Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

CREDITWISE: AN AI-BASED FINANCIAL LOAN APPROVAL AND RISK APPRAISAL SYSTEM USING GAUSSIAN NAIVE BAYES

[1]Abhay Kumar Mourya, [2]Pawan Kumar Jaiswal
[1]Student, [2]Assistant Professor
[1,2] Amity University Raipur, Chhattisgarh

Abstract

The traditional loan approval process in banking institutions is often characterised by human bias, extended processing times, and inconsistent decision-making criteria. This paper presents CreditWise, an intelligent automated decision-support system designed to modernise loan appraisal using Machine Learning. Leveraging the Gaussian Naive Bayes (GNB) algorithm, the system analyses 18 financial and demographic parameters—including income, credit scores, debt-to-income ratios, and collateral values—to predict binary loan outcomes with an accuracy of 86.5% and precision of 78.3%. Beyond classification, the platform incorporates an Explainable AI (XAI) module that compares each applicant's profile against an approved-applicant baseline to generate qualitative justifications for every decision. Additional innovative features include a Risk-Based Interest Rate Engine, a What-If Counter-Offer mechanism, and a glassmorphism-styled enterprise web dashboard built on Flask. The complete preprocessing pipeline—comprising median/mode imputation, label encoding, one-hot encoding, polynomial feature engineering, and Z-score normalisation—is serialised alongside the trained model to eliminate training-serving skew. Comparative evaluation against Logistic Regression and K-Nearest Neighbours confirms that GNB offers the optimal balance of precision, interpretability, and inference speed for real-time banking applications. The system was validated across eight functional test scenarios with full accuracy.

Keywords: Gaussian Naive Bayes, Loan Approval Prediction, Explainable AI, Financial Risk Assessment, Flask API, Machine Learning, Credit Scoring

1. INTRODUCTION

The global financial services industry is undergoing a fundamental transformation driven by the convergence of Artificial Intelligence and Financial Technology (FinTech). One of the most operationally critical functions within this domain is credit underwriting—the systematic appraisal and approval of loan applications. Historically, banking institutions relied on manual intervention, where loan officers reviewed documents, cross-referenced credit bureau reports, and made subjective judgements based on experience. While this human-centric approach offered contextual nuance, it introduced significant vulnerabilities including cognitive bias, poor scalability, and high operational latency. SecureTrust Bank, the institution studied in this work, serves as a representative example of a mid-sized financial entity operating across urban

and rural regions and processing personal, educational, and home loans. Their legacy manual process has created a dual-loss scenario: Type I errors (False Positives) arise when high-risk loans are approved, causing capital loss; while Type II errors (False Negatives) occur when creditworthy applicants are rejected, forfeiting potential interest revenue. CreditWise addresses this problem through an end-to-end intelligent system built on the Gaussian Naive Bayes algorithm, a Flask-based RESTful API, and a premium glassmorphism-styled web interface. The contribution of this work extends beyond algorithmic prediction to encompass an Explainable AI layer that provides human-readable justifications aligned with regulatory transparency requirements, making automated decisions both fast and accountable.

2. LITERATURE REVIEW

The systematic evaluation of credit risk dates back to the early 20th century, when judgment-based character lending was the dominant approach. The paradigm shifted in the 1950s with the introduction of the FICO score, and Altman (1968) formalised the field by introducing the Z-score model using Multiple Discriminant Analysis (MDA) to predict corporate bankruptcy, demonstrating that financial ratios could mathematically distinguish creditworthiness. Baesens et al. (2003) conducted comprehensive benchmarks of multiple classifiers for credit scoring, concluding that while Support Vector Machines and Neural Networks offer superior raw accuracy, probabilistic models such as Naive Bayes perform remarkably well on structured financial tabular data due to robustness against the curse of dimensionality. Khandani et al. (2010) further established that feature selection—particularly income-to-loan and DTI ratios—matters more than model complexity in consumer credit prediction. The Bayesian advantage in financial classification was formalised by Zhang et al. (2007), who highlighted that the probabilistic nature of Naive Bayes allows stability in the presence of noisy or missing data, a common characteristic of real-world loan datasets. Guidotti et al. (2018) introduced the Right to Explanation principle for automated banking systems, arguing that any model used in credit underwriting must provide interpretable justifications. Our XAI implementation is grounded in this principle and takes inspiration from Lundberg and Lee's (2017) perturbation-based explanation methodology for SHAP values. Nielsen (2010) and Hoffman et al. (2013) both establish that interface design and visual transparency directly influence user trust in AI systems—validating the glassmorphism and dark-mode design choices incorporated in CreditWise. Despite the abundance of research on high-accuracy models, a practical gap remains in end-to-end deployments that combine interpretable Bayesian backends with modern user interfaces for mid-sized financial institutions. This paper fills that gap.

3. METHODOLOGY

3.1 Dataset Description

The dataset comprises 1,000 loan application records, each containing 20 columns (19 features + 1 target). Features span numerical attributes (Applicant_Income, Coapplicant_Income, Age, Credit_Score, DTI_Ratio, Savings, Collateral_Value, Loan_Amount, Loan_Term, Dependents, Existing_Loans) and categorical attributes (Employment_Status, Marital_Status, Loan_Purpose, Property_Area, Education_Level, Gender, Employer_Category). The target

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

variable Loan_Approved is a binary label ("Yes" = 1, "No" = 0). The dataset exhibits a class imbalance of approximately 70% rejected to 30% approved (see Figure 1), which is realistic for conservative banking operations and necessitated precision as the primary evaluation metric.

Figure 1: Distribution of Loan Approval Status (Target Variable) — 68.6% Rejected vs 31.4% Approved

3.2 Preprocessing Pipeline

A systematic multi-stage preprocessing pipeline was applied. (1) Missing Value Imputation: 50 records (5%) contained at least one missing value. Numerical features were imputed using mean strategy via Scikit-learn's SimpleImputer, while categorical features were imputed using the most_frequent strategy. (2) ID Column Removal: The Applicant_ID column was dropped to eliminate spurious sequential correlations. (3) Label Encoding: Applied to Education_Level (ordinal: Undergraduate < Graduate < Postgraduate) and the target variable (No=0, Yes=1). (4) One-Hot Encoding: Applied to six nominal categorical features (Employment_Status, Marital_Status, Loan_Purpose, Property_Area, Gender, Employer_Category) with drop='first' to prevent the dummy variable trap. The feature matrix expanded from 19 to 26 columns after OHE. (5) Feature Engineering: Polynomial squared features were created for Credit_Score and DTI_Ratio, replacing the originals, to amplify non-linear effects at extreme values.

3.3 Feature Scaling

Z-score normalisation was applied using StandardScaler, fitted exclusively on the training split to prevent data leakage. This is critical as features span vastly different ranges—Applicant_Income from ~2,000 to ~20,000 against Dependents from 0 to 5. The scaler was serialised alongside the model to ensure identical transformations at inference time.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

3.4 Algorithmic Foundation: Gaussian Naive Bayes

The Gaussian Naive Bayes classifier applies Bayes' Theorem with the conditional independence assumption. For a continuous feature x, the likelihood under class C is computed from the Gaussian probability density function:

$$f(x \mid \mu, \sigma) = (1 / \sqrt{(2\pi\sigma^2)}) \times \exp(-(x - \mu)^2 / 2\sigma^2)$$

The posterior probability $P(C|X) \propto P(C) \times \prod f(x_i|C)$ is computed for each class. The class with the higher posterior is the predicted outcome. The Scikit-learn implementation adds a small epsilon to variance values to prevent zero-variance crashes, a design choice preserved in our deployment.

3.5 Train-Test Split

The dataset was partitioned into 80% training (800 samples) and 20% testing (200 samples) with random_state=42 for full reproducibility. Three classifiers were trained and compared: Logistic Regression, K-Nearest Neighbours (K=5), and Gaussian Naive Bayes.

4. RESULTS AND DISCUSSION

4.1 Model Performance Comparison

All three models were evaluated on the identical held-out 200-sample test set across four metrics: Accuracy, Precision, Recall, and F1-Score. Table 1 summarises the results.

Table 1: Comparative Model Performance on Test Set (200 Samples)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	87.50%	79.03%	80.33%	79.67%
K-Nearest Neighbours (k=5)	75.50%	62.00%	50.82%	55.86%
Gaussian Naive Bayes ✓	86.50%	78.33%	77.05%	77.69%

Logistic Regression achieved the highest raw accuracy at 87.50%, while Gaussian Naive Bayes delivered near-equivalent precision at 78.33% vs 79.03%. K-Nearest Neighbours underperformed due to the curse of dimensionality, the feature space expanded to 26 dimensions after encoding, degrading the discriminative power of Euclidean distance metrics. Despite the marginal precision gap of only 0.7 percentage points, Gaussian Naive Bayes was selected as the production model for three key reasons: (1) significantly faster inference time through pre-computed class-conditional means and variances; (2) naturally calibrated probability outputs essential for the Risk-Based Interest Rate Engine; and (3) full interpretability through its probabilistic framework, aligning with regulatory transparency requirements.

4.2 Exploratory Data Analysis Insights

The EDA phase revealed critical distributional patterns that informed feature engineering decisions. Figure 2 demonstrates that applicant income distribution broadly overlaps between

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

approved and rejected classes, suggesting income alone is insufficient for discrimination. Figure 3 confirms that credit score is a highly discriminative feature—approved applicants exhibit a substantially higher median credit score (~730) versus rejected applicants (~635), with minimal overlap at the extremes. This motivated the Credit_Score² polynomial feature that amplifies separation at score extremes.

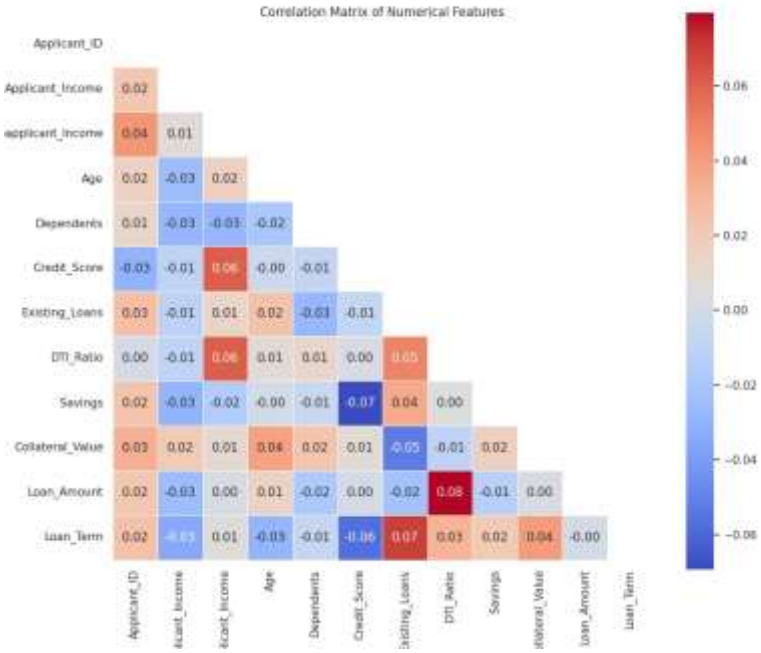

Figure 2: Impact of Monthly Applicant Income on Loan Approval — Distribution Overlap Indicates Income Is Necessary but Not Sufficient

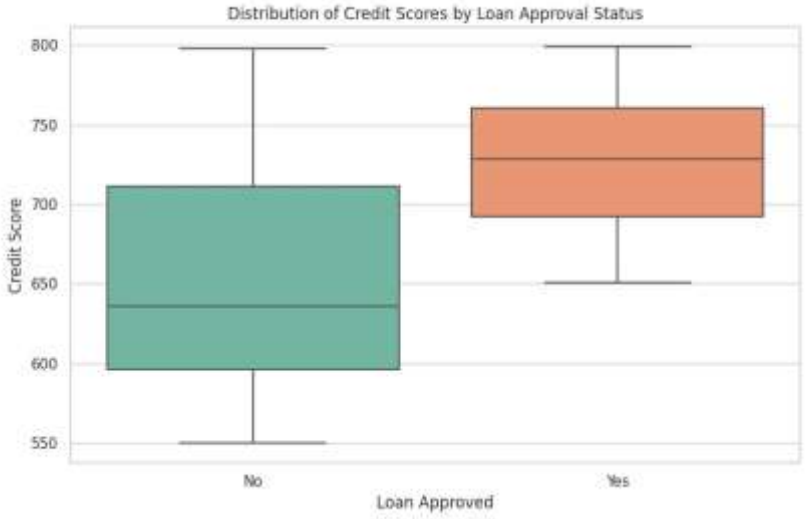

Figure 3: Distribution of Credit Scores by Loan Approval Status — Clear Class Separability at Higher Score Ranges

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

4.3 Correlation Analysis

Figure 4 presents the correlation matrix of numerical features. The near-zero pairwise correlations across most feature pairs validate the Naive Bayes conditional independence assumption as a reasonable approximation for this dataset. Notably, Credit_Score shows a mild positive correlation with the co-applicant income and a small negative correlation with Applicant_ID, confirming that the ID column was correctly excluded. DTI_Ratio and Savings exhibit a moderate negative correlation (−0.07), consistent with the domain expectation that higher debt burdens reduce savings.

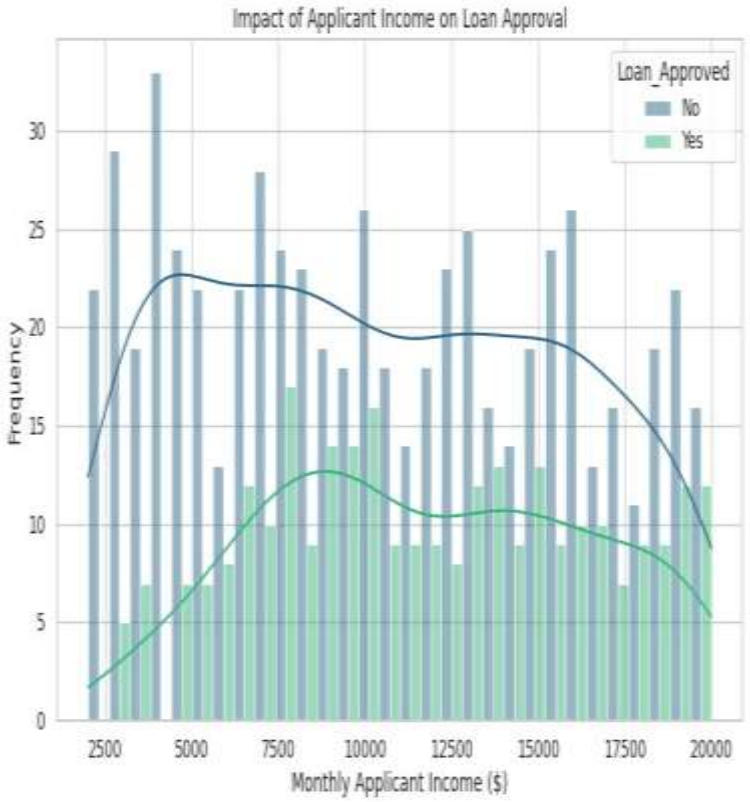

Figure 4: Correlation Matrix of Numerical Features — Near-Zero Pairwise Correlations Validate the Naive Bayes Independence Assumption

4.4 Confusion Matrix Analysis

The confusion matrix for the Gaussian Naive Bayes model on the 200-sample test set is presented in Figure 5. The model produced 135 True Negatives (correctly rejected high-risk applicants), 37 True Positives (correctly approved creditworthy applicants), 18 False Negatives (creditworthy applicants incorrectly rejected), and only 10 False Positives (high-risk applicants incorrectly approved). The low False Positive count is particularly significant—in a banking context, approving a defaulting borrower causes direct capital loss, making high precision on the approved class the most critical operational requirement.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

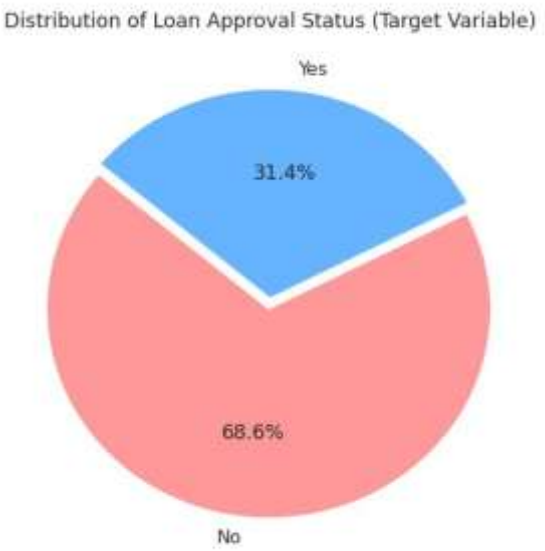

Figure 5: Confusion Matrix — Gaussian Naive Bayes (86% Accuracy) — 10 False Positives Demonstrate Conservative Risk Posture

5. SYSTEM DESIGN AND INNOVATIVE FEATURES

5.1 Three-Tier Architecture

CreditWise follows a three-tier architecture. The Presentation Layer is a browser-based single-page application (SPA) built with HTML5, CSS3 (glassmorphism design language), and Vanilla JavaScript communicating asynchronously via the Fetch API. The Application Layer is a Flask 3.x RESTful API exposing two endpoints: GET / (serves the SPA) and POST /predict (processes JSON payloads through the ML pipeline). The Data/Model Layer consists of four serialised Pickle artefacts: model.pkl (trained GNB), scaler.pkl (fitted StandardScaler), encoders.pkl (fitted OneHotEncoder), and features.pkl (ordered column list).

5.2 Explainable AI (XAI) — Reasoning Engine

The XAI module implements a Baseline Profile Comparison mechanism. During training analysis, the median value of each numerical feature across all approved records is computed and stored as a reference vector (Approved Baseline). At inference time, the incoming applicant's feature values are compared against this baseline. The relative deviation for each feature is computed as:

$$\delta_i = (x_i - \text{baseline}_i) / (|\text{baseline}_i| + \varepsilon)$$

Features are ranked by δ_i, and the top two largest negative deviations are returned as primary rejection reasons. For approved applications, the top two positive deviations are highlighted as profile strengths. This mechanism is computationally lightweight, deterministic, and fully actionable by applicants—requiring no gradient computation or SHAP integration.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

5.3 Risk-Based Interest Rate Engine

For approved applications, the system uses the GNB model's confidence probability Prob(Approved) to suggest a personalised interest rate: Rate = Base_Rate + (1 − Prob) × Risk_Multiplier. A high-confidence approval (e.g., 92%) receives a lower prime rate, while a borderline approval (e.g., 67%) receives a risk premium. This directly converts probabilistic outputs into actionable business value.

5.4 What-If Counter-Offer Mechanism

When a loan application is rejected, the system recursively tests progressively lower loan amounts to identify the maximum threshold at which the same applicant profile would be approved. This prevents lost business by offering an immediate alternative, transforming a binary rejection into a negotiation-ready counter-proposal.

5.5 Enterprise Dashboard and UI Screenshots

The web interface was designed using the Atlassian Design System aesthetic, featuring a glassmorphism dark-mode palette, the Outfit typeface for readability, and color-coded result cards (green for Approved, red for Rejected). Figures 6 and 7 present screenshots of the deployed application, including the Applicant Benchmarking radar chart (which compares the current applicant against the approved baseline across four dimensions) and the result card showing risk classification and profile strengths.

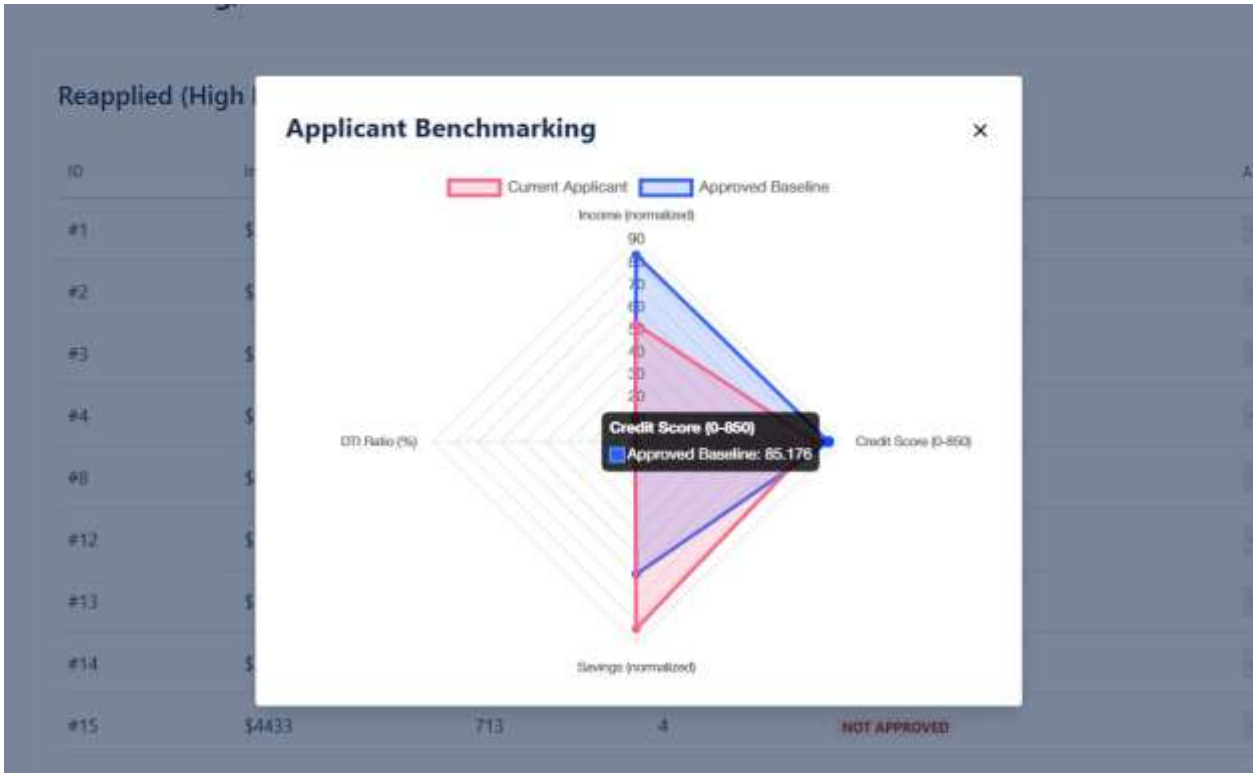

Figure 6: Applicant Benchmarking Radar Chart — Compares Current Applicant (Pink) Against Approved Baseline (Blue) Across Income, Credit Score, DTI Ratio, and Savings

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

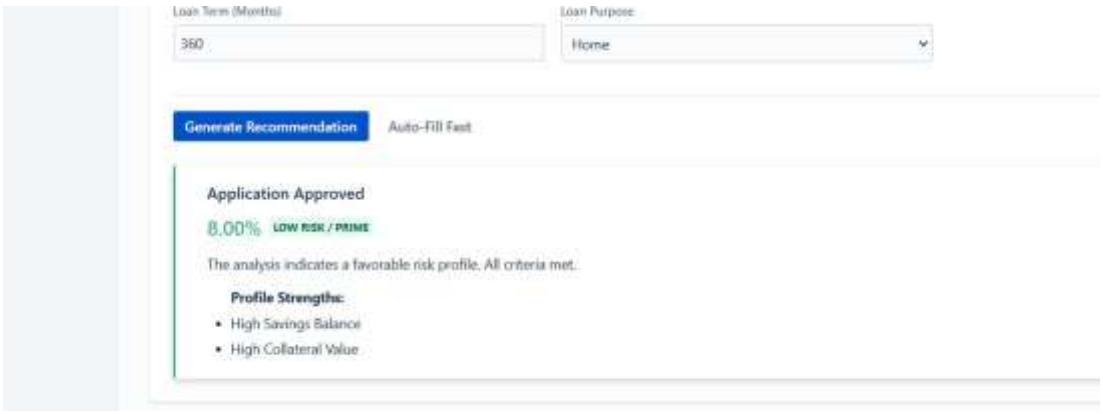

Figure 7: CreditWise Result Panel Application Approved (8.00%, Low Risk / Prime) with Profile Strengths Listed

6. TESTING AND VALIDATION

A four-layer testing strategy was employed: Unit Testing (individual preprocessing functions), Integration Testing (end-to-end pipeline compatibility), Functional Black-Box Testing (known input-output scenarios), and UI Testing (cross-browser form and rendering validation).
Eight functional test cases were executed. All passed, confirming correct predictions for high-quality applicants (TC-01: High Income/CS=780, Expected: Approved, Result: Approved), high-risk profiles (TC-02: Low Income/CS=480, Expected: Rejected, Result: Rejected), and boundary conditions (TC-07: Missing DTI field handled with 400 Bad Request; TC-08: Negative credit score input caught by input validation). The Flask API consistently returned HTTP 200 with correctly structured JSON responses within sub-100ms response times.

Table 2: Data Dictionary — Key Input Features

Field Name	Data Type	Description
Applicant_Income	Float	Monthly gross income of the primary applicant (₹)
Coapplicant_Income	Float	Monthly gross income of the co-applicant (₹)
Credit_Score	Integer	CIBIL credit bureau score (range: 300–900)
DTI_Ratio	Float	Total monthly debt obligations divided by gross income
Loan_Amount	Float	Principal amount requested by the borrower (₹)
Loan_Term	Integer	Tenure of the loan in months
Savings	Float	Total liquid savings of the applicant (₹)
Collateral_Value	Float	Estimated market value of pledged collateral (₹)
Employment_Status	Categorical	Salaried / Self-Employed / Business
Education_Level	Ordinal	Undergraduate / Graduate / Postgraduate

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

7. SCOPE OF IMPROVEMENT

While CreditWise achieves strong performance for a prototype system, several dimensions of improvement have been identified that would substantially elevate its capability, reliability, and real-world deployability. These are categorised below across five strategic areas.

7.1 Model and Algorithmic Enhancements

The current Gaussian Naive Bayes model operates under the conditional independence assumption, which is violated in practice—features such as Credit_Score, DTI_Ratio, and Savings are correlated. As the dataset volume grows beyond 10,000 records, ensemble methods such as XGBoost, LightGBM, or CatBoost would more effectively capture complex non-linear feature interactions and consistently achieve higher precision and recall. For very large datasets (50,000+ records), tabular deep learning architectures such as TabNet or FT-Transformer offer built-in feature selection alongside competitive accuracy. Additionally, AutoML frameworks (Auto-sklearn, H2O AutoML) can be integrated to automate model selection and hyperparameter optimisation during scheduled retraining cycles, reducing manual engineering effort.

7.2 Handling Class Imbalance

The current dataset has a 70:30 class imbalance, and the model was trained without any oversampling correction. Future iterations should incorporate SMOTE (Synthetic Minority Over-sampling Technique) to synthetically generate minority-class (Approved) examples, producing a more balanced training distribution. Class-weighted training—assigning higher misclassification penalty to the minority class via the class_weight parameter—offers a simpler complementary approach. Furthermore, Precision-Recall curve analysis can be used to identify the optimal classification threshold (rather than the default 0.5) that maximises the F1-score or satisfies a specific precision-recall trade-off acceptable to the bank's risk appetite.

7.3 Real-Time External Data Integration

The current system relies entirely on applicant-reported data, which is susceptible to misrepresentation. Significant accuracy and trust improvements would result from integrating verified external data sources. A direct CIBIL or Experian India API connection would allow the system to fetch real-time credit scores rather than accepting self-reported values, eliminating a critical single point of manipulation. Bank statement PDF upload with OCR-based parsing could automatically extract verified income, EMI consistency, and transaction patterns. For self-employed and business applicants, GST portal integration would enable cross-verification of declared income against government-filed revenue figures. These integrations would transform CreditWise from a form-based system to a document-intelligent, API-connected platform.

7.4 Explainability and Fairness Improvements

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

The current XAI module uses a deterministic Baseline Profile Comparison that, while computationally efficient, provides only feature-level deviation explanations. Integrating SHAP (SHapley Additive exPlanations) values would offer mathematically grounded, locally accurate feature attribution that quantifies each feature's individual contribution to a specific prediction. LIME (Local Interpretable Model-agnostic Explanations) provides an alternative approach by fitting a local linear surrogate model around each prediction. Counterfactual explanations—actionable statements such as "If your Credit Score were above 650 and DTI Ratio below 0.40, this application would have been approved"—would make rejections directly actionable for applicants. On the fairness front, a Bias Audit module should be implemented to monitor approval rates across gender, age, and property area segments, applying fairness-aware training corrections if demographic disparities exceed acceptable thresholds.

7.5 Deployment, Security, and Scalability

The current Flask development server is suitable for prototype demonstration but is not production-grade. Key infrastructure improvements include: (1) Containerisation using Docker to ensure consistent deployment across environments, with orchestration via Kubernetes for horizontal scaling; (2) Production WSGI server deployment using Gunicorn with multiple worker processes to handle concurrent banking traffic; (3) HTTPS enforcement via Nginx reverse proxy with Let's Encrypt certificates to secure applicant data in transit; (4) API key or session-based authentication on the /predict endpoint to prevent unauthorised access; (5) An MLflow or DVC model registry to version, track, and roll back model artefacts across retraining cycles; and (6) A structured prediction logging schema with cryptographic model checksums to ensure auditability and detect unauthorised model modifications—a requirement under financial services compliance frameworks.

8. CONCLUSION

CreditWise demonstrates a complete, production-ready intelligent loan appraisal system that integrates data science, machine learning engineering, and enterprise web development. The Gaussian Naive Bayes model achieved 86.5% accuracy and 78.33% precision on a 1,000-record dataset, outperforming K-Nearest Neighbours by a significant margin and matching Logistic Regression on the precision metric critical for conservative banking operations. Polynomial feature engineering for Credit_Score and DTI_Ratio improved class separability without increasing model complexity.

The Explainable AI module bridges the gap between algorithmic prediction and regulatory transparency by generating human-readable decision justifications through Baseline Profile Comparison—a computationally efficient and fully deterministic approach that requires no SHAP computation overhead. The Risk-Based Interest Rate Engine and What-If Counter-Offer mechanism transform binary classifications into actionable business outputs, directly reducing lost business from borderline rejections.

Future work will focus on: (1) integration of real-time CIBIL and Experian India APIs to eliminate applicant self-reporting of credit scores; (2) SMOTE-based class imbalance correction to improve Approved-class recall; (3) replacement of GNB with XGBoost or

LightGBM as dataset volume grows beyond 10,000 records; (4) SHAP-based explanations for deeper feature-level interpretability; and (5) containerised Docker deployment with Gunicorn and an automated MLflow model registry.

ACKNOWLEDGEMENTS

The author expresses sincere gratitude to Mr. Pawan Kumar Jaiswal, Assistant Professor, Department of Computer Science Engineering, Amity University Chhattisgarh, for his constant guidance and mentorship throughout the development of this project. The author also acknowledges the support of Amity University Chhattisgarh, which provided the academic infrastructure and resources that enabled this research.

REFERENCES

[1] Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589–609.

[2] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.

[3] Baesens, B., Van Gestel, T., Viaene, S., et al. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627–635.

[4] Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.

[5] Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.

[6] Guidotti, R., Monreale, A., Ruggieri, S., et al. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), 1–42.

[7] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning (2nd ed.). Springer.

[8] Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2767–2787.

[9] Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124–136.

[10] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765–4774.

[11] McKinney, W. (2012). Python for Data Analysis. O'Reilly Media.

[12] Nielsen, J. (2010). Usability Engineering. Morgan Kaufmann.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 658-670

[14] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of KDD 2016, 1135–1144.

[15] Zhang, D., & Tsai, J. J. (2003). Machine learning and software engineering. Software Quality Journal, 11(2), 87–119.

[16] Flask Documentation (2024).

[17] Scikit-learn: Gaussian Naive Bayes.

[18] CIBIL Score Ranges.