e-ISSN: 2686-2573
DOI: 10.21512/emacsjournal.v8i2.16455

Walk-Forward Evaluation of ROI-Based Decision Thresholds for Random Forest Loan Default Prediction

Henry Lucky[1*], Anderies[2]
[1,2]Computer Science Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
henry.lucky@binus.ac.id; anderies@binus.ac.id

*Correspondence: henry.lucky@binus.ac.id

Abstract – *Default prediction models for peer-to-peer lending are often tested on random data splits and judged by accuracy, which can hide temporal leakage and says little about investor returns. This study evaluates a random forest on 1,020,768 resolved 36-month Lending Club loans issued from 2007 to 2018, using 30 walk-forward windows with a 36-month gap between validation and test periods and only information available when a loan is funded. A decision threshold chosen by validation return on investment (ROI) was compared with the default threshold of 0.5, lending to every loan a grade-based rule and logistic regression, with full and downsampled training data, five random seeds, and paired tests across windows. The random forest reached a mean AUC of 63.7–64.6%, close to logistic regression (64.2–65.0%). Overall windows, the ROI-based threshold raised mean window ROI from 4.04% (lend-all) to 4.97% (difference 0.92 percentage points, 95% CI −0.29 to 2.23) while lending 51% of the capital; no difference remained significant after Holm adjustment. The share of test loans resolved by the end of the data fell from over 99% in windows 1–19 to 7% in window 30, so the negative returns of late windows reflect censoring. In the 19 matured windows, lend-all earned 9.08% on average and the random forest did not improve on it: ROI-based thresholds lowered mean ROI by 0.92–1.15 percentage points and profit by 40–54%. Return-based evaluations of default models therefore need time-ordered tests that account for unresolved loans.*

Keywords: *Lending Club; loan default prediction; random forest; return on investment; right-censoring; walk-forward validation*

I. INTRODUCTION

Peer-to-peer (P2P) lending platforms like Lending Club enable individual investors to invest directly in consumer loans. This means the individual investor, not the bank, takes on the risk of losses if a borrower defaults on their payments (Serrano-Cinca et al., 2015). To manage this risk, investors use default prediction models to assess potential investments in the available loan listings. As a result, the Lending Club 2007-2018 dataset has emerged as the most widely used baseline for testing these types of models (Noriega et al., 2023; Nguyen et al., 2024).

A frequently used classifier for the problem of identifying whether or not an individual will default on their loan is tree ensembles. Random forests and gradient boosting are generally found to be the most effective in large-scale evaluations of credit risk scoring data (Lessmann et al., 2015; Gunnarsson et al., 2021). However, when a logistic regression model is properly specified using non-linear variables, the results are comparable to those obtained by these more advanced machine learning methods (Dumitrescu et al., 2022). Random forests have been applied to Lending Club data as far back as when the platform first began to gain traction in the academic community (Malekipirbazari & Aksakalli, 2015) and are still commonly employed as a baseline comparison for boosting models (Liu et al., 2022; Ko et al., 2022; Li et al., 2025). The performance reported varies greatly; some researchers have reported macro-F1 scores greater than 90% (Núñez Mora et al., 2023), while others report accuracy levels of approximately 87% with significantly reduced capability to distinguish between those who defaulted and those who did not (Li et al., 2025).

This difference is due to how models are evaluated. Most studies utilizing Lending Club data select random samples of training and test loans from the complete 2007-2018 sample. Therefore, loans for the same month will be included in the training and testing sets, and information about the loan obtained after the loan has been issued (e.g., payments received or last payment date) may provide insight into the outcome of the loan. Kapoor and Narayanan (2023) identified two problems related to the use of random samples — temporal leakage and the use of post-outcome variables — as being among the main factors contributing to the

Copyright © 2026

failure of machine learning results to replicate. Credit data also changes over time because of changes in underwriting standards, types of borrowers applying for credit, and economic conditions (Nikolaidis & Doumpos, 2022; Bayram et al., 2022). Randomly selecting samples from a dataset that is not exchangeable over time will result in biased estimates of future performance (Hornung et al., 2026). Additionally, there is another problem specific to loan data; the outcome of a loan is determined only after the loan either matures or defaults. Therefore, a snapshot of the platform will include many active loans, and those studies that include only closed loans will evaluate relatively young cohorts of borrowers on a sample that over-represents early default and early repayment. Such biased samples used for evaluation purposes can influence which model performs better (Kozodoi et al., 2025).

There are two other design considerations that may affect investors. The first is class imbalance: most loans are paid off, so when classifiers are trained using the raw data, they tend to assume each loan will be paid off. Random undersampling and SMOTE are common ways to fix this problem (Zhao et al., 2024; Liu, 2025), but simply changing the decision threshold without resampling is another option that can provide similar performance (Yang et al., 2022; Chen et al., 2024). A second design consideration is the objective of the model. Many metrics used to assess the performance of a machine learning model, such as accuracy, F1-score, and area under the ROC curve, cannot be directly translated into a monetary return for an investor. Instead, profit-based classification measures (Verbraken et al., 2014), cost-sensitive learning optimized for profitability (Petrides et al., 2022; Vanderschueren et al., 2022), and direct return or profit scoring (Lyócsa et al., 2022; Rong et al., 2023) assess models according to how much money they make or save through their decision-making process. For instance, Lyócsa et al. (2022) demonstrated that profit scoring generated 15.5 percent greater returns than default scoring on Lending Club loans. Alonso Robisco and Carbó Martínez (2022) pointed out that when assessing different approaches to evaluating performance, we must also consider the level of risk associated with each approach.

These four lines of research are often conducted independently. For example, research into the validity of time-ordered testing rarely looks at financial performance. Similarly, research into the effectiveness of algorithmic models for maximizing profits rarely employs strictly time-ordered testing. We are unaware of any of the Lending Club studies referenced in this work evaluating whether unresolved loans negatively impact the returns generated from recent test periods. Therefore, we believe that the significance of our research lies in its experimental design and the associated results rather than the development of a novel machine-learning algorithm. This study will focus on: (1) evaluating a random forest default classifier on 30 contiguous out-of-time test windows with a 36-month lag between the validation and test periods to ensure all training labels were available prior to the deployment of the model; (2) determining the optimal decision threshold that yields the highest return on investment (ROI) during a validation period and comparing that decision threshold to several alternatives (a default threshold of 0.5, a "lend-all" strategy, a grade-based rule, and a logistic regression model), utilizing paired comparisons among windows; (3) comparing the performance of models trained on the complete imbalanced dataset versus random undersampling at two different ratios, evaluating those comparisons across multiple random seeds, training-window schemes, and hyperparameter settings; and (4) assessing the number of unresolved loans within each test window as of the last date represented in the dataset, then repeating the financial comparisons solely on those test windows whose loans have reached maturity.

The study focuses on three research questions: RQ1: How well can a random forest differentiate between defaulted and non-defaulted loans using walk-forward evaluation; and how does this performance compare to that of logistic regression? RQ2: Will choosing a threshold based on the validation ROI lead to higher test ROI and risk adjusted return than using a standard threshold, lending to all loans, or a grade-based rule? RQ3: Are the above findings robust when considering the effect of dealing with class imbalances, the random seed, the length of the training window, and including loans that have not yet matured.

II. METHODS

Figure 1 summarizes the research steps. The first step was to collect a dataset for this research: the Lending Club Loan dataset. Then, a pre-processing of the dataset was performed. The pre-processed dataset is then split, trained, validated, and tested using a walk-forward method based on the time window used. The results would then be evaluated using the most commonly used metrics in this domain, such as accuracy, AUC, precision, sensitivity, specificity, G-mean, and real financial metrics such as ROI and monetary profit. All experiments were run in Python with cuDF and cuML (RAPIDS 26.02) on two NVIDIA T4 GPUs, with the windows distributed over the GPUs by dask-cuda, and the code is available from the authors on request.

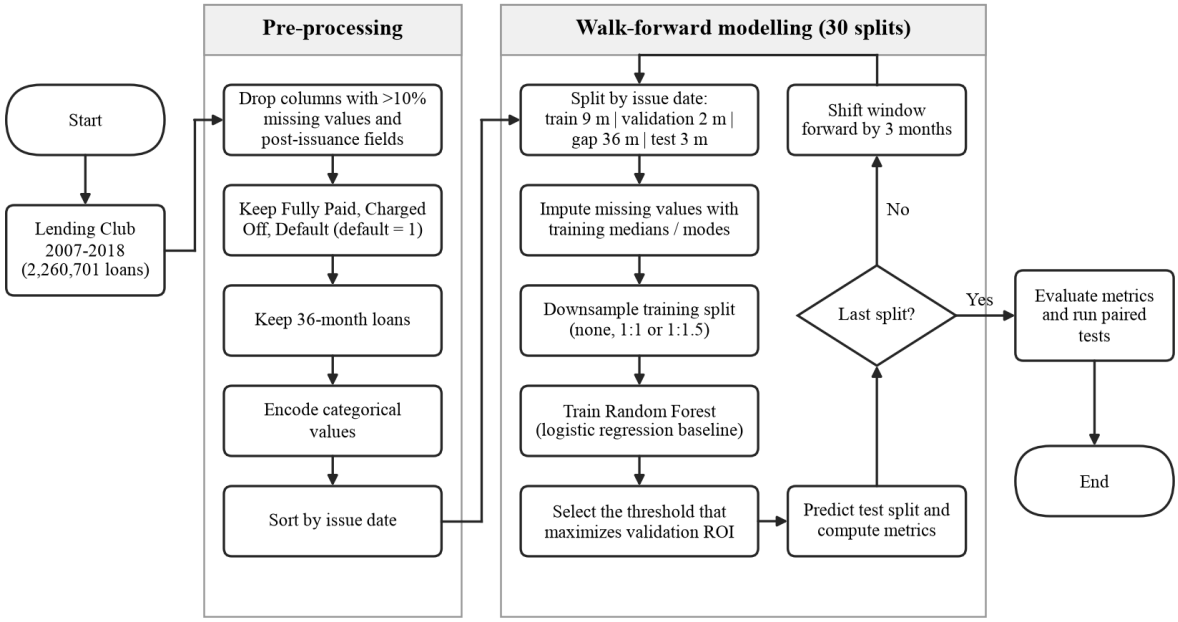


Figure 1. Research methodology

A. Lending Club Loan 2007–2018 Dataset

The dataset used in this research is the Lending Club Loan dataset, which spans from June 2007 to December 2018. Specifically, the accepted loan set is the main task in this research: predicting loan default. The dataset consists of 2,260,701 loans with 151 columns, including borrowers' and loan profiles. The loan status acts as the target with nine values: "Fully Paid", "Current", "Charged Off", "Late (31–120 days)", "In Grace Period", "Late (16–30 days)", "Does not meet credit policy. Status: Fully Paid", "Does not meet credit policy. Status: Charged Off", and "Default".

B. Pre-processing

The dataset needs to be pre-processed first before being fed into a machine learning model. Pre-processing starts with feature selection and then handling categorical values, as detailed below. After that, the dataset is filtered only to have Fully Paid, Charged Off, and Default as the target, and the other loan statuses are omitted. Charged Off and Default are set as the positive value (defaulted loan), and Fully Paid is set as the negative value for this task. The final data is then sorted by loan issue date to retain temporal information before training. Missing values are handled later, inside each walk-forward window (Section II.C.2).

1. Feature Selection

For feature selection, features with more than 10% missing values are omitted, as they are considered not helpful. This filter does not use the target, but it is computed once on the whole dataset rather than inside each window. Then, the features that have information after the loans are issued (e.g., total_pymnt, recoveries, last_pymnt_d, last_credit_pull_d, last_pymnt_amnt, last_fico_range_high, etc.) are also omitted from the model inputs due to the possibility of data leakage. The features that are deemed as not helpful, like id, url, desc, emp_title, zip_code, and addr_state, are also omitted.

In total, 64 variables are used as model inputs (85 columns after encoding), which are shown in Table 1. A few columns stay in the data only for bookkeeping and are never passed to the model (last row of Table 1): loan_status is the label, issue_d is only used to sort and split the data, funded_amnt_inv and total_pymnt_inv are only used to compute ROI after prediction, and funded_amnt and last_pymnt_d are not used. Two inputs, pymnt_plan and policy_code, are constant in the final sample. The 32 extended bureau attributes were only reported by Lending Club from 2012 onward, so they are completely missing in the training periods of windows 1 to 17, partly missing in window 18, and available from window 19 onward.

Table 1. Variables used as model inputs, grouped by type

Group (number of variables)	Variables
Loan and listing (11)	loan_amnt, int_rate, installment, grade, sub_grade, purpose, initial_list_status, application_type, disbursement_method, pymnt_plan[a], policy_code[a]
Borrower profile (6)	emp_length, home_ownership, annual_inc, verification_status, dti, earliest_cr_line (as year and month)
Credit history at application (15)	delinq_2yrs, fico_range_low, fico_range_high, inq_last_6mths, open_acc, pub_rec, revol_bal, revol_util, total_acc, collections_12_mths_ex_med, acc_now_delinq, chargeoff_within_12_mths, delinq_amnt, pub_rec_bankruptcies, tax_liens
Extended bureau attributes, reported from 2012 onward (32)[b]	tot_coll_amt, tot_cur_bal, total_rev_hi_lim, acc_open_past_24mths, avg_cur_bal, bc_open_to_buy, bc_util, mo_sin_old_il_acct, mo_sin_old_rev_tl_op, mo_sin_rcnt_rev_tl_op, mo_sin_rcnt_tl, mort_acc, num_accts_ever_120_pd, num_actv_bc_tl, num_actv_rev_tl, num_bc_sats, num_bc_tl, num_il_tl, num_op_rev_tl, num_rev_accts, num_rev_tl_bal_gt_0, num_sats, num_tl_120dpd_2m, num_tl_30dpd, num_tl_90g_dpd_24m, num_tl_op_past_12m, pct_tl_nvr_dlq, percent_bc_gt_75, tot_hi_cred_lim, total_bal_ex_mort, total_bc_limit, total_il_high_credit_limit
Kept for bookkeeping only, never passed to fit()	loan_status (label), issue_d (ordering and window assignment), funded_amnt_inv and total_pymnt_inv (ROI after prediction), funded_amnt and last_pymnt_d (unused), term (filter; constant after filtering)

[a] Constant in the modelling sample. [b] Entirely missing in the training periods of windows 1–17.

2. Handling Categorical Values

There are two types of categorical values present in the dataset: nominal and ordinal. To handle nominal data such as 'home_ownership', 'verification_status', and 'purpose', one-hot encoding is used. Ordinal data are changed into a single numerical value that contains the order information. These are fixed mappings of the category labels, so they do not use any outcome information. Employment length is first mapped to years (0 for less than one year, 10 for ten or more years, −1 when unknown) and then coded in the lexical order of these labels, so the code for ten years falls between those for one and two years. The earliest credit line date is split into year and month.

3. Target Mapping and Data Filtering

As the dataset now only has three labels, we map Fully Paid as a negative label, then Charged Off and Default as positive label. After that, we filter the dataset based on term length. There are two term lengths: 36 and 60 months. We focus on the 36-month term because waiting for the 60-month loans to conclude may be too long, and we cannot model the time shift properly. The dataset is then sorted based on the earlier issue date. Throughout this process, the remaining samples in the dataset are 857,491 negative samples and 163,277 positive samples, which is imbalanced with a ratio negative:positive of 84:16. Because loans that were still running in December 2018 are dropped, the most recent issue months are represented only by loans that ended early; Section III.D measures this effect.

C. Walk-Forward Modelling

Walk-Forward modelling is used to obtain realistic model outputs, since random splits of data that change over time give overly optimistic estimates of future performance (Hornung et al., 2026).

1. Data Splitting

For data splitting, we use nine months' worth of data to train, another two months for validation, and three months for testing. The sliding window to the next time window is three months (a quarter), making the test loans completely distinct throughout each time window. Regarding the applicability of the research, it is noted that in real life, the loan should mature before the loan_status is available to train (Fully Paid / Charged Off / Default). Ordering the loan data by last payment date is an option, but it creates a survivorship bias, where the model learns from loans that are paid earlier; the last payment date is also leaked information, so it should not be used even to order the dataset (Kapoor & Narayanan, 2023). Therefore, we provide a 36-month buffer, aligned with the loan terms, to allow the loans to mature without bias, which is achieved by introducing a 36-month gap between the validation and test splits (Figure 2). The gap protects the training and validation labels; it does not make the test labels complete, which is examined in Section III.D.

Table 2 provides details on the train, validation, and test months of each split, along with the number of samples in each split and the share of test loans that had resolved by December 2018. This gives 30 splits, with test periods from May 2011 to October 2018. The early splits are small, with 596 to 2,745 training loans in splits 1 to 7, so an expanding training window that keeps all earlier loans was also evaluated as a sensitivity check.

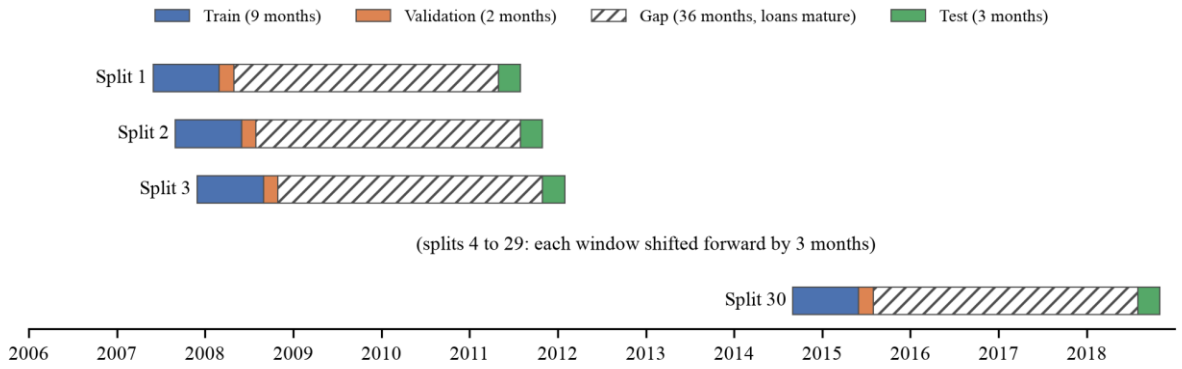


Figure 2. Walk-forward windows. Each window trains on 9 months of loans, selects the threshold on the next 2 months, skips 36 months so that training and validation loans can mature, and tests on the following 3 months

Table 2. Walk-forward windows: issue months, sample sizes and share of test loans resolved by December 2018

Split	Training months	Validation months	Test months	Train n	Validation n	Test n	Resolved (%)
1	Jun 07–Feb 08	Mar 08–Apr 08	May 11–Jul 11	596	391	2,547	100.0
2	Sep 07–May 08	Jun 08–Jul 08	Aug 11–Oct 11	994	149	4,074	100.0
3	Dec 07–Aug 08	Sep 08–Oct 08	Nov 11–Jan 12	1,112	128	5,017	100.0
4	Mar 08–Nov 08	Dec 08–Jan 09	Feb 12–Apr 12	994	462	7,120	100.0
5	Jun 08–Feb 09	Mar 09–Apr 09	May 12–Jul 12	1,530	609	9,570	100.0
6	Sep 08–May 09	Jun 09–Jul 09	Aug 12–Oct 12	1,919	730	14,743	100.0
7	Dec 08–Aug 09	Sep 09–Oct 09	Nov 12–Jan 13	2,745	994	15,516	100.0
8	Mar 09–Nov 09	Dec 09–Jan 10	Feb 13–Apr 13	3,343	1,187	27,460	100.0
9	Jun 09–Feb 10	Mar 10–Apr 10	May 13–Jul 13	5,285	1,580	17,105	100.0
10	Sep 09–May 10	Jun 10–Jul 10	Aug 13–Oct 13	5,727	1,347	28,333	100.0
11	Dec 09–Aug 10	Sep 10–Oct 10	Nov 13–Jan 14	6,147	1,343	33,482	100.0
12	Mar 10–Nov 10	Dec 10–Jan 11	Feb 14–Apr 14	5,648	1,814	48,952	100.0
13	Jun 10–Feb 11	Mar 11–Apr 11	May 14–Jul 14	7,550	1,904	31,449	100.0
14	Sep 10–May 11	Jun 11–Jul 11	Aug 14–Oct 14	7,438	2,547	47,227	100.0
15	Dec 10–Aug 11	Sep 11–Oct 11	Nov 14–Jan 15	9,228	2,779	47,078	100.0
16	Mar 11–Nov 11	Dec 11–Jan 12	Feb 15–Apr 15	10,069	3,473	78,095	100.0
17	Jun 11–Feb 12	Mar 12–Apr 12	May 15–Jul 15	13,721	5,037	49,928	100.0
18	Sep 11–May 12	Jun 12–Jul 12	Aug 15–Oct 15	17,657	6,829	75,889	100.0
19	Dec 11–Aug 12	Sep 12–Oct 12	Nov 15–Jan 16	24,624	10,282	78,015	99.7
20	Mar 12–Nov 12	Dec 12–Jan 13	Feb 16–Apr 16	34,525	10,341	86,705	85.4
21	Jun 12–Feb 13	Mar 13–Apr 13	May 16–Jul 16	49,539	15,009	47,438	66.1
22	Sep 12–May 13	Jun 13–Jul 13	Aug 16–Oct 16	53,258	17,105	45,406	62.0
23	Dec 12–Aug 13	Sep 13–Oct 13	Nov 16–Jan 17	64,087	19,152	43,200	55.1
24	Mar 13–Nov 13	Dec 13–Jan 14	Feb 17–Apr 17	71,345	22,584	33,725	48.1
25	Jun 13–Feb 14	Mar 14–Apr 14	May 17–Jul 17	101,499	26,373	35,415	42.2
26	Sep 13–May 14	Jun 14–Jul 14	Aug 17–Oct 17	101,586	31,449	30,438	35.2
27	Dec 13–Aug 14	Sep 14–Oct 14	Nov 17–Jan 18	116,268	33,944	22,362	27.5
28	Mar 14–Nov 14	Dec 14–Jan 15	Feb 18–Apr 18	121,622	30,505	20,021	17.8
29	Jun 14–Feb 15	Mar 15–Apr 15	May 18–Jul 18	158,691	45,158	6,845	11.5
30	Sep 14–May 15	Jun 15–Jul 15	Aug 18–Oct 18	159,117	49,928	5,965	6.6

Resolved: share of all 36-month loans issued in the test months whose status was Fully Paid, Charged Off or Default in December 2018. Windows 1–19 are at least 99% resolved and are treated as matured in Section III.D.

2. Data Sampling and Handling Missing Values

Due to the dataset's imbalance, downsampling is performed to achieve a 1:1 ratio of positive to negative labels (Zhao et al., 2024; Liu, 2025). However, to keep the distribution so the model is aware that there are far more negative than positive samples, we also experimented with a 1:1.5 ratio of positive:negative. Together with the full training data, this gives three training sets per window, called Full, Downsampling and Downsampling 1.5. Only the training split is resampled; the validation and test splits keep their original class mix. Median and mode imputations are used to handle numerical and categorical missing values. We ensure we get the median and mode from the training split and apply them to the validation and test splits to avoid data leakage.

3. Training, Tuning, and Testing

For the training phase, the model used is a Random Forest (Breiman, 2001) with hyperparameters n_estimators = 300, max_depth = 64, max_features = √p, bootstrap sampling, and random_state = 42, in the cuML implementation, which searches for splits over 128 quantile bins per feature. A depth limit of 64 is rarely reached, so the trees are grown almost fully, as is usual for random forests; Section III.E reports results with 100 trees and with a depth of 16. The random seed controls both the bootstrap samples and the downsampling, and seeds 0, 1, 2 and 3 were run in addition to 42 to measure seed variability. Two baselines were trained in the same windows: logistic regression (L2 penalty, C = 1) on the same three training sets, with inputs standardized by training-split means and standard deviations and entirely missing inputs set to zero, and a grade-based rule that lends to grades A up to a cutoff grade chosen by validation ROI. A lend-all baseline funds every loan in the test split.

To balance the imbalanced dataset, we also do threshold tuning. However, the tuning is done based on real financial metrics, ROI, not some superficial metrics such as F1, precision, and recall. A loan is funded when its predicted default probability is at or below the threshold τ, and the non-optimized models use the default threshold of 0.5. We use 100 evenly spaced numbers over the interval of 0.01 to 0.99 for the threshold. The best threshold on the validation split is then used for the test split:

$$\tau_w^* = \arg\max_{\tau \in T} \mathrm{ROI}_w^{\mathrm{val}}(\tau) \quad (1)$$

where T is the grid of 100 thresholds and $\mathrm{ROI}^{\mathrm{val}}{}_{\mathrm{w}}(\tau)$ is the ROI of Eq. (3) computed over the validation loans with predicted default probability at or below τ. When several thresholds give the same ROI, the smallest is taken. Validation periods are short, with 128 to 994 loans in splits 1 to 7, so the selected threshold can be noisy in early splits; Figure 4 shows the thresholds that were selected.

D. Evaluation Metrics

The results are evaluated using classic metrics such as accuracy, AUC, precision, sensitivity (recall), specificity, and G-mean, and using real financial metrics such as ROI, profit, and Sharpe ratio. Default is the positive class. Sensitivity (true positive rate, TPR) is TP/(TP+FN+1), specificity (true negative rate, TNR) is TN/(TN+FP+1) and precision is TP/(TP+FP+1), where TP, FP, TN and FN are the counts of the confusion matrix; the constant of one keeps the ratios defined in windows where a class is never predicted. G-mean is given by Eq. (2). Metrics are reported as the mean (standard deviation) over the 30 windows and as pooled values over all test loans.

$$\mathrm{G-mean} = \sqrt{\mathrm{TPR} \times \mathrm{TNR}} \quad (2)$$

The financial metrics follow the cash flows of investors. For window w, let A_{w} be the set of funded test loans, F_{i} the amount funded by investors in loan i (funded_amnt_inv), and P_{i} the total payments investors received from it (total_pymnt_inv, which includes principal, interest, late fees and post-default recoveries). The ROI of the window and the total ROI over all windows are

$$\mathrm{ROI}_w = \frac{\sum_{i \in A_w} P_i}{\sum_{i \in A_w} F_i} - 1 \quad (3)$$

$$\mathrm{ROI}^{\mathrm{total}} = \frac{\sum_w \sum_{i \in A_w} P_i}{\sum_w \sum_{i \in A_w} F_i} - 1 \quad (4)$$

This is the cumulative, undiscounted return over the life of each loan. It does not deduct the platform's service fee or collection fees, and it does not account for the shorter holding period of loans that were repaid early. When a strategy funds no loan in a window, its ROI for that window is set to zero. Profit is the sum of P_{i} minus the sum of F_{i} over funded loans. Because selective strategies invest less money than lend-all, the share of the lend-all amount that is actually invested (capital deployed) is reported, together with an idle-capital-adjusted ROI that assumes the money not lent in window w is held in a 3-year US Treasury note:

$$\mathrm{ROI}_w^{\mathrm{adj}} = \frac{\sum_{i \in A_w} P_i + \left(L_w^{\mathrm{all}} - L_w\right)(1 + r_w)}{L_w^{\mathrm{all}}} - 1 \quad (5)$$

where L_{w} is the amount lent by the strategy, $L^{\mathrm{all}}{}_{\mathrm{w}}$ the amount lent by the lend-all baseline, and r_{w} the risk-free return over the same horizon,

$$r_w = \left(1 + y_w\right)^3 - 1 \quad (6)$$

with y_{w} the 3-year Treasury constant-maturity yield (FRED series GS3) in the month the test window starts. A 3-year rate is used because the loans have a 36-month term, so the benchmark covers the same horizon as the loan returns. The Sharpe ratio (Sharpe, 1994) is computed over the 30 windows as the mean excess return divided by its sample standard deviation:

$$\mathrm{SR} = \frac{\overline{\mathrm{ROI}_w - r_w}}{s_{(\mathrm{ROI}_w - r_w)}} \quad (7)$$

Differences in window-level ROI between two strategies were tested with the Wilcoxon signed-rank test and the paired t-test over the 30 windows, and a 95% bootstrap confidence interval (10,000 resamples) was computed for the mean difference. Wilcoxon p-values were adjusted with the Holm procedure within each family of comparisons. Finally, the share of each test window's 36-month loans that had resolved by December 2018 was computed from all loan statuses, and the financial comparison was repeated on the matured windows only.

III. RESULTS AND DISCUSSION

All results in this section come from the procedure of Section II with seed 42 unless stated otherwise. τ^* denotes the threshold selected by validation ROI, and differences in ROI are given in percentage points.

A. Classification Performance

Table 3 reports the classification metrics averaged over the 30 windows and Table 4 the metrics pooled over all test loans. The mean AUC of the random forest was 63.69–64.64% for the three training sets and that of logistic regression 64.19–64.95%; pooled over all test loans, the values were 64.36–64.94% and 64.10–65.24%. The two classifiers rank loans about equally well, so the limit seems to lie in the information carried by the inputs more than in the learner. The AUC of the random forest trained on full data rose from 0.55–0.62 in windows 1 to 8, which had fewer than 3,400 training loans, to 0.69–0.70 in windows 27 to 30. Larger training sets and the extended bureau attributes, which enter the training data from window 18, explain part of this rise. The test sets of the late windows, which contain only loans that ended early, may also be easier to rank (Section III.D). Over the 19 matured windows alone, the mean AUC was 61.26% for the random forest and 62.88% for logistic regression.

With full training data and threshold 0.5, the random forest reached an accuracy of 85.01% but a sensitivity of only 0.11%. It approved almost every loan, and its accuracy is close to the 85.0% share of repaid loans in the test windows. Downsampling to 1:1 raised the sensitivity to 64.83% and the G-mean to 59.72% at the cost of specificity (56.04%), and the 1:1.5 ratio fell between the two settings. Logistic regression followed the same pattern. The ROI-based threshold moved the operating point from window to window: the standard deviation of sensitivity across windows was 27 to 37 percentage points, because some selected thresholds approved almost no loans and others approved most of them (Section III.F). The AUC does not depend on the threshold and is therefore identical for both threshold strategies.

Table 3. Classification results averaged over the 30 test windows (%); standard deviation across windows in brackets

Model	Training data	Thres-hold	Accuracy	AUC	Precision	Sensitivity	Specificity	G-mean
RF	Full	0.5	**85.01 (3.80)**	63.69 (4.28)	14.81 (19.65)	0.11 (0.24)	**99.95 (0.09)**	1.85 (2.83)
RF	Full	τ^*	43.84 (25.80)	63.69 (4.28)	19.99 (8.18)	72.17 (27.31)	40.04 (34.20)	38.66 (23.09)
RF	Downsampling	0.5	57.25 (7.22)	64.31 (4.05)	20.99 (6.41)	64.83 (7.83)	56.04 (9.30)	**59.72 (3.28)**
RF	Downsampling	τ^*	45.06 (31.05)	64.31 (4.05)	20.72 (11.21)	64.45 (36.60)	43.75 (42.20)	27.46 (24.89)
RF	Downsampling 1.5	0.5	72.93 (4.00)	64.64 (3.94)	24.35 (8.05)	36.63 (7.99)	79.43 (5.94)	53.42 (4.72)
RF	Downsampling 1.5	τ^*	47.22 (30.44)	64.64 (3.94)	20.87 (10.81)	63.46 (35.10)	46.21 (40.90)	30.85 (25.37)
LR	Full	0.5	84.50 (3.57)	**64.95 (3.70)**	**31.02 (15.03)**	2.19 (2.20)	99.03 (1.31)	12.87 (7.09)
LR	Full	τ^*	34.35 (26.53)	**64.95 (3.70)**	19.30 (9.67)	79.53 (29.13)	27.57 (36.54)	26.55 (22.11)
LR	Downsampling	0.5	58.34 (9.77)	64.19 (4.42)	21.38 (7.19)	62.38 (11.41)	57.87 (12.83)	58.86 (4.89)
LR	Downsampling	τ^*	19.78 (15.79)	64.19 (4.42)	15.85 (5.31)	**94.89 (17.04)**	6.55 (21.22)	9.55 (13.72)
LR	Downsampling 1.5	0.5	70.52 (8.53)	64.53 (4.16)	24.61 (8.54)	41.15 (12.96)	76.02 (12.14)	54.32 (6.21)
LR	Downsampling 1.5	τ^*	23.10 (18.80)	64.53 (4.16)	16.27 (5.35)	92.33 (19.05)	10.67 (25.34)	14.05 (18.29)

Default is the positive class. τ^*: threshold selected by validation ROI. AUC does not depend on the threshold.

Table 4. Classification results pooled over all test loans of the 30 windows (%)

Model	Training data	Thres-hold	Accuracy	AUC	Precision	Sensitivity	Specificity	G-mean
RF	Full	0.5	83.87	64.36	35.39	0.07	99.98	2.60
RF	Full	τ^*	52.99	64.36	19.75	62.56	51.15	56.57
RF	Downsampling	0.5	57.30	64.56	22.20	65.80	55.66	60.52
RF	Downsampling	τ^*	59.75	64.56	18.58	44.26	62.72	52.69
RF	Downsampling 1.5	0.5	71.98	64.94	25.47	38.31	78.46	54.82
RF	Downsampling 1.5	τ^*	61.65	64.94	19.80	45.20	64.82	54.13
LR	Full	0.5	83.53	65.24	31.60	1.86	99.23	13.57
LR	Full	τ^*	37.25	65.24	16.82	73.28	30.33	47.14
LR	Downsampling	0.5	58.79	64.10	22.37	62.99	57.98	60.43
LR	Downsampling	τ^*	24.22	64.10	16.48	90.96	11.40	32.20
LR	Downsampling 1.5	0.5	70.25	64.23	24.70	41.27	75.82	55.94
LR	Downsampling 1.5	τ^*	24.91	64.23	16.55	90.45	12.31	33.37

B. Financial Performance over All Windows

Table 5 gives the financial results over all 30 windows, and Figure 3 shows the test ROI of each window. In this paper, one strategy outperforms another when it earns a higher ROI on the capital it actually lends; capital deployed, profit and idle-capital-adjusted ROI show what this means in

money. Lending to every test loan returned 4.04% per window on average (SD 7.28) and 4.11% in total, a profit of USD 517.6 million on USD 12,590 million lent. The random forest trained on full data with threshold 0.5 lent 100.0% of that capital and gave practically the same result. Downsampling with threshold 0.5 raised the mean ROI to 4.96% (SD 5.24) and the total ROI to 5.20%, while lending 57.1% of the capital.

The ROI-based threshold did not improve the random forest in every setting. It raised the mean ROI from 4.05% to 4.97% for full data and from 4.70% to 4.92% for Downsampling 1.5, but lowered it from 4.96% to 4.68% for Downsampling, and its total ROI was lower than that of threshold 0.5 for both downsampling ratios. Logistic regression with an ROI-based threshold had the highest mean ROI (6.38% for Downsampling), but it lent only 12.2% of the capital and its total ROI was 6.00%. Mean and total ROI differ because the mean gives every window the same weight, whereas the total weights windows by the money lent, and the test sets range from 2,547 to 86,705 loans.

The grade-based rule earned the highest profit, USD 530.5 million with 95.8% of the capital. The random forest and logistic regression trained on full data with threshold 0.5 lent 99.03–99.96% of the capital and earned about the same profit as lend-all; every other strategy earned less, down to USD 92.5 million for logistic regression with Downsampling data and τ^*. When the capital left unlent is assumed to earn the 3-year Treasury return, the ROI-based thresholds of the random forest fall below lend-all (3.20–3.62% against 4.04%), while the forests with threshold 0.5 stay close to it (3.77–4.19%). The Sharpe ratios were low for all strategies, from 0.11 for lend-all to 0.31 for the random forest with Full data and τ^* and 0.47 for logistic regression with Downsampling data and τ^*, because window returns varied widely. These Sharpe ratios are computed over windows whose loans overlap in time; they describe the spread of window returns and are not annualized.

Table 5. Financial results over all 30 test windows

Model	Training data	Thres-hold	Avg ROI (SD)	95% CI	Total ROI	Sharpe	Capital (%)	Lent (USD M)	Profit (USD M)	Idle-adj. ROI	Neg.
Lend-all	–	–	4.04 (7.28)	1.33, 6.76	4.11	0.11	100.0	12,590.0	**517.6**	4.04	10
Grade rule	–	Grade cutoff	4.30 (6.29)	1.95, 6.65	4.40	0.15	95.8	12,060.2	530.5	**4.50**	10
RF	Full	0.5	4.05 (7.28)	1.33, 6.77	4.11	0.11	100.0	12,585.5	517.6	4.04	10
RF	Full	τ^*	4.97 (4.61)	3.25, 6.69	4.97	0.31	51.3	6,462.8	321.4	3.55	6
RF	Downsampling	0.5	4.96 (5.24)	3.00, 6.92	5.20	0.27	57.1	7,186.5	373.4	3.77	7
RF	Downsampling	τ^*	4.68 (4.92)	2.84, 6.52	4.13	0.25	64.6	8,127.4	335.5	3.20	7
RF	Downsampling 1.5	0.5	4.70 (6.13)	2.41, 6.99	4.85	0.21	79.1	9,963.7	483.0	4.19	8
RF	Downsampling 1.5	τ^*	4.92 (5.02)	3.05, 6.80	4.66	0.29	65.9	8,302.1	387.2	3.62	6
LR	Full	0.5	4.08 (7.21)	1.39, 6.77	4.16	0.11	99.0	12,467.6	518.5	4.01	10
LR	Full	τ^*	5.48 (5.03)	3.60, 7.36	3.02	0.37	31.0	3,901.8	117.7	3.14	6
LR	Downsampling	0.5	5.05 (5.42)	3.03, 7.08	4.99	0.28	58.1	7,316.2	364.8	3.63	8
LR	Downsampling	τ^*	**6.38 (5.76)**	4.23, 8.54	6.00	**0.47**	12.2	1,541.0	92.5	3.40	4
LR	Downsampling 1.5	0.5	4.74 (6.13)	2.45, 7.03	4.68	0.21	75.6	9,515.0	445.0	3.85	8
LR	Downsampling 1.5	τ^*	5.72 (6.95)	3.13, 8.32	**6.25**	0.32	13.0	1,631.7	102.0	3.72	3

ROI values in %. Avg ROI: mean window ROI (Eq. 3) with the standard deviation across windows in brackets; 95% CI: bootstrap interval of the mean; Total ROI: Eq. (4); Sharpe: Eq. (7) with the 3-year Treasury return of Eq. (6); Capital: amount lent as a share of the lend-all amount; Lent and Profit in millions of US dollars, amount received = lent + profit; Idle-adj. ROI: mean of Eq. (5); Neg.: windows with negative ROI; τ^*: threshold selected by validation ROI (Eq. 1).

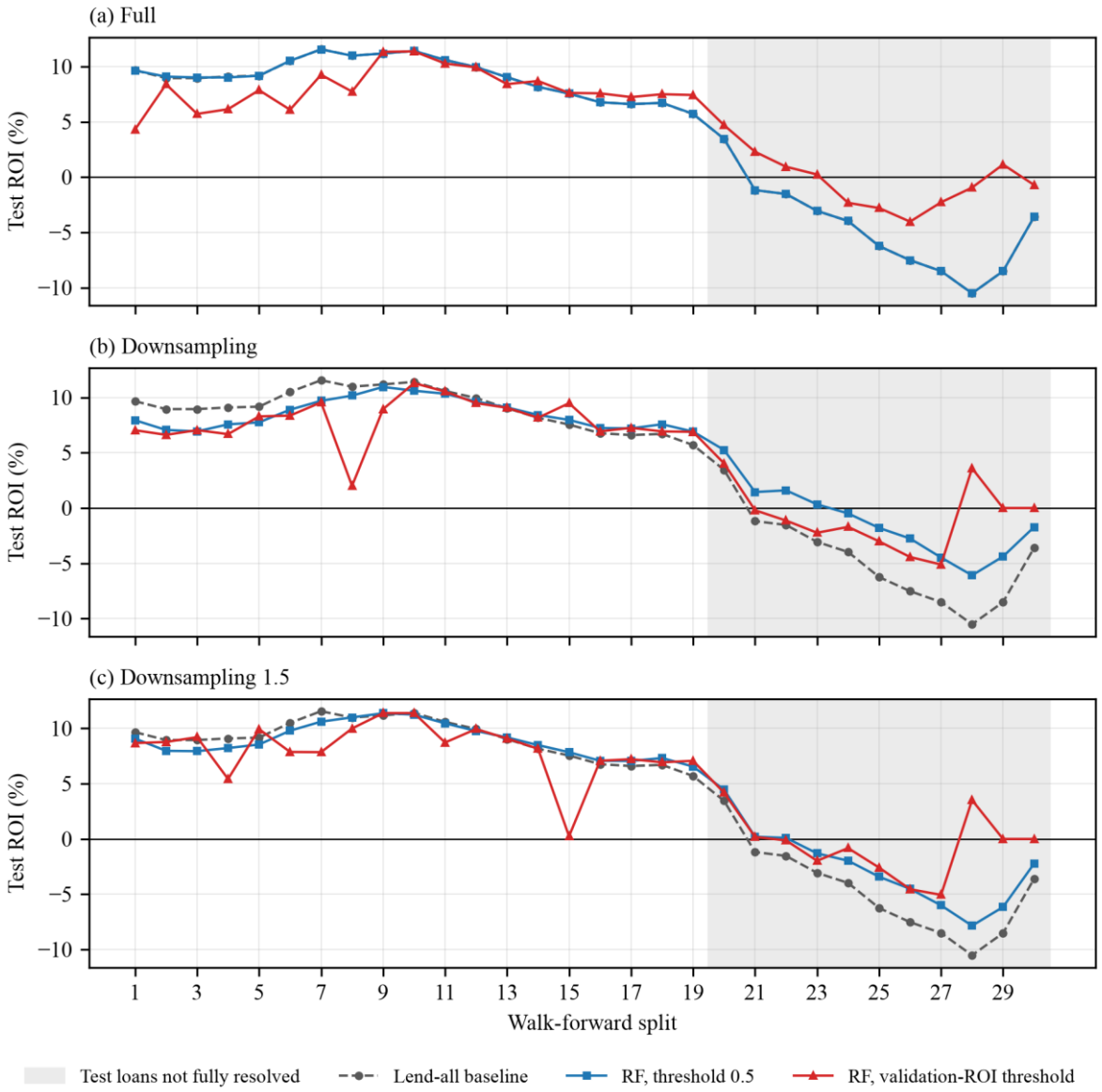


Figure 3. Test ROI (%) of each walk-forward split for the lend-all baseline and the random forest with threshold 0.5 and τ^*, trained on (a) Full, (b) Downsampling and (c) Downsampling 1.5 data. Shaded splits have fewer than 99% of their test loans resolved. In panel (a) the line for threshold 0.5 lies on top of the lend-all line. ROI is set to zero in splits where no loan was funded

C. Paired Comparisons

Table 6 tests the main differences in window ROI. Over all windows, the random forest with full data and τ^* earned 0.92 percentage points more than lend-all per window (95% CI −0.29 to 2.23) and was better in 18 of 30 windows (Wilcoxon p = 0.171). Downsampling with threshold 0.5 earned 0.92 points more (95% CI 0.14 to 1.71; p = 0.077). Compared with threshold 0.5, τ^* changed the mean ROI by 0.92 points for full data, 0.23 points for Downsampling 1.5 and −0.28 points for Downsampling, where it was better in only 9 of 30 windows. After the Holm adjustment over the 19 comparisons of each family, no difference was significant at the 5% level. Differences of about one percentage point are therefore within the variation between windows.

Table 6. Paired comparisons of window-level ROI over all 30 windows and over the matured windows 1–19

Comparison (A − B)	All: Δ (95% CI)	W/L	p	p_{Holm}	Matured: Δ (95% CI)	W/L	p	p_{Holm}
RF Full, 0.5 − Lend-all	0.00 (−0.01, 0.02)	10/11	0.689	1.000	0.01 (−0.01, 0.03)	7/7	0.683	1.000
RF Downsampling, 0.5 − Lend-all	0.92 (0.14, 1.71)	18/12	0.077	0.921	−0.55 (−1.01, −0.10)	7/12	0.066	1.000
RF Downsampling 1.5, 0.5 − Lend-all	0.65 (0.22, 1.10)	20/10	0.019	0.334	−0.15 (−0.41, 0.10)	9/10	0.352	1.000
RF Full, τ^* − Lend-all	0.92 (−0.29, 2.23)	18/12	0.171	1.000	−1.04 (−1.92, −0.19)	7/12	0.123	1.000
RF Downsampling, τ^* − Lend-all	0.64 (−0.66, 2.07)	17/12	0.405	1.000	−1.15 (−2.29, −0.26)	6/12	0.035	0.659
RF Downsampling 1.5, τ^* − Lend-all	0.88 (−0.37, 2.24)	19/10	0.127	1.000	−0.92 (−1.90, −0.08)	8/10	0.286	1.000
RF Full, τ^* − RF Full, 0.5	0.92 (−0.30, 2.17)	18/12	0.177	1.000	−1.04 (−1.93, −0.23)	7/12	0.113	1.000
RF Downsampling, τ^* − RF Downsampling, 0.5	−0.28 (−1.20, 0.74)	9/21	0.070	0.909	−0.60 (−1.59, 0.06)	6/13	0.134	1.000
RF Downsampling 1.5, τ^* − RF Downsampling 1.5, 0.5	0.23 (−0.76, 1.37)	15/15	0.968	1.000	−0.77 (−1.77, 0.01)	9/10	0.258	1.000
RF Full, τ^* − Grade-based	0.66 (−0.23, 1.55)	20/10	0.064	0.890	−0.66 (−1.53, 0.12)	9/10	0.623	1.000
RF Full, τ^* − LR Full, τ^*	−0.52 (−1.36, 0.32)	13/17	0.318	1.000	−0.55 (−1.32, 0.15)	8/11	0.275	1.000
LR Downsampling, τ^* − Lend-all	2.34 (0.55, 4.26)	21/9	0.013	0.244	0.38 (−0.96, 1.58)	12/7	0.312	1.000
Grade-based − Lend-all	0.26 (−0.26, 0.83)	12/8	0.411	1.000	−0.37 (−0.82, −0.06)	4/8	0.050	0.848

Δ: mean difference in window ROI, A minus B, in percentage points, with 95% bootstrap confidence interval; W/L: windows in which A had a higher/lower ROI than B (ties not counted); p: Wilcoxon signed-rank test; p_{Holm}: Holm-adjusted over the 19 comparisons of each family. τ^*: threshold selected by validation ROI.

D. Effect of Unresolved Loans

The data end in December 2018, and loans that were still running were excluded (Section II.B.3). Table 2 shows the consequence. At least 99.66% of the 36-month loans issued in the test months of windows 1 to 19 had resolved by then, but only 85.4% in window 20 and 6.6% in window 30, whose test set therefore holds 5,965 of the 90,907 loans issued in its months. Loans that default are charged off within months, whereas repaid loans resolve only at maturity or when the borrower prepays, so the resolved part of a recent vintage over-represents defaults. Among resolved test loans, the default rate was 10.3–15.1% in windows 1 to 19 but 16.2–22.6% in windows 20 to 28. This is why the ROI of lend-all is negative from window 21 onward (Figure 3). The negative returns of the late windows describe an incomplete sample, not the final return of these vintages, which cannot be measured from these data.

Table 7 repeats the financial comparison on the 19 matured windows. Lend-all earned 9.08% per window (SD 1.78), had no negative window and made a profit of USD 640.0 million. The random forest did not improve on it. With full data and threshold 0.5 it lent 99.95% of the capital and matched lend-all (difference 0.01 points); every other random forest strategy had a lower mean ROI. Relative to lend-all, τ^* lowered the mean ROI by 1.04 (full data), 1.15 (Downsampling) and 0.92 (Downsampling 1.5) percentage points and was worse in 10 to 12 of the 19 windows (Table 6). The forest with full data and τ^* reached a higher total ROI than lend-all (8.72% against 8.15%), because it lent little in the early windows, where its selections did worse, and more in the later ones. It deployed only 43.3% of the capital, and its profit was 54% lower. The highest Sharpe ratios among the forests came from the forests trained on Downsampling and Downsampling 1.5 data with threshold 0.5 (3.58 and 3.33 against 2.94), whose window returns varied less, although their mean ROI was lower than that of lend-all.

The per-window results suggest that the models need larger training sets. In windows 1 to 9, with at most 5,285 training loans, the ROI-based thresholds lent only 0.4–20.2% of the capital and earned less than lend-all in most windows. In windows 14 to 19, with 7,438 to 24,624 training loans, the downsampled random forest with threshold 0.5 earned more than lend-all in 6 of 6 windows (by 0.26 to 1.22 points), and the full-data forest with τ^* did so in 6 of 6. This pattern comes from six windows and was not tested formally.

Table 7. Financial results over the matured windows 1–19

Model	Training data	Thres-hold	Avg ROI (SD)	95% CI	Total ROI	Sharpe	Capital (%)	Lent (USD M)	Profit (USD M)	Idle-adj. ROI	Neg.
Lend-all	–	–	9.08 (1.78)	8.22, 9.93	8.15	2.94	100.0	7,848.9	640.0	9.08	0
Grade rule	–	Grade cutoff	8.70 (1.70)	7.89, 9.52	8.03	2.95	96.9	7,602.7	610.7	8.08	0
RF	Full	0.5	9.09 (1.77)	8.23, 9.94	8.16	2.94	99.9	7,844.6	640.0	9.07	0
RF	Full	τ^*	8.04 (1.84)	7.15, 8.93	8.72	2.88	43.3	3,398.4	296.5	4.22	0
RF	Downsampling	0.5	8.52 (1.35)	7.87, 9.18	8.21	3.58	54.7	4,294.9	352.8	5.28	0
RF	Downsampling	τ^*	7.92 (2.00)	6.96, 8.89	7.83	2.79	56.5	4,431.9	347.1	4.04	0
RF	Downsampling 1.5	0.5	8.92 (1.52)	8.19, 9.66	8.33	3.33	77.7	6,097.2	507.7	7.25	0
RF	Downsampling 1.5	τ^*	8.16 (2.46)	6.97, 9.34	8.11	2.18	60.5	4,751.9	385.2	4.52	0
LR	Full	0.5	9.07 (1.77)	8.21, 9.92	8.17	2.95	98.9	7,760.6	634.0	8.91	0
LR	Full	τ^*	8.59 (1.89)	7.68, 9.51	9.16	2.89	14.5	1,141.4	104.6	3.34	0
LR	Downsampling	0.5	8.76 (1.26)	8.16, 9.37	8.18	3.67	54.7	4,291.5	351.0	5.29	0
LR	Downsampling	τ^*	9.45 (3.63)	7.70, 11.21	9.19	1.90	6.6	520.9	47.8	2.52	1
LR	Downsampling 1.5	0.5	8.96 (1.38)	8.29, 9.63	8.25	3.52	72.2	5,665.4	467.7	6.76	0
LR	Downsampling 1.5	τ^*	9.05 (2.84)	7.68, 10.41	9.62	2.25	7.6	596.2	57.4	3.02	0

ROI values in %. Avg ROI: mean window ROI (Eq. 3) with the standard deviation across windows in brackets; 95% CI: bootstrap interval of the mean; Total ROI: Eq. (4); Sharpe: Eq. (7) with the 3-year Treasury return of Eq. (6); Capital: amount lent as a share of the lend-all amount; Lent and Profit in millions of US dollars, amount received = lent + profit; Idle-adj. ROI: mean of Eq. (5); Neg.: windows with negative ROI; τ^*: threshold selected by validation ROI (Eq. 1).

E. Robustness

Table 8 repeats the random forest with five seeds. With threshold 0.5 the seed hardly mattered: the mean ROI over all windows varied by at most 0.11 percentage points between seeds. With τ^* the spread was much wider, for example from 4.24% to 4.92% for Downsampling 1.5. In the matured windows, all 15 combinations of seed and training set with τ^* had a lower mean ROI than lend-all, by 0.38 to 1.81 percentage points. The random forest results were identical when the whole experiment was run a second time. The logistic regression solver in cuML is not bit-for-bit reproducible on the GPU: in the repeated run its AUC changed by at most 0.01 percentage points, but the selected threshold changed in a few windows, which moved the mean ROI of logistic regression with τ^* by up to 0.21 points.

Table 9 changes the training window and the forest. An expanding training window, which keeps all earlier loans, did not raise the AUC (63.6–64.5% for the random forest and 63.2–64.2% for logistic regression), so the older loans added little information about later ones. With 100 trees the AUC fell by 0.7 percentage points, and with a maximum depth of 16 it rose by at most 0.2 points. The mean ROI of τ^* over all windows moved by up to 0.6 percentage points between these variants, but the matured-window comparison did not change. In every variant the forest with full data and threshold 0.5 matched lend-all, because it funds almost every loan, and every other random forest strategy stayed below it (highest 8.95% against 9.08%). Some logistic regression strategies had a higher mean ROI than lend-all in the matured windows, by up to 0.95 points, but those with τ^* lent only 7–14% of the capital, and the one tested in Table 6 (logistic regression with Downsampling and τ^*) did not differ significantly from lend-all.

Table 8. Random forest results across five random seeds (0, 1, 2, 3 and 42)

Model	Training data	Thres-hold	All: Avg ROI mean (SD)	All: Avg ROI range	All: Total ROI range	Matured: Avg ROI mean (SD)	Matured: Avg ROI range	Matured: Capital (%)
RF	Full	0.5	4.05 (0.00)	4.04–4.05	4.11–4.11	9.08 (0.00)	9.08–9.09	99.9–99.9
RF	Full	τ^*	5.14 (0.13)	4.97–5.32	4.87–5.18	8.44 (0.25)	8.04–8.70	43.3–48.0
RF	Downsampling	0.5	5.02 (0.05)	4.96–5.08	5.20–5.26	8.59 (0.08)	8.50–8.67	54.6–55.4
RF	Downsampling	τ^*	4.71 (0.20)	4.44–4.98	3.42–4.72	7.86 (0.46)	7.30–8.44	38.5–67.1
RF	Downsampling 1.5	0.5	4.70 (0.02)	4.67–4.74	4.85–4.93	8.92 (0.04)	8.86–8.97	77.7–78.0
RF	Downsampling 1.5	τ^*	4.60 (0.25)	4.24–4.92	3.43–4.66	7.59 (0.34)	7.27–8.16	32.0–60.5

ROI values in %. All: 30 windows; Matured: windows 1–19. Mean (SD) and range are taken over the five seeds. Lend-all: 4.04% (all) and 9.08% (matured).

Table 9. Sensitivity of the random forest to the training window and hyperparameters (seed 42)

Model	Training data	Thres-hold	Sliding, 300 trees, depth 64	Expanding window	100 trees	Maximum depth 16
RF	Full	AUC (%)	63.69	63.58	62.97	63.92
RF	Full	0.5	9.09 / 4.05	9.08 / 4.04	9.08 / 4.04	9.08 / 4.05
RF	Full	τ^*	8.04 / 4.97	8.11 / 5.07	8.08 / 4.81	7.90 / 4.57
RF	Downsampling	0.5	8.52 / 4.96	8.60 / 5.09	8.58 / 4.91	8.54 / 4.99
RF	Downsampling	τ^*	7.92 / 4.68	7.16 / 4.67	8.05 / 5.27	7.80 / 4.67
RF	Downsampling 1.5	0.5	8.92 / 4.70	8.95 / 4.75	8.86 / 4.61	8.92 / 4.71
RF	Downsampling 1.5	τ^*	8.16 / 4.92	8.20 / 4.83	7.75 / 4.62	8.64 / 5.26

Cells give the mean window ROI in % over the matured windows 1–19 / over all 30 windows; the first row gives the mean AUC with full training data. Lend-all: 9.08 / 4.04.

F. Selected Thresholds

Figure 4 shows the selected thresholds and the share of test loans funded with them. With full training data the threshold ranged from 0.01 to 0.27 (mean 0.13), far below 0.5, because a forest trained on imbalanced data assigns low default probabilities to most loans. With downsampling the thresholds were higher and much less stable (0.03–0.92, mean 0.38, for the 1:1 ratio): they jumped between very low values, which fund almost nothing, and values around 0.6. The forest funded no loan in 2 windows with Downsampling and in 2 with Downsampling 1.5. This instability matches the small or negative effect of τ^* on the downsampled forests in Table 5.

Validation ROI was a weak guide to test ROI. Across windows, the correlation between the best validation ROI and the resulting test ROI was 0.13 for full data, 0.13 for Downsampling and 0.16 for Downsampling 1.5. Two reasons are likely. The validation sets of windows 1 to 7 hold only 128 to 994 loans, so the best of 100 thresholds is partly chosen on noise. In addition, the validation loans were issued more than three years before the test loans, and the relation between predicted default probability and return can change in that time (Nikolaidis & Doumpos, 2022; Bayram et al., 2022).

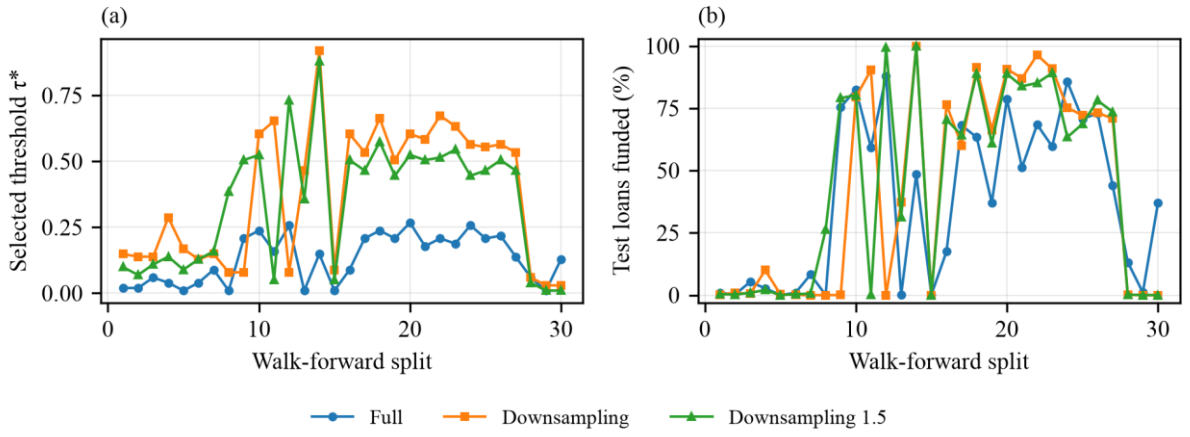


Figure 4. (a) Threshold τ^* selected on the validation split and (b) share of test loans funded with τ^*, random forest, seed 42

G. Discussion

The AUC of about 0.64 is far below the scores above 90% reported in some Lending Club studies (Núñez Mora et al., 2023). Several choices in this study lower the measured AUC: the models use only information available at funding, the test loans were issued at least three years after the training loans, the early training sets are small, and default rates differ between periods. Random splits and post-issuance variables, by contrast, inflate measured performance (Kapoor & Narayanan, 2023), and random resampling overstates future performance when the data are not stable over time (Hornung et al., 2026). The similar AUC of the random forest and logistic regression agrees with benchmarks in which a well-specified logistic regression comes close to tree ensembles on credit data (Dumitrescu et al., 2022).

For investors, the results separate return per dollar lent from total profit. Lending to fewer loans raised the return per dollar in some windows, but it left much of the capital unused, and the gain did not hold in the matured windows. Profit-scoring studies report larger gains on Lending Club data (Lyócsa et al., 2022). They model returns directly instead of tuning the threshold of a default classifier on a short validation period, so the results are not directly comparable. The present results indicate that a threshold chosen on two months of loans issued three

years earlier is too noisy to improve on broad lending. Profit-oriented learning (Petrides et al., 2022; Vanderschueren et al., 2022), longer validation periods and risk-adjusted comparisons (Alonso Robisco & Carbó Martínez, 2022) are more promising routes.

The censoring result also matters for other studies. Any time-ordered evaluation on Lending Club data that keeps only resolved loans will see test returns fall in the most recent vintages, whatever the model. If such windows are included, a strategy that lends less will appear safer simply because it avoids part of an artificially poor sample. Biased evaluation samples of this kind can change which model appears best (Kozodoi et al., 2025), so the share of resolved loans should be reported with every walk-forward result.

IV. CONCLUSION

This study evaluated ROI-based decision thresholds for a random forest default classifier on Lending Club 36-month loans. The design used 30 walk-forward windows with a 36-month gap between validation and test loans, inputs limited to information available at funding, and a check of how many test loans had resolved. For RQ1, the random forest reached a mean AUC of 63.7–64.6%, about the same as logistic regression. For RQ2, the ROI-based threshold raised the mean window ROI over all windows by 0.92 percentage points with full training data but lowered it with 1:1 downsampling. None of the differences was significant after adjustment for multiple tests, and the grade-based rule and the models that lent almost every loan produced the highest profit. For RQ3, the conclusions depend on the censored windows. In the 19 matured windows, lend-all earned 9.08% per window, the random forest at best matched it by funding almost every loan, and the selected thresholds varied strongly across windows and seeds.

For investors, these results do not support replacing broad lending with a threshold tuned on a short validation period. Some selective strategies reduced the variation of returns between windows, but they left much of the capital unlent and earned less profit. For researchers, walk-forward tests of return-based strategies on Lending Club data should report how many test loans had resolved and should check their conclusions on matured vintages.

The study has several limitations. It uses one platform and only 36-month loans, with no external validation on other lenders or periods. Only 19 windows had matured, so the recent vintages could not be evaluated. The ROI is undiscounted, excludes platform fees, taxes and the reinvestment of early repayments, and assumes that the investor could fund every approved loan. The missing-value filter was computed on the full dataset, employment length was coded in lexical order, and the forest hyperparameters were not tuned per window. The main classifier is a single random forest, with logistic regression as the only other learner, and the thresholds were selected on small validation sets. The Sharpe ratios are computed over windows with overlapping loan lives and are descriptive only.

Future work could select thresholds on longer or pooled validation periods, model returns or survival times directly so that loans still running can be used, compare gradient boosting and cost-sensitive learners in the same design, and test the design on other platforms and loan terms.

AUTHOR'S CONTRIBUTION

Henry Lucky: Conceptualization, Data curation, Formal analysis, Methodology, Validation, Visualization, Writing; Anderies: Validation, Writing.

AVAILABILITY DATA AND MATERIALS

The dataset used in this research can be accessed via the link: *https://www.kaggle.com/datasets/wordsforthewise/lending-club/data*. The 3-year Treasury yields (series GS3) are available from FRED at *https://fred.stlouisfed.org/series/GS3*.

REFERENCES

Alonso Robisco, A., & Carbó Martínez, J. M. (2022). Measuring the model risk-adjusted performance of machine learning algorithms in credit default prediction. *Financial Innovation, 8*(1), 70. https://doi.org/10.1186/s40854-022-00366-1

Bayram, F., Ahmed, B. S., & Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems, 245*, 108632. https://doi.org/10.1016/j.knosys.2022.108632

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, Y., Calabrese, R., & Martin-Barragan, B. (2024). Interpretable machine learning for imbalanced credit scoring datasets. *European Journal of Operational Research, 312*(1), 357–372. https://doi.org/10.1016/j.ejor.2023.06.036

Coenen, L., Verbeke, W., & Guns, T. (2022). Machine learning methods for short-term probability of default: A comparison of

classification, regression and ranking methods. *Journal of the Operational Research Society, 73*(1), 191–206. https://doi.org/10.1080/01605682.2020.1865847

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research, 297*(3), 1178–1192. https://doi.org/10.1016/j.ejor.2021.06.053

Gunnarsson, B. R., vanden Broucke, S., Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research, 295*(1), 292–305. https://doi.org/10.1016/j.ejor.2021.03.006

Hornung, R., Nalenz, M., Schneider, L., Bender, A., Bothmann, L., Dumpert, F., Bischl, B., Augustin, T., & Boulesteix, A.-L. (2026). Evaluating machine learning models in nonstandard settings: An overview and new findings. *Statistical Science, 41*(3), 524–560. https://doi.org/10.1214/25-STS984

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns, 4*(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Ko, P.-C., Lin, P.-C., Do, H.-T., & Huang, Y.-F. (2022). P2P lending default prediction based on AI and statistical models. *Entropy, 24*(6), 801. https://doi.org/10.3390/e24060801

Kozodoi, N., Lessmann, S., Alamgir, M., Moreira-Matias, L., & Papakonstantinou, K. (2025). Fighting sampling bias: A framework for training and evaluating credit scoring models. *European Journal of Operational Research, 324*(2), 616–628. https://doi.org/10.1016/j.ejor.2025.01.040

Kriebel, J., & Stitz, L. (2022). Credit default prediction from user-generated text in peer-to-peer lending using deep learning. *European Journal of Operational Research, 302*(1), 309–323. https://doi.org/10.1016/j.ejor.2021.12.024

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*(1), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030

Li, L.-H., Sharma, A. K., & Cheng, S.-T. (2025). Explainable AI based LightGBM prediction model to predict default borrower in social lending platform. *Intelligent Systems with Applications, 26*, 200514. https://doi.org/10.1016/j.iswa.2025.200514

Liu, W., Fan, H., & Xia, M. (2022). Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Systems with Applications, 189*, 116034. https://doi.org/10.1016/j.eswa.2021.116034

Liu, Z. (2025). Evaluate Lending Club loan status by machine learning. *Advances in Economics, Management and Political Sciences, 154*(1), 51–57. https://doi.org/10.54254/2754-1169/2024.19542

Lyócsa, Š., Vašaničová, P., Hadji Misheva, B., & Vateha, M. D. (2022). Default or profit scoring credit systems? Evidence from European and US peer-to-peer lending markets. *Financial Innovation, 8*(1), 32. https://doi.org/10.1186/s40854-022-00338-5

Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications, 42*(10), 4621–4631. https://doi.org/10.1016/j.eswa.2015.02.001

Nguyen, L., Ahsan, M., & Haider, J. (2024). Reimagining peer-to-peer lending sustainability: Unveiling predictive insights with innovative machine learning approaches for loan default anticipation. *FinTech, 3*(1), 184–215. https://doi.org/10.3390/fintech3010012

Nikolaidis, D., & Doumpos, M. (2022). Credit scoring with drift adaptation using local regions of competence. *Operations Research Forum, 3*(4), 67. https://doi.org/10.1007/s43069-022-00177-1

Noriega, J. P., Rivera, L. A., & Herrera, J. A. (2023). Machine learning for credit risk prediction: A systematic literature review. *Data, 8*(11), 169. https://doi.org/10.3390/data8110169

Núñez Mora, J. A., Moncayo, P., Franco, C., Madrazo-Lemarroy, P., & Beltrán, J. (2023). Loan default prediction: A complete revision of LendingClub. *Revista Mexicana de Economía y Finanzas, 18*(3), 1–13. https://doi.org/10.21919/remef.v18i3.886

Petrides, G., Moldovan, D., Coenen, L., Guns, T., & Verbeke, W. (2022). Cost-sensitive learning for profit-driven credit scoring. *Journal of the Operational Research Society, 73*(2), 338–350. https://doi.org/10.1080/01605682.2020.1843975

Rong, Y., Liu, S., Yan, S., Huang, W. W., & Chen, Y. (2023). Proposing a new loan recommendation framework for loan allocation strategies in online P2P lending. *Industrial Management & Data Systems, 123*(3), 910–930. https://doi.org/10.1108/IMDS-07-2022-0399

Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in P2P lending. *PLOS ONE, 10*(10), e0139427. https://doi.org/10.1371/journal.pone.0139427

Sharpe, W. F. (1994). The Sharpe ratio. *The Journal of Portfolio Management, 21*(1), 49–58. https://doi.org/10.3905/jpm.1994.409501

Vanderschueren, T., Verdonck, T., Baesens, B., & Verbeke, W. (2022). Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies. *Information Sciences, 594*, 400–415. https://doi.org/10.1016/j.ins.2022.02.021

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research, 238*(2), 505–513. https://doi.org/10.1016/j.ejor.2014.04.001

Yang, C., Dong, Y., Lu, J., & Peng, Z. (2022). Solving imbalanced data in credit risk prediction: A comparison of resampling strategies for different machine learning classification algorithms, taking threshold tuning into account. In *Proceedings of the 2022 5th International Conference on Machine Learning and Machine Intelligence* (pp. 30–40). ACM. https://doi.org/10.1145/3568199.3568204

Zhao, Z., Cui, T., Ding, S., Li, J., & Bellotti, A. G. (2024). Resampling techniques study on class imbalance problem in credit risk prediction. *Mathematics, 12*(5), 701. https://doi.org/10.3390/math12050701