Taiwan Credit Default Prediction: Feature Engineering, Class Imbalance, and Model Comparison

Felix Reibold
03769602
M.Sc. Management and Technology
TUM School of Management
Munich, Germany
felix.reibold@tum.de

Beyza Eren
03767348
M.Sc. Management and Technology
TUM School of Management
Munich, Germany
beyza.eren@tum.de

Mehmet Emin Öztürk
03791891
M.Sc. Management and Technology
TUM School of Management
Munich, Germany
mehmet.oeztuerk@tum.de

Abstract— **We study credit-risk prediction on the UCI Default of Credit Card Clients dataset (30,000 accounts; ~22% default). Two model families are compared under a common pipeline: Logistic Regression (LR) and XGBoost (XGB), each with/without feature engineering (utilization ratios, delinquency counts, aggregates) and with/without class-imbalance handling. Models are tuned via cross validation and evaluated on a hold-out test set using ROC-AUC, PR-AUC, recall, accuracy, balanced accuracy, calibration, and confusion matrices. We find that feature engineering and imbalance handling improve minority-class recall. XGB variants achieve the strongest overall curves (PR and ROC) and the best recall, while LR remains close-behind; regression possesses the advantage that it is easier to interpret (standardized coefficients, odds ratios). Given the small performance gap, LR with FE and class weighting offers a strong baseline for credit scoring.**

Keywords - credit scoring, probability of default, logistic regression, XGBoost, class imbalance, feature engineering

I. Introduction

The ability to reliably assess credit risk has long been central to the stability of modern financial systems. This report examines the development and application of statistical and machine learning methods for credit scoring, with a particular focus on evaluating different approaches on the ML-repositories data on Taiwan credit card defaults.

Motivation: Credit scoring is about predicting whether a borrower will repay their debt. Before computers, bankers relied on relationship banking: they knew their customers personally, or judged them on subjective impressions (character, reputation, family background) [1]. As lending scaled up in the 20th century, this was no longer practical. The first formal scoring models emerged in the 1950s, with Fair Isaac Corporation (FICO) developing automated scores based on statistical analysis [2].

Problem statement: Modern credit scoring is a supervised machine learning task, using demographic and economic data of an individual to determine their credit risk. Input(s) are borrower characteristics (demographic, financial, behavioral). The output is a numerical score summarizing default risk, with the objective to minimize losses for lenders while maximizing access to credit. This is beneficial to banks and financial institutions.

Research questions: This report aims to answer the question: *"Which machine learning method provides the best predictive performance for credit scoring on the ML-repositories data on Taiwan credit card defaults, and what are the most important features?".* The data is public domain and part of the research by Yeh, I. (2009) [11]. The experimental approach documented in this report aims to verify and investigate the findings of recent publications on credit scoring.

Report structure: The report is structured as follows. Section 2 provides literature review on credit scoring approaches used in practice. Section 3 describes the available data and the methodological approach to conduct the credit scoring task. Section 4 presents the modelling procedure, including data pre-processing, algorithm selection, hyperparameter tuning, and performance evaluation, and discusses the corresponding results. And it provides a conclusion and an outlook on the topic of credit scoring.

II. Background

In this literature review, total 23 of papers founded and 11 papers have been screened. The papers were published between 2003 and 2025 years and generally they were written in English.

Credit scoring has been an important area of research and practice for decades, as banks and financial institutions need reliable ways to decide whether to grant credit to customers. Historically, the most common approaches have been based on statistical models, especially Logistic Regression (LR). Logistic regression has been widely used because of its simplicity, its ability to provide clear probability estimates, and the interpretability of its coefficients. Thomas [10] explains that logistic regression became the "industry standard" in part because it satisfies regulatory requirements for transparency and is relatively robust on structured tabular data. Other traditional methods such as Linear Discriminant Analysis (LDA) and simple scorecards were also popular in early applications of credit scoring.

From the early 2000s onwards, however, researchers began to systematically investigate whether machine learning (ML) algorithms could improve predictive accuracy. Baesens [1] conducted one of the first large benchmarking studies and showed that classification algorithms like Neural Networks, Support Vector Machines, and Decision Trees often outperformed logistic regression on a range of credit datasets. These methods are more flexible in capturing non-linear patterns and complex interactions between variables, which are often present in financial data.

Among machine learning methods, tree-based models have attracted particular interest. Decision Trees alone are often too unstable to be used in practice, but Random Forests and Gradient Boosting methods combine many trees to produce more accurate and stable predictions. Brown [3], in a widely cited comparative study, found that ensemble methods consistently outperform logistic regression, particularly when datasets are imbalanced (i.e. when defaults are relatively rare compared to non-defaults, which is the usual case).

More recently, specific implementations of boosting such as XGBoost [4] and LightGBM have become the tools of choice in many applied settings. These algorithms are not only highly accurate but also computationally efficient, which makes them attractive when working with large financial datasets such as those available on Kaggle or in credit bureau records. Many Kaggle competitions on credit default prediction have been won using these boosting algorithms, which reflects their dominance in practice.

At the same time, there is a major challenge in adopting complex models in credit risk management: interpretability. Financial institutions and regulators require that credit decisions can be explained to the customer and justified in case of disputes. Logistic regression remains attractive because coefficients are easy to interpret, while models like XGBoost act more like a "black box." In response to this, researchers have developed model interpretation methods such as SHAP values [8], which allow practitioners to identify which features contributed most to an individual prediction. Martens [9] also explored ways of extracting rules from Support Vector Machines to make them more comprehensible.

The literature overall paints a picture of a trade-off: logistic regression is interpretable and still competitive in some cases, but modern ensemble methods usually deliver better predictive performance. The question of whether to prioritize accuracy or interpretability remains open, and many researchers argue for a middle ground, where advanced ML models are combined with post-hoc interpretation methods. For our project, this tension between accuracy and interpretability will be particularly relevant, as we aim to compare traditional and modern methods on a publicly available credit dataset.

III. DATA AND METHODS

This study utilizes the *Default of Credit Card Clients* dataset from the UCI Machine Learning Repository [11]. The dataset contains information on 30,000 credit card holders in Taiwan, including demographic attributes, credit history, bill statements, and repayment records. The goal is to predict whether a client will default on their payment in the following month, making the task a binary classification problem. The methodology comprises automatic data import and preprocessing (column standardization), feature engineering, stratified train–test splitting, model selection and hyperparameter tuning (logistic regression and XGBoost with class-imbalance handling), followed by training and evaluation using accuracy, ROC-AUC, precision/recall, and confusion matrices.

The dataset includes variables such as credit limit, age, sex, marital status, and education level, as well as detailed financial records covering bill amounts and repayment status over six consecutive months. The target variable (target) is imbalanced, with approximately 22% of clients defaulting and 78% not defaulting.

An overview of the main variables is provided in Table 1.

Variable	**Description**
LIMIT_BAL	Credit limit assigned to the individual (NT dollars).
SEX	Gender (1 = male, 2 = female).
EDUCATION	Level of education (1 = graduate school, 2 = university, 3 = high school, 4 = others).
MARRIAGE	Marital status (1 = married, 2 = single, 3 = others).
AGE	Age in years.
PAY_0–PAY_6	Repayment status from April–September 2005 (-1 = pay duly, 1 = payment delay of 1 month, 2 = delay of 2 months, …, 9 = delay ≥9 months).
BILL_AMT1–6	Amount of bill statement from April–September 2005 (NT dollars).
PAY_AMT1–6	Amount of previous payment from April–September 2005 (NT dollars).
target	Default payment (1 = yes, 0 = no).

Table 1: Variables in the dataset and their description

The data was loaded from a single comprehensive CSV file.

To obtain a better understanding of the data composition and insights for feature engineering a the data was initially visualized using Seaborn and Matplotlib libraries. Fig. 1 visualizes the default rate by education it shows a negative correlation between default rate and level of education, with the unspecified category (comprised of doctors, professors, and others) being a rather good indicator for creditworthiness. Fig. 2 stratifies by gender, females are 3.4 percentage points, or about 14% less likely to default then males.

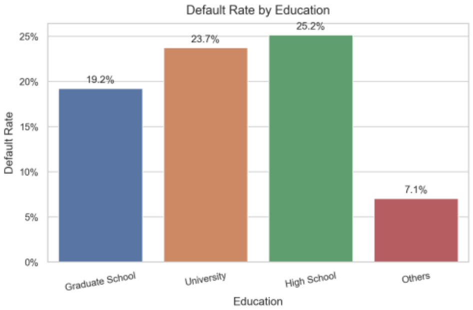

Figure 1: Default rate by education

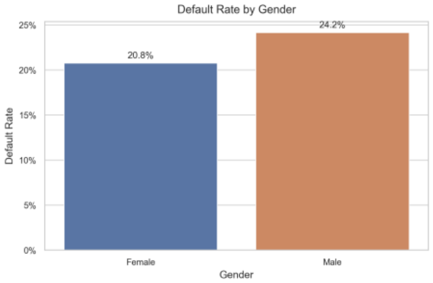

Figure 2: Default rate by gender

Fig. 3 displays the association between default rate the number of delinquent months, defined as months with unpaid credit obligations (where PAY_* > 0). The relationship is strongly monotonic and approximately linear: borrowers with 4–5 delinquent months have an observed default rate of ≈50%, which escalates to ≈70% for 6 delinquent months, indicating a substantive incremental risk to creditors.

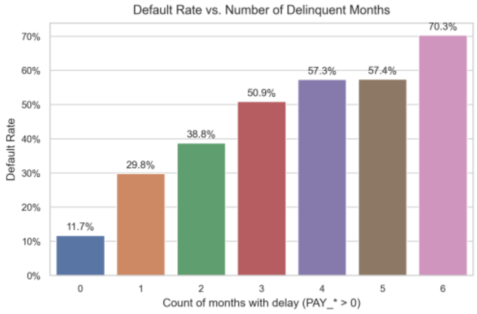

Figure 3: Default rate by number of delinquent months

Fig. 4. Shows the default rate associated with binned credit utilization. The utilization was computed as the statement balance divided by the credit limit (BILL_AMT/LIMIT_BAL, averaged over recent months). It captures the share of available credit in use, with higher values indicating tighter liquidity and elevated default risk.

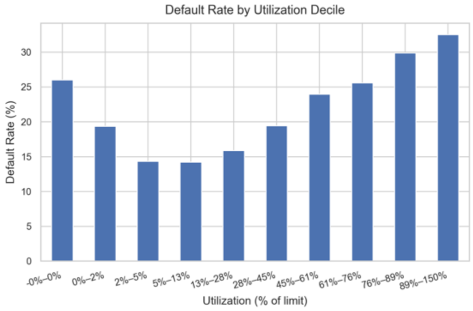

Figure 4: Default rate by binned credit utilization

Fig. 5 shows the univariate correlations of the top features with default. Delinquency signals stand out: late_payment_count (shown in more detail in Fig. 3) has the strongest correlation (r≈0.40). Recent repayment-status codes (PAY_0…PAY_6) show positive associations that decay with time (strongest for PAY_0). The decay is rather slow indicating a strong recenct effect even for a past delinquency. In contrast, LIMIT_BAL has a mild negative correlation with default (r≈−0.15) and; this is similar for the repayment_ratio_mean (r≈−0.12). Higher limits and consistent repayment (or even prepayment) may be a proxy for wealth or higher financial status. (Correlations are unadjusted and capture linear association only, the non-linear relationship in Fig. 4 regarding credit utilization may not be well captured.). The findings depicted in these figures informed the decisions made in the feature engineering.

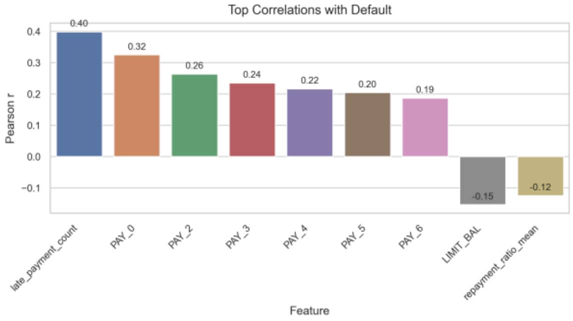

Figure 5: Top correlations with default rate

We now describe the experimental setup used to quantify model performance and the effect of feature engineering (FE) and imbalance handling. We used the following steps to construct an automated experimental pipeline to compare model performance. The investigated methods, logistic regression and extreme gradient boosting (XGB), were tuned using a parameter grid and cross validation. They were evaluated on an experiment grid executing eight experiments:

{LR, XGB} x {with FE, without FE} x {balance-aware, not balance-aware}

Step I: Data acquisition and schema standardization.

We load the Default of Credit Card Clients dataset directly from the UCI repository. Columns are normalized to the official descriptive names (e.g., X12 ⇨ BILL_AMT1) to avoid downstream key errors. The target is defined as (1=default, 0=non-default). No imputation is required as the dataset has no missing values in the used fields.

Step II: Exploratory data analysis (EDA) and cleaning.

Input variables are mapped to readable labels (e.g., education, marriage, sex). We compute quick descriptive statistics and class balance (≈22% default). These summaries are used only for understanding and are not fed back into training. (The full unabridged visual analysis presented above was conducted in a separate script)

Step III: Feature engineering (FE).

Based on EDA, we derive ratios and aggregated features making sure to avoid multicollinearity (linear dependencies with other input variables from the raw data) :

- Utilization ($UT_1 \ldots UT_6$): $BILL\_AMT_t$ / LIMIT_BAL for t=1…6.
- Repayment ratio ($RR_1 \ldots RR_6$): PAY_AMT_t / $BILL\_AMT_t$ (clipped to handle outliers seen in Fig. 4 on the left and right tail).
- Delinquency statistics: late_payment_count (count months with outstanding payments: PAY_* > 0) and max_delay.
- Aggregated features: total billed and total repaid, and a coarse age bin.

These variables measure the amount of debt, timing, and the ability to pay down the debt in a punctual manner.

Step IV: Train/test split.

We create an 80/20 “stratified” split to preserve class balance, meaning the target variable occurs at the same ratio in the training-set and the holdout-set. The variables are scaled (independently) in both sets. Scaling means making all features similar in size. This helps logistic regression but is also helpful to other supervised and unsupervised learning algorithms (e.g. k-means clustering, PCA). XGBoost does not require scaling, but it also does not affect the performance negatively, so we apply it to the data used by both methods to keep preprocessing path uniform.

Step V: hyperparameter tuning.

We evaluate two model classes:

- Logistic Regression (LR): The code runs a grid search over a small, set of settings. We vary the regularization strength C which results in simpler models for small values C = (0.01, 0.1) and more complex models for large values C = (1, 10). Simpler models have the benefit that they generalize better to unseen data, Complex models are more flexible to fit complex customer patterns, but they are more prone to overfit on unseen data which may degrade real-world performance. The best value is picked comparing performance on via a 5-fold cross-validation on the training set. We used the default performance metric which is ROC-AUC. Another parameter that we varied is the penalty (L1, L2). L1 causes weak regression coefficients to disappear to zero. Another parameter wich we varied not during tuning, but in the experiments is the class_weight = (None, balanced). In logisic regression highly unbalanced datasets during training give too much importance to the majority class. In our case most people do not default, and do not pose a rısk to the creditor, so that the model likely underestimates the risk. We quantify the difference in the experiments.

- XGBoost (XGB): We applied a randomized search (conducting 12 trials) over the parameters: n_estimators, max_depth, learning_rate, subsample, colsample_bytree, min_child_weight, with three-fold CV. Class rebalancing is automatically enabled by the XGBoost python implementation. Class imbalance can be enabled manually via scale_pos_weight ≈ (#neg / #pos) ≈ 3.5; the inverse ratio of our class imbalance 22/78.

Step VI: Class-imbalance strategies.

We compare balance-aware and non-balance-aware versions. LR uses class_weight = 'balanced'. XGB uses scale_pos_weight = 1. While balancing usually yields more desired results, it is important to be aware of the tradeoffs. Balancing increases the predictive accuracy on the minority class: the recall. That’s good for predicting more credit defaults. However, it lowers the overall accuracy of the model. More creditworthy people are falsely rejected, thus denying them a loan. Usually the cost of a credit default is higher for a creditor, than the lost intrest-revenue on the loan. Therefore not the model with the highest accuracy, but the most cost efficient model is the best.

Step VII: Training and model selection.

For each configuration, the best estimator from Step V (determined via cross-validation) is refit on the full training set and then evaluated once on the hold-out test set. This preserves a clean separation between model selection and final evaluation.

Step VIII: Evaluation on the hold-out set.

We report Accuracy, ROC-AUC, precision, recall, and the confusion matrix at the default threshold 0.5. ROC-AUC is the primary selection metric due to class imbalance; Recall quantifies the practical detection performance for defaults.

Step IX: Experimental study

An automated runner executes eight experiments:

FE = with feature engineering

NoFE = without feature engineering (raw features only)

Bal = balance-aware (class_weight = "balanced" for LR; scale_pos_weight ≈ neg/pos ≈ 3.5 for XGB)

NoBal = not balance-aware (class_weight = "none" for LR; scale_pos_weight = 1 for XGB)

1. **LR + FE + Bal** → *LR-FE-Bal*
2. **LR + FE + NoBal** → *LR-FE-NoBal*
3. **LR + NoFE + Bal** → *LR-NoFE-Bal*
4. **LR + NoFE + NoBal** → *LR-NoFE-NoBal*
5. **XGB + FE + Bal** → *XGB-FE-Bal*
6. **XGB + FE + NoBal** → *XGB-FE-NoBal*
7. **XGB + NoFE + Bal** → *XGB-NoFE-Bal*
8. **XGB + NoFE + NoBal** → *XGB-NoFE-NoBal*

Metrics, best hyperparameters, confusion counts, and runtimes are saved to CSV for reproducibility and comparative analysis.

Step X — Interpretation and diagnostics.

We inspect balanced accuracy, confusion matrices and recall for the for determining which method performed best in predicting the minority class (defaulting). Then we investigate which features arr the most imoprant for prediction. For LR, we inspect standardized coefficients and odds ratios to assess multiplicative risk effects (i.e by which factor does an increase in overdue payments by one month increase the odds of default, see Fig. 9). For XGB, we inspect feature importances (see Fig. 9)

IV. Results and Discussions

The results of our experiments are explored and analysed trough visualizations and aim to answer our research question which method performed best on the credit scoring task and which features hold the most predictive power.

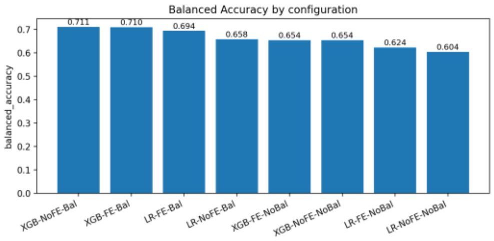

Figure 6a: Balanced accuracy comparison of each method

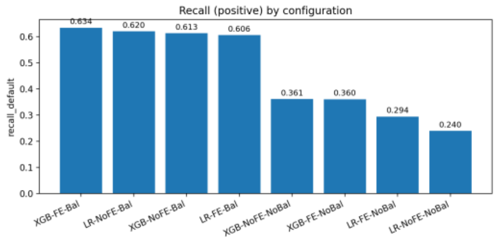

Figure 6b: Recall comparison of each method

Fig. 6 provides a comparison for all approaches for the accuracy metrics balanced accuracy and recall, as discussed in the previous section (Step VI), we are more interested in scoring high on recall rather than overall accuracy. The XGB performs slightly better than other methods applied on balanced data. A 63.4% recall indicates that about two-thirds of defaults are correctly predicted. Logistic regression on the raw data performs surprisingly well, indicating that the sub-selection of predictor variables and engineered features should be improved.

Fig. 7 shows in the red box the nominal number of defaults correctly predicted. The top two confusion matrices show the algorithms applied to unprocessed data (NoFE + NoBal), the bottom two confusion matrices show the algorithms applied to pre-processed data (FE + Bal): A substantial improvement can be seen in terms of true positives, however most of it should be attributed to balancing, showing how essential this preprocessing step is.

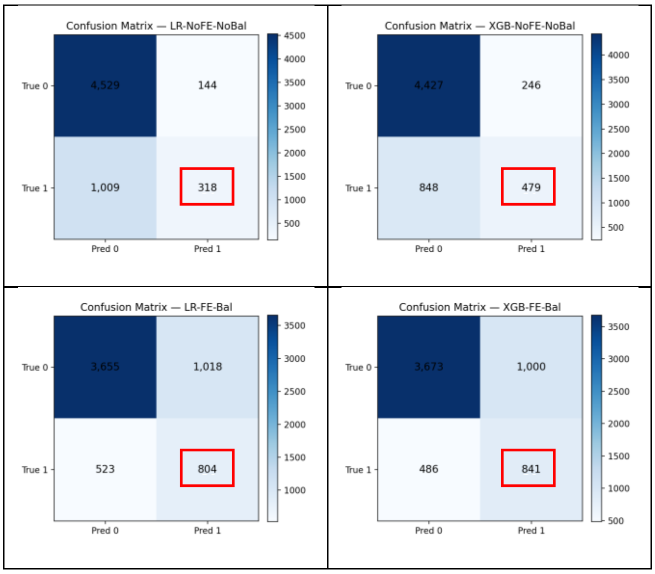

Figure 7: Confusion Matrix for both approaches on raw data (top) and on processed data (bottom)

Fig. 8a shows the precision-recall (PR). The curve shows precision (share of flagged cases that are truly positive) against recall (share of all positives correctly predicted) as the decision threshold moves. Curves closer to the top-right are better. The horizontal "no-skill" level equals the positive rate (~22% here). This can be achieved by classifying every case as a potential default; useful models should lie above it. All XGB curves the LR models.

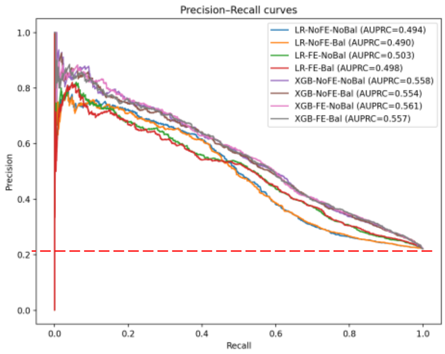

Figure 8a: Precision-recall curve

Fig. 8b shows the ROC curve. It plots true-positive rate vs false-positive rate; the diagonal is random guessing. All models beat chance, with the XGB configurations showing the highest AUCs (Area Under the Curve).

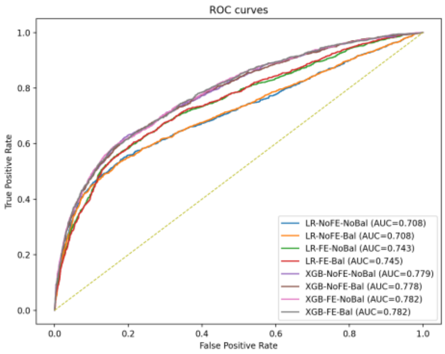

Figure 8b: ROC curve

The most interesting insights can be gained from the importance analysis (Fig. 9). During the data visualization we saw the number of late payments (delinquencies in the recent 6 month) correlate strongly with defaults. It is reflected in both models that this engineered feature captured indeed a strong predictive power. Generally, any metric capturing a past delinquency, scores high in predicting the inability to pay off the debt in the future. The chart regarding the coefficients of the logistic regression is interesting because it shows the direction of the association; LR is well interpretable. For example, customers with higher limits have a lower probability of default (all else equal).

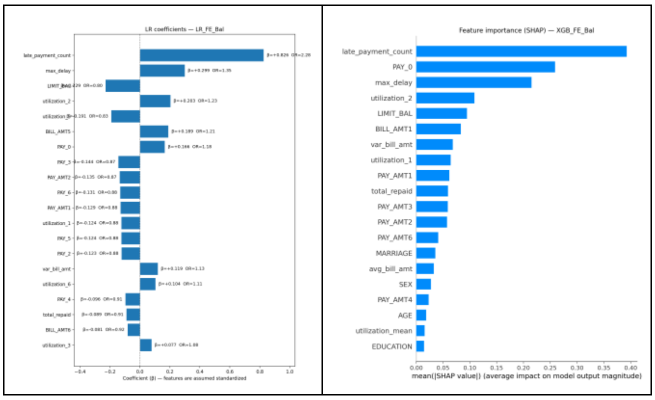

Figure 9: Importance analysis: LR-best (left), XGB-best (right)

In conclusion, XGBoost (XGB) performs best overall. Its PR/ROC curves sit closer to the top-right and it has the highest recall but the edge over Logistic Regression (LR) is small. Given the need for clear explanations, LR with feature engineering and class weighting is a strong default choice: competitive recall with interpretable odds-ratio interpretations.

Key signals are recent delinquency (late_payment_count). Higher credit limits and stronger repayment ratios are linked to lower risk.

Limitations

Limitations. This study uses a single dataset and time period (Taiwan credit cards, 2005–2006), so generalizability to other geographies, products, and cycles is limited. We did not run statistical significance tests, so we avoid strong claims about small performance gaps. Finally, our feature scope is limited: engineered variables are sensible but not exhaustive; interactions and temporal patterns beyond six months were not modeled.

REFERENCES

[1] 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

[2] Boydon, K., & Boydon, K. (2025, May 22). *From trust to tech: The evolution of credit scoring*. Credit Sesame. https://www.creditsesame.com/blog/credit-score/from-trust-to-tech-the-evolution-of-credit-scoring

[3] Brown, I., & Mues, C. (2011). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems With Applications*, *39*(3), 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033

[4] Chen, T., & Guestrin, C. (2016). XGBoost. *The 22nd ACM SIGKDD International Conference*, 785–794. https://doi.org/10.1145/2939672.2939785

[5] FRB. (2007, August). *Report to the Congress on credit scoring and its effects on the availability and affordability of credit*. www.federalreserve.gov. Retrieved August 16, 2025, from https://www.federalreserve.gov/boarddocs/rptcongress/creditscore/general.htm

[6] Kraus, A. (2014). *Recent Methods from Statistics and Machine Learning for Credit Scoring* [PhD dissertation, Ludwig–Maximilians–Universität München]. https://d-nb.info/1054598924/34

[7] Li, M., Mickel, A., & Taylor, S. (2018). “Should This Loan be Approved or Denied?”: A Large Dataset with Class Assignment Guidelines. *Journal of Statistics Education*, *26*(1), 55–66. https://doi.org/10.1080/10691898.2018.1434342

[8] Lundberg, S. M., & Lee, S. (2017). A unified approach to interpreting model predictions. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1705.07874

[9] Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, *183*(3), 1466–1476. https://doi.org/10.1016/j.ejor.2006.04.051

[10] Thomas, L. C. (2009). *Consumer credit models*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199232130.001.1

[11] Yeh, I. (2009). Default of Credit Card Clients [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H.