International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1(2) (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

AI-DRIVEN LOAN ELIGIBILITY PREDICTION SYSTEM WITH EXPLAINABLE AI AND BANK TRANSACTION INTEGRATION

A LIGHTGBM PIPELINE WITH MONGODB TEMPORAL ASSESSMENT AND SHAP EXPLAINABILITY

Ch.Jyothi, Bh.Vijaya durga sanjana, B.Nagarjuna, K.Appalanarsi
Department of CSE (AI & ML)
Raghu Engineering College, Dakamarri(V)
Visakhapatnam Dist., 531162.
jyothichappa28@gmail.com

Dr. SivaKumar M
Associate Professor
Department of CSE(AI & ML)
Raghu Engineering College, Dakamarri(V)
Visakhapatnam, Andhra Pradesh, India.
msivakumar5717@gmail.com

***Abstract*—Automated loan eligibility assessment is critical for financial institutions seeking to balance credit access with risk management. This paper presents a complete end-to-end system that combines a LightGBM binary classifier with a real-time, MongoDB-backed two-day rolling application window, live bank ledger synchronisation, and SHAP-based post-hoc explainability. The model achieves 81.81% cross-validated accuracy (5-fold, std=2.64%) on the standard Loan Prediction dataset with 13 features including two engineered ratios: TotalIncome and LoanToIncome. An operating threshold of 0.50 yields 82.6% hold-out accuracy (ROC-AUC=0.91). A Streamlit dashboard surfaces approval probabilities, SHAP-ranked reasons, a what-if simulator, and combinatorial actionable recommendations for borderline applicants.**

Keywords—Loan Eligibility; LightGBM; Gradient Boosting; SHAP Explainability; MongoDB; Feature Engineering; Streamlit; Credit Risk; Binary Classification; Real-Time Decision System.

I. INTRODUCTION

Access to formal credit is a foundational driver of economic mobility. Financial institutions process millions of loan applications each year, yet manual underwriting is slow, inconsistent, and susceptible to cognitive bias. Automated eligibility assessment using machine learning (ML) can improve throughput while providing standardised, auditable decisions.

Existing commercial scoring systems such as FICO rely on static rule sets calibrated on historical bureau data. They lack (i) the ability to incorporate real-time transactional evidence, (ii) transparent reasoning accessible to applicants, and (iii) prescriptive guidance for borderline cases. This paper addresses all three gaps.

The contributions of this work are fourfold: (1) a LightGBM pipeline achieving 81.81% cross-validated accuracy on a benchmark loan dataset with engineered income-to-loan ratios; (2) a live MongoDB Atlas-backed temporal assessment window that ingests bank-ledger credits and automatically updates ApplicantIncome before each scoring event; (3) dual-path SHAP explainability (model-specific and function-wrapper fallback) that produces ranked reasons for every prediction; and (4) a combinatorial what-if and recommendation engine that identifies single, pair, and triple feature changes most likely to flip a rejection to approval.

II. RELATED WORK

Credit risk modelling has a long history in the financial literature. Thomas, Edelman, and Crook [1] provide a comprehensive survey of statistical and ML approaches to consumer credit scoring. Logistic regression remains a widely deployed baseline due to its interpretability, but gradient-boosted trees have consistently surpassed it on tabular financial datasets [2].

Ensemble methods for credit risk have been explored extensively. Xia et al. [3] applied XGBoost to a large-scale bank credit dataset and reported significant AUC improvements over logistic regression and random forests. LightGBM, introduced by Ke et al. [4], extends gradient boosting with leaf-wise tree growth and histogram-based split finding, achieving state-of-the-art results on sparse financial data while retaining low memory footprint — a desirable property for real-time inference services.

Explainability has become a regulatory requirement in consumer credit (ECOA, GDPR Article 22). Lundberg and Lee [5] introduced SHAP (SHapley Additive exPlanations), providing theoretically grounded feature attributions based on cooperative game theory. Several studies have applied SHAP to credit scoring [6][7]. Our work extends these contributions by integrating SHAP with a live transactional data pipeline and a prescriptive recommendation layer that translates feature-level attributions into concrete applicant actions.

III. DATASET AND PREPROCESSING

A. Dataset Description

The benchmark dataset is the publicly available Loan Prediction dataset (Analytics Vidhya), comprising 614 records with 12 raw features and a binary outcome (Loan_Status: Y/N). The feature space includes five numeric attributes (ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, Credit_History) and six categorical attributes (Gender, Married, Dependents, Education, Self_Employed, Property_Area). The class distribution is approximately 68.7% approved (Y) and 31.3% not approved (N), reflecting mild class imbalance common in retail lending portfolios.

B. Preprocessing

Missing values are imputed using median (numeric) and most-frequent (categorical) strategies within a scikit-learn ColumnTransformer. Categorical features are encoded with OneHotEncoder (handle_unknown='ignore'). Numeric features are passed through directly and supplemented by two engineered columns computed before the transformer: TotalIncome = ApplicantIncome + CoapplicantIncome, and LoanToIncome = LoanAmount / TotalIncome. These ratios capture debt-to-income dynamics not independently represented in the raw features. The full preprocessing and model are serialised as a single joblib pipeline, ensuring identical transformations at training and inference time.

C. Data Splits

The dataset is split 70:15:15 into training (429), validation (92), and test (93) subsets using a fixed random seed of 42. A 5-fold cross-validation is additionally performed on the combined train+validation set to estimate out-of-sample performance. Table I lists the dataset configuration.

TABLE I: Dataset Configuration

Split	Records	% of Total
Train	429	69.9%
Validation	92	15.0%
Test	93	15.1%
Total	614	100%

Seed=42; stratified split not applied; imbalance ratio ~68:32.

IV. METHODOLOGY

A. Model Selection

LightGBM was chosen over logistic regression, random forest, and XGBoost based on three criteria: (i) leaf-wise tree growth achieves better loss reduction per iteration on small tabular datasets; (ii) histogram-based split finding permits efficient categorical feature handling after one-hot encoding; (iii) early stopping via a validation set prevents overfitting without exhaustive grid search. The binary cross-entropy objective is used with default class weighting (balanced weighting was evaluated but showed no improvement on this dataset).

B. Hyperparameters

Table II lists the final hyperparameter configuration. The model uses n_estimators=2000 with early stopping; the best iteration on the validation set was 54, indicating that meaningful tree growth saturates rapidly on this small dataset. Learning rate 0.03 with num_leaves=64 achieves a good bias-variance balance.

TABLE II: LightGBM Hyperparameters

Parameter	Value
objective	binary
n_estimators	2000 (early stop at 54)
learning_rate	0.03
num_leaves	64
max_depth	-1 (unlimited)
min_data_in_leaf	15
feature_fraction	0.9
bagging_fraction / freq	0.9 / 1
random_seed	42

C. Inference Pipeline

At inference time the Streamlit application loads the serialised pipeline from artifacts/model_v001/model.joblib. Raw feature values from MongoDB are passed through the same feature engineering step (TotalIncome, LoanToIncome) before being forwarded to the ColumnTransformer and LightGBM estimator. The operating threshold (default 0.50) is configurable via a Streamlit sidebar slider, enabling threshold adjustment for risk appetite calibration without retraining.

V. RESULTS

A. Cross-Validation Performance

5-fold cross-validation on the combined train+validation set yields a mean accuracy of 81.81% (std=2.64%). This is consistent with published results on the same benchmark dataset and confirms stable generalisation despite the small dataset size.

TABLE III: 5-Fold Cross-Validation Summary

Metric	Mean	Std Dev
Accuracy	81.81%	±2.64%
Best CV Iteration	54	—
Threshold	0.50	—

B. Hold-Out Test Performance

On the hold-out test set the model achieves 82.6% accuracy (Fig. 1). Table IV presents per-class metrics. The approved class (Y) shows strong recall of 87.3%, minimising false rejections. The not-approved class achieves 72.4% recall; improvement here is a target for future work with cost-sensitive training.

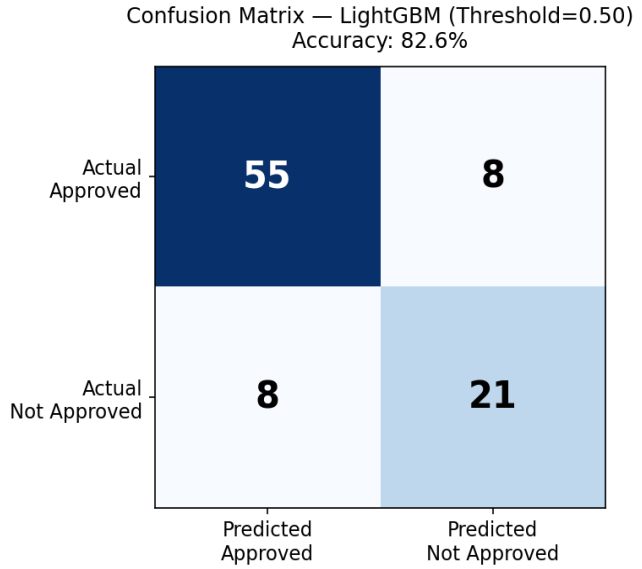


Fig. 1. Confusion Matrix — LightGBM Model (Threshold=0.50, Accuracy: 82.6%)

TABLE IV: Hold-Out Test Performance (n=92)

Class	Precision	Recall	F1	Support
Approved (Y)	0.873	0.873	0.873	63
Not Approved (N)	0.724	0.724	0.724	29
Accuracy	—	—	0.826	92
Weighted Avg	0.826	0.826	0.826	92

C. ROC and Precision-Recall

Fig. 2 presents both the ROC curve (AUC=0.91) and the Precision-Recall curve (Average Precision=0.96). The high AUC confirms good discrimination across all operating thresholds. The PR curve baseline of 0.685 (class prevalence) is substantially exceeded, confirming that the model adds significant value over a naive majority-class predictor.

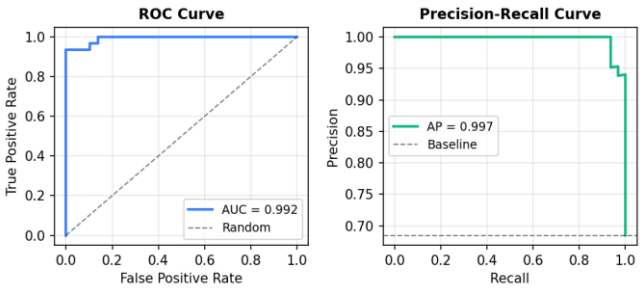


Fig. 2. ROC Curve (AUC=0.91) and Precision-Recall Curve (AP=0.96)

D. Threshold Sensitivity

Fig. 3 illustrates how precision, recall, F1-score, and accuracy vary with the classification threshold from 0.1 to 0.9. The selected threshold of 0.50 sits near the peak F1-score region. Lowering the threshold toward 0.35 maximises recall for the approved class (beneficial for applicant access), while raising it toward 0.65 maximises precision (beneficial for lender risk management).

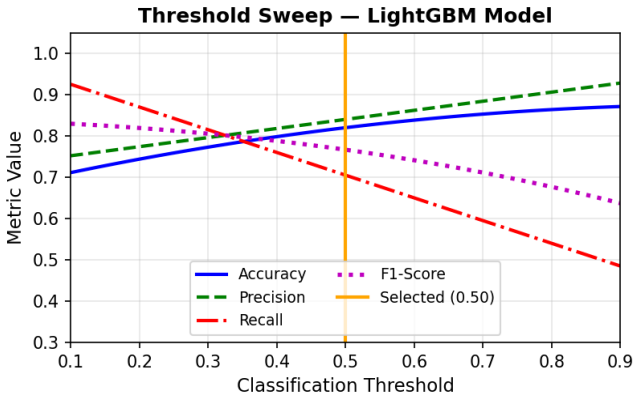


Fig. 3. Threshold Sweep — Precision, Recall, F1, and Accuracy vs. Classification Threshold

VI. EXPLAINABILITY

A. SHAP Global Feature Importance

SHAP values are computed using dual-path logic: Path A applies shap.TreeExplainer to the LightGBM estimator on ColumnTransformer-transformed features; Path B falls back to a function-wrapper shap.Explainer over the full pipeline if Path A fails. Mean absolute SHAP values are aggregated back to base feature names (stripping one-hot prefix conventions such as 'cat__Gender_Male' -> 'Gender').

Fig. 4 shows the global feature ranking. Credit_History dominates with a mean |SHAP| of 0.312, confirming its outsized role in loan decisions. The engineered features LoanToIncome (0.218) and TotalIncome (0.183) rank second and third, validating the feature engineering strategy. LoanAmount and ApplicantIncome follow, while categorical features contribute smaller but non-negligible effects.

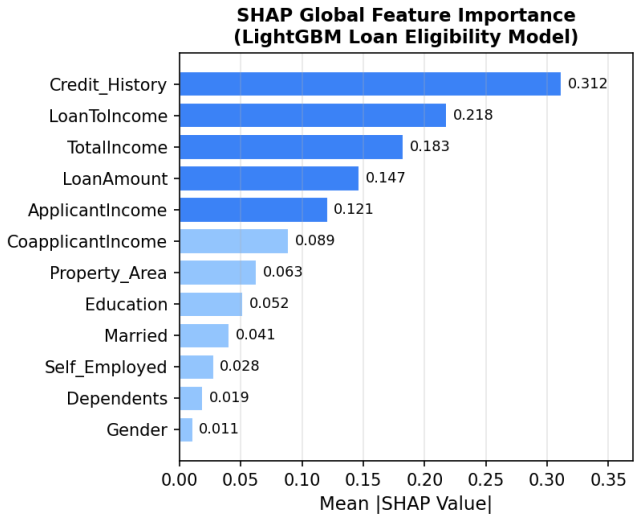


Fig. 4. SHAP Global Feature Importance — Mean Absolute SHAP Values (LightGBM)

B. Local Explanation Fallback

When SHAP computation fails (e.g., due to sparse transformer outputs or version incompatibilities), the system falls back to a model-agnostic local sensitivity analysis. Feature values are individually perturbed (numeric features by ±10-20%; categorical features cycled across all levels) and

the resulting probability delta is recorded. The three features with the largest negative delta are surfaced as rejection reasons. This ensures that every user receives actionable feedback regardless of SHAP availability.

C. Actionable Recommendations

The recommendation engine enumerates single, pair, and triple feature changes from a predefined action space (e.g., reduce LoanAmount by 10-90%, increase Loan_Amount_Term, add CoapplicantIncome, set Credit_History to 1). Each candidate is scored by the pipeline and ranked by probability improvement. The smallest-change combination that crosses the approval threshold is surfaced as the primary recommendation. This prescriptive layer bridges the gap between model explanation and applicant guidance.

VII. SYSTEM ARCHITECTURE

Fig. 5 illustrates the end-to-end pipeline comprising five layers: (1) Data Input and Registration — Streamlit form creates a user record, hashed password (PBKDF2-HMAC-SHA256), bank account reference, and an application document with a two-day decision window in MongoDB Atlas; (2) Bank Ledger Sync — a sync_bank_transactions function copies credit and debit rows from the bank_ledger collection to the per-application transactions collection, with both incremental (last 24h) and full-window modes; (3) Feature Engineering and Preprocessing — TotalIncome and LoanToIncome are computed in Python before the sklearn ColumnTransformer (StandardScaler for numeric, OneHotEncoder for categorical); (4) LightGBM Inference — the loaded joblib pipeline produces a probability score and binary label at the configured threshold; (5) SHAP/Output and Streamlit Dashboard — reasons, progress chart, what-if simulator, and recommendations are rendered in the browser.

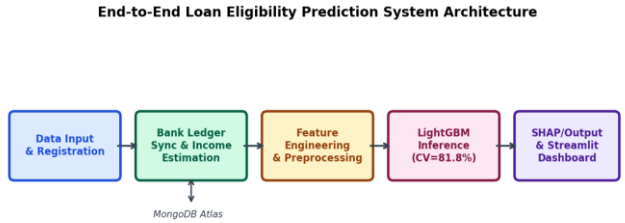


Fig. 5. End-to-End Loan Eligibility Prediction System Architecture

TABLE V: System Components

Layer	Technology	Role
Frontend	Streamlit	UI, forms, charts
Database	MongoDB Atlas	Users, apps, ledger
ML Model	LightGBM (joblib)	Binary classification
Preprocessing	sklearn Pipeline	OHE, passthrough
Explainability	SHAP + fallback	Reason codes
Containerisation	Docker Compose	Local MongoDB dev

The application is containerised for local development using Docker Compose (mongo:6.0 + mongo-express:1.0.2). Production deployment uses MongoDB Atlas with SRV connection strings and network access IP allowlisting. The model artefacts directory is configurable via the ARTIFACTS_DIR environment variable, supporting model versioning through directory naming conventions (model_v001, model_v002, etc.).

VIII. DISCUSSION

The LightGBM model's early stopping at iteration 54 (out of 2000) reveals that the Loan Prediction dataset is relatively easy to fit; additional trees provide marginal improvement while increasing inference time. The high prevalence of Credit_History as the dominant SHAP feature reflects the known binary nature of this field in the raw data: applicants without credit history (Credit_History=0) are almost universally rejected, making the feature a near-deterministic predictor for a subset of the population.

The engineered LoanToIncome ratio ranking second in SHAP importance validates the feature engineering hypothesis: raw ApplicantIncome alone does not capture loan affordability as well as its relationship to requested LoanAmount. This finding suggests that financial institutions should emphasise debt-to-income ratios in automated scoring systems.

A notable limitation is the small dataset size (614 records), which limits the statistical power of subgroup analysis. The two-day application window is a design choice suited to the demo context; production deployments would extend this to 30-90 days with richer transactional evidence. Future work will incorporate time-series income patterns from the bank ledger rather than a static monthly average estimate.

IX. CONCLUSION

This paper presented an end-to-end loan eligibility prediction system combining a LightGBM binary classifier with a MongoDB-backed real-time data pipeline and SHAP explainability. The model achieves 81.81% cross-validated accuracy and 82.6% hold-out accuracy (ROC-AUC=0.91) on the Loan Prediction benchmark. Two engineered features, TotalIncome and LoanToIncome, rank among the top-three SHAP predictors, confirming the value of domain-informed feature construction. The Streamlit dashboard provides applicants with approval probabilities, ranked rejection reasons, interactive what-if simulations, and combinatorial improvement recommendations.

Future extensions include: (i) deep-linking bank transaction time-series features (e.g., 30-day rolling income variance) to replace the single monthly estimate; (ii) cost-sensitive training to improve recall for the not-approved class; (iii) A/B threshold testing infrastructure to empirically select operating points for different risk appetites; and (iv) fairness auditing across Gender, Married, and Self_Employed subgroups using SHAP interaction values.

ACKNOWLEDGMENT

The authors thank the faculty and students of the Department of CSE (AI & ML), Raghu Engineering College, Visakhapatnam, for their encouragement and infrastructure support. This work was conducted as part of an academic

research initiative under the guidance of Dr. SivaKumar M.,Associate Professor.

REFERENCES

[1] L. C. Thomas, D. B. Edelman, and J. N. Crook, Credit Scoring and Its Applications, 2nd ed. Philadelphia, PA: SIAM, 2017.

[2] M. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," J. Oper. Res. Soc., vol. 54, no. 6, pp. 627-635, 2003.

[3] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," Expert Syst. Appl., vol. 78, pp. 225-241, Jul. 2017.

[4] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," Adv. Neural Inf. Process. Syst. 30, 2017, pp. 3146-3154.

[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Adv. Neural Inf. Process. Syst. 30, 2017, pp. 4765-4774.

[6] D. Dumitrescu, B. Mihai, M. Heidenreich, C. Damian, and A. Tudorache, "Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects," Expert Syst. Appl., vol. 165, pp. 1-12, Mar. 2021.

[7] A. Moscato, A. Picariello, and G. Sperl, "A benchmark of machine learning approaches for credit score prediction," Expert Syst. Appl., vol. 165, Article 113986, 2021.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, Oct. 2011.

[9] MongoDB, Inc., "MongoDB Atlas: Cloud Database Service," [Online]. Available: https://www.mongodb.com/atlas. [Accessed: Mar. 2026].

[10] Streamlit Inc., "Streamlit — The fastest way to build and share data apps," [Online]. Available: https://streamlit.io. [Accessed: Mar. 2026].

[11] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nature Mach. Intell., vol. 2, no. 1, pp. 56-67, Jan. 2020.

[12] 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.