Interpretable Natural Language Concept Bottleneck Models for Default Risk estimation

Andrea Conti* Giacomo Morelli†

July 30, 2026

Abstract

The estimation of the probability of default is a core competence of banks and lending financial institutions. A central challenge in probability of default models is the trade-off between predictive performance and explainability because traditional models are easily interpretable, whereas complex models achieve better forecasting performance at the cost of transparency. We leverage Natural Language Processing (NLP) in two complementary ways: (i) we build an inherently interpretable framework that translates the decision logic into human-readable language and (ii) we incorporate textual data of borrowers in a simple way that blends seamlessly into the framework. We illustrate the strength of our model on a large portfolio of retail loans, showing that it simultaneously improves the estimation accuracy and enhances the explainability.

Keywords: Concept Bottleneck Model, Credit Scoring, Natural Language Processing.

JEL classification: C38, C45, D83, G21.

*Department of Social and Economic Sciences, Sapienza University of Rome, 00185 Rome, Italy E-mail: andrea.conti@uniroma1.it.

†Corresponding author: Department of Statistical Sciences, Sapienza University of Rome, 00185 Rome, Italy. E-mail: giacomo.morelli@uniroma1.it

1 Introduction

Models that forecast the probability of default are crucial to minimize risk exposure and ensure the quality of credit outstanding. Probability of default models measure the probability borrower will default on a credit loan, and financial institutions use these models for assessing and quantifying the risk associated with loan applicants (Blöchlinger and Leippold, 2006). Beyond the estimation of the probability of default, these models serve operational purposes as providing informed credit approval processes and regulatory compliance with interpretable decisions to internal stakeholders, supervisors, and customers. In this setup, explainability is a fundamental requirement of credit scoring models that supports forecasts, but also transparent and operationally reliable decision making.

In credit scoring, the volume of available data makes algorithmic methods essential for predicting default risk, as algorithmic decision making yields higher returns and lower classification error (Jansen et al., 2025). However, most machine-learning models are opaque which conflicts with credit scoring regulations that require transparent and explainable decisions (EBA, 2020; EC, 2020).

Traditional frameworks like Logit (Wiginton, 1980; Deo and Juneja, 2021), Linear Discriminant Analysis (Taffler, 1982), and Decision Trees (Khandani et al., 2010) are common for their inherent interpretability. The rise of high-performance approaches such as Bagging, Boosting (Paleologo et al., 2010; Kündig and Sigrist, 2025) and Deep Learning (Lahmiri and Bekiros, 2019; Gunnarsson et al., 2021) has introduced a trade-off because they yield better performance at the cost of transparency, necessitating specific approaches that allow for the interpretation of the results. This is central in banking and lending activity, where institutions must estimate the probability of default for each borrower to support core practices, such as credit approval, while ensuring that model decisions remain interpretable to satisfy regulatory requirements. In addition, the performance of these complex models strongly depends on the chosen set of hyperparameters selected. This contributes to additional opacity because it is often difficult to justify selecting one configuration over another as it yields better performance. Moreover, hyperparameters impose an additional and practical constraint that, for business purposes, frequently switching configurations to chase marginal gains is rarely optimal, because it undermines the consistency and comparability of results. Accordingly, probability of default models must yield strong predictive accuracy but also sufficient explainability to operational lending decisions.

An additional factor that hinders explainability in credit scoring models is the nature of the applicant financial information, as it is numerical, categorical, and textual. In fact, the techniques required to process textual data (the input text collected from borrowers during the application process) rely on models and algorithms that introduce additional opacity. For example, common approaches to Natural Language Processing involve the use of Encoders, Deep Learning architectures, or Large Language Models (Mai et al., 2019; Kriebel and Stitz, 2022; Xia et al., 2025; Wu et al., 2025), that are black-box models, do not provide guarantees on fairness (Hurlin et al., 2026), and it is not clear how the features and specific bits of textual data affect the decision process because the link between unstructured textual inputs and final predictions is often difficult to trace and interpret (Stevenson et al., 2021; Korangi et al., 2023). This is a crucial gap. Existing methods that leverage textual data are not able to link the textual information with default risk drivers, even if it improves performance.

In general, to address the interpretability gap introduced by complex models, there exist specific frameworks such as post-hoc explanation methods and models that are inherently interpretable. Specifically, post-hoc explanation methods such as rule extraction (Martens et al., 2007) derive decision rules from complex models, while Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) (Chen et al., 2024; Ballegeer et al., 2025) quantify feature contributions to model predictions. This class of methods can be further distinguished into local explanations, which clarify individual predictions, and global explanations, which summarize overall model behavior. Inherently interpretable approaches, by contrast, rely on transparent model families such as linear models (Dumitrescu et al., 2022; Li et al., 2022; Glasserman and Li, 2025), decision trees (Tu and Wu, 2025), or generalized additive models whose structure is intended to be understandable by design. However, a key gap remains because post-hoc explanations can be unstable and may not faithfully reflect the model true decision process, while intrinsically interpretable models often require sacrificing predictive performance or expressive power, limiting their applicability. Furthermore, post-hoc explanation methods are often not enough because they do not explain the reasons and the process that lead a model to learn a particular decision function. Rather, they explain the output as a sum of contributions by features, leaving unclear how these contributions are formed and preferred to alternative representations.

The purpose of this work is to fill the aforementioned gap. We introduce a new interpretable predictive modeling framework that strikes a balance between performance and explainability, providing a flexible class of models that are inherently explainable based on human language and

that include textual data with a clear link to the decision-making process.

First, we propose an interpretable framework based on Concept Bottleneck Model (CBM) for credit risk prediction. In order to illustrate the framework, we nest it in a Logistic Regression for the default risk estimation of retail borrowers in an explainable setting. This approach allows us to obtain an inherently explainable framework that is flexible enough to accommodate the use of other models, and we illustrate how to extend it to Neural Networks. Crucially, by grounding predictions in a structured CBM framework, we mitigate the limitations of post-hoc explanation methods by relying instead on a model that is interpretable by design. Specifically, this approach is inherently explainable because it works by first defining or extracting human-readable rules and then mapping them to default risk. The crucial idea of the CBM is that explanations are not post-hoc. The model is forced to reason through the bottleneck that are human-readable explanations. As a consequence, each output is a combination of explanations, and it is easy to connect each output to human-readable rules/explanations. The CBM provides a flexible, modular framework in which the input rules are human-defined, making the reasoning process transparent while remaining readily extensible to alternative predictive models as well as to alternative scoring rules.

Second, we integrate textual borrower information into interpretable risk factors within the CBM framework. In our framework, the use of textual data (the borrower input text) is integrated seamlessly into the model, and it is possible to directly link risk drivers from textual data to default risk in a way that is clear and explainable manner. Importantly, the same human-readable risk drivers defined in the CBM can be defined to capture and explain signals in the borrower text, allowing textual information to be fully exploited while the link to default risk remains transparent and interpretable. With this approach, we allow a broader representation of potential risk drivers from textual data, rather than collapsing all textual information into a single variable. Overall, textual data is crucial not only because it provides additional information about borrowers characteristics, but also because the CBM framework leverages natural language to guide the model reasoning in a human-readable and interpretable way.

Third, we empirically evaluate the framework on a large retail loan dataset and show improvements in both predictive accuracy and interpretability. To achieve this, we compare the performance of the CBM framework, including Logistic Regression and Neural Network, with respect to classical Logistic Regression and Decision Tree. In addition, within the CBM framework, we compare model performance with and without access to risk drivers extracted from textual

data. Overall, the framework we introduce is a novel approach to credit scoring so that banks, financial, and lending institutions can benefit in the support of operational decisions by producing default probabilities together with human-readable reasons. The model makes interpretability easy to validate because its predictions are compositions of human-readable rules. This allows us to check whether the explanations are adequate for the decision process, and experts can use their domain knowledge to define and integrate new scoring rules into the model. The same rules are easy to audit because it is possible to observe if the same borrower receives stable explanations across similar cases. In addition, the framework and idea of language bottleneck can be extended to other machine learning algorithms and models. This has important managerial value because understandable explanations can increase trust in automated credit scoring among managers and regulators.

The rest of the paper is organized as follows. In Section 2, we introduce the CBM framework, the differences, and benefits with respect to post-hoc explanation methods. In Section 3, we outline the CBM theoretical framework, illustrate how we apply it using a Logistic Regression model, and extend it to Neural Networks. We discuss in Section 4 the dataset and the preprocessing, while Section 5 presents the empirical analysis where we compare the performance of the CBM framework with respect to other intrinsically explainable models.

2 Concept Language Bottleneck for Explainable models

In this section, we illustrate the framework. Our goal is to build a credit scoring model that is explainable by construction. The model produces a prediction together with an explanation that is (i) human-readable, (ii) directly connected to the decision logic so that it is possible to clearly audit the decision logic.

The estimation of the probability of default is not only a predictive task but also a decision process in which fundamental requirement are clarity and the understanding of what supports any given decision. In practice, explanations need to be:

- Actionable: explanations must link the rationale to the main default risk drivers.
- Stable and consistent: similar applicants should receive similar explanations, and small differences in borrowers feature should not change the outcome.
- Auditable: the reasoning should be traceable and decision process must agree with each

specific firm policy. This allows firms and institutions to understand the rationale that led the model to produce any given forecast.

- Communicable: explanations should be clear to borrowers and lenders.

Our framework is inspired by the notion of a bottleneck representation. Probability of default forecasts must pass through a restricted representation (bottleneck) of rules and decision functions that are designed to be interpretable. The bottleneck results inherently interpretable because the rules and decision functions are framed in human language. The key idea is that the model is not allowed to base its prediction on arbitrary latent representations but instead, its decision process and forecast must pass through concepts (the bottleneck) that are auditable, can be inspected, named, and explained in plain language. This design makes the decision process more transparent, because each intermediate component has a semantic meaning, such as "low income", "high debt burden", or "GDP at originiation time is low". As a result, the final prediction is a composition of simple human-readable rules, rather than opaque feature interactions learned by complex models.

The model consists in two steps:

1. **Rationale generation (explanation first).** The model takes as input default risk drivers that are also human-readable rules (such as "high utilization", "recent delinquencies", "short credit history", "stable income relative to obligations"). The rules are mapped to the borrower on a numerical scale, indicating how relevant that rule is for the borrower.

2. **Scoring from the rationale (decision from explanation).** A second model maps the score of the rule to a default probability, and classifies each borrower as high or low default risk.

The intrinsic design of the model is one of the benefit of the approach. In fact, the model works through the explanations and in order to produce any forecast the model leverages the rules of the language bottleneck. In turn, these rules are human-readable and are explainable by construction. Overall, any model forecasts is a composition of explainable and human readable rules, without possibility to deviate from the same rules defined in the language bottleneck, and this is what makes the approach explainable-by-design, rather than post-hoc. We illustrate in Algorithm 1 the steps of the CBM framework.

Algorithm 1: Credit Scoring Model with a Concept Language Bottleneck

Input: Loan application features x; rationale template $\mathcal{T}$, a function f_θ that computes the relevance between rationale and borrowers features, and a final scoring model g.

Output: Default probability $\hat{p}$ and rationale z

Step 1: Construct the language bottleneck

Fill a structured rationale using $\mathcal{T}$: $z \leftarrow f_\theta(\mathcal{X})$

Step 2: Map the language bottleneck to a risk score

Compute risk score $\hat{p} \leftarrow g(z)$

return $(\hat{p}, z)$

3 Theoretical Framework

We discuss the human-readable decision rules that the CBM f leverages to produce explanations and predictions. Overall, to illustrate how the framework operates and generates explanations, we develop two different approaches. The first is a CBM based on a two-step logistic regression. The second is a CBM that performs both steps simultaneously through a single loss function, and we implement this approach by testing both logistic regression and a neural network.

3.1 Human-readable decision rules

Let the dataset be a set of rows $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where x_i denote the features (such as income and age) of borrower i and $y_i \in \{0, 1\}$ is the binary target that indicate the default event. Let the training set be $\mathcal{D}_{\text{tr}}$ and test set $\mathcal{D}_{\text{te}}$. A concept rule (or decision function) is indexed by $k \in \{1, \ldots, K\}$, and each rule k is a function

$$r_k : \mathcal{X} \to \{0, 1, \perp\},$$

where $\perp$ denotes the action of abstention. For any instance x, evaluating rule k returns a label $r_k(x) \in \{0, 1, \perp\}$ which we use for explanation within the bottleneck. Within our framework, we define different rule and decision functions, depending for the nature of borrower features.

Consider a numeric borrower feature c, with value v, then for each c we define the training sample for these features as

$$S_c = \{v = x_i[c] : x_i \in \mathcal{D}_{\text{tr}}\},$$

then, the first class of rule that we define is a decision function that uses a three-way outcome based on an uncertainty band around a threshold t. Given numeric column c and threshold t, we define

$$r_{c,t}^{\text{high}}(x) = \begin{cases} \perp, & \text{if } v \text{ is missing,} \\ 1, & \text{if } v \geq t + m_c, \\ 0, & \text{if } v \leq t - m_c, \\ \perp, & \text{if } t - m_c < v < t + m_c, \end{cases}$$

where t is quantile function Q at level q, which is a threshold we define as $t_{c,q} = \mathrm{Q}_q(S_c)$, and m_c is the interquantile ratio (IQR) using $\alpha = 0.1$

$$m_c = \alpha \cdot \mathrm{IQR}(S_c) = \alpha \cdot \big(\mathrm{Q}_{0.75}(S_c) - \mathrm{Q}_{0.25}(S_c)\big).$$

As an example, the rule yield "confident high" if the feature is above $t + m_c$, "confident not-high" if it is below $t - m_c$, and it abstains in any other case. Similarly we can define other bottleneck rationale by changing t and link it to a "low" state

$$r_{c,t}^{\text{low}}(x) = \begin{cases} \perp, & \text{if } v \text{ is missing,} \\ 1, & \text{if } v \leq t - m_c, \\ 0, & \text{if } v \geq t + m_c, \\ \perp, & \text{otherwise.} \end{cases}$$

For categorical borrower feature, we define a strict AND operator on tri-state labels

$$\mathrm{AND}_{\text{strict}}(\ell_1, \ldots, \ell_m) = \begin{cases} 0, & \text{if } \exists j \text{ such that } \ell_j = 0, \\ 1, & \text{if } \forall j,\ \ell_j = 1, \\ \perp, & \text{otherwise.} \end{cases}$$

We then define composite interaction terms as

$$r(x) = \mathrm{AND}_{\text{strict}}\big(r_a(x),\, r_b(x),\, \ldots\big),$$

thereby mixing numeric rules, which may abstain, with hard rules. Regarding textual data, or a set of regex patterns $P = \{p_1, \ldots, p_M\}$, we define

$$\mathrm{match}_P(x) = \bigvee_{j=1}^{M} \mathbf{1}\{\text{regex } p_j = (x[\text{text}])\}\,,$$

and then positive-only text rule is

$$r_P^{\text{text}+}(x) = \begin{cases} 1, & \text{if match}_P(x) = 1, \\ \perp, & \text{otherwise.} \end{cases}$$

Then, the bottleneck is the representations of rule based concepts that link the mapping from the features of borrowers to prediction. For any borrower with features $x \in \mathcal{X}$, we define the bottleneck as the composition of human-readable rules

$$\phi(x) = \big(r_1(x), \ldots, r_K(x)\big) \in \{0, 1, \perp\}^K,$$

where each component corresponds to the output of one concept token, which in turn is a human-readable rule. The prediction function is then a two step approach in which $\hat{y} = g(\phi(x))$ is estimated from the bottleneck, rather than as a direct function of borrower features x. Therefore, with this framework, the model can only base its decision on this explicit, human-interpretable bottleneck representation. In Table 1, we report a sample of three human-readable rules that the model uses to score borrowers into low and high-risk of default. The entire model reasoning passes through this language bottleneck of human-readable rules.

Table 1: Examples of human-readable rules for the language bottleneck

	Explanation/Rule
1	**Low income and high debt burden:** the borrower presents low income together with a high debt-to-income ratio.
2	**Financial distress in the description:** the borrower presents textual signals of financial distress in the loan description.
3	**Revolving credit stress:** the borrower presents both high revolving utilization and high revolving balance.

For completeness, in Appendix A, we report the full set of human-readable rules used in the language bottleneck. In addition to these rules, we enhance the bottleneck of rules with 50 auto-mined risk rules and 30 auto-mined protective rules. These terms are mined from a shallow tree trained on $\mathcal{D}_{\text{tr}}$ after imputing numeric missing values with a median only on the train sample. We

define for a generic numeric feature c

$$\tilde{x}[c] = \begin{cases} x[c], & \text{if finite,} \\ \text{median}(S_c), & \text{if missing,} \\ 0, & \text{otherwise.} \end{cases}$$

A mined rule is a conjunction of conditions,

$$C = \{(f_j, \text{op}_j, \tau_j)\}_{j=1}^{d}, \qquad \text{op}_j \in \{\leq, >\},$$

and its rule is evaluated using the same imputation:

$$r_C^{\text{auto}}(x) = \mathbf{1}\left\{ \bigwedge_{j=1}^{d} \left(\tilde{x}[f_j] \ \text{op}_j \ \tau_j\right) \right\}.$$

Hence, $r_C^{\text{auto}}(x)$ always returns 0 or 1, never $\perp$. The rule selection is then based on the training data evidence (support) and conditional bad rate. Specifically,

$$\text{supp}(C) = \Pr_{\mathcal{D}_{\text{tr}}}\left[r_C^{\text{auto}}(x) = 1\right], \qquad \text{prec}(C) = \Pr\left[y = 1 \mid r_C^{\text{auto}}(x) = 1\right].$$

We the compare these quantities against

$$\pi = \Pr_{\mathcal{D}_{\text{tr}}}[y = 1].$$

We keep "risk" rules when $\text{prec}(C) \geq \pi + \delta$ and "protective" rules when $\text{prec}(C) \leq \pi - \delta$.

3.2 Approach 1 - Two Stage LR-CBM

The first approach we consider is a CBM framework built on a two-step logistic regression.

In the first step, we construct the CBM bottleneck by estimating the weight of each rule, for each borrower. Specifically, for each concept k, we train a classifier

$$f_k : \mathbb{R}^d \to [0, 1], \qquad z_k(x) = f_k(\mathcal{X}_{-S_k}) = \mathbb{P}\big(r_k(x) = 1 \mid \mathcal{X}_{-S_k}\big),$$

where $\mathcal{X}_{-S_k}$ is the borrower information excluding the variable(s) that define rule k, and each f_k is an L1-regularized logistic regression

$$f_k(u) = \sigma(\alpha_k + \beta_k^\top u), \qquad \sigma(t) = \frac{1}{1 + e^{-t}}.$$

Since some rules may abstain, we estimate the binary concept model for rule k only on the non-abstaining observations with a weighted logistic loss and L1 penalty (concept loss L_c)

$$\min_{\alpha,\beta} L_y = \min_{\alpha_k,\beta_k} \sum_{\mathcal{D}_{tr}^{(k)}} w_{ik}\Big(- \, r_k(x_i) \log f_k(x_{i,-S_k}) - \, (1 - r_k(x_i)) \log\big(1 - f_k(x_{i,-S_k})\big)\Big) \; + \; \lambda_k \|\beta_k\|_1,$$

where $\mathcal{D}_{\text{tr}}^{(k)} = \{i \in \mathcal{D}_{\text{tr}} \mid r_k(x_i) \in \{0,1\}\}$. We use balanced weights w_{ik} that give more importance to the class with less observations for each concept. The bottleneck, after estimation, is the vector of predicted concept probabilities

$$z(x) = \big(z_1(x), \ldots, z_K(x)\big)^\top \in [0,1]^K.$$

In the second step we score the loans by using only the concept of the bottleneck. To achieve this, we train a second model that predicts default only from the bottleneck $z(x)$:

$$g : [0,1]^K \to [0,1], \qquad \hat{p} = g(z(x)) = \mathbb{P}(y = 1 \mid z(x)).$$

Our scorer is a second logistic regression, with L1 regularization

$$g(z) = \sigma\big(b + w^\top z\big), \qquad w \in \mathbb{R}^K, \; b \in \mathbb{R}.$$

so that we minimize the label loss L_y

$$\min_{b,w} L_y = \min_{b,w} \sum_{i \in \mathcal{D}_{tr}} \Big(- \, y_i \log g\big(z(x_i)\big) - \, (1 - y_i) \log\big(1 - g(z(x_i))\big)\Big) \; + \; \lambda \|w\|_p,$$

and the final decision threshold is $\hat{y}(x) = \mathbf{1}[\hat{p} \geq 0.5]$.

Overall, the model is a composition of two learned maps. The first map constructs the concept bottleneck

$$x \longmapsto z(x) = (f_1(\mathcal{X}_{-S_1}), \ldots, f_K(\mathcal{X}_{-S_K}))^\top,$$

where each component is

$$f_K(\mathcal{X}_{-S_K}) = \sigma\Big(\alpha_k + \beta_k^\top x_{-S_k}\Big).$$

The second map predicts default from the bottleneck

$$z(x) \longmapsto \widehat{p} = \sigma\Big(b + w^\top z(x)\Big).$$

Thus, the full model can be written as

$$\widehat{p} = \sigma\left(b + \sum_{k=1}^{K} w_k \sigma\Big(\alpha_k + \beta_k^\top x_{-S_k}\Big)\right).$$

3.3 Approach 2 - Joint CBM

We propose a second approach to obtain a more general form of the concept bottleneck that would the development of further deep learning architectures. Differently from the two-stage LR-CBM in Section 3.2, the model does not first estimate concept probabilities ($x \to z$) and then fit an additional scoring model ($z \to \hat{p}$).

In this case, the model learns the concept probabilities and the default prediction function through a single objective. We achieve this by using a single objective function in which there are two loss terms, namely the concept loss (L_c) and the label loss (L_y).

To make our framework more general and include deep learning approaches, we define a concept representation map that maps borrower data into a representation that we use to estimate the concept probability

$$h_{\theta,k} : \mathcal{X}_{-S_k} \to \mathbb{R}^{H_k}, \qquad h_{\theta,k}(x_{-S_k}) \in \mathbb{R}^{H_k}.$$

We consider two alternative specifications

1. *Linear/logistic CBM*: $h_{\theta,k}(x_{-S_k}) = x_{-S_k}$.
2. *Neural-network CBM*: $h_{\theta,k}$ is an L-layer feedforward neural network with ReLU activations.

As a stylized overview, we illustrate with the following scheme the logic of the two approaches.

Approach	**Map**	**Prediction**	**Objective**
Two-step CBM	$x_{-S} \longrightarrow z$	$z = \mathbb{P}(r(x) = 1 \mid x_{-S})$	Estimate z from x_{-S_k} using L_c,
	$z \longrightarrow \hat{p}$	$\hat{p} = \mathbb{P}(y = 1 \mid z)$	then estimate $\hat{p}$ from z using L_y
Joint CBM	$x_{-S} \longrightarrow z \longrightarrow \hat{p}$	$z = \mathbb{P}(r(x) = 1 \mid h_\theta(x_{-S}))$	estimate jointly $L = L_y + \lambda_c L_c$
		$\hat{p} = \mathbb{P}(y = 1 \mid z)$	

Within the joint approach, the model learns in a semi-supervised way, using the available rule based concept information together with direct borrower default/non-default information, and the concept loss plays the role of a regularizer.

As in the two-stage LR-CBM, in Section 3.2, the model that predicts concept probabilities k does not receive the variables that directly define the rule. Therefore, concept k is inferred from x_{-S_k} that is, from the borrower data excluding the variable(s) S_k directly related to the.

Regarding the concept loss, we first define the concept logit

$$\ell_{\theta,k}^{(c)}(x_i) = a_k + v_k^\top h_{\theta,k}(x_{i,-S_k}),$$

where $v_k \in \mathbb{R}^{H_k}$ and $a_k \in \mathbb{R}$. The corresponding concept probability is

$$z_{\theta,k}(x_i) = \sigma\Big(\ell_{\theta,k}^{(c)}(x_i)\Big) = \sigma\Big(a_k + v_k^\top h_{\theta,k}(x_{i,-S_k})\Big)\,,$$

which we interpret as the probability, given borrower data, that the model infer a concept is applicable to a given borrower

$$z_{\theta,k}(x_i) = \mathbb{P}(r_k(x_i) = 1 \mid x_{i,-S_k})\,.$$

The bottleneck representation is then

$$z_\theta(x_i) = (z_{\theta,1}(x_i), \ldots, z_{\theta,K}(x_i))^\top \in (0,1)^K.$$

Since some rules may abstain, similarly to the two-step approach, we evaluate concept based only on non-abstaining observations. So, for each concept k let $\mathcal{D}_{\text{tr}}^{(k)} = \{i \in \mathcal{D}_{\text{tr}} \,:\, r_k(x_i) \in \{0,1\}\}$, and the observation mask $o_{ik} = \mathbf{1}\{r_k(x_i) \in \{0,1\}\}$.

The concept loss aligns the learned bottleneck with the human-readable rule labels. For each concept k, define the number of non-abstaining positive and negative labels in the training set as

$$n_k^+ = \sum_{i \in \mathcal{D}_{\text{tr}}} o_{ik}\, r_k(x_i), \qquad n_k^- = \sum_{i \in \mathcal{D}_{\text{tr}}} o_{ik}\,\big(1 - r_k(x_i)\big).$$

To reduce impact of imbalance between concepts of different classes we use the weights

$$\omega_k = \text{clip}\left(\frac{n_k^-}{n_k^+ + \varepsilon}, 1, 50\right),$$

where $\varepsilon > 0$ is used for numerical stability. Concepts that are constant among non-abstaining observations do not provide a learnable supervised signal. We therefore define the variability mask

$$m_k = \mathbf{1}\left\{n_k^+ > 0 \text{ and } n_k^- > 0\right\} \in \{0,1\}.$$

For each non abstaining observation-concept pair, the weighted concept loss is

$$\mathcal{L}_{c,k}^{(i)} = -\left[\omega_k\, r_k(x_i) \log z_{\theta,k}(x_i) + \big(1 - r_k(x_i)\big) \log(1 - z_{\theta,k}(x_i))\right].$$

The masked batch concept loss is

$$\mathcal{L}_c = \frac{\sum_{i\in\mathcal{D}_{tr}} \sum_{k=1}^{K} m_k\, o_{ik}\, \mathcal{L}_{c,k}^{(i)}}{\sum_{i\in\mathcal{D}_{tr}} \sum_{k=1}^{K} m_k\, o_{ik}}.$$

Regarding the label loss, we build the model so that the final scoring model receives only the bottleneck vector $z_\theta(x_i)$, and we define the label logit as

$$\ell^{(y)}(x_i) = b + w^\top z_\theta(x_i), \qquad w \in \mathbb{R}^K,\ b \in \mathbb{R},$$

and the predicted probability of default is

$$\widehat{p}_i = g(z_\theta(x_i)) = \sigma\Big(\ell^{(y)}(x_i)\Big) = \sigma\Big(b + w^\top z_\theta(x_i)\Big).$$

Thus, the borrower data x_i enters the prediction only through the estimated concept probabilities $z_\theta(x_i)$. Then, the loss for each observation is

$$\mathcal{L}_y^{(i)} = -\left[y_i \log \widehat{p}_i + (1-y_i)\log(1-\widehat{p}_i)\right].$$

In the end, the joint CBM is estimated by minimizing

$$\mathcal{L}(B) = \mathcal{L}_y(B) + \lambda_c\, \mathcal{L}_c(B).$$

Therefore, the joint CBM can be written as the composition

$$x \longmapsto (x_{-S_1}, \ldots, x_{-S_K}) \longmapsto z_\theta(x) \longmapsto \widehat{p},$$

where

$$\widehat{p} = \sigma\left(b + \sum_{k=1}^{K} w_k \sigma\Big(a_k + v_k^\top h_{\theta,k}(x_{-S_k})\Big)\right).$$

In the linear/logistic case, this becomes

$$\widehat{p} = \sigma\left(b + \sum_{k=1}^{K} w_k \sigma\Big(a_k + v_k^\top x_{-S_k}\Big)\right).$$

In the neural-network case, the same bottleneck structure is preserved, but each concept probability is obtained through a more flexible representation $h_{\theta,k}(x_{-S_k})$:

$$\widehat{p} = \sigma\left(b + \sum_{k=1}^{K} w_k \sigma\Big(a_k + v_k^\top h_{\theta,k}(x_{-S_k})\Big)\right).$$

The intermediate vector $z_\theta(x)$ provides the concept-level explanation of the prediction because each component $z_{\theta,k}(x)$ is aligned, through $\mathcal{L}_c$, with the corresponding human-readable rule r_k, while the final score is computed only from these concept probabilities.

3.4 Faithfulness

Within the CBM framework, model forecasts are interpretable only if the prediction is mediated by the bottleneck representation used to explain the decision. We refer to this property as faithfulness.

Let $B_\theta(x)$ denote the complete bottleneck representation passed to the final prediction model. In the baseline binary-concept case,

$$B_\theta(x) = z_\theta(x) = \big(z_{1,\theta}(x), \ldots, z_{K,\theta}(x)\big)^\top,$$

where

$$z_{k,\theta}(x) = \widehat{\Pr}(r_k(x) = 1 \mid \mathcal{X}_{-S_k}) \,.$$

The requirements to obtain a faithful model are

C1 The final scoring model g receives only the bottleneck representation $B_\theta(x)$ as input. It does not receive the original borrower features x or any of the preprocessed features $\varphi(x)$. Formally,

$$\widehat{p} = g\big(B_\theta(x)\big),$$

and not

$$\widehat{p} = g\big(B_\theta(x), x\big) \quad \text{or} \quad \widehat{p} = g\big(B_\theta(x), h_\theta(x)\big).$$

C2 The same bottleneck representation $B_\theta(x)$ must be used during training and inference phase.

C3 The final classification rule must depend only on the bottleneck probability:

$$\widehat{y}(x) = \mathbf{1}\{g(B_\theta(x)) \geq \tau\},$$

where τ is fixed threshold that does not depend on borrower data.

C4 The components of the bottleneck must correspond to declared human-readable concepts. In our case, in the CBM framework, each component $z_{k,\theta}(x)$ corresponds to the predicted activation probability of a concept rule r_k:

$$z_{k,\theta}(x) = \widehat{\Pr}(r_k(x) = 1 \mid \mathcal{X}_{-S_k}) \,,$$

and each component has an explicit semantic interpretation.

Proposition 3.1 (Faithfulness of the concept bottleneck)**.** *Suppose Conditions C1-C3 hold. Then the CBM prediction is fully mediated by the bottleneck representation $B_\theta(x)$. That is, for any two borrowers $x, x' \in \mathcal{X}$,*

$$B_\theta(x) = B_\theta(x') \implies \widehat{p}(x) = \widehat{p}(x').$$

If Condition C4 also holds, then the prediction is fully mediated by a human-readable concept bottleneck.

Proof. By Condition C1, the prediction has the form

$$\widehat{p}(x) = g_\psi\big(B_\theta(x)\big).$$

Take any two borrowers x and x' such that

$$B_\theta(x) = B_\theta(x').$$

Then

$$\widehat{p}(x) = g\big(B_\theta(x)\big) = g\big(B_\theta(x')\big) = \widehat{p}(x').$$

Therefore, the interpretation is that once the bottleneck representation is fixed, the original borrower features contain no additional information for the fitted prediction. In other words, we say that the model-implied conditional prediction satisfies

$$\Pr(\widehat{Y} = 1 \mid X = x, B = b) = g(b),$$

which does not depend on x. Hence, at the level of the fitted model we have that $\widehat{Y} \perp X \mid B$. □

In our framework, for the two-stage LR-CBM, faithfulness follows by construction because the second-stage logistic regression uses only the predicted concept probabilities

$$\widehat{p}(x) = \sigma\big(b + w^\top z(x)\big).$$

Similarly, for the joint CBM, faithfulness is satisfied because the label logit depends only on the concept representation

$$\ell_i^{(y)} = w_y^\top z_i + b_y.$$

In contrast, our approach would no longer be faithful if the model is no longer fully mediated by the concept bottleneck. An example is a situation in which the final layer concatenates the concept vector with the hidden representation $h_i = g_\theta(x_i)$

$$\ell_i^{(y)} = w^\top [z_i, h_i] + b.$$

Here part of the prediction can be driven by non-concept information, and the reported concept explanation is not fully faithful to the estimated model.

4 Data

We collect data on peer-to-peer loans facilitated by Lending Club, which is a major lending institution that allows retail borrowers and investors to perform lending on an online platform. For each loan, the borrower must make payments according to the schedule, and if payments are late, default occurs 90 days after the first recorded late payment. Overall, the objective is to distinguish between high and low-risk borrowers, which is a supervised classification problem where the binary target variable indicates the default event. The dataset contains more than 100 thousand observations of loans issued in a period that spans from 2007 to 2014, and it is very rich with both categorical and numerical features such as the age of the borrower, income, information that denotes if the borrower rents or owns the house of his address, macroeconomic data such as the Consumer Price Index (CPI) and the Gross Domestic Product (GDP) in the United States at origination date, as well as other relevant financial and personal features. Most importantly, it provides information about loan performance, including whether the loan is fully paid or the borrower incurred in financial distress and ultimately defaults. In addition, the dataset contains borrower input text collected during the application process. The dataset is publicly available in the UCI Library[1].

4.1 Preprocessing

We split the dataset into train and test sets. Training is done on an undersampled/balanced train set, and evaluation is on the original test distribution. This approach ensures that the model learns features from both default and non-default observations. In fact, a model that learns from data that consists predominantly of one class, during inference, yields poor performance on the minority class. Since an imbalance between the proportion of default and non-default events is a typical

[1] https://archive.ics.uci.edu/

characteristic of credit scoring datasets, addressing it is a crucial step because the Lending Club dataset has an imbalance of 79% non-default and 21% default. In addition, to avoid leakage, we first split the dataset into training and test set, and then select relevant variables with chi-square test and F-test.

Our raw tabular input x contains both continuous variables and one-hot encoded categorical features, so we apply a deterministic preprocessing map $\varphi : \mathcal{X} \to \mathbb{R}^d$ before model training. For continuous columns, we impute missing values with the median computed on the training set, and then we standardize the same numerical variable. For binary and one-hot encoded columns, we instead apply most-frequent imputation and do not perform any scaling, to preserve the original interpretability of these indicators.

Regarding textual data, Let d_i denote the input text of borrower i, and let $\tilde{d}_i$ be the clean version of the text. We map the cleaned corpus into a count matrix $X^{\text{text}} \in \mathbb{N}^{n \times M}$ over a vocabulary $V = \{v_1, \ldots, v_M\}$ of unigrams and bigrams, where X_{ij}^{text} is the number of occurrences of phrase v_j in $\tilde{d}_i$. Then we score each phrase through its association with the target, using both a chi-square statistic and the contrast $\text{lift}_j = P(v_j \text{ present} \mid y = 1) - P(v_j \text{ present} \mid y = 0)$. In the end, we convert relevant bits of borrower text into positive-only textual rules $r_{ik} \in \{1, \text{NaN}\}$ through regex phrase matching on $\tilde{d}_i$. For example, if the expression "debt consolidation" is common, we assign a positive outlook to it, because it reduces debt, and flag each borrower that manifests the intent to do debt consolidation. Separately, we use the same text and transform it into a TF–IDF vector $x_i^{\text{text}} \in \mathbb{R}^p$, which is not used to define rules, but only as input to models that estimate the probability of activation of each concept.

5 Empirical Analysis

We test our framework and compare the performance versus other inherently explainable models such as Logistic Regression and Decision Tree. Then, we introduce the additional variable of the user input text and the associated rule for the bottleneck, to study their role in highlighting default risk drivers.

In Table 2, we report the results of the first analysis where CBM LR1 denotes a simple CBM model with base model a logistic regression, and Simple CBM is a concept bottleneck model with a Logistic Regression in which we use just simple human-defined rules without auto-mined rules. In contrast, CBM LR2 denotes the CBM logistic regression model with the training procedure (the

two step) is joint and concur as two different terms in the loss function, and CBM NN denotes the Neural Network CBM model. In the latter case, the extension is making the base model deep with multiple layers, compared to a logistic regression by using the joint inference procedure of Section 3.3. We use a CBM NN with 1 hidden layer and 30 neurons, while using a learning rate of 0.001 for all models.

Table 2: Results of the empirical analysis among CBM models.

	Simple CBM	CBM LR1	CBM LR2	CBM NN
Accuracy	0.678	0.667	0.635	0.638
B. Acc.	0.651	0.656	0.672	0.681
Precision	0.306	0.305	0.299	0.305
Recall	0.621	0.648	0.731	0.748
F1	0.410	0.414	0.429	0.433

In Table 3, we illustrate an example of a borrower scored with the CBM LR2 and a rich bottleneck. The borrower eventually incurred in financial distress and defaulted, which is the correct forecast of the CBM model as it assigned a probability of 75.8% to the default event of the same borrower. The benefit of our model is the immediate human-readable default risk drivers and rules that lead to a model forecast, allowing for a preciseness and explainability. The CBM model identifies as relevant default risk drivers for this borrower the advanced age of the loan, the borrower's low income, the high loan-to-income ratio, the borrower's employment length, and the timing of the latest credit inquiry.

Table 3: Example of borrower scoring with human-readable rules sorted by importance. The borrower incurred in default and the CBM forecasted probability of default = 75.8%)

	Explanation/Rule
1	The loan of the borrower has an age of 219 months (confident not-low)
2	The income of the borrower is confident low (48000<52000)
3	The borrower has an employment length of 10 years (confident not-low)
4	The borrower ratio loan to income is confident high (0.44>0.30)

We complement the empirical analysis by reporting, in Table 4, the results of simple benchmark models on the same dataset. We observe that the Multilayer Perceptron, besides the CBM models,

outperforms all other benchmarks, which instead shows mixed performance across the different performance measures.

Table 4: Model Benchmarking between alternative benchmark models.

	Logistic R.	Decision Tree	Linear SVM	LDA	Naive Bayes	k-NN	MLP
Accuracy	0.671	0.615	0.674	0.671	0.660	0.679	0.672
B. Acc.	0.656	0.611	0.655	0.651	0.616	0.643	0.667
Precision	0.309	0.263	0.309	0.306	0.282	0.307	0.314
Recall	0.632	0.605	0.625	0.619	0.547	0.586	0.658
F1	0.415	0.367	0.414	0.410	0.372	0.403	0.425

While the CBM models still outperforms all benchmark in Table 4, our approach still balances explainability, and their direct connection with models outputs, rather than performance. For this reason, we present in Appendix C an extension of our CBM approach to improve performance even further with bagging and boosting techniques.

In Table 5, we report the results of the empirical analysis on the same CBM models, but with the addition of textual data, so the language bottleneck is enriched with additional human-readable rules extracted from borrowers input text.

Table 5: Model Benchmarking of CBM models with the addition of human-readable rules extracted from textual data.

	Simple CBM	CBM LR1	CBM LR2	CBM NN
Accuracy	0.671	0.663	0.721	0.665
Balanced Acc.	0.652	0.656	0.683	0.685
Precision	0.307	0.305	0.354	0.318
Recall	0.627	0.645	0.621	0.717
F1	0.410	0.414	0.451	0.441

The addition of the user input text and associated human-readable rules extracted from it does not seem to bring a relevant impact. Nonetheless, we illustrate how text signals are useful in some cases and for some borrowers. In fact, in Table 6, we report a borrower that did not experienced default event.

Table 6: Example of borrower scoring with human-readable rules sorted by importance. The borrower did not incurred in default and the CBM forecasted probability of default = 27.6%)

Input text:	"This loan is to payoff a high-interest credit card."
	Explanation/Rule
1	The GDP of the United States, at origination date, is 17500 (confident-high)
2	The borrower indicates the intent of debt consolidation (positive outlook)
3	The borrower ratio debt to income is confident high (26.9>18.4)
4	The loan is 156 months old (confident-low)
5	The borrower has income of 42000$ (confident-low)

Although the structured borrower features suggest a relatively weak applicant profile, the textual information provides an important positive signal. From the tabular features, we observe several risk-related characteristics, such as low income and a high debt-to-income ratio, which would ordinarily indicate a higher likelihood of default and might even have led to the rejection of the loan application. However, the input text states that the purpose of the loan is to pay off a high-interest credit card, which the model interprets as an intention to consolidate debt. This is a favorable signal, as it may reflect a borrower's effort to improve their financial situation. In this case, that textual information helps shift the model toward a forecast of non-default, which is indeed the correct prediction. More generally, textual information is useful for identifying risky applicants: borrowers have little incentive to explicitly reveal negative information about themselves, so when such signals do appear in the text, they may be especially informative. By contrast, explicitly positive self-descriptions are generally less reliable, since they may simply reflect strategic self-presentation.

6 Operational decision and economic evaluation

In credit scoring, the model is ultimately used to support an operational decision, assessing whether a loan application should be accepted or rejected. Therefore, predictive accuracy alone is not sufficient. A model may have good classification performance but still lead to poor lending decisions if its probability estimates are poorly calibrated, if it approves too many high-loss borrowers, or if it rejects too many profitable borrowers. To address this issue, we extend the empirical analysis by evaluating each model as a credit allocation policy. Each model produces a predicted probability

of default for each borrower. This predicted probability is then translated into an accept/reject decision using an expected-profit rule. We then compare the economic consequences of the lending policies induced by the competing models.

This addition is particularly important for our framework because the CBM does not only produce a probability of default. It also provides the human-readable concepts that justify the decision. Therefore, the model can be evaluated both as a predictive model and as an explainable operational system.

Consider a lender that receives a set of loan applications indexed by $i = 1, \ldots, n$. For each applicant, the lender observes a vector of borrower and loan characteristics x_i at the time of origination. These variables include borrower income, debt-to-income ratio, employment length, credit history, loan amount, interest rate, loan term, and, when available, borrower-provided textual information.

Let $y_i \in \{0, 1\}$ denote the realized default outcome, where $y_i = 1$ if borrower i defaults and $y_i = 0$ otherwise, and the predicted probability of default for loan i is

$$\widehat{PD}_i = \widehat{p}_(x_i).$$

For the CBM models, this prediction is mediated by the concept bottleneck. That is, the model first maps the borrower into human-readable concepts and then predicts default from those concepts only. The lender must choose a binary decision

$$a_i = \begin{cases} 1, & \text{if loan } i \text{ is accepted,} \\ 0, & \text{if loan } i \text{ is rejected.} \end{cases}$$

The goal is not only to minimize classification error, but to maximize the economic value of the lending portfolio. For this simulation setup, we assume

A1 Loss given default is assumed to be constant across borrowers.

A2 The lender takes the loan amount, contractual interest rate, installment, and term as given. The model is used only to decide whether to accept or reject the loan.

For accepted loans, if borrower i does not default, the lender receives the contractual repayment according to schedule and we define the overall revenue as

$$G_i(c) = \text{installment}_i \times \text{term}_i - EAD_i - c \times EAD_i \times \frac{\text{term}_i}{12},$$

and this is the total contractual payment minus the funded amount, while c is the annual cost of funds. Then, the expected profit from accepting loan i under model m is:

$$\widehat{\Pi}_{im} = (1 - \widehat{PD}_{im})G_i - \widehat{PD}_{im} LGD_i EAD_i.$$

The first term is the expected gain if the borrower does not default, while the second term is the expected loss if the borrower defaults.

For the binary approval decisions $a_i \in \{0, 1\}$ for each loan i, the approval is then a score-based portfolio selection problem

$$a^\star(s, \mathcal{A}) \in \arg\max_{a \in \{0,1\}^n} \sum_{i=1}^{n} a_i s_i \qquad \text{s.t. } a \in \mathcal{A},$$

where s_i is the score used to rank loans and $\mathcal{A}$ defines the admissible set of approval decisions. We define two different approval policies, the expected-profit policy in which loans are acept if the expected profit is greater than zero, and the fixed approval-rate policy in which K% loans are accepted by selecting them among the lowest estimated probability of default.

Under the expected-profit policy, we set $s_i = \widehat{\Pi}_i$ and impose no approval-size constraint. The resulting solution is

$$a_i^{EP} = \mathbf{1}\{\widehat{\Pi}_i \geq 0\},$$

so the bank approves all loans with non-negative expected profit. This decision rule can also be written as a borrower-specific PD threshold:

$$a_i = \mathbf{1}\left[\widehat{PD}_i \leq \frac{G_i}{G_i + LGD_i EAD_i}\right].$$

This highlights that the acceptance threshold is not necessarily the same for every borrower. A borrower with a high contractual return may be acceptable even with a higher PD, whereas a borrower with a low contractual return may need a much lower PD to be profitable.

Under the fixed approval-rate policy, for a target approval rate q, we set $K_q = \lceil qn \rceil$ and $s_i = -\widehat{p}_i$, so that

$$a^{(q)} \in \arg\max_{a \in \{0,1\}^n} \sum_{i=1}^{n} a_i(-\widehat{p}_i) \qquad \text{s.t. } \sum_{i=1}^{n} a_i = K_q,$$

and this yields the approval of the K_q loans with the lowest predicted PD.

After the model makes accept/reject decisions, we evaluate the realized profit on the test set using the observed default outcome y_i. The realized profit is then

$$\Pi = \sum_{i=1}^{n} a_i \left[(1 - y_i) G_i - y_i LGD_i EAD_i \right].$$

This measures the actual economic value of the lending policy on unseen data. A model is better if it produces a higher realized portfolio profit, lower realized losses, and a better risk-return trade-off.

Among the CBM specifications, Table 7 shows that the expected-profit approval rule generates different balancing between lending volume and realized profitability. The joint LR-CBM achieves the highest realized profit within the CBM family, equal to $28.2 million, while maintaining a high approval rate of 84.5%. The joint NN-CBM is more selective, approving 70.6% of applications, but delivers the lowest realized default rate, equal to 12.1%, and the highest profit per unit of accepted exposure, equal to 10.8%. By contrast, the two-stage specifications remain more volume-oriented, with approval rates above 86%, but they generate lower realized profits and weaker profit efficiency.

Table 7: Operational portfolio evaluation of CBM models under the expected-profit approval rule

Model	**Accepted**	**Approval (%)**	**Default (%)**	**Avg. PD (%)**	**Exposure ($m)**	**Realized loss ($m)**	**Realized profit ($m)**	**Profit/Exposure (%)**
Simple CBM	24,087	86.9	16.1	15.7	316.6	33.6	25.1	7.9
CBM LR1	23,964	86.4	15.9	15.4	312.9	33.0	24.9	8.0
CBM LR2	23,433	84.5	14.6	12.2	308.7	30.6	28.2	9.1
CBM NN	19,576	70.6	12.1	11.2	254.6	21.3	27.6	10.8

Notes: The table reports operational portfolio performance of the CBM specifications under the expected-profit approval rule. A loan is accepted whenever its model-implied expected profit is non-negative. "Default" is the realized default rate among accepted loans, while "Avg. PD" is the average predicted probability of default among accepted loans. Monetary values are reported in millions of dollars.

The fixed approval-rate analysis in Table 8 further confirms the strength of the joint CBM design as the joint CBM-LR and CBM-NN deliver the highest realized profit among the fixed approval-rate policies.

Table 8: Realized portfolio profit of CBM models under fixed approval-rate policies

Model	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Simple CBM	2.4	4.9	7.2	9.4	11.5	14.2	16.8	19.5	22.4
CBM LR1	2.0	3.9	6.5	9.5	12.2	15.2	18.0	20.2	22.6
CBM LR2	2.6	5.6	9.2	12.3	15.7	18.6	21.5	23.8	25.7
CBM NN	3.4	7.2	10.9	14.5	17.8	21.0	23.9	26.4	27.2

Notes: The table reports realized portfolio profit when each CBM model is constrained to approve a fixed percentage of loan applications. Columns correspond to the approval-rate grid from 10% to 90%. Values are reported in millions of dollars.

Similarly for benchmark models, we report in Table 9 the results of the positive expected-profit policy and in Table 10 the results for the fixed approval threshold policy. Overall, we observe that the Multilayer Perceptron outperforms other benchmark models under both policies, while still not reaching the performance of the CBM models.

Table 9: Operational portfolio evaluation of simple benchmark models under the expected-profit approval rule

Model	**Accepted**	**Approval (%)**	**Default (%)**	**Avg. PD (%)**	**Exposure ($m)**	**Realized loss ($m)**	**Realized profit ($m)**	**Profit/Exposure (%)**
Logistic R.	24,705	89.1	16.0	15.8	320.7	33.5	25.5	7.9
Decision Tree	15,983	57.6	13.0	8.2	199.0	17.1	16.9	8.5
Linear SVM	25,299	91.2	19.1	17.9	337.7	42.1	23.6	7.0
LDA	24,776	89.4	16.0	15.7	321.7	33.7	25.3	7.9
Naive Bayes	18,948	68.3	14.0	6.5	191.8	15.6	12.7	6.6
k-NN	23,841	86.0	16.4	14.3	310.2	33.6	24.0	7.7
MLP	23,124	83.4	14.9	14.6	301.8	30.0	26.3	8.7

Notes: The table reports out-of-sample operational portfolio performance of the simple benchmark models under the expected-profit approval rule. A loan is accepted whenever its model-implied expected profit is non-negative. "Default" is the realized default rate among accepted loans, while "Avg. PD" is the average predicted probability of default among accepted loans. Monetary values are reported in millions of dollars.

Table 10: Realized portfolio profit of simple benchmark models under fixed approval-rate policies

Model	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Logistic R.	2.1	4.3	6.5	9.0	11.8	14.4	17.2	19.9	22.3
Decision Tree	3.0	6.1	8.9	11.7	14.6	17.3	19.2	20.8	22.1
Linear SVM	2.1	4.2	6.5	9.0	11.8	14.4	17.1	19.7	22.3
LDA	2.0	4.1	6.2	8.9	11.6	14.1	16.7	19.7	21.8
Naive Bayes	1.8	3.2	4.9	6.7	8.5	10.6	12.8	15.6	19.6
k-NN	2.2	4.8	7.5	10.2	12.9	14.8	17.5	20.2	22.5
MLP	2.2	4.5	7.0	9.8	12.7	15.9	19.0	21.5	23.6

Notes: The table reports realized portfolio profit when each simple benchmark model is constrained to approve a fixed percentage of loan applications. Columns correspond to the approval-rate grid from 10% to 90%. Values are reported in millions of dollars.

In this experiment, we test the two policies using the standard assumption of LGD= 60% and $c = 0.01$. In Appendix D.1, we test different setting such as LGD= $(30\%, 75\%)$ together with $c = (0.02, 0.05)$. Additionally, in Appendix D.2 we test the performance of frontier machine learning algorithms including bagging and boosting approaches, evaluating the out of sample performance as well as their performance under the positive expected profit and fixed threshold approval policies.

References

Ballegeer, M., M. Bogaert, and D. F. Benoit (2025). Evaluating the stability of model explanations in instance-dependent cost-sensitive credit scoring. *European Journal of Operational Research 326*(3), 630–640.

Blöchlinger, A. and M. Leippold (2006). Economic benefit of powerful credit scoring. *Journal of Banking & Finance 30*(3), 851–873.

Chen, Y., R. Calabrese, and B. Martin-Barragan (2024). Interpretable machine learning for imbalanced credit scoring datasets. *European Journal of Operational Research 312*(1), 357–372.

Deo, A. and S. Juneja (2021). Credit risk: Simple closed-form approximate maximum likelihood estimator. *Operations Research 69*(2), 361–379.

Dumitrescu, E., S. Hué, C. Hurlin, and S. Tokpavi (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research 297*(3), 1178–1192.

EBA (2020). Report on big data and advanced analytics. Tech. rep., European Banking Authority.

EC (2020). White paper on artificial intelligence: A european approach to excellence and trust. Tech. rep., European Commission.

Glasserman, P. and M. Li (2025). Linear classifiers under infinite imbalance. *Operations Research 73*(2), 1075–1101.

Gunnarsson, B. R., S. Vanden Broucke, B. Baesens, M. Óskarsdóttir, and W. Lemahieu (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research 295*(1), 292–305.

Hurlin, C., C. Pérignon, and S. Saurin (2026). The fairness of credit scoring models. *Management Science 72*(1), 406–425.

Jansen, M., H. Q. Nguyen, and A. Shams (2025). Rise of the machines: The impact of automated underwriting. *Management Science 71*(2), 955–975.

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

Korangi, K., C. Mues, and C. Bravo (2023). A transformer-based model for default prediction in mid-cap corporate markets. *European Journal of Operational Research 308*(1), 306–320.

Kriebel, J. and L. Stitz (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.

Kündig, P. and F. Sigrist (2025). A spatio-temporal machine learning model for mortgage credit risk: Default probabilities and loan portfolios. *European Journal of Operational Research*.

Lahmiri, S. and S. Bekiros (2019). Can machine learning approaches predict corporate bankruptcy? evidence from a qualitative experimental design. *Quantitative Finance 19*(9), 1569–1577.

Li, W., F. Paraschiv, and G. Sermpinis (2022). A data-driven explainable case-based reasoning approach for financial risk detection. *Quantitative Finance 22*(12), 2257–2274.

Mai, F., S. Tian, C. Lee, and L. Ma (2019). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research 274*(2), 743–758.

Martens, D., B. Baesens, T. Van Gestel, and J. Vanthienen (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research 183*(3), 1466–1476.

Paleologo, G., A. Elisseeff, and G. Antonini (2010). Subagging for credit scoring models. *European Journal of Operational Research 201*(2), 490–499.

Stevenson, M., C. Mues, and C. Bravo (2021). The value of text for small business default prediction: A deep learning approach. *European Journal of Operational Research 295*(2), 758–771.

Taffler, R. J. (1982). Forecasting company failure in the uk using discriminant analysis and financial ratio data. *Journal of the Royal Statistical Society: Series A (General) 145*(3), 342–358.

Tu, J. and Z. Wu (2025). Inherently interpretable machine learning for credit scoring: Optimal classification tree with hyperplane splits. *European Journal of Operational Research 322*(2), 647–664.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis 15*(3), 757–770.

Wu, Z., Y. Dong, Y. Li, and B. Shi (2025). Unleashing the power of text for credit default prediction: Comparing human-written and generative ai-refined texts. *European Journal of Operational Research 326*(3), 691–706.

Xia, Y., Z. Han, Y. Li, and L. He (2025). Credit scoring model for fintech lending: An integration of large language models and focalpoly loss. *International Journal of Forecasting 41*(3), 894–919.

A Supplemental Material

Table 11: Lending Club dataset selected variables.

Feature	Type	Description
acc_now_delinq	Integer	The number of credit lines where the borrower is delinquent.
annual_inc	Real	The self-reported annual income of the borrower at the time of registration.
CPI	Real	Quarterly growth rate of the Consumer Price Index (CPI) of the country where the loan originated.
credit_age_months	Integer	Number of months since the borrower was issued their first loan.
delinq_2yrs	Integer	Number of times the borrower was late in the payment schedule for more than 30 days, in the last 2 years.
dti	Real	Ratio of the borrower total monthly debt payments, excluding mortgages, over the self-reported monthly income.
emp_length	Integer	Employment length of the borrower in years.
funded_amnt	Real	The total committed amount for the loan at that time.
funded_amnt_inv	Real	The total committed amount for the loan at that time by investors.
GDP	Real	Real and seasonally-adjusted Gross Domestic Product (GDP) of the country where the loan originated.
home_ownership	Categorical	Indicates the type of home ownership of the borrower: Own, Mortgage, Rent, None, Other and Any.
installment	Real	Monthly payment the borrower is required to render if the loan originates.
int_rate	Real	Represents the interest rate on the loan.
inq_last_6mths	Integer	Indicates the number of time the borrower received a credit inquiry in the last 6 months.
loan_amnt	Real	Indicates the listed loan amount requested by the borrower.
mths_since_last_delinq	Integer	Reports the number of months since the borrower last inquiry.
open_acc	Integer	Number of active credit lines of the borrower.
pub_rec	Integer	The number of public records flagged as derogatory.
revol_util	Real	Revolving line utilization rate.
term	Integer	Number of monthly payments of the loan.
total_acc	Integer	Total number of active and expired credit lines of the borrower.
verification_status	Categorical	Indicates if the self-reported monthly income of the borrower is: verified, source verified or not verified.

Table 12: Examples of human-readable rules used in the language bottleneck (1–19)

	Explanation/Rule
1	**High interest rate:** the borrower presents a clearly high loan interest rate relative to the training distribution.
2	**Very high interest rate:** the borrower presents an extremely high loan interest rate compared with typical borrowers in the training data.
3	**High debt-to-income ratio:** the borrower presents a high debt-to-income ratio indicating a heavy repayment burden.
4	**Very high debt-to-income ratio:** the borrower presents an extremely high debt-to-income ratio compared with most borrowers.
5	**High revolving credit utilization:** the borrower presents a high fraction of available revolving credit already used.
6	**Very high revolving credit utilization:** the borrower presents extremely high utilization of revolving credit lines.
7	**High revolving balance:** the borrower presents a large outstanding balance on revolving credit accounts.
8	**High loan amount:** the borrower presents a loan request that is high relative to the distribution observed in the training data.
9	**High loan-to-income ratio:** the borrower presents a loan amount that is large compared with their reported annual income.
10	**High installment burden:** the borrower presents a monthly installment that is large relative to their monthly income.
11	**Low income:** the borrower presents a clearly low annual income compared with the training population.
12	**Very low income:** the borrower presents extremely low annual income relative to the training data distribution.
13	**Short credit history:** the borrower presents a short credit history measured in months since the earliest credit line.
14	**Very short credit history:** the borrower presents extremely limited credit history.
15	**Recent credit inquiries:** the borrower presents a very recent credit inquiry indicating possible credit-seeking behavior.
16	**High number of recent inquiries:** the borrower presents multiple credit inquiries within the last six months.
17	**Employment instability:** the borrower presents a short employment length relative to typical borrowers.
18	**Long loan term:** the borrower presents a loan with a long repayment term compared with other loans in the dataset.
19	**Low income and high debt burden:** the borrower presents low income together with a high debt-to-income ratio.

Table 13: Examples of human-readable rules used in the language bottleneck (20–38)

	Explanation/Rule
20	**Revolving credit stress:** the borrower presents both high revolving utilization and high revolving balance simultaneously.
21	**High rate and long maturity:** the borrower presents both a high interest rate and a long loan term.
22	**Thin credit file with recent inquiry:** the borrower presents a short credit history and a very recent credit inquiry.
23	**Multiple derogatory signals:** the borrower presents at least two among delinquency history, public records, or current delinquency.
24	**Low income renter:** the borrower presents both rental housing status and low annual income.
25	**Missing credit history information:** the borrower presents missing information about past delinquencies or credit history length.
26	**Missing income information:** the borrower presents missing or unavailable annual income data.
27	**Not verified income:** the borrower presents income that has not been verified by the platform.
28	**Verified income:** the borrower presents income that has been verified by the lender or data source.
29	**Home ownership − rent:** the borrower presents a housing status indicating that they rent their residence.
30	**Home ownership − own:** the borrower presents a housing status indicating that they fully own their residence.
31	**Home ownership − mortgage:** the borrower presents a housing status indicating that the residence is financed by a mortgage.
32	**Financial distress in the description:** the borrower presents textual signals of financial distress in the loan description.
33	**Job or income shock mentioned:** the borrower presents textual signals of unemployment or job loss in the loan description.
34	**Medical expenses mentioned:** the borrower presents textual signals related to medical costs or hospital expenses.
35	**Education purpose mentioned:** the borrower presents textual signals indicating education-related borrowing.
36	**Debt consolidation mentioned:** the borrower presents textual signals indicating that the loan will consolidate existing debt.
37	**Empty or very short description:** the borrower presents little or no textual explanation in the loan description.
38	**Accumulation of multiple risk signals:** the borrower presents several risk conditions simultaneously among interest rate, debt burden, utilization, and credit inquiries.

B Bottleneck Explainability

The proposed CBM framework is interpretable by construction because the prediction is produced from the concept bottleneck rather than directly from the original borrower features. For each borrower i, the model maps the application data into a vector of concept activations

$$z_i = (z_{i1}, \ldots, z_{iK}) \in [0,1]^K,$$

where each component corresponds to a human-readable credit-risk concept. The final predicted probability of default is

$$\widehat{p}_i = \sigma\left(b + \sum_{k=1}^{K} w_k z_{ik}\right).$$

Therefore, the concepts are not post-hoc explanations but rather risk driver that are easy to recover and sort them by importance based on w_k.

We report an analysis in Table 14, focusing on metrics and measures that denote the different levels of explainability of alternative CBMs.

First, we analyze the row median active concepts per borrower that measures the typical size of the borrower-level explanation. We define an active concept as

$$a_{ik} = \mathbf{1}\left\{z_{ik} \geq \tau_z \text{ and } |w_k| > \varepsilon_w\right\},$$

and the number of active concepts for borrower i as

$$S_i = \sum_{k=1}^{K} a_{ik}.$$

We report the median of S_i across borrowers which captures explanation sparsity so that a lower values correspond to more concise explanations.

Second, we analyze the rows Mean PD change after top-concept ablation and 90th percentile PD change after top-concept ablation. This measures whether the most important concept in the local explanation is relevant from a decision making point of view. For each borrower, we identify the concept with the largest absolute logit contribution,

$$k_i^\star = \arg\max_k |w_k z_{ik}|.$$

We then replace $z_{ik_i^\star}$ with its training-set mean $\bar{z}_{k_i^\star}$, recompute the predicted probability of default, and measure

$$A_i = \left|\widehat{p}_i - \widehat{p}_i^{(-k_i^\star)}\right|.$$

The table reports both the mean of A_i and its 90th percentile. These quantities indicate how much the predicted probability changes when the leading explanatory concept is neutralized.

Third, the row Mean top-q explanation overlap among nearest neighbours measures local explanation stability. Let nearest neighbours be defined in bottleneck space by

$$d(i,j) = \|z_i - z_j\|_2,$$

and let $\mathcal{P}$ denote the set of nearest-neighbour pairs. Here the overlap is measured by the Jaccard index, defined for two sets A and B as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

For each borrower i, define $\mathcal{E}_i^{(q)}$ as the set of the q concepts with the largest absolute contributions $|w_k z_{ik}|$. The overlap diagnostic is

$$\mathcal{J}^{(q)} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} J\left(\mathcal{E}_i^{(q)}, \mathcal{E}_j^{(q)}\right) = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \frac{|\mathcal{E}_i^{(q)} \cap \mathcal{E}_j^{(q)}|}{|\mathcal{E}_i^{(q)} \cup \mathcal{E}_j^{(q)}|}.$$

Higher values indicate that similar borrowers receive more similar explanations.

Finally, we also analyze the row Mean PD difference among nearest neighbours that measures prediction stability in the same bottleneck neighbourhoods

$$\Delta^{PD} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} |\widehat{p}_i - \widehat{p}_j|.$$

This diagnostic checks whether borrowers that are close in concept space also receive similar predicted probabilities of default.

Table 14: Compact explanation across CBM specifications.

Diagnostic	**Simple CBM**	**CBM LR1**	**CBM LR2**	**CBM NN**
Median active concepts per borrower	9	15	13	13
Mean PD change after top-concept ablation	7.3%	6.5%	12.9%	12.7%
90th percentile PD change after top-concept ablation	14.6%	12.2%	24.2%	23.9%
Mean top-q explanation overlap among nearest neighbours	0.635	0.530	0.584	0.543
Mean PD difference among nearest neighbours	6.2%	7.4%	16.1%	16.6%

Overall, the audit indicates that the CBM explanations are available in a compact form and are materially connected to the predicted probability of default. The Simple CBM specification is the most sparse and locally stable, while the joint CBM specifications display stronger ablation

effects, suggesting that their leading concepts have larger influence on the final score, at the cost of higher local variation in predicted probabilities.

C Bagged and Boosted CBM

The two CBM approaches can be extended in a unified way by keeping the same concept bottleneck structure and modifying only the way the final concept-mediated prediction is aggregated. In both cases, the final prediction depends on borrower information only through the concept vector $z_\theta(x)$. Therefore, the model remains of the form

$$x \longmapsto z_\theta(x) \longmapsto \widehat{p}.$$

For bagging, we estimate B CBM models on bootstrap samples of the training data. Each model produces a concept-mediated prediction

$$\widehat{p}^{(b)}(x) = g_{(b)}(z_{\theta^{(b)}}(x)), \qquad b = 1, \ldots, B.$$

The bagged CBM prediction is then obtained by averaging the predicted probabilities

$$\widehat{p}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} g_{(b)}(z_{\theta^{(b)}}(x)).$$

For boosting, instead, we keep the learned concept representation $z_\theta(x)$ as the only input to the final scorer and replace the linear bottleneck scorer with an additive model over concepts

$$F_M(z_\theta(x)) = \gamma_0 + \sum_{m=1}^{M} \nu h_m(z_\theta(x)),$$

where h_m are individual CBMs defined only on the concept vector and ν is the learning rate. The boosted CBM prediction is

$$\widehat{p}_{\text{boost}}(x) = \sigma(F_M(z_\theta(x))).$$

We apply boosting only after the bottleneck and the final decision rule remains

$$\widehat{y}(x) = \mathbf{1}[\widehat{p}(x) \geq 0.5].$$

Table 15: Bagged and Boosted Simple CBM.

	Bagged CBM	Boosted CBM
Accuracy	0.663	0.669
B. Acc.	0.661	0.650
Precision	0.307	0.305
Recall	0.657	0.620
F1	0.418	0.409

D Extended Empirical Analysis

D.1 Operational decision and economic evaluation details

Table 16: Operational robustness of CBM models across LGD and cost-of-funds assumptions

Model	**LGD (%)**	**Cost (%)**	**Accepted**	**Approval (%)**	**Default (%)**	**Profit ($m)**	**Profit/Exposure (%)**
Simple CBM	30	2	27,157	97.9	18.2	35.7	10.0
CBM LR1	30	2	27,092	97.7	18.0	35.5	10.0
CBM LR2	30	2	26,299	94.8	17.0	36.2	10.4
CBM NN	30	2	23,112	83.4	14.7	34.4	11.3
Simple CBM	30	5	14,616	52.7	20.8	10.4	4.9
CBM LR1	30	5	14,132	51.0	20.5	10.6	5.3
CBM LR2 joint LR	30	5	15,166	54.7	17.8	11.9	5.6
CBM NN	30	5	12,315	44.4	15.0	11.4	6.6
Simple CBM	75	2	16,325	58.9	14.5	9.8	4.4
CBM LR1	75	2	16,950	61.1	13.8	10.2	4.5
CBM LR2	75	2	19,348	69.8	12.5	14.2	5.6
CBM NN	75	2	15,565	56.1	10.7	14.6	7.2
Simple CBM	75	5	2,225	8.0	15.4	0.9	2.4
CBM LR1	75	5	2,239	8.1	15.7	1.0	2.7
CBM LR2	75	5	6,956	25.1	12.1	1.6	1.7
CBM NN	75	5	6,256	22.6	10.1	2.8	3.2

Notes: The table reports the operational performance of each CBM specification under alternative assumptions on loss given default and cost of funds. The approval rule accepts a loan whenever its model-implied expected profit is non-negative under the corresponding scenario. Monetary values are reported in millions of dollars.

Table 17: Operational robustness of simple benchmark models across LGD and cost-of-funds assumptions

Model	LGD (%)	Cost (%)	Accepted	Approval (%)	Default (%)	Profit ($m)	Profit/Exposure (%)
Logistic R.	30	2	27,227	98.2	18.2	35.6	10.0
Decision Tree	30	2	15,983	57.6	13.0	19.9	10.0
Linear SVM	30	2	26,909	97.0	18.7	35.7	10.0
LDA	30	2	27,322	98.5	18.1	35.6	9.9
Naive Bayes	30	2	20,573	74.2	14.7	17.7	8.2
k-NN	30	2	26,679	96.2	17.9	35.2	10.0
MLP	30	2	26,529	95.7	17.6	35.4	10.1
Logistic R.	30	5	14,796	53.4	20.8	10.4	5.0
Decision Tree	30	5	11,580	41.8	15.4	5.4	3.6
Linear SVM	30	5	14,679	52.9	24.4	10.0	4.7
LDA	30	5	14,851	53.6	20.7	10.5	5.0
Naive Bayes	30	5	12,453	44.9	16.8	2.7	2.3
k-NN	30	5	14,803	53.4	20.7	10.1	4.9
MLP	30	5	13,992	50.5	18.8	11.1	5.6
Logistic R.	75	2	16,872	60.8	13.5	10.9	4.9
Decision Tree	75	2	15,983	57.6	13.0	7.0	3.5
Linear SVM	75	2	15,028	54.2	21.8	5.9	2.7
LDA	75	2	17,096	61.7	13.6	10.6	4.8
Naive Bayes	75	2	17,442	62.9	13.6	3.4	2.0
k-NN	75	2	17,806	64.2	15.0	9.8	4.3
MLP	75	2	17,369	62.6	12.4	12.7	5.6
Logistic R.	75	5	1,573	5.7	16.3	0.5	1.8
Decision Tree	75	5	11,580	41.8	15.4	-5.9	-4.0
Linear SVM	75	5	2,892	10.4	33.2	-2.6	-4.2
LDA	75	5	1,501	5.4	15.9	0.6	2.3
Naive Bayes	75	5	8,538	30.8	15.4	-2.9	-3.9
k-NN	75	5	4,721	17.0	17.3	-0.1	-0.1
MLP	75	5	2,809	10.1	14.4	1.1	2.7

Notes: The table reports the operational performance of each simple benchmark model under alternative assumptions on loss given default and cost of funds. The approval rule accepts a loan whenever its model-implied expected profit is non-negative under the corresponding scenario. Monetary values are reported in millions of dollars.

D.2 Bagged and Boosted Benchmark Models

In Table 18, we report the results of the empirical analysis in which we benchmark black-box machine learning algorithms on the same Lending Club dataset.

Table 18: Model Benchmarking between bagged and boosted classification algorithms.

	LightGBM	CatBoost	XGBoost	Bagged DT	R. Forest	G. Boosting	AdaBoost
Accuracy	0.693	0.696	0.670	0.686	0.692	0.684	0.636
B. Acc.	0.702	0.702	0.669	0.688	0.690	0.681	0.643
Precision	0.342	0.343	0.314	0.331	0.335	0.327	0.287
Recall	0.715	0.711	0.668	0.693	0.686	0.677	0.654
F1	0.462	0.463	0.427	0.448	0.451	0.441	0.399

The benchmark results in Table 19 show that gradient-boosted tree models perform very strongly in operational terms. CatBoost and LightGBM obtain the highest realized profits under the expected-profit rule, with LightGBM also reaching the highest profit per unit of exposure, equal to 10.3%.

The fixed approval-rate frontier in Table 20 confirms LightGBM as the strongest benchmark across the approval rate grid.

Table 19: Operational portfolio evaluation under the expected-profit approval rule

Model	**Accepted**	**Approval (%)**	**Default (%)**	**Avg. PD (%)**	**Exposure ($m)**	**Realized loss ($m)**	**Realized profit ($m)**	**Profit/Exposure (%)**
CatBoost	22,863	82.5	13.7	13.8	302.6	28.3	29.7	9.8
LightGBM	21,872	78.9	12.7	12.7	289.4	25.4	**30.2**	**10.4**
Bagged Decision Tree	21,825	78.7	13.9	13.1	289.9	28.0	27.7	9.5
Random Forest	23,689	85.4	15.3	15.1	314.6	32.7	27.3	8.7
Gradient Boosting	24,227	87.4	15.4	15.5	322.3	33.5	27.5	8.5
XGBoost	26,473	95.5	17.5	17.2	351.2	41.0	25.2	7.2
AdaBoost	21,567	77.8	14.0	14.2	268.7	23.8	17.9	6.7

Notes: The table reports out-of-sample operational portfolio performance under the expected-profit approval rule. A loan is accepted whenever its model-implied expected profit is non-negative. "Default" is the realized default rate among accepted loans, while "Avg. PD" is the average predicted probability of default among accepted loans. Monetary values are reported in millions of dollars.

Table 20: Realized portfolio profit under fixed approval-rate policies

Model	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
CatBoost	3.2	6.6	10.1	13.7	16.9	20.1	22.3	24.6	26.4
LightGBM	3.8	7.4	10.9	14.3	17.4	20.7	23.6	26.2	27.0
Bagged Decision Tree	3.3	7.1	10.4	13.4	16.5	19.7	22.8	24.9	25.6
Random Forest	2.3	5.3	8.3	11.8	14.7	18.2	20.8	23.1	24.8
Gradient Boosting	2.3	5.3	8.5	11.8	14.7	17.7	20.4	22.5	24.5
XGBoost	1.9	4.4	7.2	10.1	13.0	15.6	18.2	20.1	23.8
AdaBoost	1.5	2.5	4.6	6.2	8.7	10.9	13.4	15.9	18.1

Notes: The table reports realized portfolio profit when each benchmark model is constrained to approve a fixed percentage of loan applications. Columns correspond to the approval-rate grid from 10% to 90%, and values are reported in millions of dollars.

In Table 21 and Table 22 we report the analysis with different assumptions on the LGD values.

Table 21: Operational robustness across cost-of-funds assumptions: LGD = 30%

Model	**Cost (%)**	**Accepted**	**Approval (%)**	**Default (%)**	**Profit ($m)**	**Profit/Exposure (%)**
CatBoost	2	26,559	95.8	17.1	36.6	10.4
LightGBM	2	25,847	93.2	16.5	36.7	10.7
Bagged Decision Tree	2	25,458	91.8	16.6	35.7	10.6
Random Forest	2	26,943	97.2	17.7	36.0	10.1
Gradient Boosting	2	27,525	99.3	18.3	35.9	9.9
XGBoost	2	27,663	99.8	18.5	35.8	9.8
AdaBoost	2	27,669	99.8	18.4	35.7	9.8
CatBoost	5	14,050	50.7	17.3	12.6	6.3
LightGBM	5	13,650	49.2	16.3	12.6	6.5
Bagged Decision Tree	5	13,665	49.3	17.3	11.7	6.0
Random Forest	5	14,324	51.7	19.2	11.6	5.6
Gradient Boosting	5	14,192	51.2	19.6	11.7	5.7
XGBoost	5	14,699	53.0	21.8	11.0	5.1
AdaBoost	5	11,958	43.1	18.9	8.0	4.6

Table 22: Operational robustness across cost-of-funds assumptions: LGD = 75%

Model	Cost (%)	Accepted	Approval (%)	Default (%)	Profit ($m)	Profit/Exposure (%)
CatBoost	2	16,964	61.2	10.7	16.8	7.5
LightGBM	2	16,768	60.5	10.3	16.0	7.3
Bagged Decision Tree	2	16,660	60.1	11.7	14.3	6.5
Random Forest	2	16,944	61.1	12.6	14.2	6.3
Gradient Boosting	2	16,361	59.0	12.5	14.2	6.3
XGBoost	2	16,057	57.9	14.8	12.4	5.3
AdaBoost	2	15,437	55.7	11.3	2.6	1.4
CatBoost	5	4,465	16.1	7.9	4.3	7.0
LightGBM	5	5,388	19.4	8.1	4.1	5.6
Bagged Decision Tree	5	5,602	20.2	10.8	2.7	3.4
Random Forest	5	3,577	12.9	13.0	2.0	3.8
Gradient Boosting	5	3,319	12.0	11.9	2.9	5.7
XGBoost	5	1,868	6.7	19.9	1.3	3.4
AdaBoost	5	1	0.0	0.0	2.2×10^{-3}	11.1