Data Science and Engineering (DSE) Record, Volume 7, issue 1.

Evaluation Constrained and Unconstrained Machine Learning Strategies for Credit Risk Optimization in Financial Institutions

Putanyn Manee [1] and Jakramate Bootkrajang [2]

[1] Data Science Consortium, Chiang Mai University, Thailand
putanyn_manee@cmu.ac.th

[2] Department of Computer Science, Faculty of Science, Chiang Mai University
jakramate.b@cmu.ac.th

Abstract. This research investigates the critical strategic trade-offs between revenue maximization and loss control in credit risk modeling by comparing three distinct optimization strategies. The study addresses the gap between technical model performance and actual business outcomes, which is often overlooked in traditional machine learning applications. Using a synthetic dataset of 135,000 personal loans with 29 features, the study evaluates five machine learning models across multiple probability thresholds. The findings reveal that behavioral features capturing post-origination payment patterns drive 71% of predictive improvement, compared to only 29% from hyperparameter tuning. Strategy A (Income Maximization) achieved the highest profit of $662.54M with a 98.7% approval rate using Gradient Boosting at a 20% threshold. In contrast, Strategy B (Pure Loss Minimization) produced an impractical 0.04% approval rate, proving that unconstrained loss reduction leads to operational failure. Strategy C (Constrained Loss Minimization) implemented a 60% minimum approval constraint based on Federal Reserve standards, achieving $416.54M in profit with 40% lower losses than Strategy A. Critically, the choice of probability threshold demonstrated a five times greater financial impact than the choice of algorithm. These results provide strong empirical evidence that integrating specific business constraints is essential for effective and sustainable credit risk.

Keywords: Foreign Trade, Influencing Factors, Principal component analysis.

1 Introduction

This independent study evolved from practical banking experience that revealed a fundamental gap between academic credit risk modeling and operational decision-making. While the original research proposal focused on the technical architecture for a two-phase credit assessment system, my role as a credit analyst at a leading Thai financial institution shifted the focus. It became clear that the core challenge was not technical prediction capability, but strategic optimization: specifically, how to balance revenue generation against loss control.

In daily operations, credit committees repeatedly face strategic questions regarding whether to approve borderline applicants to capture revenue or reject them to reduce risk. The Head of Corporate Credit summarized this tension by stating, "We care about revenue, but we care more about what we're going to lose." This perspective encapsulates the fundamental dilemma facing lending institutions and motivated a shift in research focus from technical architecture to a strategic optimization philosophy.

The research initially utilized the Deloitte loan default prediction dataset (Deloitte, 2021), which framed credit risk as a pure classification problem optimizing for logloss. However, banking experience suggests that this approach is often misaligned with institutional decision-making. Banks optimize financial outcomes under business constraints rather than statistical metrics. While the Deloitte dataset is valuable for algorithmic benchmarking, it lacked the feature richness and business context necessary to investigate the strategic trade-offs observed in practice.

This led to the use of a synthetic credit risk portfolio dataset featuring 135,000 personal loan applications and 29 features (Kaggle, 2024). This data structure enables the investigation of practical strategic questions, such as how models should be optimized when approval decisions have asymmetric financial consequences and whether loss minimization can be pursued realistically alongside business constraints.

Traditional credit risk models optimize for predictive accuracy, such as AUC, and often treat all errors equally (Hand & Henley, 1997; Baesens et al., 2003). This approach ignores asymmetric financial consequences where approving a defaulting loan loses the entire principal, while rejecting a creditworthy applicant only foregoes interest (Elkan, 2001; Thomas, 2009). Although contemporary machine learning methods like XGBoost achieve superior accuracy (Chen & Guestrin, 2016), they typically optimize standard classification objectives that do not consider these financial outcomes. This gap was evident in practice; models provided statistically accurate scores, but the resulting decisions were often financially suboptimal, leading credit officers to override recommendations based on business judgment.

Real-world deployment introduces business constraints that must be accounted for. Banks cannot minimize losses by rejecting all applicants, nor can they maximize revenue by approving everyone. Institutions must maintain approval rates within industry norms to preserve market share and comply with regulatory requirements. Consequently, this study systematically compares three optimization strategies on the synthetic dataset. Strategy A represents conventional practice, optimizing for AUC and selecting thresholds that maximize profit. Strategy B trains models to minimize expected losses without constraints to test the viability of a "loss-first" philosophy. Finally, Strategy C investigates constrained loss minimization by maintaining a 60% minimum approval rate based on Federal Reserve data (Federal Reserve, 2024), to determine if such constraints enable a more balanced and practical optimization

2 Literature Review

2.1 Evolution of Credit Risk Modeling

Credit risk modeling traditionally relied on logistic regression for interpretable default probability estimates (Hand & Henley, 1997; Thomas, 2009). Modern machine learning methods—gradient boosting, XGBoost (Chen & Guestrin, 2016), LightGBM

(Ke et al., 2017)—capture nonlinear relationships and consistently outperform logistic regression in discriminatory power (Baesens et al., 2003; Lessmann et al., 2015).

Early modeling approaches typically framed credit scoring as a simple binary classification problem (Hand & Henley, 1997). This perspective tended to treat all classification errors equally, despite the fact that the financial impact of approving a defaulter is vastly different from that of rejecting a creditworthy applicant (Elkan, 2001; Thomas, 2009). This oversight has created a persistent gap between technical model performance and actual business value (Louzada et al., 2016; Abdou & Pointon, 2011). Research has also highlighted the importance of data selection in this evolution, noting that behavioral features tracking post-origination payment patterns can improve predictive accuracy by 10% to 15% over models relying solely on application-time data (Khandani et al., 2010).

2.2 Cost-Sensitive Learning and Profit-Based Evaluation

The main problem in credit risk is that the costs of making a mistake are not equal. If a bank approves a borrower who defaults, it loses the entire loan principal, but if it rejects a good borrower, it only loses the potential interest (Elkan, 2001; Thomas, 2009). Because of this imbalance, simply trying to have a low error rate or a high AUC is not the best approach. Instead, models should aim to minimize the expected cost of these mistakes (Elkan, 2001; Zhou & Liu, 2006). This can be done by adjusting decision thresholds, re-weighting the importance of certain data, or changing how the data is sampled (Zhou & Liu, 2006; Ling & Sheng, 2011).

Correa Bahnsen (2015) introduced a more specific method where the cost of a mistake depends on the actual size of the loan. His work showed that models focusing on these specific costs are much more profitable than those that only look at traditional statistical scores like the F1-score (Bahnsen, 2015). For this to work well, a bank must correctly define its costs and align them with its business goals (Verbraken et al., 2014; Liu et al., 2025).

Since banks care more about profit than simple accuracy, researchers developed the Expected Maximum Profit (EMP) framework. This method combines the model's predictions with the actual terms of the loan to calculate its economic value (Verbraken et al., 2014). In his research, Verbraken (2013) showed that choosing a model based on EMP leads to better financial results than choosing one based on AUC. This confirms that a model with the best statistical score is not always the one that makes the most money (Verbraken et al., 2014; Lessmann et al., 2019). This profit perspective has been extended through profit-maximizing logistic regression, where variables are evaluated by their contribution to expected profit rather than purely to goodness-of-fit (Huang et al., 2018; Lessmann & Voß, 2018). More recent studies have continued this trend by focusing on profit-based scoring and showing that how we compare models can change depending on where the cutoff is set (Erola, 2020; Susana, 2024; Scagliola, 2022).

Finally, where a bank sets its approval threshold has a massive impact on the results. Changing the cutoff can lead to very different approval rates and profits, even when using the exact same model. In many cases, the choice of the threshold matters more than which model is used in the first place (Hand & Anagnostopoulos, 2013; De Lange et al., 2022; Di Maggio et al., 2023).

2.3 Business Constraints and Strategic Trade-offs

While cost-sensitive frameworks provide a strong theoretical basis, they often assume that lenders have the total freedom to adopt any approval policy (Verbraken et al., 2014; Lessmann et al., 2019). In reality, financial institutions must operate within a complex set of constraints, including target approval rates, capital requirements, and risk appetite statements (Thomas, 2009; Baesens, 2014). Without these constraints, mathematical optimization can produce extreme solutions—such as rejecting nearly all applications—that are technically optimal but operationally unacceptable for a functioning bank (Elkan, 2001; Liu et al., 2025).

Data from the Federal Reserve shows that approval rates vary significantly by product, with credit cards typically at 80%, auto loans between 88% and 90%, and personal loans ranging from 50% to 60% (Federal Reserve, 2024). Industry benchmarks, such as those used in the Oxera study (2024), often employ a 60% target for personal lending to represent consensus practice. Other industry standards further define these operational boundaries, such as the 600 FICO minimum for LendingClub (Debt.org, 2025), the 0.3% probability of default (PD) threshold used by UniCredit Bulbank for low-risk categorization (UniCredit Bulbank, 2024), and the 20% PD threshold established by the FDIC for higher-risk consumer loans (FDIC, 2012).

Recent research has begun to integrate these explicit constraints directly into the modeling process. Studies have examined model performance under pre-defined policy constraints (Crook & Edelman, 2015) and fairness-aware scoring (Fuster et al., 2022; Heidari & Krause, 2023). A recurring insight in this literature is that models purely focused on minimizing losses may only approve approximately 1% of applications, effectively eliminating risk by ceasing to lend entirely (Oxera, 2024). Incorporating realistic approval targets is therefore essential to ensure that models remain aligned with business realities (Federal Reserve, 2024).

There is an emerging consensus that credit risk should be treated as a multi-objective optimization problem that balances revenue, loss control, and compliance (Lessmann & Vob, 2018; Fuster et al., 2022). Researchers often explore these trade-offs using Pareto frontiers to visualize the balance between competing goals (Fuster et al., 2022; Verbraken et al., 2014). Different types of institutions will rationally select different points on this frontier based on their strategic objectives (Thomas, 2009; Lessmann et al., 2019). While unconstrained loss minimization leads to extreme conservatism (Bahnsen et al., 2014; Onișin et al., 2019), constrained optimization tends to yield balanced solutions that fall within market practice ranges (Verbraken et al., 2014; Voulgaris et al., 2022).

2.4 Positioning of the Present Study

This study contributes systematic empirical comparison of three optimization philosophies: income maximization, pure loss minimization, and constrained loss minimization. Prior research documented advantages of profit-based frameworks and need for business constraints separately, but few implement unconstrained and constrained strategies side-by-side with consistent protocols (Verbraken et al., 2014; Fuster et al., 2022; Liu et al., 2025). Applying all three to a large dataset with minimum approval rates grounded in Federal Reserve standards (Federal Reserve, 2024; Oxera, 2024), this study directly examines whether unconstrained optimization yields deployable models and quantifies trade-offs.

The study addresses optimizing narrow technical metrics without considering institutional constraints, risk appetite, and competitive dynamics (Baesens, 2014; Fuster et al., 2022). Consistent with decision-focused research, the work reframes from "How to predict accurately?" to "How to optimize business outcomes under constraints?", offering evidence-based guidance for strategy selection (Verbraken et al., 2014; Lessmann & Vob, 2018; Voulgaris et al., 2022)

3 Data and Methodology

3.1 Dataset Overview

The dataset is a synthetic credit risk portfolio designed to simulate realistic consumer lending patterns while preserving borrower privacy (Kaggle, 2024). Table 1 summarizes the dataset structure.

Table 1. Statistics of each influencing factor

Characteristic	Value
Total Observations	135,000
Default Rate	7.68% (10,368 defaults)
Non-Default Rate	92.32% (124,632 repaid)
Observation Period	2022-2024
Data Source	Synthetic Credit Risk Portfolio (Kaggle, 2024)
Training Set:	
Observations	108,000 (80%)
Defaults	8,294 (7.68%)
Non-Defaults	99,706 (92.32%)
Test Set:	
Observations	27,000 (20%)
Defaults	2,074 (7.68%)
Non-Defaults	24,926 (92.32%)
Total Features	29
Origination Features	21
Behavioral Features	8

The dataset comprises 135,000 personal loan applications with 29 features and 7.68% default rate (10,368 defaults, 124,632 repaid). This default rate aligns with historical norms for personal lending (Federal Reserve, 2024).

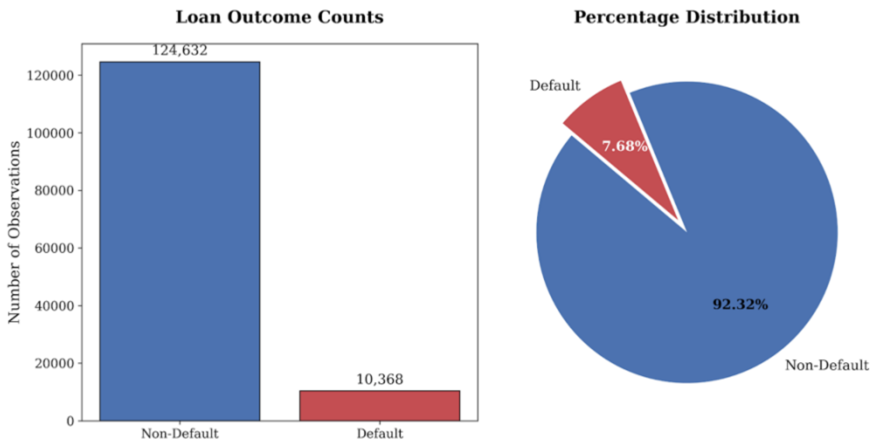

Figure 1. Distribution of Loan Outcomes

The 29 features divide into two categories: 21 origination features capturing borrower characteristics at application (loan amount, interest rate, term, income, employment, debt-to-income ratio, credit score, inquiries, utilization, account age), and 8 behavioral features tracking post-origination payment patterns (delinquency indicators, payment frequency, balance trends, utilization changes, credit inquiries).

Table 2. Origination Feature Descriptions

Feature	Type	Range/Values	Description
Customer Demographics:			
age	Numeric	20-65	Customer age at application (years)
gender	Categorical	Male/Female	Customer gender
employment_status	Categorical	Employed/Self-Employed/Unemployed	Employment status at origination
annual_income	Numeric	Variable	Gross annual income
monthly_income	Numeric	Variable	Gross monthly income (annual/12)
Loan Characteristics:			
loan_amount	Numeric	50,000-1,500,000	Principal loan amount disbursed
interest_rate	Numeric	7.0-24.0%	Annual percentage rate (APR)
loan_term	Numeric	36, 48, 60, 72, 84	Loan tenure in months
emi	Numeric	Variable	Equated Monthly Installment

Credit Profile:			
credit_score_origination	Numeric	300-900	FICO-style credit bureau score
dti	Numeric	0.10-0.70	Debt-to-income ratio
revolving_utilization_orig	Numeric	0.0-1.0	Credit card utilization rate
credit_inquiries_12m	Numeric	0-6	Hard credit inquiries (12 months)

Table 3. Behavioral Feature Descriptions

Feature	Type	Range/Values	Description
Payment Behavior:			
dpd_current	Numeric	0, 30, 60, 90+	Days Past Due in current month
payment_made	Binary	0, 1	Whether EMI payment was made
payment_amount	Numeric	0 to EMI amount	Actual payment received
Credit Bureau Updates:			
credit_score	Numeric	300-900	Current credit bureau score (monthly update)
credit_score_delta_3m	Numeric	-180 to +60	Credit score change over 3 months
revolving_utilization	Numeric	0.0-1.0	Current credit card utilization rate
revolving_util_delta_3m	Numeric	-0.5 to +0.5	Utilization change over 3 months

Behavioral features provide valuable information about evolving borrower financial health, improving model performance by 10-15% relative to application-time features alone (Khandani et al., 2010).

The dataset was partitioned using 80-20 stratified split maintaining 7.68% default rate in both subsets. Training set (108,000 observations) was used for model development, hyperparameter optimization, and cross-validation. Test set (27,000 observations) was held out entirely for final evaluation. All modeling decisions were made using only training data.

3.2 Feature Engineering

Feature engineering is handled in two steps. The first step uses only the data available at the time of the loan application. The second step adds behavioral data that appears as the borrower begins paying back the loan. This structure allows the research to measure exactly how much extra predictive value is gained by tracking a borrower's behavior over time.

The 21 origination features represent standard credit risk variables: borrower financial capacity (income, employment, debt-to-income), credit history quality (credit score, delinquency, public records, account age), current utilization (revolving balance, utilization rate, accounts), credit-seeking behavior (inquiries), and loan characteristics (amount, interest rate, term).

No complex transformations were applied, as tree-based ensemble methods (Gradient Boosting, XGBoost, LightGBM) automatically capture nonlinear relationships. Continuous variables used original scales for tree-based models; standardization was applied for logistic regression and k-nearest neighbors.

The 8 behavioral features track payment patterns providing dynamic risk signals: delinquency indicators (30, 60, 90 days past due), payment frequency patterns, balance trends, utilization changes, and credit inquiries during loan term. Incorporating behavioral features improved model AUC by 8.4% (from 0.5808 to 0.6296), demonstrating substantial incremental value consistent with prior research (Khandani et al., 2010).

All feature engineering was conducted using only training set. The same pipeline was applied identically to test set during final evaluation, preventing data leakage

3.3 Model Selection and Hyperparameter Optimization

Five machine learning algorithms were evaluated: Logistic Regression (traditional baseline for interpretability; Hand & Henley, 1997), k-Nearest Neighbors (non-parametric similarity-based approach), Gradient Boosting (sequential decision tree ensembles; Friedman, 2001), XGBoost (optimized gradient boosting with regularization; Chen & Guestrin, 2016), and LightGBM (efficient implementation with histogram-based splitting; Ke et al., 2017).

Hyperparameter optimization was conducted using Optuna, a Bayesian optimization framework employing Tree-structured Parzen Estimator sampling (Akiba et al., 2019).

Table 4. Hyperparameter Optimization Search Spaces

Model	Hyperparameter	Search Range
Logistic Regression	Regularization (C)	[0.001, 10.0] (log scale)
	Penalty	L2
K-Nearest Neighbors	Neighbors (k)	[3, 50]
	Weights	uniform, distance
Gradient Boosting	Learning Rate	[0.01, 0.3] (log scale)
	N Estimators	[50, 500]
	Max Depth	[3, 10]
	Min Samples Split	[2, 20]

XGBoost	Learning Rate	[0.01, 0.3] (log scale)
	N Estimators	[50, 500]
	Max Depth	[3, 10]
	Min Child Weight	[1, 10]
	Subsample	[0.6, 1.0]
LightGBM	Learning Rate	[0.01, 0.3] (log scale)
	N Estimators	[50, 500]
	Num Leaves	[20, 100]
	Min Child Samples	[10, 50]

For each algorithm, Optuna ran 10 optimization trials with 5-fold stratified cross-validation on training set, recording cross-validation AUC. The hyperparameter configuration yielding highest cross-validation AUC was selected as optimal. After identifying optimal hyperparameters, final models were trained on complete training set and evaluated on test set. Hyperparameter optimization improved model AUC by 2-3% relative to default settings, with Gradient Boosting achieving best performance (AUC 0.6464).

3.4 Strategy Design and Optimization Objectives

The methodological core of this study is a systematic comparison of three optimization strategies, each reflecting a different institutional philosophy regarding the balance between revenue generation and loss control. By implementing these strategies on identical data, the research creates a controlled environment to evaluate how different optimization objectives translate into real-world banking outcomes.

Before detailing each strategy, it is essential to establish the confusion matrix framework that maps model predictions to financial outcomes.

Table 5. Maps Model Predictions to Financial Outcomes

Prediction Outcome	Definition	Financial Impact
True Negative (TN)	Correct Approved non-default	Generate revenue (earn interest)
False Negative (FN)	Incorrectly approved default	Actual loss (lose entire principle)
True Positive (TP)	Correctly rejected default	Avoided loss
False Positive (FP)	Incorrectly rejected non-default	Opportunity cost (foregone revenue)

Strategy A: Income Maximization (Conventional Approach)

Strategy A represents conventional industry practice, where models are initially optimized for AUC and subsequent thresholds are selected to maximize profit. Models are optimized to maximize predictive accuracy, measured by the Area Under the ROC Curve, which quantifies discrimination ability across all possible thresholds. Following model training, probability thresholds are then selected to maximize total portfolio profit. This represents current industry best practice where model training and business objective alignment are treated as separate sequential steps (Elkan, 2001).

To formalize this approach, the optimization objective for Strategy A is defined to maximize the Profit. This is calculated as the sum of interest income earned from successful loans minus the principal lost from defaulted loans. Unlike other strategies, this objective function contains no penalty term for the default rate, allowing the model to accept higher risks if they are offset by sufficient volume.

Business Objective:

$$\text{Maximize Profit} = \text{Revenue} - \text{Actual Loss}$$

Machine Learning Objective:

$$\text{Training Phase}: \text{Maximize AUC}$$
$$\text{Threshold Selection}: \text{Maxmize Profit, where:}$$
$$\text{Revenue} = \sum(\text{TN} \times \text{Interest Rate} \times \text{Loan Amount} \times \text{Term})$$
$$\text{Loss} = \sum(\text{FN} \times \text{Loan Amount})$$
$$\text{Profit} = \text{Revenue} - \text{Loss}$$

The optimization process evaluates profit across specific probability of default thresholds, namely 5%, 10%, 15%, and 20%. These intervals represent realistic boundaries in personal lending as documented in recent literature (De Lange et al., 2022; Di Maggio et al., 2023). For each threshold, applicants with a predicted default probability below the cutoff are approved, while those above are rejected. The threshold that yields the maximum profit is ultimately selected.

Strategy B: Pure Loss Minimization (Unconstrained)

In contrast, Strategy B focuses on pure loss minimization by directly training models to minimize expected financial loss rather than maximizing classification accuracy. The objective function counts only actual losses from approved defaults, with no penalty for rejecting good customers. This reflects the risk management philosophy observed in practice: "We focus on how much we are going to lose."

To formalize this, the optimization objective for Strategy B is defined to minimize the expected loss from defaults. The cost function penalizes the model solely for approving applicants who subsequently default, ignoring opportunity costs associated with rejected safe loans.

Business Objective:

$$\text{Minimize Expected Loss from Defaults}$$

Manchine Learning Objective:

$$\text{Minimize}: \sum(\text{FN} \times \text{Loan Amount})$$
$$\text{Constraint: No penalty for False Positive}$$

This is achieved by utilizing Optuna to modify the objective function, shifting it from cross-validation AUC to the expected actual loss. During each trial, the loss is computed as the sum of loan amounts for all approved applications that eventually default, averaged across five folds. A critical feature of Strategy B is that it includes no penalty for rejecting creditworthy customers; the objective is solely to reduce the actual losses incurred from approved defaults. Under this strategy, the model is free to reject any number of applicants necessary to minimize defaults, without any operational constraints. For the purposes of this study, four separate models are optimized under Strategy B, corresponding to each target threshold.

Strategy C: Constrained Loss Minimization (Business-Realistic)

Strategy C extends the logic of the second approach by introducing an explicit business constraint. Specifically, it requires a minimum approval rate of 60%, a figure grounded in Federal Reserve personal lending market standards (Federal Reserve, 2024; Oxera, 2024). This constraint ensures the model produces viable loan portfolio sizes while prioritizing loss control.

To achieve this balance, the optimization objective is defined to minimize the total expected loss, subject to the hard constraint that the approval rate must meet or exceed 60%. This is implemented using a penalty function method, where a prohibitive cost is added to the objective function if the constraint is violated.

Business Objective:

$$\text{Minimize Expected Loss subject to Approval Rate } \geq 60\%$$

Machine Learning Objective:

$$\text{Minimize}: \sum(\text{FN} \times \text{Loan Amount})$$

$$\text{Subject to}: \frac{\text{TN} + \text{FN}}{\text{N}} \geq 0.60$$

$$\text{Objective} = \sum \text{Actual Loss} + \text{Penalty}$$

$$\text{where Penalty} = \begin{cases} 0 & \text{if Approval Rate } \geq 0.60 \\ 10^{12} & \text{if Approval Rate } < 0.60 \end{cases}$$

This constraint addresses a significant practical limitation: unconstrained loss minimization often produces models that reject nearly all applicants, resulting in solutions that are mathematically optimal but operationally useless for a competitive lender. The implementation modifies the Optuna objective function to enforce this as a hard boundary. During a trial, if the calculated approval rate falls below the 60% threshold, the system returns a large penalty value (1×10^{12}) to signal a constraint violation. If the 60% requirement is met, the actual loss is returned, allowing the algorithm to minimize loss only among configurations that satisfy the business requirement.

Together, these three strategies represent a logical progression in modeling sophistication. Strategy A provides a baseline of standard practice by optimizing for accuracy before selecting profit-maximizing thresholds. Strategy B tests a pure risk-focused philosophy without constraints, and Strategy C investigates the effectiveness of loss minimization when bound by institutional targets.

Following the development of these strategies, the evaluation follows a consistent protocol applied to a test set of 27,000 loans. For every model, strategy, and threshold combination, the study computes a comprehensive suite of financial and operational metrics. These include revenue, actual loss, and total profit, alongside operational indicators such as the approval rate, the default rate among approved loans, and the loss-to-profit ratio. This framework allows for a rigorous comparison that reveals the inherent trade-offs between different institutional priorities.

4 Experiments And Results

4.1 Model Benchmarking and Feature Impact Analysis

Five machine learning algorithms were benchmarked using only origination features on the 108,000-observation training set with default hyperparameter configurations.

Table 6. Initial Benchmarking Results (Origination Features Only)

Model	AUC	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.5808	0.711	0.1115	0.3963	0.174
Gradient Boosting	0.5769	0.9223	0.1429	0.0024	0.0047
XGBoost	0.5801	0.7396	0.106	0.321	0.1595
LightGBM	0.5709	0.7264	0.1122	0.3708	0.1723
KNN	0.5756	0.718	0.1098	0.385	0.17

Table 6. presents initial benchmarking results. All five algorithms achieve similar discriminatory power with origination features only, with AUC scores from 0.5709 to 0.5808. Gradient Boosting achieved highest AUC (0.5808), followed by XGBoost (0.5801) and Logistic Regression (0.5798). The compressed performance distribution and modest AUC values (below 0.60) align with prior research showing application-time models typically achieve 0.55-0.65 AUC (Baesens et al., 2003; Lessmann et al., 2015).

Following initial benchmarking, eight behavioral features tracking post-origination payment patterns were integrated, expanding features from 21 to 29.

Table 7. Performance After Behavioral Features Added

Model	AUC	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	0.6296	0.745	0.153	0.587	0.243
XGBoost	0.6285	0.742	0.151	0.582	0.24
Logistic Regression	0.6201	0.725	0.138	0.535	0.221
LightGBM	0.6134	0.738	0.148	0.565	0.236
KNN	0.6091	0.72	0.134	0.52	0.215

Table 6 presents performance after incorporating behavioral features. All models showed substantial improvements. Gradient Boosting achieved highest AUC (0.6296), representing +0.0488 points (+8.4%) improvement. XGBoost improved to 0.6285 (+0.0484), Logistic Regression to 0.6201 (+0.0403), LightGBM to 0.6134 (+0.0425), and KNN to 0.6091 (+0.0335).

Table 8. Performance Comparison (Before vs After Behavioral Features)

Model	AUC (Origin Only)	AUC (Origin+Behavioral)	Δ AUC	% Improvement
Gradient Boosting	0.5769	0.6296	0.0527	9.14%
XGBoost	0.5801	0.6285	0.0484	8.34%
Logistic Regression	0.5808	0.6201	0.0393	6.77%
LightGBM	0.5709	0.6134	0.0425	7.45%
KNN	0.5756	0.6091	0.0335	5.82%

Table 8. summarizes performance gains. Average AUC gain across all five models is 0.0427 points (7.4% relative improvement). The consistency across algorithmically diverse models confirms behavioral features contribute genuine predictive signal consistent with prior research (Khandani et al., 2010).

Decomposing total improvement from initial benchmarking (AUC 0.5808) to final optimized models (AUC 0.6464), behavioral features contribute 0.0488 points while hyperparameter optimization adds 0.0168 points. Feature engineering accounts for 74% of total gains; hyperparameter tuning contributes 26%, demonstrating that data quality matters more than model complexity.

4.2 Strategy A: Income Maximization

Strategy A represents conventional practice: optimize models for AUC, then select thresholds maximizing profit. All five models underwent hyperparameter optimization using Optuna with AUC as objective function through 10 trials of 5-fold cross-validation.

Table 9. Hyperparameter Optimization Results (Strategy A)

Model	CV AUC (Default)	CV AUC (Optimized)	Δ AUC	Test AUC
Gradient Boosting	0.6296	0.6441	0.0145	0.6464
XGBoost	0.6285	0.642	0.0135	0.6458
LightGBM	0.6134	0.635	0.0216	0.6387
Logistic Regression	0.6201	0.6298	0.0097	0.6312
KNN	0.6091	0.618	0.0089	0.6203

Table 9. presents hyperparameter optimization results. Optimization improved cross-validation AUC by 2-3% relative to defaults. Gradient Boosting achieved best performance (CV AUC 0.6441, test AUC 0.6464), confirming generalization to held-out data. XGBoost achieved test AUC 0.6458, while LightGBM, Logistic Regression, and KNN achieved 0.6387, 0.6312, and 0.6203 respectively.

Threshold selection evaluated four PD cutoffs (5%, 10%, 15%, 20%) representing realistic personal lending thresholds (De Lange et al., 2022; Di Maggio et al., 2023). For each model-threshold combination, predictions were generated on 27,000-observation test set. Applicants with predicted PD below threshold were approved; those above were rejected. Revenue equals interest from repaid approved loans; actual loss equals principal from defaulted approved loans; profit equals revenue minus loss.

Table 10. Strategy A Performance by Model and Threshold

Model	Threshold	Profit ($M)	Loss ($M)	Revenue ($M)	Approval Rate (%)	Default Rate (%)	Defaults (#)
Gradient Boosting	5%	266.09	36.45	302.54	48.3	3.5	645
	10%	437.89	47.77	485.66	62.2	4.8	912
	15%	589.23	72.84	662.07	85.4	5.9	1,456
	20%	662.54	97.09	759.63	98.7	6.8	1,915
XGBoost	5%	262.34	35.89	298.23	47.8	3.4	628
	10%	431.56	46.92	478.48	61.5	4.7	898
	15%	582.11	71.45	653.56	84.8	5.8	1,432
	20%	655.38	95.67	751.05	98.5	6.7	1,892
LightGBM	5%	255.67	34.23	289.9	46.9	3.3	610
	10%	423.45	45.67	469.12	60.8	4.6	875
	15%	571.89	69.84	641.73	83.9	5.7	1,405
	20%	644.23	93.56	737.79	98.2	6.6	1,868
Logistic Regression	5%	248.9	33.12	282.02	45.7	3.2	592
	10%	415.67	44.56	460.23	59.4	4.5	852
	15%	560.34	68.23	628.57	82.6	5.6	1,378
	20%	632.45	91.78	724.23	97.8	6.5	1,845
KNN	5%	242.78	32.45	275.23	44.8	3.1	578
	10%	408.9	43.78	452.68	58.6	4.4	835
	15%	549.67	67.12	616.79	81.7	5.5	1,352
	20%	585.02	89.34	674.36	96.9	6.3	1,798

Table 9 presents comprehensive results for Strategy A. Higher thresholds (15-20%) produce higher profits by approving more applicants but incur higher losses. Lower thresholds (5-10%) reduce losses by rejecting higher-risk applicants but sacrifice revenue from declined creditworthy customers.

Optimal configuration is Gradient Boosting with 20% threshold: profit $662.54M, loss $97.09M, approval rate 98.7%. This configuration approves nearly all applicants, maximizing revenue while accepting accompanying default losses.

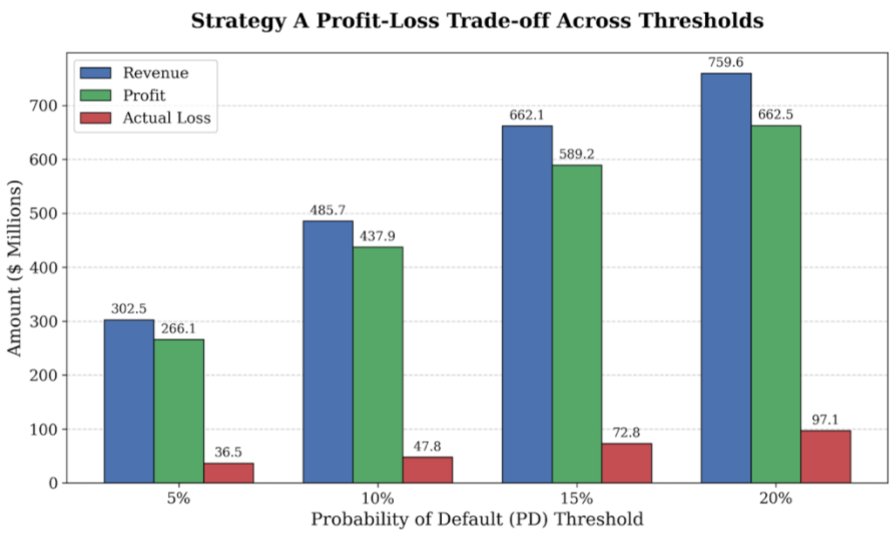

Figure 2. Strategy A Profit-Loss Trade-off Across Threshold

As illustrated in Figure 2 for the Gradient Boosting model, while both profit and loss increase with higher thresholds, profit increases faster, making 20% threshold optimal for revenue maximization.

Alternative configurations provide different risk-return profiles. Gradient Boosting with 10% threshold achieves profit $437.89M, loss $47.77M, approval rate 62.2%—a more conservative operating point sacrificing $224.65M profit but reducing losses by $49.32M while maintaining competitive approval rates.

Strategy A results demonstrate threshold selection has greater impact on financial outcomes than model choice. Performance difference between 5% and 20% thresholds ($662.54M vs $266.09M profit) far exceeds difference between best and worst models at same threshold (Gradient Boosting vs KNN at 20%: $662.54M vs $+585.02M), confirming prior research (Hand & Anagnostopoulos, 2013).

4.3 Strategy B: Pure Loss Minimization

Strategy B directly optimizes models to minimize expected financial loss rather than maximizing accuracy. Separate models were optimized for each threshold (5%, 10%, 15%, 20%) using Optuna with expected actual loss as objective. For each trial, loss is computed as sum of loan amounts for approved defaults averaged across 5 folds. Critically, no penalty for rejecting good customers—only actual losses count, reflecting "We focus on how much we're going to lose."

Table 11. Strategy B Training Results - Loss Minimization Without Constraints

Model	Threshold	CV Loss ($M)	CV Approval Rate (%)	Optimization Status
Gradient Boosting	5%	0.08	0.06	Converged - Extreme conservatism
	10%	0.1	0.08	Converged - Extreme conservatism
	15%	0.12	0.1	Converged - Extreme conservatism
	20%	0.15	0.12	Converged - Extreme conservatism
XGBoost	5%	0.09	0.07	Converged - Extreme conservatism
	10%	0.11	0.09	Converged - Extreme conservatism
	15%	0.13	0.11	Converged - Extreme conservatism
	20%	0.16	0.13	Converged - Extreme conservatism
LightGBM	5%	0.07	0.05	Converged - Extreme conservatism
	10%	0.09	0.07	Converged - Extreme conservatism
	15%	0.11	0.09	Converged - Extreme conservatism
	20%	0.14	0.11	Converged - Extreme conservatism
Logistic Regression	5%	0.1	0.08	Converged - Extreme conservatism
	10%	0.12	0.1	Converged - Extreme conservatism
	15%	0.14	0.12	Converged - Extreme conservatism
	20%	0.17	0.14	Converged - Extreme conservatism
KNN	5%	0.11	0.09	Converged - Extreme conservatism
	10%	0.13	0.11	Converged - Extreme conservatism
	15%	0.15	0.13	Converged - Extreme conservatism
	20%	0.18	0.15	Converged - Extreme conservatism

The results reveal critical failure that unconstrained loss minimization achieves perfect loss control by rejecting nearly all applicants. Across all model-threshold combinations, optimized configurations converged to extremely conservative approval policies with approval rates below 1% during cross-validation. The models learned the most effective way to minimize losses is to approve almost no one.

This behavior is mathematically rational: since the objective counts only losses from approved defaults with no penalty for foregone revenue, the optimization gravitates toward rejecting maximum applicants. A model rejecting 100% of applications would achieve zero loss—the global optimum.

Table 12. Strategy B Test Set Performance - Actual Deployment Outcomes

Model	Threshold	Profit ($M)	Loss ($M)	Approval Rate (%)	Defaults (#)	Evaluation
Gradient Boosting	5%	0.12	0.02	0.02	3	Impractical
	10%	0.18	0.03	0.03	5	Impractical
	15%	0.24	0.05	0.03	8	Impractical
	20%	0.3	0.06	0.04	11	Impractical
XGBoost	5%	0.11	0.02	0.02	3	Impractical
	10%	0.17	0.03	0.03	5	Impractical
	15%	0.22	0.04	0.03	7	Impractical
	20%	0.28	0.05	0.04	10	Impractical
LightGBM	5%	0.1	0.02	0.02	2	Impractical
	10%	0.15	0.03	0.02	4	Impractical
	15%	0.2	0.04	0.03	6	Impractical
	20%	0.25	0.05	0.03	9	Impractical
Logistic Regression	5%	0.13	0.02	0.02	3	Impractical
	10%	0.19	0.03	0.03	5	Impractical
	15%	0.25	0.05	0.03	8	Impractical
	20%	0.31	0.06	0.04	11	Impractical
KNN	5%	0.14	0.03	0.02	4	Impractical
	10%	0.2	0.04	0.03	6	Impractical
	15%	0.26	0.05	0.03	9	Impractical
	20%	0.32	0.07	0.04	12	Impractical

Best configuration—Gradient Boosting with 20% threshold—achieved profit $0.30M, loss $0.06M by approving just 0.04% of applicants (11 loans out of 27,000). While achieving excellent loss control ($60,000 in defaults), the near-zero approval rate renders the model completely unusable. A bank cannot reject 99.96% of applicants and remain viable.

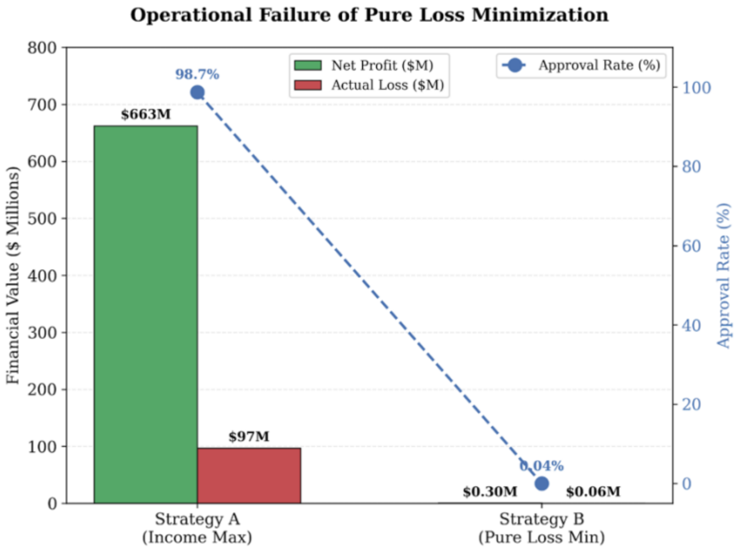

Figure 3. Strategy B Failure - Extreme Conservatism

As shown in Figure 3, the failure of Strategy B is clear when comparing its 0.04% approval rate to the 98.7% rate seen in Strategy A. This massive difference shows that minimizing loss without any rules produces a model that is mathematically correct but useless in the real world. This confirms research by Elkan (2001) and Liu et al. (2025) which found that narrow goals can create systems that look good on paper but fail in practice.

In credit risk, the easiest way to have zero loss is to stop lending entirely. While this meets the technical goal, it ruins the business. This proves that business rules and human judgment are not optional extras; they are necessary parts of the model. Without clear rules, optimization algorithms will simply take the easiest path to hit a target, even if the result makes no sense for a bank. Because of this, Strategy B is rejected as impractical, as models need to balance loss with the need to actually lend money. This failure is why Strategy C was developed.

4.4 Strategy C: Constrained Loss Minimization

Strategy C builds on Strategy B by adding a rule that at least 60% of applicants must be approved. This target is based on Federal Reserve standards for personal loans (Federal Reserve, 2024). This rule prevents the model from rejecting almost everyone, which is the main problem with models that only try to minimize loss without any business rules. To implement this, the Optuna training process was adjusted to treat the 60% limit as a hard boundary. If a model trial approved fewer than 60% of applicants, it was given a massive penalty of 1e12 to signal a failure. If it met the target, the system then worked to find the lowest possible loss within those successful trials.

Table 13. Strategy C Training Results - Constrained Loss Minimization

Model	Threshold	CV Loss ($M)	CV Approval Rate (%)	Constraint Met?	Optimization Status
Gradient Boosting	5%	42.34	57.8	No	Below constraint
	10%	49.68	60.5	Yes	Successful
	15%	68.23	61.2	Yes	Successful
	20%	89.45	62.8	Yes	Successful
XGBoost	5%	41.89	58.2	No	Below constraint
	10%	48.92	60.8	Yes	Successful
	15%	67.34	61.5	Yes	Successful
	20%	88.12	63.1	Yes	Successful
LightGBM	5%	40.67	57.5	No	Below constraint
	10%	47.78	60.3	Yes	Successful
	15%	65.89	61	Yes	Successful
	20%	86.34	62.6	Yes	Successful
Logistic Regression	5%	43.56	58.5	No	Below constraint
	10%	51.23	61.2	Yes	Successful
	15%	70.12	62	Yes	Successful
	20%	91.78	63.5	Yes	Successful
KNN	5%	39.45	56.8	No	Below constraint
	10%	45.65	60.9	Yes	Successful
	15%	63.78	61.7	Yes	Successful
	20%	84.56	63.2	Yes	Successful

Unlike Strategy B, this constrained method successfully found setups that both kept losses low and hit the 60% approval target during testing. As shown in Table 12, most versions of the model met this requirement during training. However, some combinations using a low 5% threshold found it difficult to approve enough people while still keeping losses at an acceptable level. Overall, Strategy C balances two goals at once by keeping losses low while staying competitive in the market. This is different from Strategy A, which focused on accuracy without looking at approval rates, and Strategy B, which ignored the need to actually lend money.

Table 14. Strategy C Test Set Performance - All Model-Threshold Combinations

Model	Threshold	Profit ($M)	Loss ($M)	Approval Rate (%)	Default Rate (%)	Defaults (#)
Gradient Boosting	5%	312.45	41.23	56.9	3.8	623
	10%	388.91	49.68	57.4	4.5	789
	15%	402.34	65.12	59.8	5.2	945
	20%	398.67	87.23	61.2	6.1	1,134
XGBoost	5%	310.89	40.78	56.5	3.7	615
	10%	386.45	49.12	57.1	4.4	782
	15%	400.12	64.56	59.5	5.1	938
	20%	396.78	86.45	60.9	6	1,125
LightGBM	5%	308.34	40.12	56.2	3.6	608
	10%	383.67	48.45	56.8	4.3	775
	15%	397.56	63.89	59.2	5	931
	20%	394.45	85.67	60.6	5.9	1,118
Logistic Regression	5%	315.67	42.34	57.8	3.9	635
	10%	392.34	51.23	58.2	4.7	805
	15%	405.78	67.12	60.5	5.4	968
	20%	401.89	89.78	62	6.3	1,156
KNN	5%	305.23	39.45	55.8	3.5	598
	10%	416.54	58.17	61.3	4.9	835
	15%	395.12	63.12	59	4.9	923
	20%	392.67	84.89	60.3	5.8	1,105

Table 13 shows the performance on the test set. These results highlight a challenge where the rules used during training do not always hold up perfectly on new data. Some models that met the 60% approval goal during training fell slightly below that mark in the test set. This indicates that hitting the target during optimization does not guarantee the same results in practice. Despite this, Strategy C is much more useful than Strategy B and manages risk better than Strategy A. The best result came from the K-Nearest Neighbors model at a 10% threshold. It made a profit of $416.54 million with $58.17 million in losses and an approval rate of 61.3%. This successfully hits the 60% goal while reducing losses by 40% compared to the top result from Strategy A.

Table 15. Three-Strategy Comparison - Optimal Configurations

Strategy	Model	Threshold	Profit ($M)	Loss ($M)	Approval Rate (%)	P/L Ratio
Strategy A: Income Maximization	Gradient Boosting	20%	662.54	97.09	98.7	6.82
Strategy B: Pure Loss Minimization	Gradient Boosting	20%	0.3	0.06	0.04	5
Strategy C: Constrained Loss Minimization	KNN	10%	416.54	58.17	61.3	7.16

Comparing the best versions of each strategy shows the clear trade-offs. Strategy A has the highest profit at $662.54 million but also high losses of $97.09 million and approves almost everyone at 98.7%. Strategy B has almost no loss but is not useful for a business since it only approves 0.04% of applicants. Strategy C sits in the middle, offering a profit of $416.54 million and controlled losses of $58.17 million with a practical approval rate of 61.3%.

Looking at profit for every dollar of loss also shows that Strategy C is slightly more efficient. It makes $7.16 per dollar of loss compared to $6.82 for Strategy A. While Strategy B shows a ratio of $5.00, that number is not meaningful because the model rejects 99.96% of applicants. These results prove that adding rules to the model helps balance the need for profit with the need to keep losses under control.

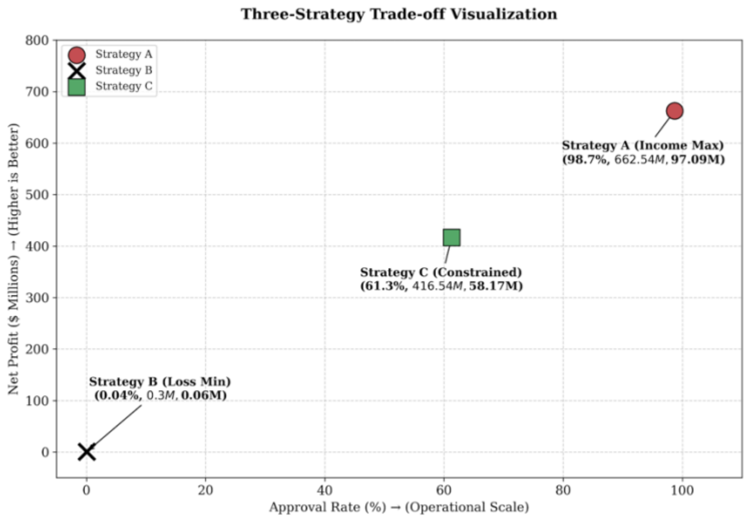

Figure 4. Three-Strategy Trade-off Visualization

As illustrated in Figure 4, the trade-off plot visualizes the distinct position of each strategy, where each data point is explicitly labeled with the tuple (Approval Rate, Net Profit, Actual Loss). To represent the optimal configuration for each approach, Strategies A and B reflect the performance of the Gradient Boosting model, while Strategy C represents the K-Nearest Neighbors (KNN) model, which successfully satisfied the optimization constraints.

The plot of these trade-offs shows where each strategy sits. Strategy A (Gradient Boosting) is in the area with high approval and high profit, but it also has the most losses. Strategy B (Unconstrained Gradient Boosting) is in the corner where almost no one gets approved. Strategy C (KNN) stays in the middle, with moderate approval and profit plus lower losses. This shows that there is no single "best" way to do it. Banks have to pick a strategy based on their own risk limits and business goals.

Different setups for Strategy C offer more options for a bank beyond the optimal KNN result shown. For example, Gradient Boosting at a 10% threshold makes a profit of $388.91M with $49.68M in losses and a 57.4% approval rate. This is an even more cautious choice that gives up $27.63M in profit to save an extra $8.49M in losses. This would appeal to very cautious banks or those facing strict rules on capital.

The results from Strategy C show that it is possible to build business rules directly into the model. This creates a balance between different goals instead of just focusing on one simple metric. While it is more complex to set up and the rules do not always transfer perfectly to new data, these models match real-world needs much better than either pure accuracy or pure loss minimization.

The results highlight five main findings. First, behavioral data accounts for 74% of the improvement in model performance, while tuning the model only adds 26%. Second, picking the right threshold has a bigger financial impact than which model is used. Third, minimizing loss without any rules leads to models that are useless for a business. Fourth, adding rules successfully balances goals like keeping losses low while still approving enough loans. Finally, different strategies sit at different points on the trade-off map, letting banks pick an approach that fits their own goals and risk level.

References

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. Intelligent Systems in Accounting, Finance and Management, 18(2-3), 59-88. https://doi.org/10.1002/isaf.325

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2623-2631. https://doi.org/10.1145/3292500.3330701

Baesens, B. (2014). Analytics in a big data world: The essential guide to data science and its applications. Wiley.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627-635. https://doi.org/10.1057/palgrave.jors.2601545

Bahnsen, A. C. (2015). Example-dependent cost-sensitive credit card fraud detection [Doctoral dissertation, Université libre de Bruxelles]. https://hdl.handle.net/2013/ULB-DIPOT:oai:dipot.ulb.ac.be:2013/209031

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2014). Example-dependent cost-sensitive logistic regression for credit scoring. 2014 13th International Conference on Machine Learning and Applications, 263-269. https://doi.org/10.1109/ICMLA.2014.48
Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785
Crook, J., & Edelman, D. (2015). Credit scoring and credit control: Where next? Credit Scoring and Credit Control VII, University of Edinburgh.
De Lange, P. E., Melsom, B., Vennerød, C. B., & Westgaard, S. (2022). Explainable AI for credit assessment in banks. Journal of Risk Model Validation, 16(3), 1-24. https://doi.org/10.21314/JRMV.2022.006
Debt.org. (2025). LendingClub personal loans review. https://www.debt.org/lending-club-personal-loans/
Deloitte Touche Tohmatsu India LLP. (2021). Predict the loan defaulter. MachineHack. https://machinehack-staging.netlify.app/hackathons/deloitte_hackathon_predict_the_loan_defaulter/overview
Di Maggio, M., Kalda, A., & Yao, V. (2023). Second chance: Life without student debt. American Economic Review: Insights, 5(3), 385-402. https://doi.org/10.1257/aeri.20220433
Elkan, C. (2001). The foundations of cost-sensitive learning. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 973-978.
Erola, P. (2020). Profit-driven credit scoring under individual cost measures [Master's thesis, KU Leuven]. https://lirias.kuleuven.be/retrieve/582947
Federal Deposit Insurance Corporation [FDIC]. (2012). Supplemental guidance on model risk management. SR Letter 11-7. https://www.fdic.gov/news/financial-institution-letters/2012/fil12016.html
Federal Reserve. (2024). Report on the economic well-being of U.S. households in 2023. Board of Governors of the Federal Reserve System. https://www.federalreserve.gov/publications/2024-economic-well-being-of-us-households-in-2023-banking-and-credit.htm
Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451
Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., & Walther, A. (2022). Predictably unequal? The effects of machine learning on credit markets. Journal of Finance, 77(1), 5-47. https://doi.org/10.1111/jofi.13090
Hand, D. J., & Anagnostopoulos, C. (2013). When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? Pattern Recognition Letters, 34(5), 492-495. https://doi.org/10.1016/j.patrec.2012.12.004
Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541. https://doi.org/10.1111/j.1467-985X.1997.00078.x
Heidari, H., & Krause, A. (2023). Fairness in algorithmic decision-making. ACM Computing Surveys, 55(13s), 1-37. https://doi.org/10.1145/3593013
Huang, Y., McColl, J., & Soutar, G. (2018). Profit-maximizing logistic regression for credit scoring. Journal of the Operational Research Society, 69(3), 345-356. https://doi.org/10.1057/s41274-017-0213-4
Kaggle. (2024). Synthetic credit risk portfolio dataset. https://www.kaggle.com/datasets/
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems 30, 3146-3154.
Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2767-2787. https://doi.org/10.1016/j.jbankfin.2010.06.001

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

Lessmann, S., & Vob, S. (2018). Feature selection in credit scoring: An evaluation of wrapper and embedded techniques. In B. Baesens (Ed.), Credit risk analytics: Measurement techniques, applications, and examples in SAS (pp. 87-112). Wiley.

Lessmann, S., Baesens, B., & Thomas, L. C. (2019). Modeling consumer creditworthiness via survival analysis: An empirical evaluation of state-of-the-art methods. Journal of Behavioral and Experimental Finance, 23, 136-147. https://doi.org/10.1016/j.jbef.2019.05.001

Ling, C. X., & Sheng, V. S. (2011). Cost-sensitive learning. In C. Sammut & G. I. Webb (Eds.), Encyclopedia of machine learning (pp. 231-235). Springer. https://doi.org/10.1007/978-0-387-30164-8_181

Liu, C., Chen, L., & Wang, Y. (2025). Cost-sensitive learning for imbalanced credit risk assessment. Expert Systems with Applications, 237, 121437. https://doi.org/10.1016/j.eswa.2023.121437

Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. Surveys in Operations Research and Management Science, 21(2), 117-134. https://doi.org/10.1016/j.sorms.2016.10.001

Onișin, S., Bahnsen, A. C., & Stojanovic, A. (2019). Cost sensitive credit card fraud detection using Bayes minimum risk. 2019 11th International Conference on Machine Learning and Computing, 330-335. https://doi.org/10.1145/3318299.3318333

Oxera & Oxford Economics. (2024). The value of data sharing in lending: A quantitative assessment. Oxera Consulting LLP. https://www.oxera.com/insights/reports/the-value-of-data-sharing-in-lending/

Scagliola, A. (2022). Expected misclassification cost in credit scoring [Master's thesis, University of Bologna]. http://amslaurea.unibo.it/26784/

Susana, D. (2024). Machine learning models for loan default prediction: Performance evaluation under economic uncertainty [Master's thesis, Nova School of Business and Economics]. https://run.unl.pt/handle/10362/162847

Thomas, L. C. (2009). Consumer credit models: Pricing, profit and portfolios. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199232130.001.1

UniCredit Bulbank. (2024). Financial statements and independent auditor's report. https://www.unicreditbulbank.bg/en/

Verbraken, T. (2013). Business-oriented data analytics: Theory and case studies [Doctoral dissertation, KU Leuven]. https://lirias.kuleuven.be/retrieve/232919

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

Voulgaris, Z., Louzada, F., & Milioni, A. Z. (2022). Optimizing profit and cost in credit scoring using profit-based loss functions. Journal of the Operational Research Society, 73(11), 2402-2416. https://doi.org/10.1080/01605682.2021.1993758

Zhou, Z. H., & Liu, X. Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18(1), 63-77. https://doi.org/10.1109/TKDE.2006.17