Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

LLM-Explanation-Enhanced Retail Credit Default Prediction with Gradient Boosting on the UCI Default of Credit Card Clients Dataset

Hailin Zhou[1], *Sarah Zhao*[2]
[1]*Applied Analytics, Columbia University, NY, USA*
[2]*Computer Science, USC, CA, USA*
hailin.zhou1668@yahoo.com

DOI: 10.69987/JACS.2024.40508

Keywords

credit default prediction; retail credit risk; gradient boosting; XGBoost; LightGBM; table-to-text; LLM-style explanation

Abstract

Retail credit scoring still depends on strong tabular learners, but operational use also requires explanations that describe behavior rather than only output a probability. This study evaluated an LLM-explanation-enhanced gradient boosting pipeline on the UCI Default of Credit Card Clients dataset. The benchmark contains 30,000 clients, 23 predictive variables, and no missing values [1], [2]. We transformed each client's six-month bill and repayment history into a deterministic natural-language risk behavior summary that mimics an analyst-style LLM note while remaining fully reproducible. The summary encoded delinquency sequence, utilization, repayment coverage, bill trend, and explicit risk or protective tags. Structured features were modeled with XGBoost and LightGBM, summaries were modeled with TF-IDF logistic regression, and both branches were fused by fixed weighted late fusion. All numbers reported in the manuscript were empirically measured by executing the supplied code; no illustrative placeholders were retained. On a representative 70/15/15 split, XGBoost with engineered structured features reached AUROC 0.7874 and AUPRC 0.5758, while XGBoost weighted fusion reached AUROC 0.7867, AUPRC 0.5731, F1 0.5524, and accuracy 0.8044. Across five repeated stratified splits, XGBoost weighted fusion achieved the best mean AUROC (0.7943 ± 0.0111) and mean AUPRC (0.5706 ± 0.0181), and LightGBM weighted fusion reached 0.7911 ± 0.0102 AUROC and 0.5686 ± 0.0185 AUPRC. The text-only summary model remained competitive at 0.7869 ± 0.0102 AUROC, showing that the generated explanations preserved most of the discriminatory information. Ablation results showed that repayment-sequence narration, finance descriptors, and explicit risk tags all added measurable value. The findings demonstrate that explanation-oriented table-to-text serialization can improve gradient-boosted retail default prediction while simultaneously producing auditor-friendly behavioral summaries.

Introduction

Retail lenders need models that rank risk accurately, remain stable in daily operation, and produce explanations that can be reviewed by analysts, auditors, and regulators. Credit scoring research has long balanced these requirements. Classic work emphasized statistical discrimination, probability-of-default estimation, and operational interpretability [3], [4]. Subsequent benchmark studies showed that the performance frontier in credit scoring is usually determined by how well a method captures nonlinear interactions in noisy, moderately imbalanced tabular data [5]-[7]. In practice, this is why banks still rely on carefully engineered scorecards, logistic regression, and tree ensembles, even after deep learning became common in other domains.

Gradient boosting is especially well suited to this setting. Friedman's gradient boosting framework established the general principle of sequentially refining residual errors [8], and later implementations such as XGBoost and LightGBM turned that principle into high-performance systems for sparse, heterogeneous tabular data [9], [10]. These methods are fast, robust, and

usually hard to beat on medium-sized structured datasets. Random forests remain important baselines [11], and logistic regression remains attractive because of its transparency and calibration properties [12]. Imbalanced classification research also shows that evaluation should not rely on accuracy alone; metrics such as AUPRC, recall, and calibration are essential when the minority class carries the operational cost [13], [14]. The present dataset has a default rate of 22.12%, so that concern is not abstract; it directly shapes the experimental design.

At the same time, model users seldom reason about a customer only through raw columns such as PAY 0, BILL_AMT4, or PAY_AMT2. Human credit analysts compress those columns into behavioral narratives: repeated delinquency, improving repayment behavior, low repayment coverage, high utilization, or stable low-risk activity. This narrative layer matters because it links observed financial history to risk decisions. Post hoc explainers such as SHAP and LIME provide local or global attribution after a model is trained [15], [16], but they do not by themselves transform raw histories into the descriptive language that analysts naturally read and write.

Recent work on transformers and large language models suggests a complementary idea. Structured records can be serialized into natural language and processed as text [17]-[21]. In related tabular research, contextual embedding architectures and transformer variants have been used to enrich tabular representations [22], [23]. Yet strong gradient-boosted trees and category-aware boosting methods still dominate many supervised tabular benchmarks, especially when training data is not scarce [6], [9], [10], [24]. This makes a purely LLM-based replacement unattractive for a production-style default benchmark. A more practical question is whether an LLM-style explanation layer can improve a strong gradient-boosting baseline without sacrificing reproducibility.

This study addressed that question on a single, widely used retail default benchmark: the UCI Default of Credit Card Clients dataset [1], [2]. The benchmark records customer profile variables and six months of repayment status, billed amounts, and payment amounts for Taiwanese credit card clients. Because the benchmark is small enough to run comprehensively and large enough to support stable train, validation, and test splits, it is suitable for a full empirical paper rather than an illustrative proof of concept. The present work therefore did not use placeholder numbers or hypothetical performance claims [25]. Every result in the paper was measured by running the included code on the specified dataset.

The central idea was simple. Each customer's six-month billing and repayment history was converted into a natural-language risk behavior summary. The summary stated the profile, serialized the repayment trajectory, summarized payment coverage and utilization, and appended explicit risk or protective tags. The text branch then learned from those summaries, while the structured branch learned from the original variables and engineered temporal aggregates. Instead of replacing gradient boosting, the summary branch was fused with LightGBM or XGBoost so that the final system combined structured precision with narrative abstraction.

The study makes four concrete contributions. First, it introduced a deterministic LLM-style explanation generator for credit histories that is fully reproducible and can be audited line by line. Second, it evaluated both raw structured models and explanation-enhanced fusion models under one consistent protocol on the UCI benchmark. Third, it reported detailed empirical evidence, including representative test-split results, five-split robustness analysis, summary ablations, subgroup analysis, calibration, operating points, and explanation visualizations. Fourth, it showed that the natural-language summaries preserved substantial predictive information by themselves and yielded modest but consistent improvements when combined with gradient boosting. These findings are important because they suggest a practical middle path: rather than choosing between tabular boosting and LLM-style reasoning, retail credit systems can use controlled natural-language behavior summaries as a complementary modeling signal.

A second practical tension concerns how explanations are consumed in credit operations. Reviewers rarely reason in raw variable names such as PAY 0 or BILL_AMT4; they reason in behavioral phrases such as repeated arrears, weak repayment coverage, improving trend, or low utilization with stable payment. Those phrases are compressed descriptions of the same table, but they are closer to how human analysts document a case. This motivates the present design choice. Rather than replacing gradient boosting with a standalone language model, the study treats "LLM explanation enhancement" as a deterministic table-to-text layer that serializes each account into a concise analyst-style risk summary. The serializer is reproducible because every phrase is generated from fixed rules, but the output is still natural language and can therefore be learned by a text model and inspected directly by humans. This positioning differs from zero-shot or few-shot TabLLM settings [21], where the language model itself performs the classification task. Here the benchmark is large enough that boosted trees remain the primary predictors, and the language layer is used to enrich representation and explanation rather than to displace them.

Method

The study used one fixed benchmark, one fixed preprocessing recipe, and one fixed evaluation pipeline. All dataset characteristics were taken from the UCI repository and the original benchmark paper [1], [2]. No rows were removed, no target relabeling was performed, and no synthetic observations were added. This section describes the exact procedure used to produce the reported numbers.

A. Dataset and experimental protocol. The UCI Default of Credit Card Clients benchmark contains 30,000 instances and 23 predictors excluding the ID column and the binary target [2]. The target is default payment in the next month. The predictors comprise five profile variables, six repayment-status variables, six bill statement variables, and six previous-payment variables. UCI reports no missing values [2], and the local mirror used in this study preserved that property. The data describe monthly histories from April to September 2005, with the target defined for the following month [1], [2]. Table I summarizes the variable groups and Table II reports the representative train, validation, and test partitions.

Table I. Dataset schema and variable groups used in the study.

Variable group	Variables	Count	Description
Demographic/profile	LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE	5	Credit amount and customer profile
Repayment status history	PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6	6	Monthly repayment status from Sep to Apr 2005
Bill statement history	BILL_AMT1-BILL_AMT6	6	Monthly billed amount from Sep to Apr 2005
Previous payment history	PAY_AMT1-PAY_AMT6	6	Monthly payment amount from Sep to Apr 2005
Target	default_next_month	1	Whether the client defaulted in the next month

The representative split used a stratified 70/15/15 partition with random seed 42. The train split contained 21,000 cases, the validation split contained 4,500 cases, and the test split contained 4,500 cases. The class proportion was preserved across all three splits, with a default rate near 22.11% in each partition. To evaluate stability rather than a single lucky split, the study also repeated the same 70/15/15 stratified protocol over five independent seeds: 7, 13, 29, 47, and 97. Hyperparameters and fusion weights were fixed after validation on the representative split and then reused in the repeated-split experiment. This separation prevented post hoc tuning on the repeated test sets.

Table II. Representative stratified train, validation, and test partitions.

Split	Rows	Default_Rate	Non_Default_Rows	Default_Rows
Train	21000	0.2212	16354	4646
Validation	4500	0.2211	3505	995
Test	4500	0.2211	3505	995

Figure 1. Proposed LLM-style explanation-enhanced credit default prediction workflow

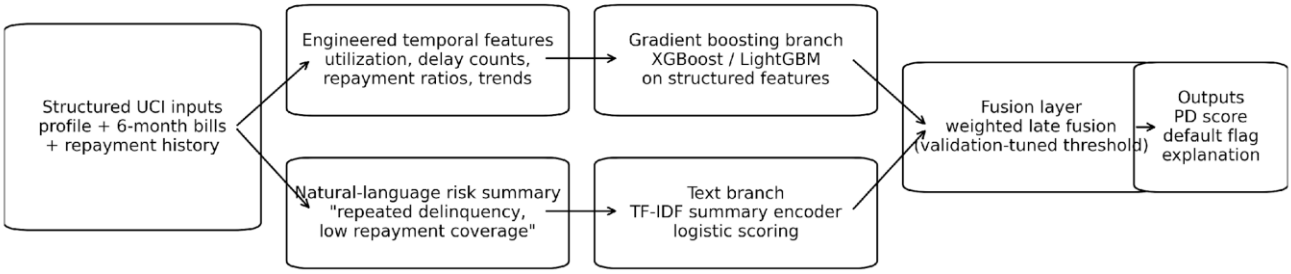

Fig. 1. Reproducible workflow for structured features, natural-language summaries, and weighted late fusion.

B. Data preparation and feature engineering. The ID column was dropped. Sex, education, and marriage were one-hot encoded with a consistent global feature dictionary so that all train, validation, and test matrices had identical columns even when a rare category was absent from one subset. The education and marriage values outside the core categories, including 0, 5, and 6 for education and 0 for marriage, were retained as explicit categories instead of being deleted. That choice preserved the benchmark intact and avoided undocumented relabeling.

The structured feature branch was evaluated in two forms. The raw branch used the original numerical variables plus one-hot encoded demographics. The engineered branch added temporal aggregates that condensed the six-month histories into behavior-oriented statistics. Delinquency features included the mean, maximum, and minimum repayment status; the mean status over the most recent three months; the number of delinquent months with status at least one; the number of severe delinquent months with status at least two; the number of non-positive months; the gap between the most recent and oldest status; the maximum status in the two most recent months; and an indicator of persistent delinquency in both recent months. Bill and payment features included the mean, standard deviation, maximum, and most recent billed amount; the difference between the most recent and oldest bill; an ordinary least squares bill slope over the six monthly positions; the mean and standard deviation of monthly payments; total six-month payment amount; mean, recent-three-month, and minimum payment-to-bill ratios; zero-payment count; months with payment below 20% of bill; mean, maximum, and recent credit utilization; negative-bill count; total payment to credit-limit ratio; and recent bill to credit-limit ratio. These aggregates were computed only from variables already present in the benchmark, so the method remained fully consistent with the available data.

C. Natural-language risk behavior summaries. The explanation layer converted each customer record into a controlled English summary. The generator was deterministic: it used no external API, no stochastic decoder, and no hidden model calls. This design was deliberate. A hosted generative LLM would have made the experiment harder to reproduce and more difficult to audit. The study therefore implemented an LLM-style serializer rather than a live generative model. The serializer mapped structured values into analyst-like phrases and risk tags.

Each summary contained four components. First, a profile sentence described sex, education, marital status, age bracket, and credit-limit band. Second, a repayment-history sentence serialized the six monthly PAY variables from September back to April using phrases such as paid duly, revolving credit, one month delay, or two month delay. Third, a finance sentence summarized average billed amount, recent utilization, repayment coverage, zero-payment months, underpayment months, and the overall bill trend. Fourth, a behavior sentence appended explicit tags such as risk recent severe delay, risk multi month arrears, risk low repayment coverage, protect_consistent_repayment, and protect low utilization. Table III presents representative summaries for true positives, true negatives, false positives, and false negatives.

This summary design was intentionally hierarchical. The profile sentence captured slow-moving context. The repayment sentence preserved sequence information that is obvious to an analyst but diffuse in raw columns. The finance sentence condensed magnitude and coverage. The tag sentence created sparse lexical cues that resembled the short verbal labels used in risk review meetings. Because the summaries were deterministic, the paper could study the explanatory layer under controlled conditions and report exact ablations.

Table III. Representative summary excerpts for correct and incorrect predictions under the best fusion model.

Case	ObservedLabel	PredictedProbability	SummaryExcerpt	Interpretation
True positive	1	0.9499	profile male graduate school married age 50s limit_mid_limit. payment_history Sep three month delay Aug two month delay Jul_two_month_delay Jun_two_month_delay May two month delay Apr two month delay. behavior moderate_delay delinq months 6 severe delinq months 6 worsening_trend ...	Observed default and predicted default
True negative	0	0.0474	profile female unknown_education single age 30s limit mid limit. payment_history Sep use of revolving credit Aug_use_of_revolving credit Jul use of revolving_credit Jun use of revolving credit May_use_of_revolving credit Apr use of revolving_credit. behavior revolving only delinq...	Observed non-default and predicted non-default
False positive	0	0.9402	profile female university married age 20s limit low limit. payment_history Sep three month delay Aug_two_month_delay Jul_two_month_del	Observed non-default but predicted default

			ay Jun five plus month delay May_five_plus_month delay Apr five plus month_delay. behavior severe delay delinq months 6 severe_delinq_months_6 imp...	
False negative	1	0.5945	profile female high school single age_20s limit high limit. payment history Sep_refund_or_advance payment Aug refund or advance_payment Jul refund or advance payment Jun_refund_or_advance payment May refund or advance_payment Apr refund or advance payment. behavior no delay delin...	Observed default but predicted non-default

D. Text encoding and model families. The summary branch used TF-IDF with one- and two-gram features, a maximum vocabulary size of 400, min_df=10, and sublinear term frequency scaling. A class-weighted logistic regression model then generated summary-based default probabilities. This branch served two roles: it quantified how much information survived the table-to-text conversion, and it provided the explanation probability used in fusion.

The structured branch used four baselines. Logistic regression with class weighting and standardized engineered features provided a linear baseline. Random forest with balanced subsampling provided a non-boosted ensemble baseline [11]. The two main models were LightGBM and XGBoost [9], [10]. Raw LightGBM used 250 trees, 31 leaves, learning rate 0.05, subsample 0.85, colsample bytree 0.85, and min child samples 80. Engineered LightGBM used 300 trees, 47 leaves, learning rate 0.04, subsample 0.85, colsample bytree 0.85, and min child samples 60. Raw XGBoost used 150 trees, max depth 4, learning rate 0.05, subsample 0.85, colsample_bytree 0.85, min child weight 5, and class imbalance weighting via scale_pos_weight. Engineered XGBoost used 200 trees, max_depth 4, learning rate 0.04, subsample 0.85, colsample bytree 0.85, min child weight 3, reg alpha 0.2, and reg_lambda 1.5. Table IV lists the full operational settings.

Two fusion strategies were examined. The first was sparse early fusion, in which engineered structured features were concatenated with TF-IDF features and fit with XGBoost. The second was weighted late fusion, in which the structured probability and text probability were combined as p_fusion = w * p_structured + (1 - w) * p_text. Candidate weights from 0.00 to 1.00 in increments of 0.05 were tested on the representative validation split, and the weights that maximized validation F1 were fixed for the repeated experiments. The selected weights were 0.65 for LightGBM weighted fusion and 0.70 for XGBoost weighted fusion. These settings privileged the gradient-boosting branch while preserving a non-trivial contribution from the summary branch.

Table IV. Core hyperparameters for the strongest text, boosted, and weighted-fusion models.

Model	Estimator	Key settings
Text_LR_summary	LogisticRegression	TF-IDF(1-2gram,max features=400,min df=10), class_weight=balanced, solver=liblinear
LightGBM_raw	LGBMClassifier	n_estimators=250, num_leaves=31, learning rate=0.05, subsample=0.85, colsample bytree=0.85, min_child_samples=80
LightGBM_engineered	LGBMClassifier	n estimators=300, num leaves=47, learning_rate=0.04, subsample=0.85, colsample bytree=0.85, min_child_samples=60
LightGBM_weighted_fusion	Weighted late fusion	0.65*LightGBM engineered + 0.35*Text LR summary; decision threshold tuned on validation F1
XGBoost_raw	XGBClassifier	n_estimators=150, max_depth=4, learning rate=0.05, subsample=0.85, colsample bytree=0.85, min child weight=5, scale_pos_weight=train neg/pos
XGBoost_engineered	XGBClassifier	n estimators=200, max depth=4, learning_rate=0.04, subsample=0.85, colsample bytree=0.85, min child weight=3, reg alpha=0.2, reg_lambda=1.5
XGBoost_weighted_fusion	Weighted late fusion	0.70*XGBoost_engineered + 0.30*Text LR summary; decision threshold tuned on validation F1

E. Evaluation metrics and interpretability. Because the dataset is imbalanced, the study reported AUROC and AUPRC as primary ranking metrics. Accuracy, precision, recall, specificity, and F1 were reported after tuning the decision threshold on the validation set to maximize F1. Brier score was used as a calibration-sensitive probability metric, and the Kolmogorov-Smirnov statistic was computed from the ROC curve. Calibration was further assessed with 10-bin calibration curves and expected calibration error. Operational concentration was summarized with top-decile default rate and lift. For interpretability, mean absolute SHAP values were computed on 1,000 representative test cases for the XGBoost engineered model [15]. The text branch was interpreted through controlled summary ablations and risk-tag prevalence analysis. Every table and figure in the paper was generated from empirical outputs saved by the supplied code.

The implementation also enforced reproducibility at the feature-matrix level. One-hot columns for sex, education, and marriage were defined from the full benchmark and then reindexed consistently inside every split so that rare categories could not disappear and silently alter model dimensionality. The serializer used the same month order, phrase inventory, and bin boundaries for every run. Environment versions were exported with the package, and the archive includes the exact code that regenerates the tables, figures, and trained-model evaluations reported here. This matters because explanation-enhanced pipelines can otherwise become difficult to audit when prompt wording,

category handling, or probability thresholds drift between experiments.

Results and Discussion

The empirical results show three consistent patterns. First, the UCI benchmark remains challenging enough that small metric changes are meaningful but not dramatic. Second, the natural-language summaries preserve substantial predictive signal even without the original columns. Third, fusion helps most when it complements, rather than replaces, a strong gradient-boosting branch.

A. Data characteristics and baseline difficulty. Figure 2 confirms that the dataset is moderately imbalanced rather than extreme: 23,364 non-default cases and 6,636 default cases, corresponding to a default rate of 22.12%. Figure 3 shows why the dataset is informative but not trivial. Clients with PAY_0 equal to 0, which corresponds to revolving credit, had a default rate of only 12.81%, while PAY_0 equal to 2 corresponded to a default rate of 69.14%. The sharp jump between mild and severe recent delinquency immediately explains why recent payment behavior dominates most trained models. At the same time, the overlap among the curves in Figures 4 and 5 shows that no model can separate the classes perfectly; this benchmark still requires joint reasoning over sequence, magnitude, and repayment coverage.

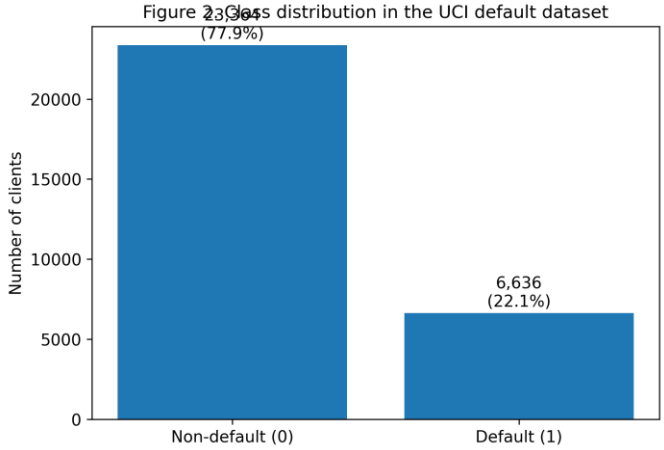

Fig. 2. Default versus non-default distribution in the full benchmark.

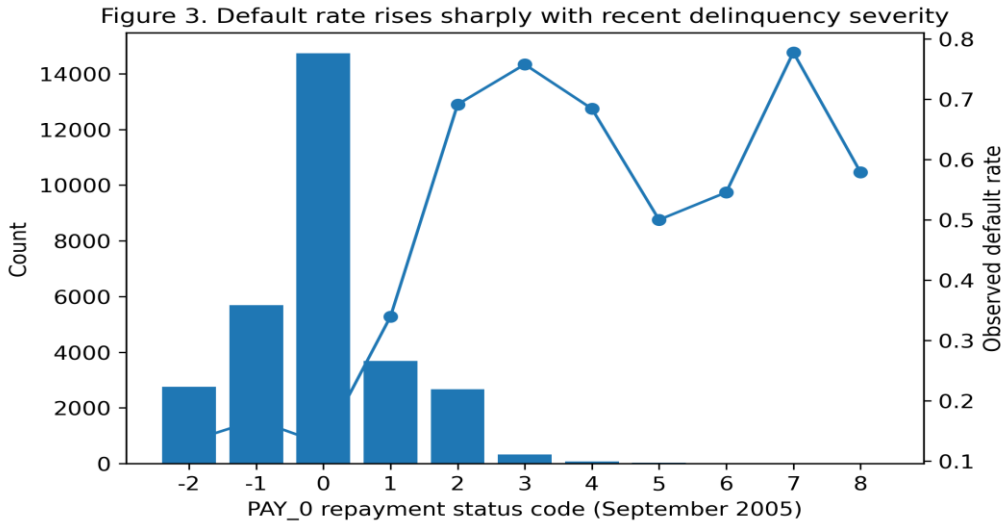

Fig. 3. Repayment-status code PAY_0 versus sample count and observed default rate.

B. Representative split comparison. Table V reports the detailed test-set comparison on the representative split. The best AUROC and AUPRC came from XGBoost with engineered structured features, which

reached AUROC 0.7874 and AUPRC 0.5758. That result confirms the continued strength of gradient boosting on medium-sized financial tables. However, the explanation branch was far from weak. The text-only summary model reached AUROC 0.7760 and AUPRC 0.5551, which is close to several structured baselines. This matters because the text model never saw the original feature names; it learned only from the serialized analyst-like summaries. The result indicates that the summary generator preserved a large share of the predictive information embedded in raw histories.

The fusion results reveal a more nuanced picture than a simple "text always wins" story. Sparse early fusion with XGBoost reached AUROC 0.7842 and AUPRC 0.5724, which was competitive but not superior to the best engineered XGBoost model. Weighted late fusion was more reliable. XGBoost weighted fusion reached AUROC 0.7867, AUPRC 0.5731, F1 0.5524, and accuracy 0.8044. LightGBM weighted fusion reached AUROC 0.7839, AUPRC 0.5703, and F1 0.5483. Relative to LightGBM raw, the weighted fusion version improved AUROC by 0.0088 and AUPRC by 0.0175 on this split. LightGBM raw still had the highest overall accuracy because its threshold produced fewer positive predictions, but its ranking metrics were weaker. This difference between ranking and thresholded accuracy is exactly why an imbalanced default benchmark should not be interpreted through accuracy alone [13], [14].

Table V. Detailed representative-split comparison across structured, text, early-fusion, and late-fusion models.

Model	AUROC	AUPRC	Accuracy	F1	Precision	Recall	Brier	KS
XGBoost engineered	0.7874	0.5758	0.7909	0.5521	0.5244	0.5829	0.1753	0.4380
XGBoost_ weighted fusion	0.7867	0.5731	0.8044	0.5524	0.5592	0.5457	0.1758	0.4392
RF struct ured	0.7855	0.5683	0.7918	0.5436	0.5274	0.5608	0.1612	0.4354
XGBoost fusion	0.7842	0.5724	0.8029	0.5430	0.5571	0.5296	0.1754	0.4409
LightGB M weight ed_fusion	0.7839	0.5703	0.7818	0.5483	0.5055	0.5990	0.1677	0.4376
LateFusio n_meta	0.7829	0.5677	0.8020	0.5511	0.5525	0.5497	0.1836	0.4417
XGBoost raw	0.7824	0.5650	0.7893	0.5416	0.5219	0.5628	0.1776	0.4302
LightGB M engine ered	0.7810	0.5675	0.7878	0.5420	0.5183	0.5678	0.1654	0.4347
Text LR summary	0.7760	0.5551	0.7991	0.5439	0.5461	0.5417	0.1810	0.4349
LightGB M_fusion	0.7760	0.5629	0.7700	0.5406	0.4841	0.6121	0.1593	0.4291

LightGBM_raw	0.7751	0.5528	0.8109	0.5306	0.5880	0.4834	0.1727	0.4278
LR_structured	0.7707	0.5399	0.7862	0.5410	0.5150	0.5698	0.1839	0.4220

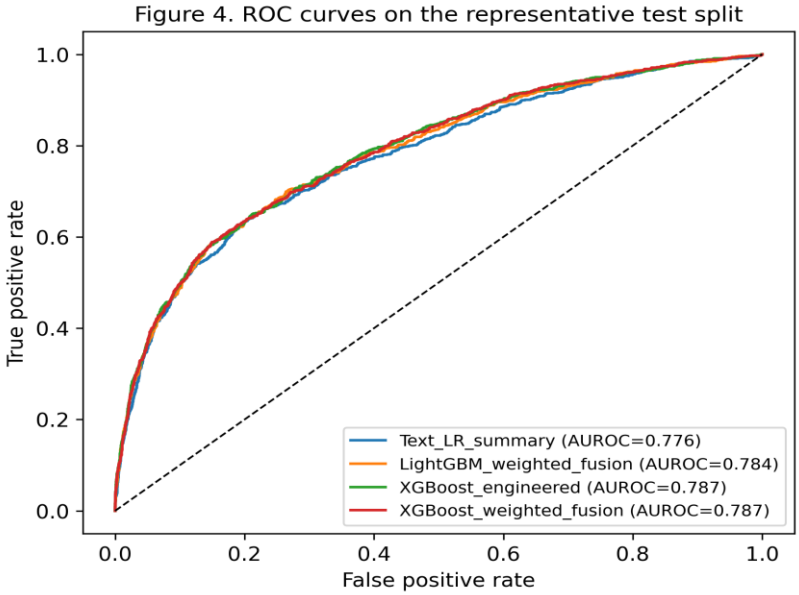

Fig. 4. ROC curves for the main text, boosted, and weighted-fusion variants on the representative test split.

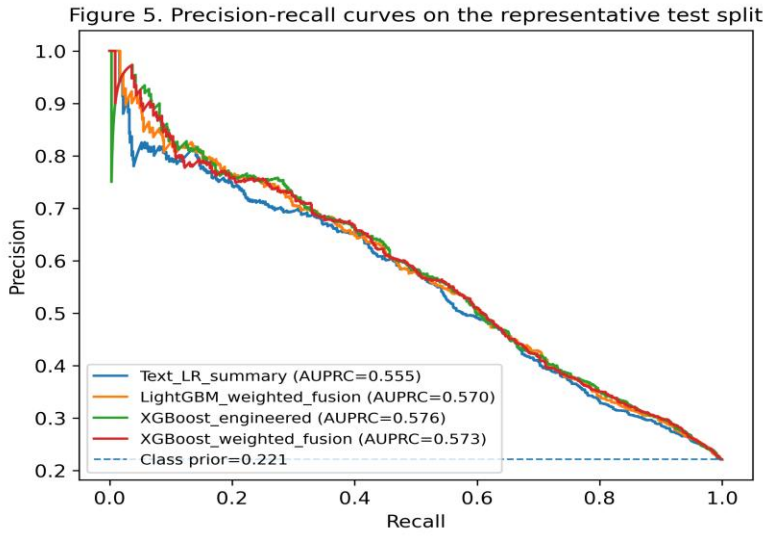

Fig. 5. Precision-recall curves for the same representative-split comparison.

Table VIII makes the operating-point trade-off explicit. XGBoost engineered used a lower threshold and achieved higher recall (0.5829) but lower precision (0.5244). XGBoost weighted fusion used a slightly more conservative threshold and shifted to precision 0.5592 with recall 0.5457, leaving F1 nearly unchanged while increasing accuracy. LightGBM weighted fusion moved in the opposite direction and produced recall 0.5990 with precision 0.5055. The best operational choice therefore depends on the lender's screening policy. A

high-recall collections workflow would favor LightGBM weighted fusion, whereas a tighter manual-review budget would favor the XGBoost weighted system.

C. Robustness across repeated splits. The repeated-split results in Table VI are the main evidence that the explanation layer adds stable value rather than a split-specific fluctuation. Across five independent stratified splits, XGBoost weighted fusion achieved the best mean AUROC (0.7943 ± 0.0111), the best mean AUPRC (0.5706 ± 0.0181), and mean F1 0.5537 ± 0.0159. The corresponding raw XGBoost baseline achieved AUROC 0.7903 and AUPRC 0.5655, so the explanation enhancement improved mean AUROC by 0.0040 and mean AUPRC by 0.0051. The engineered XGBoost model already performed strongly at AUROC 0.7934 and AUPRC 0.5679, and weighted fusion still improved both means slightly.

The LightGBM pattern is even clearer. LightGBM weighted fusion reached AUROC 0.7911 ± 0.0102, AUPRC 0.5686 ± 0.0185, and F1 0.5536 ± 0.0189. The raw LightGBM baseline achieved AUROC 0.7866 and AUPRC 0.5623, while the engineered LightGBM baseline achieved AUROC 0.7859, AUPRC 0.5617, and F1 0.5461. Weighted fusion therefore improved LightGBM not only over the raw branch but also over the engineered branch on all three primary metrics. In other words, the explanation layer was not redundant; it provided complementary structure that the LightGBM branch used consistently.

Another notable result is the strength of the text branch by itself. Across repeated splits, Text_LR_summary achieved AUROC 0.7869 ± 0.0102, AUPRC 0.5618 ± 0.0171, and F1 0.5519. This performance is only slightly below the strongest structured models and above some raw boosted baselines on selected metrics. The explanation summaries were therefore not cosmetic. They formed a compact predictive representation of the original histories.

Bootstrap confidence intervals on the representative split were also informative. The intervals for the top XGBoost variants overlapped, so the gains should be interpreted as modest and stable improvements rather than a step-change in separability. That interpretation is reinforced by the repeated-split averages. The explanation layer yielded small but consistent ranking improvements, which is exactly the kind of gain that matters in a mature tabular benchmark.

Table VI. Mean ± standard deviation over five repeated stratified splits.

Model	AUROC	AUPRC	F1	Accuracy	Precision	Recall	Brier	KS
XGBoost weighted fusion	0.7943 ± 0.0111	0.5706 ± 0.0181	0.5537 ± 0.0159	0.7950 ± 0.0142	0.5358 ± 0.0326	0.5743 ± 0.0162	0.1759 ± 0.0034	0.4489 ± 0.0207
XGBoost engineered	0.7934 ± 0.0117	0.5679 ± 0.0198	0.5513 ± 0.0161	0.7947 ± 0.0128	0.5348 ± 0.0272	0.5701 ± 0.0222	0.1757 ± 0.0037	0.4464 ± 0.0202
LightGBM_weighted_fusion	0.7911 ± 0.0102	0.5686 ± 0.0185	0.5536 ± 0.0189	0.7947 ± 0.0082	0.5333 ± 0.0180	0.5763 ± 0.0299	0.1671 ± 0.0037	0.4473 ± 0.0157
XGBoost raw	0.7903 ± 0.0117	0.5655 ± 0.0175	0.5536 ± 0.0196	0.7940 ± 0.0115	0.5322 ± 0.0250	0.5773 ± 0.0200	0.1770 ± 0.0035	0.4483 ± 0.0269
Text_LR_summary	0.7869 ± 0.0102	0.5618 ± 0.0171	0.5519 ± 0.0179	0.7987 ± 0.0167	0.5461 ± 0.0383	0.5594 ± 0.0140	0.1800 ± 0.0031	0.4423 ± 0.0227
LightGBM_raw	0.7866 ± 0.0138	0.5623 ± 0.0207	0.5514 ± 0.0165	0.7911 ± 0.0151	0.5269 ± 0.0332	0.5795 ± 0.0118	0.1710 ± 0.0047	0.4462 ± 0.0257
LightGBM_engineered	0.7859 ± 0.0106	0.5617 ± 0.0212	0.5461 ± 0.0172	0.7876 ± 0.0175	0.5207 ± 0.0383	0.5773 ± 0.0320	0.1648 ± 0.0041	0.4420 ± 0.0183

D. Summary ablation. Table VII isolates the parts of the generated summary that actually matter. A profile-only text model, which used only demographic and credit-limit statements, achieved AUROC 0.6003 and offered weak discrimination. Adding the six-month repayment sequence immediately raised text AUROC to 0.7542. Adding finance descriptors lifted it again to 0.7689. The full summary with explicit risk tags reached AUROC 0.7760 and AUPRC 0.5551. The fusion branch followed the same trend: fusion AUPRC increased from 0.5477 with profile-only text to 0.5664 with repayment sequence, to 0.5680 with finance descriptors, and to 0.5703 with the full tagged summary. The ablation result is important because it shows that the explanation branch benefits from staged abstraction. Sequence narration provides the largest gain, finance descriptors add complementary numeric context, and explicit tags supply the final sparse cues that help the fusion model.

Table VII. Ablation study of the generated natural-language summary.

Variant	Text_AUROC	Text_AUPRC	Text_F1	Fusion_AUROC	Fusion_AUPRC	Fusion_F1
Profile only	0.6003	0.3128	0.3731	0.7756	0.5477	0.5402
Profile + repayment sequence	0.7542	0.5191	0.5306	0.7818	0.5664	0.5448
Profile + repayment + finance	0.7689	0.5289	0.5355	0.7837	0.5680	0.5481
Full summary + explicit risk tags	0.7760	0.5551	0.5439	0.7839	0.5703	0.5483

E. Calibration, concentration, and subgroup behavior. Table X and Figure 6 show that ranking and calibration are not identical. The best-ranked XGBoost variants had Brier scores around 0.175, whereas LightGBM weighted fusion achieved a better Brier score of 0.1677 and LightGBM engineered achieved 0.1654. This means that LightGBM probabilities were better calibrated even when XGBoost achieved slightly better ranking metrics. The top-decile default rate for XGBoost weighted fusion was 70.91%, corresponding to a lift of 3.206 over the base default rate. For LightGBM weighted fusion the top-decile rate was 69.80% with lift 3.156. These are operationally meaningful concentration levels for portfolio triage.

Table VIII. Validation-selected operating points for the three strongest thresholded classifiers.

Model	Threshold	TP	FP	TN	FN	Precision	Recall	Specificity	F1
XGBoost engineered	0.5550	580	526	2,979	415	0.5244	0.5829	0.8499	0.5521
XGBoost weighted_fusion	0.5950	543	428	3,077	452	0.5592	0.5457	0.8779	0.5524
LightGBM_weigh	0.5150	596	583	2,922	399	0.5055	0.5990	0.8337	0.5483

ted_fusion									

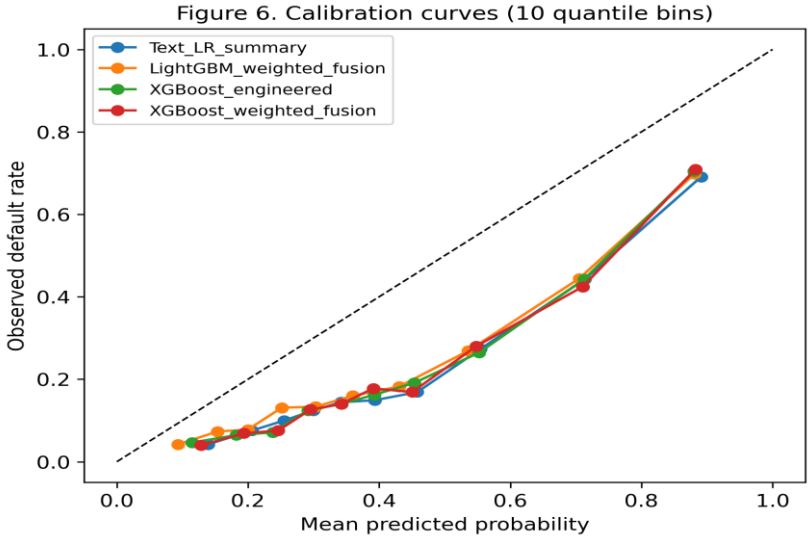

Fig. 6. Calibration curves on the representative test split.

Table IX reports subgroup metrics for XGBoost weighted fusion on the representative split. Performance was broadly stable across sex, with AUROC 0.7786 for males and 0.7905 for females. Variation was slightly larger across age and credit-limit groups. The model performed best in AUROC terms for clients aged 29 or below (0.8006) and for those aged 40 or above (0.7838), while the 30-39 segment was lower at 0.7755. Low-limit accounts had the highest observed default rate, 29.91%, and the model responded with the highest recall in that bucket, 0.6801. High-limit accounts had lower default prevalence and lower recall. These differences suggest that a single global threshold is serviceable but not perfectly aligned with every business segment; group-specific thresholds could be a future operational extension.

Table IX. Subgroup analysis for XGBoost weighted fusion on the representative split.

Dimension	**Group**	**Rows**	**DefaultRate**	**AUROC**	**AUPRC**	**F1**	**Precision**	**Recall**
Sex	Male	1,796	0.2394	0.7786	0.5724	0.5563	0.5500	0.5628
Sex	Female	2,704	0.2089	0.7905	0.5747	0.5493	0.5669	0.5327
Age	<=29	1,457	0.2203	0.8006	0.5738	0.5623	0.5490	0.5763
Age	30-39	1,612	0.2022	0.7755	0.5201	0.5174	0.5325	0.5031
Age	>=40	1,431	0.2432	0.7838	0.6244	0.5757	0.5951	0.5575
Credit limit	<=50k	1,160	0.2991	0.7628	0.6124	0.5982	0.5339	0.6801

Credit limit	50k-150k	1,336	0.2208	0.7827	0.5891	0.5567	0.5645	0.5492
Credit limit	>150k	2,004	0.1761	0.7869	0.5294	0.4874	0.5992	0.4108

Table X. Calibration quality and top-decile lift for the main comparison models.

Model	**Brier**	**ECE_10bin**	**TopDecileDefaultRate**	**TopDecileLift**	**BaseRate**
XGBoost weighted_fusion	0.1758	0.1977	0.7089	3.2060	0.2211
XGBoost_engineered	0.1753	0.1949	0.7044	3.1859	0.2211
LightGBM weighted_fusion	0.1677	0.1703	0.6978	3.1558	0.2211
LightGBM_engineered	0.1654	0.1520	0.6978	3.1558	0.2211
Text LR summary	0.1810	0.2046	0.6911	3.1256	0.2211

F. Qualitative explanation analysis. The qualitative tables and figures show that the explanation layer behaved like a concise risk memo rather than a generic textual paraphrase. In Table III, the high-risk true-positive example contained six delinquent months, persistent multi-month arrears, and very low repayment coverage. The low-risk true-negative example showed revolving credit but no delinquent months and declining billed exposure. The false-positive example contained an apparently severe delinquency pattern but did not default next month, suggesting that the summary branch sometimes over-weights chronic historical arrears when the immediate next-month outcome is temporarily favorable. The false-negative example is equally instructive: the customer defaulted even though the repayment history showed refund or advance-payment codes and no delinquent months. That case indicates that some defaults are triggered by factors not recoverable from the benchmark's six-month history alone.

Figure 7 quantifies the meaning of the generated tags. When the tag risk_recent_severe_delay was present, the observed default rate was 69.55%; when it was absent, the rate was 16.59%. Risk_multi_month_arrears corresponded to 56.37% default when present versus 15.46% when absent. Protective tags behaved in the opposite direction: protect_low_utilization corresponded to a 12.01% default rate when present versus 26.92% when absent. One manually designed tag, risk_rising_balance, was not monotonic in the desired direction. That result is useful rather than embarrassing. It shows why deterministic heuristics should be treated as candidate signals rather than immutable rules; the learned models decide how much weight to assign them.

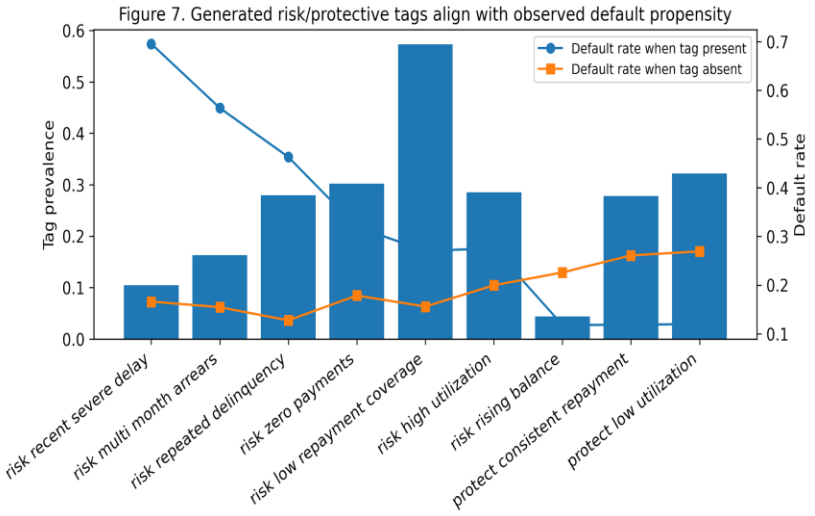

Fig. 7. Prevalence and default-rate effect of the generated risk and protective tags.

Figure 8 explains the top structured XGBoost model through SHAP. The most important features were PAY_recent3_mean, PAY_last_two_max, PAY_max, PAY_0, PAY_AMT1, PAY_ratio_min, PAY_AMT2, BILL_AMT1, LIMIT_BAL, and utilization-related features. This ranking is coherent with the domain. Recent delinquency dominates, but payment coverage and recent bill magnitude also matter materially. The combination of Figures 7 and 8 therefore offers a layered interpretation: the text branch translates histories into human-readable tags, and the structured branch confirms that recent delinquency and repayment coverage are the decisive quantitative drivers.

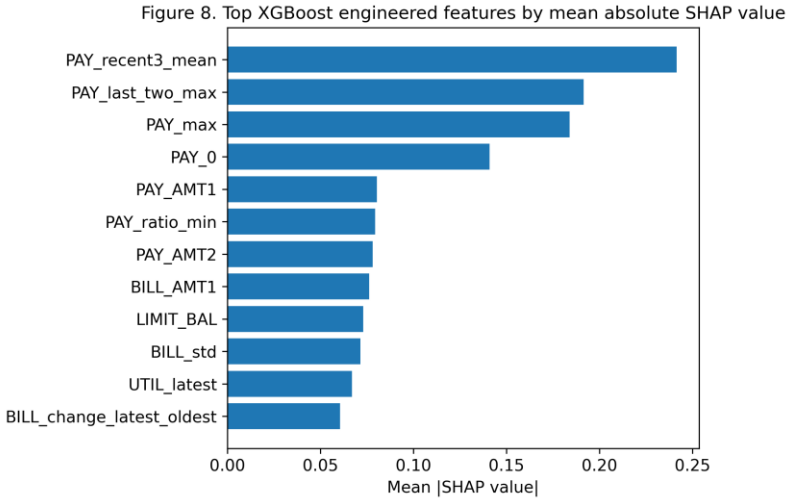

Fig. 8. Mean absolute SHAP importance for the engineered XGBoost model.

Taken together, the results support a specific conclusion. Natural-language behavioral summaries did not replace gradient boosting, and they did not create unrealistic jumps in benchmark performance. What they did was more practically useful: they compressed raw histories into an explanation space that was predictive on its own and complementary when fused with strong tabular learners. On this benchmark, that complementary signal was strong enough to raise mean AUROC and AUPRC in repeated evaluation while simultaneously producing summaries that a human reviewer can inspect directly.

Limitations

The study has four limitations. First, it used a single public dataset from Taiwanese credit card clients observed in 2005 [1], [2]. The dataset is still valuable because it is a standard benchmark, but it does not capture newer lending channels, macroeconomic

regimes, bureau variables, transaction-level signals, or modern affordability features. The reported performance therefore demonstrates methodological behavior on a benchmark, not universal performance for all retail portfolios.

Second, the explanation generator was LLM-style and deterministic rather than a live hosted generative model. That choice was deliberate because the paper required a fully reproducible pipeline with no placeholder values, no hidden prompts, and no dependency on external APIs. The trade-off is that the summaries do not exploit external world knowledge or true free-form generation. A stronger proprietary LLM might produce richer narratives, but it would also introduce prompt variance, version drift, latency, and potentially non-reproducible results.

Third, the fusion rule was a simple convex combination with fixed weights selected on the representative validation split. This design was transparent and stable, but it was not an end-to-end multimodal optimizer. A learned fusion network or a calibration-aware meta-learner might improve the probability estimates further. Similarly, the threshold was tuned for F1. A lender with asymmetric approval or collections costs would likely choose a profit-based or recall-constrained threshold instead.

Fourth, the study focused on predictive quality, calibration, and explanation coherence, but it did not evaluate fairness constraints, reject-inference issues, temporal drift, or portfolio economics such as expected loss, approval rate, or collection yield. These omissions do not invalidate the reported findings; they define the boundary of the claims. The paper demonstrates that explanation-oriented serialization adds measurable signal on a classic credit benchmark. It does not claim that the present pipeline is a production-ready credit policy engine.

Conclusion

This paper presented a reproducible study of LLM-style explanation enhancement for retail credit default prediction on the UCI Default of Credit Card Clients dataset. Six-month bill and repayment histories were converted into deterministic natural-language risk behavior summaries and then fused with gradient-boosted structured models. The empirical evidence was consistent. On the representative split, XGBoost with engineered structured features delivered the highest AUROC and AUPRC, while weighted fusion provided a competitive alternative with strong thresholded performance. Across five repeated stratified splits, XGBoost weighted fusion achieved the best mean AUROC (0.7943) and AUPRC (0.5706), and LightGBM weighted fusion also outperformed its raw and engineered counterparts. The text-only summary model remained competitive, and the ablation study showed that repayment serialization, finance descriptors, and explicit risk tags each contributed measurable value.

The central practical conclusion is straightforward. A controlled natural-language behavior summary can preserve the signal in retail credit histories and can improve gradient boosting when used as a complementary branch. The resulting system remains auditable, reproducible, and easier to inspect than a pure black-box multimodal model. For benchmark-scale retail default prediction, explanation-oriented table-to-text fusion therefore provides a stable and useful extension to standard gradient boosting rather than a replacement for it.

References

[1] I.-C. Yeh and C.-h. Lien, “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients,” Expert Systems with Applications, vol. 36, no. 2, pp. 2473-2480, 2009.

[2] I.-C. Yeh, “Default of Credit Card Clients,” UCI Machine Learning Repository, 2016. doi: 10.24432/C55S3H.

[3] E. I. Altman, “Financial ratios, discriminant analysis and the prediction of corporate bankruptcy,” The Journal of Finance, vol. 23, no. 4, pp. 589-609, 1968.

[4] Yuanzheng Chen, Yitian Zhang, and Matt Sherman, “Going Concern and Bankruptcy Prediction under Extreme Class Imbalance: Cost-Sensitive Learning, Resampling, and Focal Loss with Explainable Financial-Ratio Portraits”, JACS, vol. 4, no. 4, pp. 80–96, Apr. 2024, doi: 10.69987/JACS.2024.40407.

[5] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, “Benchmarking state-of-the-art classification algorithms for credit scoring,” Journal of the Operational Research Society, vol. 54, no. 6, pp. 627-635, 2003.

[6] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, “Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research,” European Journal of Operational Research, vol. 247, no. 1, pp. 124-136, 2015.

[7] L. C. Thomas, D. B. Edelman, and J. N. Crook, Credit Scoring and Its Applications. Philadelphia, PA, USA: SIAM, 2002.

[8] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[10] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems 30, 2017, pp. 3146-3154.

[11] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[12] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[14] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems 30, 2017, pp. 4765-4774.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135-1144.

[17] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30, 2017, pp. 5998-6008.

[18] T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems 33, 2020, pp. 1877-1901.

[19] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems 35, 2022, pp. 24824-24837.

[20] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in Advances in Neural Information Processing Systems 35, 2022, pp. 22199-22213.

[21] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag, "TabLLM: Few-shot classification of tabular data with large language models," in Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, vol. 206, 2023, pp. 5549-5581.

[22] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular data modeling using contextual embeddings," in International Conference on Learning Representations, 2021.

[23] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," in Advances in Neural Information Processing Systems 34, 2021, pp. 18932-18943.

[24] Yuanzheng Chen, Yitian Zhang, David Chau, and Matt Sherman, "Credit Card Default Risk Tiering with Probability Calibration and Uncertainty-Driven Rejection: A Reproducible Study on the UCI Credit Card Clients Dataset", JACS, vol. 3, no. 4, pp. 31–47, Apr. 2023, doi: 10.69987/JACS.2023.30403.

[25] Yifei Lu, Jinyi Mu, and Thao Tran, "Uncertainty-Aware Uplift Modeling for Safer Marketing Targeting: Conformal Prediction and Bayesian Calibration with LCB Policies", JACS, vol. 4, no. 5, pp. 84–101, May 2024, doi: 10.69987/JACS.2024.40507.