IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

Credit Score Classification And Forecasting Using Gradient-Boosted Decision Trees

Adorn S. George[1], Arpita Biswal[2]

Department of Computer Science and Engineering (Emerging Technologies),

SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[1,2]

Abstract: Credit scoring systems classify borrowers according to their likelihood of repaying credit. This paper presents a machine learning pipeline for classifying monthly credit scores into Poor, Standard, and Good categories and forecasting scores for existing customers. Experiments use the public Kaggle credit-score classification dataset, containing 100,000 labelled monthly records from 12,500 customers and 50,000 unlabelled test records. The dataset contains malformed numeric values, placeholder categories, missing fields, and implausible outliers. A customer-aware cleaning strategy repairs records using only the same customer's observations. To prevent information leakage caused by repeated customer records, the study evaluates new-customer classification with customer-grouped cross-validation and evaluates forecasting with temporal hold-outs. LightGBM is compared with logistic regression and random forest. The features-only model achieves 71.27% accuracy and 0.6922 macro-F1 on unseen customers. A second model that incorporates known score history achieves 79.50% accuracy and 0.7967 macro-F1 for one-to-four-month forecasting. Random row splitting raises the apparent accuracy to 80.31%, demonstrating the effect of customer leakage. The findings also show that 58.3% of customers change score during the observed period despite largely stable financial profiles, limiting forecastability.

Keywords: credit scoring, multi-class classification, gradient boosting, LightGBM, data cleaning, data leakage, grouped cross-validation, forecasting.

I. INTRODUCTION

Credit scores summarise a borrower's credit standing and are used by lenders when making decisions about loan and credit-card approval, interest rates, credit limits, and account monitoring. Traditional scorecards commonly use variables such as income, outstanding debt, and repayment history. As lending activity has moved online, machine learning methods have become useful for learning complex relationships from large collections of structured financial records. Tree-based ensembles are particularly effective for tabular data because they capture non-linear relationships and can work with mixed feature types [3]–[8].

The dataset considered in this study contains repeated monthly records for each customer and deliberately corrupted values. A random row-wise split can place records from the same customer in both training and validation sets, producing an overly optimistic estimate. The study therefore addresses three linked problems: repairing corrupted data without cross-customer leakage, evaluating new-customer classification fairly, and forecasting later monthly scores for customers whose earlier scores are known.

The main contributions are a per-customer data-repair procedure, an evaluation design that separates unseen-customer classification from existing-customer forecasting, and an analysis of the performance ceiling arising from score changes that are not explained by stable financial inputs.

II. RELATED WORK

Classical consumer credit scoring has used discriminant analysis, logistic regression, decision trees, and nearest-neighbour methods. Hand and Henley [1] reviewed these approaches, while Thomas et al. [2] described the practical use of scorecards and the interpretability of logistic regression. Comparative studies found that modern ensemble methods, including random forests and gradient boosting, are strong alternatives for credit scoring [3], [4].

Random forests combine multiple decision trees to reduce variance [5]. Gradient boosting builds trees sequentially, with each new tree correcting errors made by earlier trees. XGBoost introduced an efficient and regularised boosting framework [6], while LightGBM uses histogram-based splitting and leaf-wise growth to improve training efficiency [7]. Tree-based methods also remain competitive on typical tabular datasets [8].

 This work is licensed under a Creative Commons Attribution 4.0 International License

Evaluation design is critical when records share an entity. Data leakage occurs when information unavailable at prediction time enters model training or validation [9]. Since credit bands have a natural order (Poor, Standard, Good), ordinal classification is also relevant [10]. This work uses grouped validation and scikit-learn utilities [11] to avoid customer overlap in the new-customer evaluation.

III. DATASET AND METHODOLOGY

A. Dataset Description

The public Kaggle "Credit score classification" dataset [12] contains 100,000 training rows with 28 columns and 50,000 test rows with 27 columns. The records describe 12,500 customers. January–August observations are labelled training data, while September–December observations form the unlabelled test set. The target variable has three classes: Standard (53.2%), Poor (29.0%), and Good (17.8%). During the eight labelled months, 58.3% of customers change score at least once.

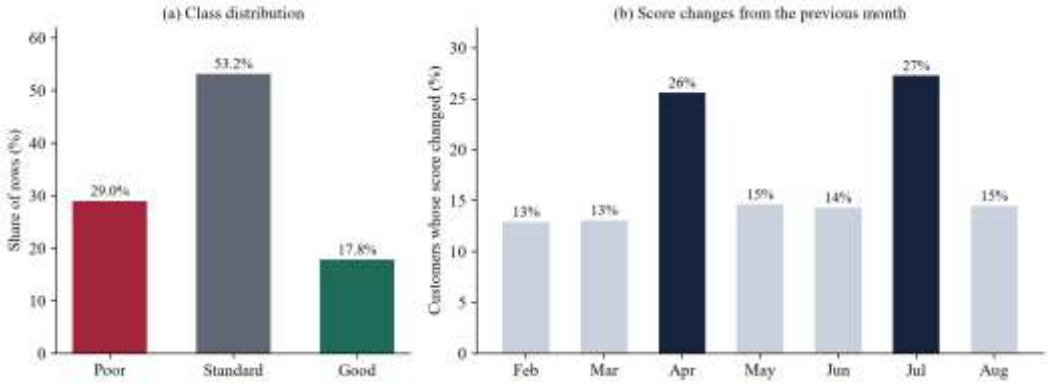


Fig. 1. Class distribution and monthly share of customers whose score changed.

B. Data Cleaning

The raw data includes numeric values stored as text with stray underscores, junk placeholders such as "_____", "_", and "!@9#%8", implausible outliers, and missing values. Before repair, 46,951 of the 100,000 training rows contain at least one missing value. Examples include ages outside 14–100, bank-account counts as high as 1,798, and interest rates as high as 5,797%.

The cleaning process converts numeric text to numbers, masks invalid ranges and sentinel values, and repairs values using the customer's own monthly observations. Stable numeric and categorical attributes are replaced by the customer-specific mode after invalid values are removed. Age is reconstructed from neighbouring months after implausible values are discarded. Slowly varying attributes are filled forward and backward within each customer. Freely varying numeric attributes use the customer-specific median. Credit-history age is converted to months and reconstructed from the customer's monthly offset. Identifiers and personal fields are not used as model inputs.

Attribute group	Examples	Repair strategy
Stable per customer	Income, bank accounts, cards, interest rate, loans, debt, occupation, credit mix	Remove implausible values; replace using customer-specific mode
Age	Age	Keep plausible values and values close to customer's typical age; fill from neighbouring months
Slowly varying	Payment delays, inquiries, credit-limit change, payment behaviour	Remove invalid values; forward-fill then backward-fill within customer
Freely varying	Investment, monthly balance, utilisation ratio	Remove sentinels; fill using customer-specific median
Credit-history age	Credit_History_Age	Convert to months and reconstruct from the customer-specific monthly offset

The repair uses only feature values belonging to the same customer and does not use the target label. Training and test records are cleaned together so that the repair process can use all twelve monthly observations for each customer. This procedure is designed to prevent information transfer between different customers or from the target label.

 This work is licensed under a Creative Commons Attribution 4.0 International License

C. Feature Engineering

ID, Customer_ID, Name, and SSN are excluded from model inputs. Credit_Mix is ordinally encoded as Bad = 0, Standard = 1, and Good = 2. Payment_Behaviour is separated into spending level and payment size; occupation and minimum-payment status are one-hot encoded; and Type_of_Loan is expanded into nine binary indicators. Additional affordability ratios include debt-to-income, EMI-to-salary, investment-to-salary, balance-to-salary, and credit cards per bank account. Past-only trend features are generated for eleven month-to-month signals. For each signal, the feature set includes the customer's expanding average, the current value's deviation from that average, and the change from the previous month. The month index is excluded because the training period ends in August and a tree model cannot extrapolate a numeric month trend to September–December. Model A contains 85 input features.

D. Model Selection and Training

Three classifiers are compared: multinomial logistic regression, random forest with 200 trees, and LightGBM gradient-boosted decision trees. LightGBM is selected as the main model because it performs best in the experiments, handles missing values natively, and trains efficiently on the dataset. The multi-class model produces a score for each of the three classes and converts the scores to class probabilities using softmax. Training minimises multi-class log loss.

The final LightGBM configuration uses a learning rate of 0.02, 31 leaves, a minimum of 100 samples per leaf, feature and row sampling of 0.8, and L2 regularisation of 5.0. Early stopping is used during model selection. The final Model A uses 627 trees and Model B uses 600 trees.

Parameter	Value
Objective	Multiclass (softmax)
Learning rate	0.02
Number of leaves	31
Minimum child samples	100
Feature / row sampling	0.8 / 0.8
L2 regularisation	5.0
Final estimators	Model A: 627; Model B: 600

IV. FORECASTING MODELS AND EVALUATION

A. Model A: Features-Only Classification

Model A predicts the credit band of a customer not previously seen by the model. It uses the 85 financial and behavioural features and is evaluated with StratifiedGroupKFold using five folds grouped by Customer_ID. All monthly records for a customer remain in the same fold. A separate customer-grouped validation subset is used for early stopping, preventing the validation fold from being used for tuning.

B. Model B: Score-History Forecasting

Model B addresses existing customers whose earlier scores are known. For a selected cutoff month, it predicts the next one to four months. Its inputs combine the target month's financial features with score-history features: last and previous score, proportions of past months in each class, average score, number of score changes, forecast horizon, quarterly change-point indicators, and changes in monthly signals since the cutoff. Training examples are created from cutoffs one to seven. For the test set, August is the cutoff and September–December are forecast.

C. Validation Strategy and Metrics

Three evaluation schemes are used. First, customer-grouped five-fold cross-validation measures generalisation to unseen customers. Second, temporal hold-outs train on earlier months and predict later months for the same customers, matching the forecasting task. Third, a random row split is used only to quantify the inflation caused by leakage. Performance is reported using accuracy and macro-F1. Macro-F1 is the unweighted mean of class-wise F1 scores and gives each class equal weight despite class imbalance.

V. SYSTEM IMPLEMENTATION

The implementation is organised into data cleaning, feature engineering, score-history generation, training and evaluation, dashboard export, and dashboard presentation modules. The pipeline is implemented in Python using pandas, NumPy, scikit-learn, LightGBM, Matplotlib, and joblib. The interactive front end uses React, Vite, and Tailwind CSS.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

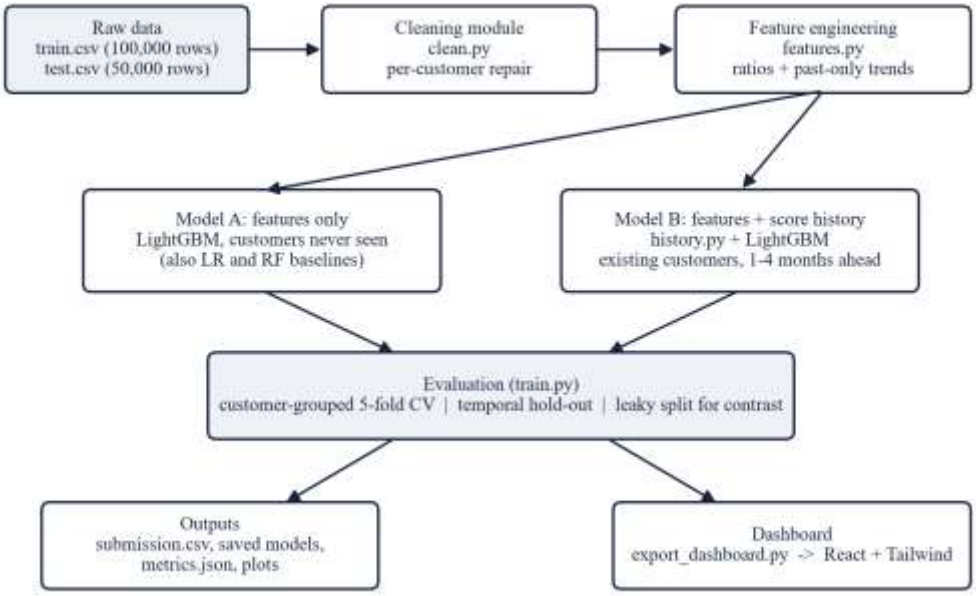


Fig. 2. System architecture: raw data is cleaned and transformed before classification, forecasting, evaluation, and dashboard export.

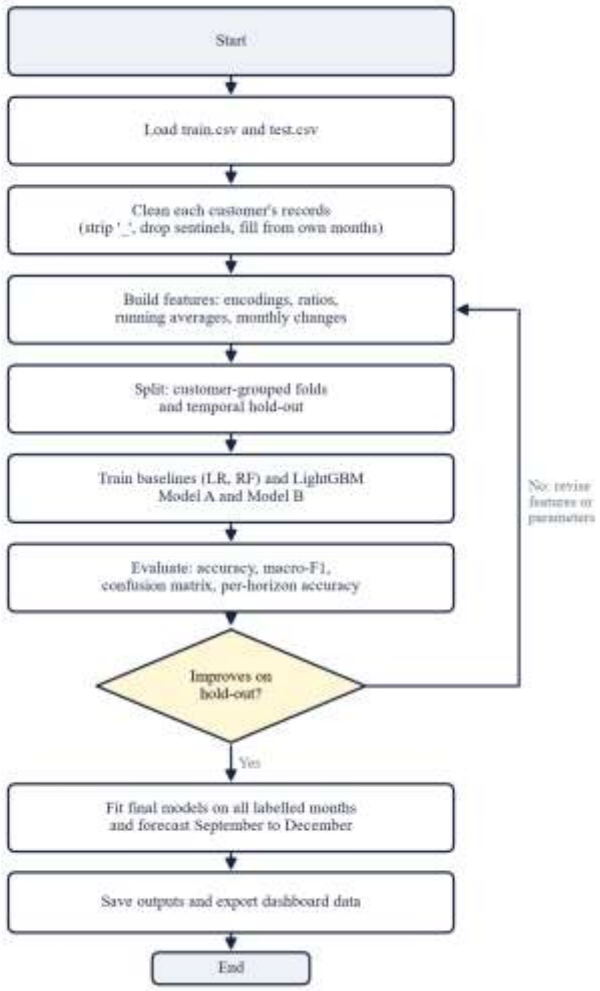


Fig. 3. Model development and evaluation workflow.

The dashboard presents a customer's January–December score sequence, distinguishing recorded scores from September–December forecasts. It also provides a mode for checking predictions against known May–August outcomes, model accuracy comparisons, accuracy by forecast horizon, feature drivers, limitations, and confusion matrices. The dashboard is responsive and supports light and dark themes.

VI. RESULTS AND DISCUSSION

A. Overall Performance

Table I summarises the results on held-out data. Among features-only models, LightGBM achieves the highest customer-grouped accuracy at 71.27%, compared with 70.76% for random forest and 66.73% for logistic regression. Model B achieves 79.50% accuracy when scores are known through April and May–August are forecast. The last-known-score baseline reaches 79.02%.

TABLE I HELD-OUT PERFORMANCE

Model	Evaluation	Accuracy	Macro-F1
Logistic regression	Customer-grouped 5-fold CV	66.73%	0.6384
Random forest	Customer-grouped 5-fold CV	70.76%	0.6862
Model A: LightGBM	Customer-grouped 5-fold CV	71.27%	0.6922
Model A	Temporal: Jan–Jun to Jul–Aug	68.79%	0.6753
Repeat last known score	Scores to Apr; predict May–Aug	79.02%	0.7866
Model B: LightGBM + history	Scores to Apr; predict May–Aug	79.50%	0.7967
Repeat last known score	Scores to Jun; predict Jul–Aug	72.86%	0.7221
Model B	Scores to Jun; predict Jul–Aug	76.62%	0.7663
Model A (leaky contrast)	Random row split	80.31%	0.7940

The random row split reports 80.31% accuracy, approximately nine percentage points above the honest customer-grouped result for Model A. This difference is attributable to the model seeing other months from the same customers during training. Consequently, random-split results on this dataset should not be interpreted as performance on new applicants.

B. Class-Wise Results

TABLE II PER-CLASS PRECISION, RECALL AND F1

Class	A Precision	A Recall	A F1	B Precision	B Recall	B F1
Poor	0.7275	0.6661	0.6955	0.7380	0.9239	0.8206
Standard	0.7480	0.7538	0.7509	0.8821	0.6972	0.7789
Good	0.5979	0.6663	0.6302	0.7297	0.8625	0.7905

Model A is weakest on the Good class, which is the smallest class. Model B has high recall for Poor and Good, but it can miss customers moving back to Standard. Across 50,000 forecasts, Model B confuses Poor with Good only 14 times; its errors are nearly always between adjacent score bands.

C. Forecast Horizon and Accuracy Limit

Model B achieves 84.27% accuracy one month ahead and 84.54% two months ahead. Accuracy falls to 75.02% at three months and 74.18% at four months. Since the test set contains all four forecast horizons, the overall accuracy is just below 80%.

A customer's score remains unchanged from the previous month in 82.5% of transitions. However, 58.3% of customers change score at least once during the year while much of their financial profile remains stable. When the model inputs do not change but the target label does, the change cannot be inferred from those inputs. Experiments using neighbouring months' inputs show no meaningful hidden time shift.

 This work is licensed under a Creative Commons Attribution 4.0 International License

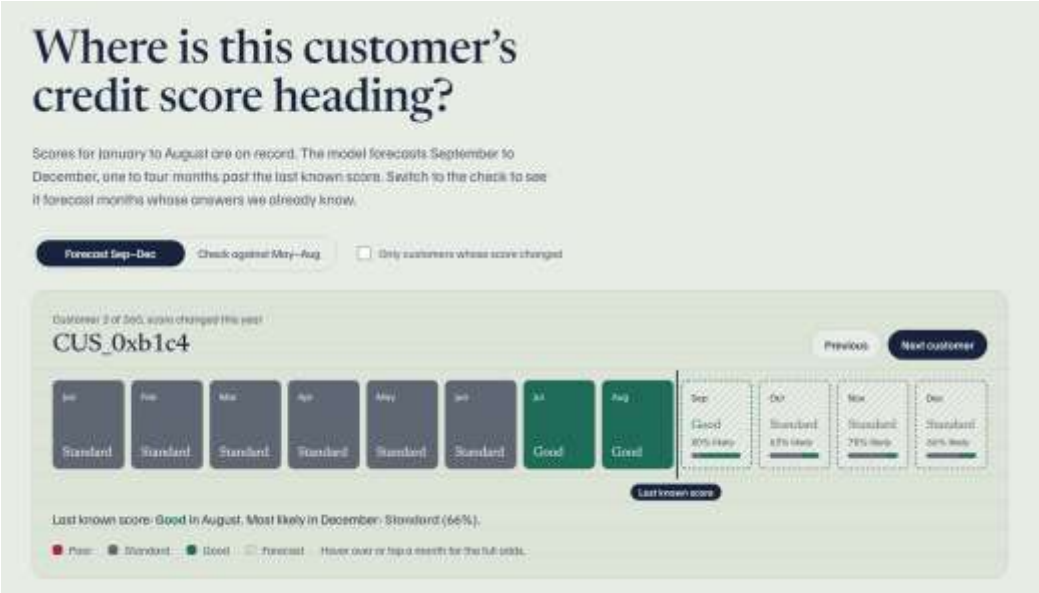


Fig. 4. Dashboard view showing an individual customer's recorded scores and forecast months.

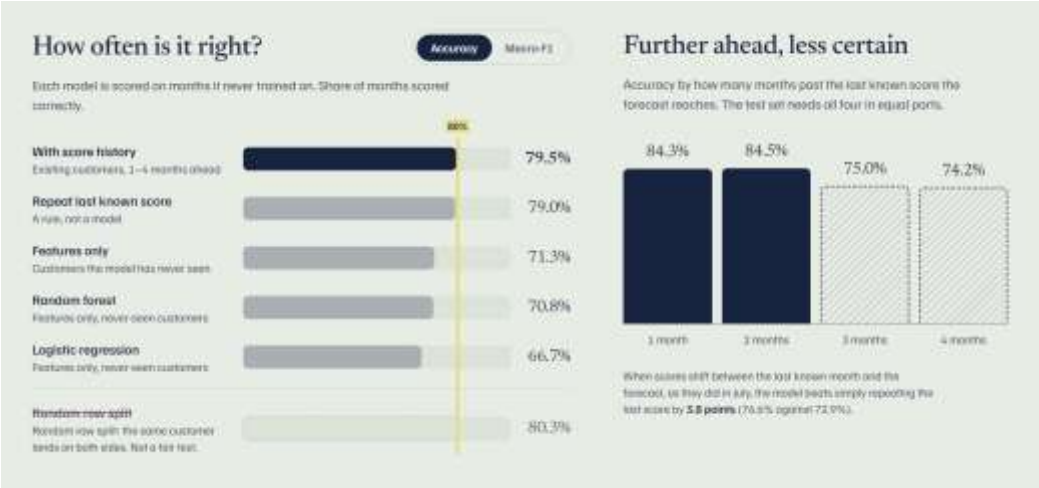


Fig. 5. Dashboard comparison of model accuracy and performance by forecast horizon.

D. Feature Interpretation

For Model A, the leading features are credit mix (17.4% share of effect), outstanding debt (12.6%), and interest rate (10.1%), followed by spending level relative to the customer's average, number of credit cards, and payment delays. A Good credit mix pushes predictions towards Good, while high debt and interest rates push towards Poor. Spending more than usual and making larger payments than usual tend to move predictions towards Good.

Model B is dominated by the customer's own score history. The share of months previously rated Poor contributes 17.0% of the effect, the last known score 12.8%, and the share of months rated Good 10.8%. Spending and payment changes provide additional adjustments to the history-based forecast. SHAP values are used to examine the direction of feature contributions [13], [14].

E. Test-Set Forecast

For the 50,000 unlabelled September–December test rows, Model B predicts 49.8% Standard, 30.4% Poor, and 19.8% Good. Since the test set contains no labels, its accuracy cannot be measured directly; the temporal hold-out results provide the estimate of expected performance.

VII. CONCLUSION AND FUTURE WORK

This study developed a customer-aware machine learning pipeline for credit score classification and forecasting. Per-customer cleaning repairs corrupted values while avoiding cross-customer and target leakage. LightGBM outperforms

logistic regression and random forest for new-customer classification, achieving 71.27% accuracy and 0.6922 macro-F1 under customer-grouped cross-validation. Incorporating known score history raises forecasting performance to 79.50% accuracy and 0.7967 macro-F1 for one-to-four-month predictions.

The experiments demonstrate that random row splitting inflates the apparent performance to 80.31%, while customer-grouped evaluation gives a more realistic estimate for unseen customers. Forecast accuracy is also limited by score changes that are not explained by stable financial inputs. The dataset is synthetic, test labels are unavailable, quarterly change-point assumptions cannot be verified on the test period, and fairness across demographic groups was not assessed. Future work includes ordinal classification, customer-level SHAP explanations, probability calibration, sequence-based models, deployment as a live scoring API, and fairness and data-drift monitoring.

ACKNOWLEDGMENT

The authors thank S. Poornima, Assistant Professor, Department of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, for guidance and support during this project.

REFERENCES

[1]. D. J. Hand and W. E. Henley, “Statistical classification methods in consumer credit scoring: a review,” Journal of the Royal Statistical Society: Series A, vol. 160, no. 3, pp. 523–541, 1997.
[2]. L. C. Thomas, D. B. Edelman and J. N. Crook, Credit Scoring and Its Applications. Philadelphia: SIAM, 2002.
[3]. 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.
[4]. 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.
[5]. L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[6]. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[7]. G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Advances in Neural Information Processing Systems 30, 2017, pp. 3146–3154.
[8]. L. Grinsztajn, E. Oyallon and G. Varoquaux, “Why do tree-based models still outperform deep learning on typical tabular data?,” in Advances in Neural Information Processing Systems 35, 2022.
[9]. S. Kaufman, S. Rosset, C. Perlich and O. Stitelman, “Leakage in data mining: Formulation, detection, and avoidance,” ACM Transactions on Knowledge Discovery from Data, vol. 6, no. 4, article 15, 2012.
[10]. E. Frank and M. Hall, “A simple approach to ordinal classification,” in Proc. 12th European Conf. Machine Learning, 2001, pp. 145–156.
[11]. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[12]. parisrohan, “Credit score classification,” Kaggle dataset. [Online]. Available: https://www.kaggle.com/datasets/parisrohan/credit-score-classification
[13]. 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.
[14]. S. M. Lundberg et al., “From local explanations to global understanding with explainable AI for trees,” Nature Machine Intelligence, vol. 2, no. 1, pp. 56–67, 2020.

 This work is licensed under a Creative Commons Attribution 4.0 International License