-0.131	-1.303	-1.682	2.985	0.777	3.021*	1.726*	0.491	0.287
		(0.706)	(0.658)	(0.726)	(0.726)	(1.371)	(0.559)	(0.814)	(0.605)	(0.241)	(0.896)
Decile 3		-0.555	-0.282	-1.133*	-1.467*	2.514*	1.455	3.599*	1.082	0.772	1.004
		(0.726)	(0.562)	(0.351)	(0.392)	(0.620)	(0.885)	(1.157)	(0.830)	(0.458)	(0.438)
Decile 4		-1.190	0.401	-1.073	-0.968	2.166	2.257*	2.091	1.595**	-0.218	1.274
		(0.458)	(0.616)	(0.525)	(0.469)	(1.015)	(0.578)	(1.214)	(0.303)	(0.487)	(0.601)
Decile 5		-0.837	0.0783	-1.149*	-0.924	1.374	1.889	2.720*	-0.0144	-0.418	-0.168
		(0.318)	(0.507)	(0.260)	(0.711)	(1.246)	(1.040)	(0.657)	(0.476)	(0.483)	(0.465)
Decile 6		-0.417	0.209	-0.0313	-1.007	0.203	1.672**	2.063	0.757	-1.719	0.674
		(0.396)	(0.715)	(0.272)	(0.558)	(1.287)	(0.341)	(1.071)	(0.909)	(1.032)	(0.619)
Decile 7		0.0759	0.389	-0.753*	-0.219	-0.880	1.208*	2.039	0.819	-1.904	0.805
		(0.270)	(0.782)	(0.234)	(0.438)	(0.476)	(0.393)	(1.092)	(0.747)	(1.363)	(0.410)
Decile 8		0.457	-0.317	-0.879	0.0801	-2.232	2.424*	2.086	1.325	-2.199*	1.660
		(0.266)	(0.349)	(0.578)	(0.480)	(1.022)	(0.770)	(1.085)	(0.635)	(0.601)	(0.686)
Decile 9		0.0769	0.441	-0.146	0.444	-1.119	0.647	0.284	1.461*	-1.089	0.879*
		(0.532)	(0.237)	(0.578)	(0.365)	(1.235)	(0.374)	(0.731)	(0.318)	(0.703)	(0.282)
Observations	694	1,160	1,160	1,160	1,160	1,160	1,160	1,154	1,160	1,160	1,160
R-squared	0.270	0.269	0.286	0.257	0.273	0.346	0.228	0.158	0.249	0.266	0.253
# cluster	5	5	5	5	5	5	5	5	5	5	5

Notes: This table summarizes the boosted tree model's AUC relative to the logistic model's by decile of borrower characteristics and local economic conditions using the OLS coefficients. "UR Diff. from US Rate": difference between unemployment rate in a 3-digit zip code minus the US unemployment rate. All the regressions include fixed effects of each training and test sample month, and standard errors are clustered by training month. ** and * denote significance at 5 percent and 10 percent critical levels, respectively.

Appendix I. The Machine Learning Methods and Their Implementation

A1.1. LASSO, Random Forest and Stochastic Gradient Boosting

This section discusses the machine learning (ML) methods applied in this study in greater detail, focusing on the key property of each method and the associated intuition. For in-depth exposition, see Hastie, Tibshirani, and Friedman (2008, second edition).

A1.1.1 Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO is used here to gain intuition about which features are important. It achieves feature selection by minimizing a loss function subject to a maximum constraint on the sum of the absolute value of all the feature coefficients.[54] LASSO was first developed for the linear regression with a quadratic loss function. The Lagrangian representation of LASSO often adds a penalty term equal to a multiple (λ) of the sum of the absolute value of all the coefficients:[55]

$$\min_{\beta \in \mathbb{R}^P} \left\{ \frac{1}{N} \| y - X\beta \|_2^2 + \lambda \| \beta \|_1 \right\}. \tag{12}$$

The constraint can in fact be more intuitively expressed as: $\sum_i \left(|\beta_i| / |\beta_i^{OLS}| \right) \leq s,\ s \in (0,1]$, with β^{OLS} being the OLS estimates. The above equation makes clear the importance of standardizing the feature vector X before estimating LASSO: Only with every feature x measured on the same scale after standardization can the elements of the coefficient vector β be compared by magnitude.

Clearly, the model output—that is, how many features will be assigned non-zero coefficients and their magnitude—is conditional on λ. A larger (smaller) λ penalizes any additional coefficients more (less) aggressively and thus tends to set more (fewer)

[54] The absolute value sum is also referred to as the L1, or ℓ_1, norm. Hence the regularization used in LASSO is referred to as the L1 regularization. By comparison, a ridge regression results from regularization with the L2 norm (that is, the sum of squared coefficients), which also shrinks the size of coefficients but does not set any to zero strictly.

[55] This is equivalent to capping the sum of the absolute value of all the feature coefficients, or capping the absolute magnitude of every individual coefficient.

coefficients to zero. For any given problem, a single or a set of λ values minimizes the loss function. Thus, λ is the hyperparameter that needs to be tuned for the LASSO regression.

Efron et al. (2004) demonstrate that the LASSO generally arrives at nearly the same solutions as forward-stagewise estimations, which are closely related to boosting, and the solutions are equivalent when the coefficients follow monotonic paths. In fact, a special stagewise procedure called least angle regressions (LARS) is an efficient method for deriving the LASSO estimates.

If LASSO is used as a feature selection tool, we find that when a separate LASSO model is estimated for each cohort of loans using all the available historical data, the number of features chosen rises with the increasing size of data over time. This is to be expected. In general, once a feature is chosen for one loan cohort, it remains in the feature set. Also as would be expected, among each group of highly correlated features, there may be some instability, in that some are selected in certain months while the others are selected in the remaining months, as their coefficients hover around the threshold and a specific coefficient can be set to zero by LASSO somewhat randomly. On the other hand, the prediction results are minimally affected.

A1.1.2 Classification and Regression Trees

The tree models systematically and repeatedly partition a sample space so that subjects are allowed to have an outcome with a potentially nonparametric heterogeneous relationship with their covariate values.[56] At each step, the algorithm selects the feature and its value over which to partition the data by optimizing the chosen criteria, such as maximizing the reduction in impurity, often measured as the Gini index gain, which for the case of a Bernoulli outcome variable is equivalent to its empirical variance. (See, for example, Strobl et al. 2007b for a derivation of the Gini index gain.) This process continues until the pre-set terminal condition is met, such as reaching the lower bound of the Gini

[56] This attribute of tree-based models is used to estimate heterogeneous treatment effects; see Athey and Imbens (2016), for example.

gain or the minimum number or fraction of observations in each terminal node. Deeper trees correspond to higher orders of interaction effects. Each observations is eventually assigned to one of the terminal nodes, which are the nodes formed at the final step of the process. All the observations in a terminal node take on a single predicted value, either the mean response in the regression tree or the class that is the majority in a classification tree. Thus, this process can obtain a different relationship between the response variable and input features within each terminal node.

A tree structure is in fact evident in the risk-grading algorithm used by LendingClub from mid-2011 to late 2012. Appendix V exhibits one of the rating algorithms typical of that period, as copied from the Web Archive. Grade adjustments are made based on partitions of the FICO score, interacted with the loan amount, the number of recent inquiries, and the chosen loan maturity.

For a classification problem, the predicted probability of an event for each observation is often computed as some form of weighted average outcome among the training data points that reach the same terminal node.[57] We adopt this approach to estimate the default probability for each loan. We estimate classification trees primarily using the classification and regression Trees (CART) algorithm, originally developed by Breiman et al. (1984). Most tree-based models, including random forests, which will be discussed next, use CART. But it is well known that the criteria used for partition in CART, either the impurity reduction or the Gini gain, can lead to biased partitions, as they tend to favor variables that are larger in scale or have more missing values or, in the case of categorical variables, have more categories. (See Strobl et al. 2007a for derivations of these biases.) We therefore also experimented with the other tree-building algorithm: the conditional inference tree (CIT) developed by Hothorn, Hornik, and Zeileis (2006). CIT is, in principle, free from CART biases (see Hothorn, Hornik and Zeileis 2006 for empirical evidence), because CIT uses the p-value of the chi-square test to choose the partition at

[57] Or a binary prediction can be created by setting a threshold value above which the predicted outcome is one, and is zero otherwise.

each step, which accounts for the number of categories and sample size explicitly. Unfortunately, applying an off-the-shelf CIT-based random forest is too time-consuming for our sample size; we therefore estimate only a CART-based random forest.

A1.1.3 Random Forest

A random forest is an ensemble of trees, each of which is built using a random subset of features over a bootstrapped (with replacement) random sample.[58] The final value of the predicted classification or probability is calculated as the average of the estimates obtained from many trees thus grown, hence the name. Using only a randomly chosen subset of the features to build each tree is meant to further lower the variance of the ultimate prediction by reducing the correlation across trees.[59] There are alternative ways to achieve the randomization over features, such as using only a subset of features to decide each split or to grow an entire tree. Regardless of the exact scheme, the CART algorithm is almost always used to build each tree. Given the superior ability of the CIT algorithm to achieve unbiased tree splitting, we also experiment with a random forest built on CITs, but only on a small random subsample. It is used only as an alternative for robustness checks, because the available R package has difficulty handling large data sets.

Another, arguably more general way to think of decision trees, and in turn the random forest method, is to regard them as making predictions using the weighted average of nearby sample data points. Following this logic, it can be shown that random forests may be regarded as a specific form of a more general class of models—potential nearest neighbor predictors (PNNs), to which k-nearest neighbor (k-NN) models also belong (Lin and Jeon 2006). Random forests are adaptive PNN estimators, in that data determine the “distance” over which to define NNs—the distance is shorter for more

[58] An ensemble of trees grown over bootstrapped samples but always the full set of features is called bagging (bootstrapped aggregating), introduced by Breiman (1996). It generally achieves less variance reduction than random forests because the trees are more correlated (Breiman 2001).
[59] Otherwise, in a sample characterized by a few important features, they will be represented in most trees, leading to high correlations across trees and potentially large bias in out-of-sample predictions. The ensemble without feature randomization is often called bagged trees.

relevant features. The regular nonadaptive k-NN method, in contrast, treats all dimensions equally, regardless of their correlations with the outcome variable. It thus almost invariably leads to inferior predictions, as demonstrated by Wager and Athey (2018). For this reason, we do not consider the regular k-NN method in this study.

A1.1.4 Stochastic Gradient Boosting

At a high level, boosting can be viewed as an additive approach to fitting data relationships flexibly. More specifically, in each successive step, boosting fits new parameters for the base learner to minimize the residual error from the previous step (by assigning greater weights to observations that have been wrongly classified, such as in AdaBoost), or to maximize the gradient descent. This, in effect, improves the model where it was particularly inaccurate. Friedman, Hastie, and Tibshirani (2000) demonstrate the equivalence between boosting and the statistical principle of additive modeling to estimate the maximum likelihood. This equivalence can explain the low bias of boosting, and it may be responsible for boosting's resistance to overfitting as well.

Friedman (2001) derives gradient boosting as an approach of greedy stepwise functional approximation; the parameters of the base learner in each step and the step size are chosen to achieve the greatest descent in gradient. In the case of linear models, Rosset, Zhu, and Hastie (2004) prove a form of equivalence between boosting and a sequence of incremental L1-regularized minimization of the loss function.

Rosset, Zhu, and Hastie (2004) further show a fundamental similarity between boosting and kernel support vector machines (SVM), for both can be characterized as methods for regularized optimization with high-dimension features, although SVM uses L2-regularization and a different loss function (the hinge loss). In the benchmarking study by Lessmann et al. (2015), SVM performs noticeably worse than random forest and boosting models. In our experiments with SVM, we find similar results (available upon request). We thus choose not to consider SVM for the bulk of our analysis. One possible reason that SVM performed worse on our data is that the dimension of features relative to sample size is not large enough to benefit from SVM, and the data structure is closer to being sparse; that is, a relatively small subset of covariates accounts for the bulk of the variation, and L2-regularized models tend to perform worse in such cases (as shown by Tibshirani 1996). If so, it is possible that the SVM may perform better given a different set of credit data, one in which many covariates matter moderately.

A1.2. List of Python Packages Used for Model Estimation

We use Python for training all the models, mainly for consistency, even though its main strength lies in the ML models. The specific packages used are listed below:

Pandas: This is for importing and other manipulations of data sets.

SKLearn: This is the machine learning library for Python. Within this library, we mainly use the following functions:

- GridsearchCV, to tune hyperparameters over a grid of possible values
- RandomForestRegressor, to estimate random forest models
- LogisticRegression, to estimate logistic regressions; this routine permits regularization
- LassoCV, to estimate LASSO regressions with cross-validation to optimize the penalty parameter
- XGBoost, to estimate CART-based gradient boosting models; within this, we have used only XGBRegressor

We use R only for assessing the degree of bias in feature importance due to the CART's intrinsic bias, using the cforest packages—the random forests corresponding to the conditional inference tree method. Unfortunately, these modules are too slow for the size of our data in later years and can be applied only to small random subsamples. We thus omit these results here, but they are available upon request.

A1.3. Feature Selection

To help gain intuition about which covariates account for most of the variation in default, we estimate a LASSO regression using all the available features, including the zip code dummy variables, on the three-year loans month by month. Covariates with non-zero coefficients should contain the most explanatory power. Later, they can be compared with the list of covariates identified as important by the other models. All the covariates are normalized to have a mean of zero and a standard deviation of one before being used in the estimation. To tune the penalty parameter α (corresponding to λ in equation [4] in Section III.1), we utilize Python's LassoCV, setting n_alphas to 50 and eps to 1e-10. N_alphas determines the number of α's to check via cross validation. Eps is the value set for the ratio between the smallest and the largest alpha. We run fivefold cross-validation and generally arrive at an optimally chosen value of alpha that is extremely small (less than 1e-3), meaning little regularization is required.

In the baseline specification of the ML models, we allow the two algorithms XGBoost and RandomForest to use all the available features (including the zip code dummy variables) within the training sample for each loan cohort. Both models have built-in regularization to deal with a large set of features that are of varying degrees of relevance. Tuning of the relevant parameters is discussed below. The features identified as important by the more flexible ML methods can then be compared with those chosen by LASSO.

A1.4. Hyperparameter Tuning

Each LASSO estimate is derived conditional on the degree of regularization, which is a hyperparameter that needs to be tuned (that is, selected to minimize the loss function). Both XGBoost and RandomForest have multiple parameters that need to be tuned. The tuning process, along with the parameter values tried, is described below, and we provide a table with the final chosen values. In addition, we provide the minimum and the maximum mean squared errors (MSEs) of all parameter combinations. We report, as an example, the MSE of the chosen model that uses data from loans originated up to and including 2010:M1.[60] The tuning of hyperparameters is carried out using the baseline

[60] We have examined other sample periods and found that the hyperparameters are highly stable.

feature set. We then also apply these hyperparameters to the alternative model specifications (many of which serve as robustness checks), so that the difference in prediction accuracy is due only to the difference in model specification.[61]

A1.4.1 Stochastic Gradient Boosting: XGBoost

Many implementation methods of the boosting model have been developed to increase the speed and sophistication of the algorithm. We use XGBoost, one of the recently popular methods for stochastic gradient boosting. Relative to the traditional boosting algorithms, such as Adaboost (see Friedman, Hastie, and Tibshirani 2000 for an exposition of Adaboost in the context of additive models), XGBoost has an advantage in terms of both execution speed and robustness of performance across different types of models. Moreover, XGBoost includes multiple regularization hyperparameters that can further limit the tendency of boosting to overfit, which is described more thoroughly below.

We use the XGBRegressor routine from the XGBoost packages (see Chen and Guestrin 2016). Compared with the standard approach, XGBoost adds a penalty term to the objective function to attain further regularization:

$$L(\Phi)=\sum_i l(y_i, y_i)+\sum_k \Omega(f_k), \text{ where } \Omega(f)=\lambda T+\frac{1}{2}\lambda||w||^2.$$

The function L sums up the prediction loss, denoted as l(.), at each data point i over all trees f(.). T is the number of leaves in each tree k, while w is the score for each leaf so that the last term further smooths the scores to avoid overfitting.

We implement the XGBregressor (from the xgboost library) because it has good compatibility with common functions in sklearn. We set the objective to be binary:logistic, given the nature of our response variable (0/1 default outcome). We hold constant n_estimators (which designates the number of trees grown in each base learner in the boosting model) at a value (500) that is found to reach the range of diminishing returns

[61] The hyperparameters are, in fact, also insensitive to the changes in the feature set.

from more trees.[62] Some hyperparameters are set at their recommended values from previous studies.[63] For example, the subsample determines the fraction of observations used for deciding each split in a tree. We set it at 0.5, in the range that Friedman (2002) finds minimizes the prediction error, although we also find that model output is insensitive in the neighborhood of 0.5. In sum, we tune the following hyperparameters:

- Learning_rate determines the shrinkage placed on each additional base learner tree in the model. A low learning rate reduces the contribution of each individual tree and is recommended to guard against overfitting (Friedman 2002). A high (low) learning_rate is paired with a small (large) number of base learner trees.
- Min_child_weight sets the minimum number of observations in each terminal node in a base learner tree. We tune this over low possible values. Because of the low percentage of defaulted loans (on average from 10 percent to 20 percent), this number is typically required to be very small to isolate those that are defaulted on.
- Max_depth sets the maximum depth of each individual tree. The greater the depth, the larger the number of potential interactive effects allowed. But deep trees may increase bias and reduce prediction accuracy. We tune this over 2, 4, 6, and 8.
- Gamma is the minimum improvement of the objective function required to split a node in each tree. Absent recommended values, we tune this over a wide range.
- Colsample_bytree is the fraction of features drawn randomly to fit each tree. We tune this from 0.3 to 1.0 and find that the predictions are insensitive to this parameter. Given the large number of zip code dummy variables (relative to other features), we thus choose to consider all the features in every tree.[64]

Of all the combinations for XGBRegressor hyperparameters using loans originated

[62] In general, model performance improves monotonically in the number of trees, but Oshiro, Perez, and Baranausaks (2012) find diminishing returns to the number of trees.

[63] We choose to impose some L2 regularization with reg_lamda = 1 but no L1 regularization (with reg_alpha = 0), because the tree model by nature already carries out feature selection.

[64] Hastie, Tibshirani, and Friedman (2008) document that the accuracy of random forests is adversely affected if the ratio of noise to signal is high among features, whereas boosting is more robust. So this choice is made more to maintain consistency between boosted trees and random forest.

in 2010:M1, we observe small variation in the MSE between the best and the worst models corresponding to hyperparameter choices: The lowest and highest values are 0.1144 and 0.1357 (differing by about 16 percent), respectively. And the combination of hyperparameters has little discernable relationship with the MSE. One exception by comparison is that a smaller learning rate, when paired with a large number of trees, is generally favored. But XGBoost otherwise prefers a minimally regularized model.[65] We thus choose to use a set of common parameter values for training models across all sample months and between the boosted trees and random forests to make comparison of results from the two models more consistent.

Parameter	**Chosen Value**	**Parameter**	**Chosen Value**
n_estimators	500	**gamma**	1.00E-06
learning_rate	0.01	**min_child_weight**	1
max_depth	6	**colsample_bytree**	1

A1.4.2 Random Forests: RandomForestRegressor

We utilize the RandomForestRegressor from the sklearn library. First, we hold constant n_estimators at 500, matching its value used in the XGBRegressor model. We set max_features at auto, which means that all the features can be considered for each split in each tree, and the program chooses to use all the features. We tune over the following hyperparameter:

- Max_depth determines the maximum depth of each individual tree. We again tune over 2, 4, 6, and 8.
- Min_impurity_decrease sets the minimum improvement in impurity required to make an additional split at a node and thus analogous to gamma in the XGBRegressor model. We tune this over a similar range to gamma.
- Min_samples_split picks the minimum fraction of observations required to split at a node. We tune this over a comparable range to min_child_weight in XGBRegressor.

[65] The result of preferring less regularization appears to be specific to our data and is not a general finding in previous studies.

Of all the hyperparameter combinations for RandomForestRegressor using the same data as XGBoost, we again find the differences in MSE to be minimal across hyperparameter choices. The optimal hyperparameter values are listed below, chosen based on the same principles as described above under XGBoost.[66]

Parameter	Chosen Value	Parameter	Chosen Value
n_estimators	500	min_samples_split	0.0001
max_features	auto (all)	min_impurity_decrease	1.00E-06
max_depth	6		

Appendix II. Lists of Covariates (Features) Used in the Estimations

A2.1. Full List of All Available Covariates (Features)

The following is the full list of covariates (that is, features) that can be used to predict the default outcome. In the baseline specification, each model is allowed to select from among the full list of features. See Appendix III for details of their construction.

Loan-, Borrower-Specific Variables	*Zip-code-Area Economic Variables*
# Delinquencies in the Past 2 years	*Real Balance of Credit Card Debt*
# Derogatory Public Records	*Real Balance of Student Loan Debt*
# Inquiries in the Past 6 Months	*Real Balance of all Other Nonmortgage Debt*
# Open Accounts	*3-, 6-, 9-, and 12-Month Lagged 3-Month Change in Unemployment Rate* [1]
# Total Accounts	*1-, 2-, 3-, and 4-Quarter Lagged Quarterly Growth Rate of FHFA House Price Index* [1]
Debt to Income Ratio	*Ratio of Prime Age Population to US Total*
Employment Indicator	*Share of Prime Age in Population*
Employment Length Indicators	*Share of Young Population (≤ 40 years old)*
FICO Score	*Local Poverty Rate*
Home Ownership Indicators	*Share of LC Loans to Consumers with Income below the Poverty Line*
Indicators That the Income or Income Sources Were Verified	*Average Rate of CRE NonPerforming Loans (NPL) in Local Banks*[2]
Indicator for Any Public Record	*Average Rate of RRE NPL in Local Banks*[2]
Indicator for Any Past Delinquencies	*Share of Deposits Held at Big-Four Banks*
If Loan Amount Is a Multiple of 1000	*Zip code Dummy Variables*

[66] We experimented with setting max_features as low as 0.333, meaning only one-third of the features are drawn randomly for fitting each tree, and again found that the prediction accuracy is insensitive. Thus, for the reason explained above, we turn off subsampling.

Length of Credit History	
Loan Purpose Indicators	**US National Economic Variables***
Log Annual Income	3-, 6-, 9-, and 12-Month Lagged 3-Month Change in Unemployment Rate
Log Loan Amount	1-, 2-, 3-, and 4-Quarter Lagged Quarterly Growth Rate of FHFA House Price Index
Month of the Year Dummy Variables	PCE Deflator Growth
No Delinquency Information	GDP Deflator Growth
Revolving Balance	Real PCE Growth
Revolving Line Utilization Rate	Real GDP Growth
Average Change in Unemployment Rate over a Loan's Life	Private Industry Employment Cost Index Growth
Average HPI Growth over a Loan's Life	Average Hourly Earnings Growth

Notes: These variables are measured as each zip code's deviation from the corresponding US value, which is discussed further in section A2.4. CRE is commercial real estate, and RRE is residential real estate. *Unless otherwise noted, all the aggregate variables are one-quarter lagged growth rate.

A2.2. The Subset of Features Selected in LendingClub's Archived Model

For comparison, we apply a small common subset of features to train all the machine learning (ML) models across all sample periods in an alternative specification. This subset of features is explicitly referenced in early vintages of LendingClub's model for assigning credit grades. Appendix IV reproduces one archived example of such models. The eight covariates are listed below:

# Inquiries in the Past 6 Months	FICO Score
# of Currently Open Credit Accounts	DTI
Revolving Line Utilization	(Log of) Requested Loan Amount
Length of Credit History	Total # of Open Credit Accounts

A2.3. The Subset of Features to Approximate Thin Credit Files

As a rough exploration of how the ML methods would perform on applications with extremely limited credit history (sometimes called thin credit files), we train models with a small set of variables that should be available for even applicants with no credit history. Below, we list the individual-specific variables, most of which can be collected in the loan application. These are then supplemented with local and national economic indicators (see the list in the table in Appendix A2.1 above).

If Loan Amount Is a Multiple of 1000
Indicators That the Income or Income Sources Were Verified
Inquiries in the Past 6 Months
(Log of) Requested Loan Amount
(Log of) Annual Income
Loan Purpose
Employment Information
Average Change in Unemployment over Loan's Life
Average Housing Price Index Growth over Loan's Life

A2.4. Transformation of Covariates for ML Models to Minimize the CART Bias

Finite-Sample Drift of the Covariates

As noted above, we include a variety of macroeconomic indicators at the zip code level as predictors. Within our sample period, most of which falls within a cyclical expansion, there is a distinct drift in a few of these variables. The unemployment rate, in particular, declined steadily after late 2010. To obtain a more stable distribution of the unemployment rate (as well as of the other variables measured in levels or growth rates such as the HPI) across zip codes, we decompose the zip-code-specific value of these variables into the national value and a zip code's deviation from the national average, since the cross-section dispersion has a largely stable distribution.

For covariates that enter in logs, such as real balances of credit card loans, student loans, or other non-mortgage debt, we measure the cross-zip-code dispersion as the log difference (equivalent to percentage difference) from the national value. Lastly, for a stationary measure of each zip code's relative population size, we use the local prime-age population as a share of total national prime-age population.

Discretization of the Covariates

It is well known (see, for example, Breiman et al. 1984) that CART models, which use Gini index gain or impurity reduction as the partition criterion, suffer from an intrinsic bias: Variables with many values or more missing observations are more likely to be selected for splitting. This means that continuous variables, such as the zip code level unemployment rate, would tend be chosen as more important even if they possess no more explanatory power than binary or categorical covariates. To minimize the impact of this bias on our feature importance analysis, we discretize these variables to restrict the number of distinct values per variable used to train the ML models. This treatment turns out to have small quantitative effects but no qualitative impact on the relative importance of predictors. We further verify that the predicted values are minimally affected by the discretization, due to the structure of tree models.

We apply the following algorithm. First, we choose a target of about 40 for the number of unique values per variable. This is comparable to the number of unique values for FICO score bins available in the data; the FICO score is generally found to be the most important individual-specific indicator of credit risk. We then separate the value range of each variable into three bins: the bottom and the top 5 percent, and all the values in between. We then compute the standard deviation (SD) separately within each of these ranges. The goal is to arrive at 10 unique values in the bottom 5 percent, 20 unique values in the middle range, and 10 values in the top 5 percent. For each range, we round each raw value to the nearest multiple of the SD. For example, if a specific range of a variable has a standard deviation of 0.6, then any original value from 0.9 (inclusive) to 1.8 (exclusive) will be discretized to 1.2. We then count the number of resulting unique values. If this step produces a number of unique values that is near the target value (that is, about 10 for the two tails of the distribution and 20 for the middle range), the resulting discretized version of this variable will then be used in fitting the random forest and the boosted tree models. However, if the number of unique values falls short, we shrink the SD by half and repeat the rounding process. We continue to iterate the shrinkage factor to one-third or one-fourth, etc., until we reach the minimum number of unique values or a little more. At the end of this process, the discretized version of each variable preserves the basic shape of the distribution of the original variable with only 40 to 45 unique values while giving the tail values more chance in the tree-splitting process. In contrast, for ratio variables such as the share of the population in a certain age range, we do not treat the tails of the distribution separately from the center, but instead just search for 40 unique observations across the entire distribution.

By the same logic, we need not discretize the national values of each economic indicator (as defined for the model estimation, which can involve taking the logarithm, a number of lags, or log difference), because each variable can take on at most one unique value per month, so that there are 27 to 50 unique values over the sample years.

Borrower-Specific Variables

Borrower-specific variables, such as the DTI, the number of inquiries within the last six months, are manually discretized into categorical values depending on each variable's range of values that are suggested by past studies. The number of unique values per variable ranges from two, which corresponds to dummy variables (such as for whether there exists public records), to 38, the number of FICO score bins.

Appendix III. LendingClub Data and Economic Variables by Zip Code

A3.1 LendingClub Data

The primary data sources for this study are the application and loan-level data of LendingClub (LC), one of the major online lenders of unsecured installment consumer loans. The first month of positive application and loan volume is June 2007, and LC originated a little more than 700 loans in 2007. This is likely too small a sample to afford reliable inference. Moreover, the lending technology was likely rather immature, adding to the volatility of ex post outcome. We therefore begin the analysis at the start of 2008, trying to strike a balance between using as much data as possible and ensuring data reliability. There is no definitive criteria for choosing a cutoff time, however. First, data underwent two relatively major changes in 2008: (1) LC stopped approving applicants with FICO scores below 660 (beginning in November 2018), and (2) it had to halt the sale of member notes after filing with the SEC to register these notes as securities (in April 2008). Thus, LC's origination volume fell as much as 85 percent, and the company had to use internal funds on more than 50 percent of loan origination in the second quarter of

2008:.[67] It took a few month for LC to recover its scale of operation after relaunching its platform on October 13, 2008, following approval from the SEC.

A3.2 Local Macroeconomic Indicators

The raw data by geography are provided at the county level: monthly unemployment rate by the US Bureau of Labor Statistics, quarterly house price index of CoreLogic, annual income per capita and poverty rate by the Bureau of Economic Analysis, annual population by age group and multiple-year average education attainment by the US Census Bureau, and fraction of population with various types of debt outstanding by age group from the Survey of Consumer Finance. For every variable, the last available data point by the month when the online loans were originated is used in the regression analysis.

A3.3 Local Consumer Credit Data

Anonymized credit data of individual consumers are provided by the Federal Reserve Bank of New York's Consumer Credit Panel (CCP), which is essentially a 5 percent random sample of the Equifax database of adults with a credit report and social security number and the other adult members of their household.[68] It reports the data from the end of each quarter. Data from 2008 to 2016 are used in this research. The utilization rate of revolving credit is calculated as the ratio of total outstanding balance over total credit limit on all the revolving accounts. Percentiles 1, 5, 25, 50, 75, 95, and 99 are then calculated for each three-digit zip code area each quarter in order to capture a more detailed picture of the local distribution of credit indicators. Statistics particular to credit card debt include only balances on bankcards, not on department store or retail credit cards. This is because in the CCP data, these other types of cards are consolidated

[67] Form S-1, Registration Statement under the Securities Act of 1933 filed on June 20, 2008. https://www.sec.gov/Archives/edgar/data/1409970/000089161808000318/f41480orsv1.htm
And LC 10-K for the fiscal year ending March 31, 2012.

[68] See Lee and van der Klaauw (2010) for more detailed information.

with installment loans from retailers and do not represent only revolving lines of credit from non-bank sources.

A3.4 Local Banking Market Indicators

Measures of bank credit constraints and market power are calculated using the FDIC's Summary of Deposits data, which contains the number of branches and the balance of deposits in each branch owned by every bank on June 30 of each year. Measurements relating to total deposits exclude those made in savings associations, because these institutions have business models that are rather different from those of commercial and savings banks.

The Big-four share is calculated as the share of deposits within each three-digit zip code area held in branches owned by the four largest banks (Bank of America, Citigroup, JPMorgan Chase and Wells Fargo):

$$s_{zt}^{\text{Big-4}} = \sum\nolimits_{j\in\text{Big-4}} D_{jzt} \Big/ \sum\nolimits_{i=1}^{N_z} D_{izt} \text{ ,}$$

where D_{jzt} denotes the balance of deposits held in branches owned by bank *j* in zip code area *z* and at time *t*.

The Herfindahl–Hirschman Index of market concentration for each three-digit zip code area is defined as the sum of each bank's share in the area's total deposits squared:

$$HHI_{zt} = \sum\nolimits_{i=1}^{N_z} s_{izt}^2 \text{ , where } s_{izt} = D_{izt} \Big/ \sum\nolimits_{i=1}^{N_z} D_{izt} \text{ .}$$

Data on bank assets, nonperforming loans, and capital are from the Call Reports that all FDIC-insured banks file quarterly. Each indicator *x* for a three-digit zip code area *z* is computed as the weighted average of *x* among banks with branches in *z*, with each bank's deposit share (s_{izt} defined above) as the weight:

$$x_{zt} = \sum\nolimits_{i=1}^{N_z} s_{izt} x_{izt} \text{ .}$$

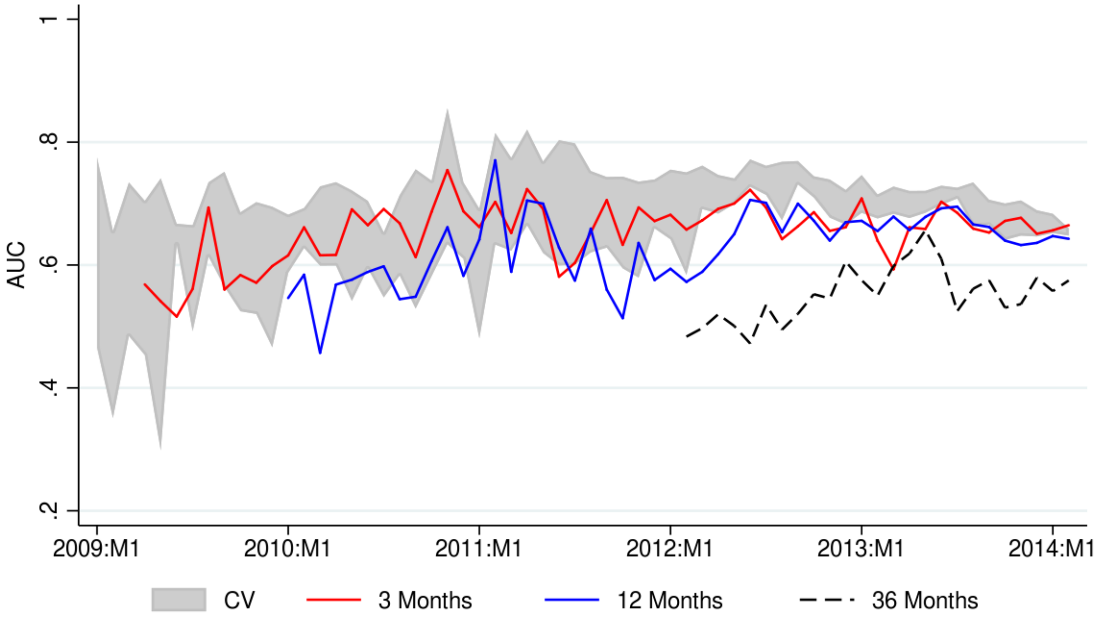
The Call Reports variables used to construct local banking indicators are listed in the following table:

Bank Balance-sheet Variables		Call Reports Mnemonics
Nonperforming Commercial Real Estate Loans	Construction NPL	rcon2759, rcon2769, rcon3492
	Nonfarm NPL	rcon3502, rcon3503, rcon3504
	Multifamily NPL	rcon3499, rcon3500, rcon3501
Residential Real-Estate NPL		rcon5399, rcon5400, rcon5402, rcon5403
Tier-one Capital		rcfd8274
Total Assets		rcfd2170

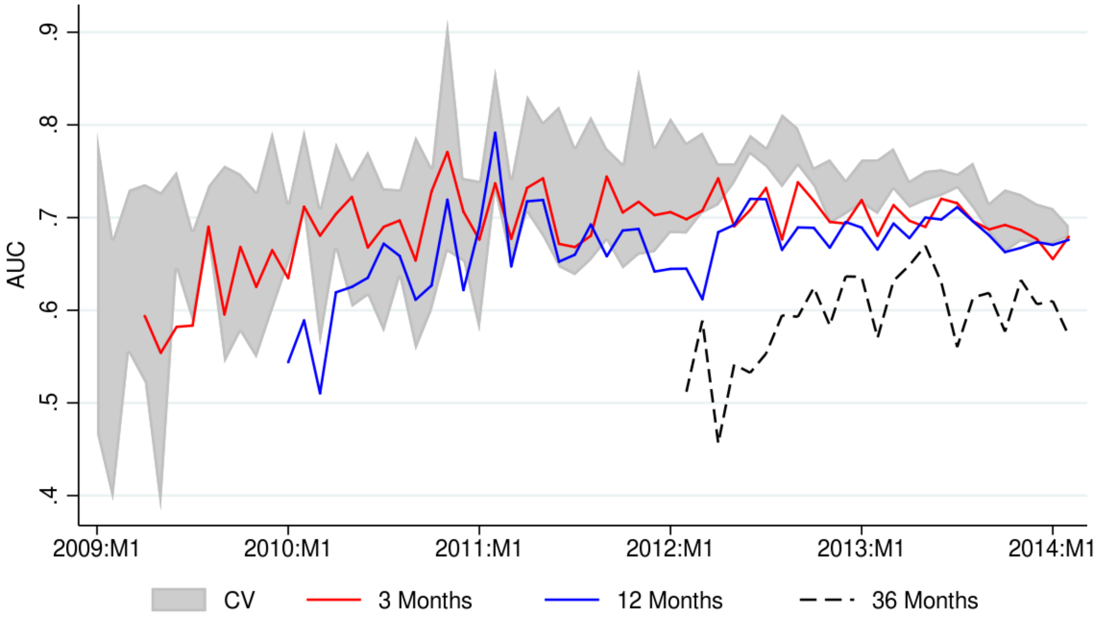
Appendix IV. Additional Empirical Results

Figure A1. AUC of the Same-loan-cohort CV Samples and Future-cohort Test Samples

Panel a. Random Forest Models

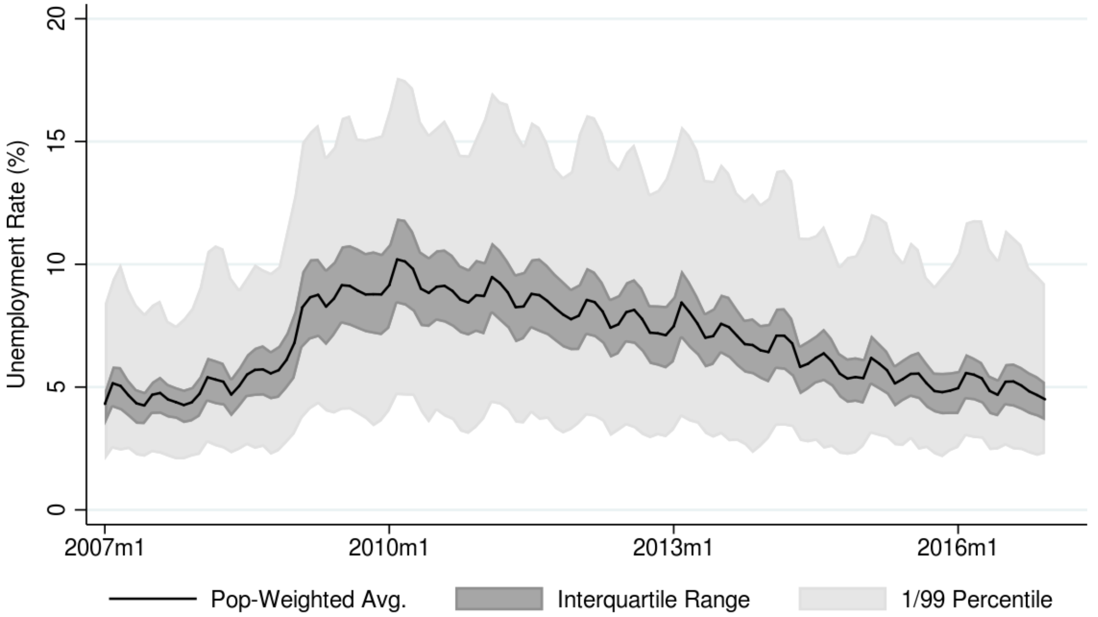


Panel b. Boosted Tree Models



Notes: Panels a and b depict the range of the AUC of the five CV samples, along with the AUC using models trained on the 3-, 12- and 36-months-earlier loan cohorts using random forests and boosted trees, respectively. The time axis indexes the month in which the data are used to test the models.

Figure A2. Unemployment Rates across 3-Digit Zip Codes Locales


Unemployment Rate (%)
2007m1
2010m1
2013m1
2016m1
Pop-Weighted Avg.
Interquartile Range
1/99 Percentile