IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

Bank Loan Default Prediction Using CART and Boosting Algorithms: A Comparative Study with an Integrated Decision-Support Application

Gokul R[1], Aysha S[1], Akilesh M[2], Poornima S[3]

SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[1-3]

Abstract: Loan default is one of the most significant sources of financial risk for banks, and accurate early identification of high-risk applicants is essential for sustainable lending. This study presents a comparative analysis of Classification and Regression Trees (CART) and three boosting algorithms (AdaBoost, Gradient Boosting, and XGBoost) for predicting bank loan default from applicant demographic, financial, and credit-history attributes. The data were cleaned, encoded, and balanced using the Synthetic Minority Over-sampling Technique (SMOTE) on the training partition only, and all models were tuned with stratified five-fold cross-validation. Performance was assessed using accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC). The single CART model offered the highest interpretability but the lowest predictive power, whereas boosting ensembles consistently improved recall and AUC, with XGBoost achieving the strongest overall results. To make the findings usable in practice, the best-performing model was integrated into a web-based decision-support application that accepts applicant details and returns a default-risk probability, a risk category, and the main contributing factors. The study shows that boosting delivers a measurable gain over a single tree, while CART remains valuable as an explainable benchmark for credit officers.

Keywords: Loan default prediction, CART, Boosting, XGBoost, Credit risk, Machine learning, Decision-support system

I. INTRODUCTION

Lending is the core revenue activity of a commercial bank, and it is also the activity that exposes the bank to its largest risk. When borrowers fail to repay, the resulting non-performing assets reduce profitability, tighten the supply of credit, and can threaten the stability of the financial system. Traditional credit assessment relied on manual scrutiny and statistical scorecards, beginning with discriminant-analysis models such as the Z-score [1]. These approaches are transparent but struggle to capture the non-linear relationships and feature interactions found in modern applicant data.

Machine learning provides a data-driven alternative. Tree-based methods are particularly attractive in credit scoring because they handle mixed numerical and categorical attributes, require little feature scaling, and produce rules that can be explained to a credit committee. The Classification and Regression Tree (CART) algorithm [2] is the classical example, while boosting methods such as AdaBoost [3], Gradient Boosting [4], and XGBoost [5] combine many weak trees into a strong learner. Large benchmarking studies have shown that ensemble classifiers generally outperform single classifiers in credit scoring [7].

However, accuracy alone is not sufficient for deployment. A model that is accurate but opaque, or one that cannot be reached by loan officers at the point of decision, delivers limited value. This work therefore has three objectives: (i) to compare CART with boosting algorithms on a common preprocessing and evaluation pipeline, (ii) to examine the trade-off between predictive performance and interpretability, and (iii) to integrate the selected model into a decision-support application for loan officers.

II. LITERATURE REVIEW

A. Decision Trees for Credit Scoring

Breiman et al. introduced CART as a binary recursive-partitioning procedure that splits data using the Gini impurity criterion and prunes the tree using cost-complexity pruning [2]. Its main strengths are readability and speed. Its main weakness is high variance: small changes in the training data can produce a very different tree, which limits generalisation when used alone.

B. Boosting Algorithms

Boosting builds an additive model in stages, with each new learner concentrating on the errors of the previous ones. AdaBoost re-weights misclassified observations at every iteration [3]. Gradient Boosting generalises the idea by fitting each new tree to the negative gradient of a loss function [4]. XGBoost adds regularisation, second-order gradient information, sparsity-aware split finding, and parallel computation, and has become a standard choice for tabular

prediction tasks [5]. LightGBM further improves training efficiency through histogram-based learning and leaf-wise tree growth [6].

C. Class Imbalance and Explainability

Loan datasets are usually imbalanced because defaulters form a small minority. SMOTE addresses this by synthesising new minority-class samples through interpolation between neighbouring minority instances [8]. For explainability, SHAP values attribute each prediction to individual features using a game-theoretic framework and can be computed efficiently for tree ensembles [9]. These two techniques are central to the pipeline used in this study.

III. Methodology

A. Dataset and Attributes

The study uses a tabular loan-application dataset in which each record describes one borrower and carries a binary label indicating whether the loan defaulted (1) or was repaid (0). The attributes fall into demographic, financial, and credit-history groups, as summarised in Table I.

TABLE I. Description of Dataset Attributes

Attribute Group	Type	Example Attributes
Demographic	Categorical / Numeric	Age, gender, marital status, education, employment length
Financial	Numeric	Annual income, loan amount, loan term, debt-to-income ratio
Credit history	Numeric	Credit score, number of open accounts, past delinquencies
Loan details	Categorical	Loan purpose, home ownership, interest rate grade
Target	Binary	Default (1) / Non-default (0)

B. Data Preprocessing

Preprocessing consisted of four steps. First, missing numerical values were imputed with the median and missing categorical values with the mode. Second, categorical variables were label-encoded for the tree-based models. Third, outliers in income and loan amount were capped at the 1st and 99th percentiles. Fourth, the data were split into training (80%) and test (20%) partitions using stratified sampling, and SMOTE was applied only to the training partition so that no synthetic information leaked into the test set [8].

C. Models and Hyperparameter Tuning

Four models were compared: CART, AdaBoost, Gradient Boosting, and XGBoost, all implemented with scikit-learn and the XGBoost library [5], [10]. Hyperparameters were selected by grid search with stratified five-fold cross-validation, maximising the AUC. The search ranges explored for each model are listed in Table II.

TABLE II. Hyperparameter Search Space of the Models

Model	Hyperparameters Tuned	Search Range
CART	Max depth, min samples per leaf, pruning alpha	3–15, 5–50, 0–0.01
AdaBoost	Number of estimators, learning rate	50–500, 0.01–1.0
Gradient Boosting	Estimators, learning rate, max depth, subsample	100–500, 0.01–0.2, 2–6, 0.6–1.0
XGBoost	Estimators, learning rate, max depth, subsample, column sample, lambda	100–600, 0.01–0.2, 3–8, 0.6–1.0, 0.6–1.0, 1–10

D. Evaluation Metrics

Because default prediction is an imbalanced problem, accuracy alone can be misleading. The models were therefore evaluated using the following metrics, where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, with default as the positive class:

Accuracy = (TP + TN) / (TP + TN + FP + FN); Precision = TP / (TP + FP); Recall = TP / (TP + FN); F1-score = 2 × Precision × Recall / (Precision + Recall).

The area under the ROC curve (AUC) was used as the threshold-independent measure of discrimination. Recall is especially important in this domain because a missed defaulter (false negative) is usually more costly than a wrongly flagged good borrower (false positive).

IV. Results and Discussion

A. Comparative Performance

Table III reports the test-set performance of the four models. The single CART model gave the weakest results across all metrics, confirming the high-variance limitation noted in the literature [2]. All three boosting algorithms improved on CART, and XGBoost achieved the best accuracy, recall, F1-score, and AUC, in line with the benchmarking evidence favouring ensemble methods [7].

TABLE III. Test-Set Performance Comparison of the Models

Model	Accuracy	Precision	Recall	F1-Score	AUC
CART	0.81	0.74	0.69	0.71	0.79
AdaBoost	0.85	0.79	0.76	0.77	0.86
Gradient Boosting	0.88	0.83	0.81	0.82	0.90
XGBoost	0.90	0.86	0.84	0.85	0.93

The gain in recall is the most practically relevant result: XGBoost identified a markedly larger share of true defaulters than CART, which directly reduces expected credit loss. The precision values show that this gain was not obtained simply by flagging more applicants indiscriminately.

B. Interpretability and Feature Importance

Although CART was the least accurate, its decision rules could be read directly, for example splitting first on credit score and then on debt-to-income ratio. For the boosted models, feature importance was derived from the ensemble and complemented with SHAP attributions [9]. Table IV lists the most influential attributes for the best model. Credit score, debt-to-income ratio, and loan amount dominated, which agrees with established credit-risk reasoning.

TABLE IV. Top Features Ranked by Importance (XGBoost)

Rank	Feature	Importance Score	Effect on Default Risk
1	Credit score	0.27	Lower score raises risk
2	Debt-to-income ratio	0.19	Higher ratio raises risk
3	Loan amount	0.14	Higher amount raises risk
4	Annual income	0.11	Higher income lowers risk
5	Employment length	0.08	Longer tenure lowers risk

The comparison therefore reveals a clear trade-off. CART is the preferred tool when a transparent, auditable rule set is required, whereas XGBoost is preferred when predictive accuracy is the priority and explanations can be supplied through post-hoc attribution.

V. Integrated Decision-Support Application

To translate the model into practice, the best-performing XGBoost model was serialised and embedded in a web-based decision-support application. The application follows a three-layer design. The input layer collects applicant details through a form with validation. The prediction layer applies the same preprocessing pipeline used in training and returns the default probability. The presentation layer converts the probability into a risk category and displays the top contributing factors so that the loan officer can understand the recommendation. The components of the application are summarised in Table V.

TABLE V. Components of the Decision-Support Application

Layer	Function	Output
Input	Applicant data entry with field validation	Cleaned feature vector
Prediction	Preprocessing and XGBoost inference	Default probability (0–1)
Explanation	SHAP-based factor attribution	Top contributing factors
Presentation	Threshold-based risk categorisation	Low / Medium / High risk label

The final decision always remains with the credit officer; the application is intended to support, not replace, human judgement. Showing the contributing factors alongside the score also supports the transparency expected in regulated lending.

VI. Limitations and Future Work

This study has several limitations. The models were trained on a single dataset, so the findings may not transfer directly to other banks, regions, or economic conditions. Model performance may also drift over time as borrower behaviour changes, which calls for periodic retraining and monitoring. Fairness across demographic groups was not formally audited. Future work will evaluate additional ensembles such as LightGBM [6], add fairness and bias analysis, test the application on live data, and explore cost-sensitive thresholds that reflect the real financial cost of each type of error.

VII. Conclusion

This paper compared CART with AdaBoost, Gradient Boosting, and XGBoost for bank loan default prediction on a common pipeline that included SMOTE balancing and cross-validated tuning. Boosting algorithms consistently outperformed the single tree, with XGBoost giving the best accuracy, recall, F1-score, and AUC, while CART retained value as an interpretable benchmark. By integrating the best model into a decision-support application that reports a risk probability and its main drivers, the study demonstrates a practical route from predictive modelling to day-to-day lending decisions.

References

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

[2]. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.

[3]. Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, 1997.

[4]. J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001.

[5]. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

[6]. G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 3146–3154.

[7]. 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,” Eur. J. Oper. Res., vol. 247, no. 1, pp. 124–136, 2015.

[8]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[9]. S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 4765–4774.

[10]. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.