IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

Credit Scoring Using Machine Learning

Dr. Prashant Wakhare, Mr. Samarth Sokashi, Ms. Vedika Sonawane
Department of AI & DS
AISSMS Institute of Information Technology, Pune
prashant.wakhare@aissmsioit.org, samarth.sokashi@aissmsioit.org, vedikasonawane345@gmail.com

Abstract: *The proliferation of digital financial services has intensified the demand for accurate, efficient, and fair credit scoring systems. Traditional credit scoring methods such as FICO scores rely on limited financial history and fail to capture the full creditworthiness of diverse borrowers, particularly thin-file and unbanked populations. This paper presents the development and project management of a machine learning-based credit scoring system capable of leveraging a broader spectrum of applicant features to predict default risk with high accuracy.*

The system allows financial institutions to input borrower attributes such as income, employment history, loan amount, repayment history, and demographic indicators. The ML engine generates creditworthiness predictions using ensemble methods including Random Forest, Gradient Boosting (XGBoost), and Logistic Regression, benchmarked against a baseline FICO-style scorecard. The backend is implemented in Python (scikit-learn, XGBoost, pandas, Flask API), with a relational database (PostgreSQL) and a clean analytical dashboard frontend.

The paper evaluates the effectiveness of structured project management methodologies in delivering a functional, explainable, and bias-aware ML credit scoring solution, concluding that systematic execution significantly improves the accuracy, fairness, and deployability of financial AI systems..

Keywords: *credit scoring systems*

I. INTRODUCTION

Credit scoring is the process of estimating the probability that a borrower will repay a loan or fulfill a financial obligation. It forms the backbone of retail banking, lending, insurance underwriting, and buy-now-pay-later platforms. Lenders use credit scores to make rapid, data-driven decisions on loan approval, interest rate assignment, and credit limit determination. Accurate credit scoring reduces non-performing assets (NPAs) for financial institutions while expanding credit access for deserving applicants.

Traditional credit scoring relies on linear statistical models applied to a narrow set of credit bureau variables — payment history, outstanding balances, length of credit history, new credit inquiries, and credit mix. While well-established, these methods struggle with non-linear feature interactions, high-dimensional data, and the increasing availability of alternative data sources such as transactional behavior, utility payments, mobile usage patterns, and psychometric assessments.

Machine learning (ML) techniques offer a compelling alternative. Algorithms such as Random Forests, Gradient Boosted Trees, and Neural Networks naturally capture complex, non-linear relationships in borrower data, yielding demonstrably superior predictive performance on standard benchmarks such as the Lending Club and Home Credit Default Risk datasets. However, deploying ML-based credit models introduces new challenges: model interpretability, regulatory compliance (ECOA, GDPR), algorithmic fairness, and the risk of encoding historical biases.

This project addresses these challenges by developing a complete ML-based credit scoring pipeline with explainability (SHAP values), fairness auditing (disparate impact analysis), and model performance monitoring — all delivered through a structured software project management approach.

DOI: 10.48175/IJARSCT-34405

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

II. PROBLEM STATEMENT

Credit risk assessment in modern financial institutions faces multiple interconnected challenges:
Limited feature coverage in traditional scorecards leads to underestimation of default risk for applicants with non-standard financial profiles.

- High false-negative rates (approving high-risk borrowers) and false-positive rates (rejecting creditworthy thin-file applicants) result in financial losses and financial exclusion.
- Lack of interpretability in black-box ML models creates regulatory barriers to deployment under ECOA (Equal Credit Opportunity Act) and GDPR Article 22, which mandate explainable automated decisions.
- Algorithmic bias — models trained on historically biased data may perpetuate or amplify discriminatory lending patterns across gender, race, and geography.
- Static scorecards fail to adapt to macroeconomic shifts such as recessions, inflation surges, or pandemic-induced financial disruptions.

The absence of an interpretable, fair, and dynamic ML credit scoring system forces financial institutions to choose between accuracy and compliance. This project directly addresses this gap by combining predictive power with transparency and fairness constraints.

III. OBJECTIVES

The primary objectives of this project are:

- To develop a complete ML-based credit scoring pipeline using Python (scikit-learn, XGBoost, LightGBM) trained on real-world lending datasets.
- To engineer a rich feature set combining traditional credit bureau variables with alternative data indicators (transactional behavior, employment stability, income volatility).
- To benchmark multiple ML algorithms (Logistic Regression, Random Forest, Gradient Boosting, Neural Networks) against a traditional FICO-style scorecard using AUC-ROC, Gini coefficient, and KS-statistic.
- To apply model explainability techniques (SHAP, LIME) to generate per-applicant decision explanations compliant with adverse action notice requirements.
- To conduct fairness auditing using disparate impact ratios and equalized odds metrics across demographic subgroups.
- To apply structured project management methodologies (planning, execution, risk management, delivery) throughout the development lifecycle.
- To deliver a REST API (Flask/FastAPI) exposing the scoring engine, integrated with a PostgreSQL database and an analytical dashboard

IV. PROJECT MANAGEMENT APPROACH

4.1 Project Planning

The project was initiated by defining clear objectives, scope, and a phased timeline. A formal Requirement Analysis phase identified core user personas: retail lending officers, risk analysts, and compliance teams at mid-sized financial institutions processing 500–5,000 loan applications per month. The formal project scope was defined to include:

- A feature engineering pipeline ingesting structured CSV applicant data with 50+ raw attributes.
- A curated training dataset of 150,000+ records sourced from the Lending Club public dataset and Home Credit Default Risk dataset (Kaggle).
- ML model training, hyperparameter optimization (Optuna), and cross-validated performance benchmarking.
- SHAP-based explainability module generating waterfall charts and summary plots per application decision.
- Fairness audit module computing disparate impact (DI) ratio and equalized odds across gender and age subgroups.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

- REST API endpoints for real-time scoring, batch scoring, and model monitoring (concept drift detection).

A five-phase timeline was created: (1) Requirement Analysis, (2) Data Engineering & Feature Design, (3) Model Development & Evaluation, (4) API Integration & Testing, and (5) Deployment & Monitoring — each with defined deliverables and acceptance criteria.

4.2 Project Execution

The execution phase involved full-stack ML engineering through structured, iterative steps:

- Data Engineering: Raw lending data was cleaned, missing values imputed (median/mode for numerical/categorical), outliers capped at the 1st–99th percentile, and categorical variables encoded using WoE (Weight of Evidence) transformation for regulatory alignment.
- Feature Engineering: 50+ raw features were transformed into 35 model-ready features including Debt-to-Income (DTI) ratio, payment-to-income ratio, delinquency recency score, and employment stability index.
- Model Development: Five classifiers were trained — Logistic Regression (baseline), Random Forest, XGBoost, LightGBM, and a 3-layer MLP Neural Network. Hyperparameters were optimized using Optuna with 5-fold stratified cross-validation. Final model selection was based on AUC-ROC and Gini coefficient on a held-out test set.
- Explainability Module: SHAP TreeExplainer was applied to the XGBoost champion model, generating global feature importance plots and per-applicant waterfall charts for adverse action notices.
- Fairness Audit: Disparate impact ratios (DI = P(positive|protected group) / P(positive|reference group)) were computed for gender and age subgroups. Models failing DI < 0.8 threshold were retrained with fairness constraints (reweighing via IBM AI Fairness 360).
- API Integration: The scoring engine is exposed via POST /api/score (real-time, single applicant), POST /api/batch-score (CSV upload), and GET /api/model-stats (performance metrics), all secured with JWT authentication.

4.3 Risk Management

Key risks were identified and managed proactively. The following risk register summarizes mitigation strategies:

Risk	Impact	Mitigation Strategy
Class imbalance (3–5% default rate)	Biased model	SMOTE oversampling + class_weight balancing + PR-AUC as primary metric
Data leakage	Overfitted, non-generalizable model	Strict train/validation/test split; time-based splitting for temporal data
Regulatory non-compliance	Deployment blocked	SHAP explanations + adverse action notice generation per ECOA
Algorithmic bias	Discriminatory lending decisions	DI ratio auditing + reweighing via AI Fairness 360
Concept drift	Model accuracy degradation over time	PSI (Population Stability Index) monitoring + scheduled model retraining
API security breach	Unauthorized score access	JWT authentication + HTTPS + rate limiting on all API endpoints

Table 1: Risk Register — Credit Scoring ML Project

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

4.4 Project Delivery

The final system was delivered as a fully functional ML credit scoring platform with the following verified features:

- Applicant input interface accepting structured feature vectors via REST API and CSV batch upload.
- ML-generated credit scores (0–1000 scale, calibrated to probability of default) with explicit confidence intervals.
- Per-applicant SHAP waterfall explanation charts suitable for adverse action notice generation.
- Fairness audit dashboard displaying DI ratios and equalized odds for protected subgroups.
- Model performance dashboard showing live AUC-ROC, Gini, KS-statistic, and PSI drift indicators.
- Batch scoring endpoint processing 10,000 applications per minute on standard cloud hardware..

V. SYSTEM DESIGN AND IMPLEMENTATION

5.1 System Architecture

The credit scoring system is designed as a three-tier ML platform:

- Data Layer: PostgreSQL database storing applicant profiles, loan application records, historical outcomes, and model predictions. SQLAlchemy ORM manages entity relationships across Applicant, LoanApplication, CreditScore, and ModelPrediction tables.
- ML Engine Layer: A modular Python pipeline (scikit-learn Pipeline API) handling preprocessing, feature engineering, model inference, SHAP explanation generation, and fairness auditing. XGBoost serves as the champion model based on benchmarking results.
- API Layer: Flask/FastAPI REST server exposing scoring endpoints with JWT authentication, request validation (pydantic schemas), and structured JSON responses including score, risk tier, and SHAP feature contributions.
- Dashboard Layer: Streamlit analytics dashboard for model performance monitoring, score distribution visualization, applicant segment analysis, and drift detection alerts.

5.2 Machine Learning Pipeline

The ML pipeline implements an end-to-end credit scoring workflow. Raw applicant data undergoes cleaning (missing value imputation, outlier treatment) followed by feature engineering to compute derived attributes including Debt-to-Income ratio, credit utilization rate, payment delinquency recency score, and employment stability index. Weight of Evidence (WoE) encoding is applied to categorical variables to produce monotonic, regulation-aligned feature transforms. The engineered feature matrix is fed into the champion XGBoost classifier, calibrated using Platt scaling to produce well-calibrated probability-of-default estimates. SHAP TreeExplainer computes Shapley values for each prediction, providing additive feature contribution explanations. The score is linearly transformed to a 0–1000 scale with higher scores indicating lower default risk, consistent with industry conventions.

5.3 Model Benchmarking

Model	AUC-ROC	Gini Coefficient	KS-Statistic	F1 Score (Default)
Logistic Regression (Baseline)	0.721	0.442	0.341	0.398
Random Forest	0.783	0.566	0.489	0.461
XGBoost (Champion)	0.812	0.624	0.531	0.489
LightGBM	0.809	0.618	0.524	0.482
Neural Network (MLP)	0.798	0.596	0.507	0.471

Table 2: Model Benchmarking Results — Credit Scoring ML System

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

5.4 Technology Stack

Layer	Technology	Purpose
Data Engineering	Python, pandas, NumPy	Data cleaning, feature engineering, WoE encoding
ML Models	scikit-learn, XGBoost, LightGBM	Classification, hyperparameter tuning (Optuna)
Explainability	SHAP, LIME	Per-applicant feature contribution explanations
Fairness Auditing	IBM AI Fairness 360	Disparate impact ratio, equalized odds computation
Database	PostgreSQL + SQLAlchemy ORM	Applicant records, loan outcomes, model predictions
API Server	Python 3.11, Flask/FastAPI	REST scoring API with JWT authentication
Dashboard	Streamlit	Model monitoring, drift detection, analytics
Infrastructure	Docker, AWS EC2/S3	Containerized deployment, model artifact storage

Table 3: Credit Scoring System Technology Stack

5.5 API Endpoints

Method	Endpoint	Description
POST	/api/score	Real-time credit scoring for a single applicant (JSON input)
POST	/api/batch-score	Batch scoring via CSV upload (up to 10,000 records)
GET	/api/model-stats	Live model performance metrics (AUC, Gini, PSI drift)
GET	/api/applicants/	CRUD — applicant profile management
GET	/api/loans/	CRUD — loan application management
GET	/api/scores/history	Historical score records per applicant
GET	/api/fairness-report	Disparate impact and equalized odds audit report

Table 4: Credit Scoring REST API Endpoints

5.6 Database Schema

The SQLAlchemy ORM defines five core models. The Applicant model stores borrower profile data including annual_income, employment_length, home_ownership, loan_purpose, and demographic attributes. The LoanApplication model records loan_amount, loan_term, interest_rate, application_date, and loan_status (approved/rejected/defaulted). The CreditScore model stores ml_score (0–1000), probability_of_default, risk_tier (A–F), and shap_values_json per scoring event. The ModelVersion model tracks trained model artifacts, training_date, feature_set_version, and performance metrics for audit trail maintenance. The FairnessAudit model logs DI ratios and equalized odds metrics per model version and demographic subgroup. The database is seeded at initialization with 150,000+ historical loan records from the Lending Club dataset, partitioned into training (70%), validation (15%), and test (15%) sets by application date to prevent temporal leakage.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 19, April 2026 Impact Factor: 8.2

5.7 Dashboard Interface — Feature Screens

Figure 1: Credit Scoring Dashboard — Home screen displaying system overview with total applications scored (150,000+), current model AUC-ROC (0.812), average default rate (4.2%), and real-time API health status. Navigation tabs provide access to Scoring, Model Performance, Fairness Audit, and Applicant Explorer modules.

Figure 2: Real-Time Scoring Interface — Input form accepting 35 applicant features (income, employment, loan amount, credit history, etc.) with validation. Output panel displays credit score (0–1000), probability of default (%), risk tier (A–F), and SHAP waterfall chart showing top 10 positive and negative feature contributions to the decision.

Figure 3: Model Performance Dashboard — Time-series charts for AUC-ROC, Gini coefficient, and KS-statistic computed on rolling 30-day windows. PSI (Population Stability Index) drift gauges highlight score distribution shift. Automated retraining alert fires when PSI > 0.20.

Figure 4: Fairness Audit Dashboard — Bar charts comparing approval rates across gender (Male / Female / Non-binary) and age group subgroups. Disparate impact ratio table with regulatory threshold (DI ≥ 0.80) traffic-light indicators. Equalized odds heatmap showing TPR and FPR parity across subgroups.

Figure 5: Batch Scoring Interface — CSV upload panel supporting up to 10,000 applicant records per batch. Progress indicator, error log for invalid records, and downloadable results CSV containing applicant_id, credit_score, risk_tier, and top_5_shap_features per applicant.

VI. RESULTS AND DISCUSSION

The developed credit scoring ML system successfully generates accurate, explainable, and fair credit predictions across all defined use cases. Key outcomes include:

Prediction Accuracy: The XGBoost champion model achieves AUC-ROC of 0.812 and Gini coefficient of 0.624 on the held-out test set, representing a 12.6% AUC improvement over the logistic regression baseline (0.721), consistent with published literature on ML credit scoring.

Explainability: SHAP TreeExplainer correctly identifies the top 5 most influential features per applicant in all test cases. Income-to-loan-amount ratio, delinquency recency, and DTI ratio are the three most globally important predictors, consistent with domain expertise.

Fairness: Post-reweighing fairness correction, all subgroup disparate impact ratios exceed the 0.80 regulatory threshold. Gender DI improved from 0.74 (pre-correction) to 0.87 (post-correction) with negligible AUC degradation (<0.5%).

System Performance: The FastAPI scoring endpoint processes single-applicant requests in <50ms p99 latency. Batch scoring handles 10,000 records in under 8 seconds on a standard 4-vCPU cloud instance.

Drift Detection: PSI monitoring correctly flags concept drift when model performance degrades beyond 0.20 PSI threshold, triggering automated retraining workflows with no manual intervention.

The application demonstrates that applying structured project management principles ensures better organization of ML development tasks, rigorous model evaluation, efficient risk mitigation, and improved final output quality. The combination of predictive accuracy, regulatory explainability, and fairness auditing positions this system for real-world financial deployment.

VII. LIMITATIONS

The system relies on the Lending Club public dataset which reflects the U.S. credit market; performance on Indian or emerging-market lending data requires retraining with locally relevant features and historical outcomes.

ML recommendations are based on pre-trained static models; the system does not yet implement online/incremental learning to adapt in real time to individual repayment behavior.

Alternative data sources (mobile usage patterns, utility payment history, psychometric scores) are not yet integrated due to data access constraints, limiting performance for thin-file applicants.

Fairness constraints are applied via reweighing; more advanced in-processing fairness methods (adversarial debiasing, constrained optimization) remain as future work.

DOI: 10.48175/IJARSCT-34405

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

The current deployment uses a single-model champion architecture; a challenger model framework for A/B testing is not yet implemented.

VIII. FUTURE SCOPE

Integration of alternative data sources — mobile transaction history, utility payments, GST filing records, and UPI payment behavior — for thin-file and new-to-credit applicant scoring in the Indian market.

Implementation of a Hybrid ML Pipeline combining supervised default prediction with unsupervised anomaly detection for early warning of stressed portfolios.

Deployment of a challenger model framework enabling live A/B testing between model versions with automatic champion promotion based on statistical significance tests.

Real-time streaming scoring pipeline using Apache Kafka and Apache Flink for sub-5ms credit decisions in buy-now-pay-later and embedded finance contexts.

Extension to multi-label credit risk classification — predicting not just default vs. non-default but distinguishing among delinquency severity levels (30-day, 60-day, 90-day+) and recovery probability.

Integration with RBI-compliant credit bureau APIs (CIBIL, Experian India, Equifax India) for automated bureau report ingestion and real-time feature enrichment.

LLM-augmented adverse action notice generation ("Why was my application declined?") providing natural language explanations derived from SHAP values for applicant-facing communications.

IX. CONCLUSION

This paper presented the development of a machine learning-based credit scoring system — covering the full lifecycle from data engineering and model development through explainability, fairness auditing, API integration, and deployment — using a structured project management approach. The platform integrates an XGBoost champion model (AUC-ROC 0.812, Gini 0.624) with SHAP-based explainability and IBM AI Fairness 360 bias correction within a Python/FastAPI/PostgreSQL architecture.

The ML integration enhances credit decision quality by providing accurate, interpretable, and bias-aware predictions that outperform traditional logistic regression scorecards by 12.6% on AUC-ROC while satisfying regulatory fairness thresholds. The SHAP explainability module enables generation of adverse action notices compliant with ECOA requirements, directly addressing the key regulatory barrier to ML adoption in credit underwriting. The fairness auditing pipeline ensures equitable access to credit across protected demographic subgroups.

The study confirms that successful application of project management practices — including formal scope definition, phased timelines, proactive risk registers (class imbalance, data leakage, concept drift, regulatory compliance), and structured evaluation — plays a crucial role in delivering reliable and deployable AI systems for high-stakes financial applications. The credit scoring platform provides a solid technical foundation for future integration of alternative data, real-time streaming inference, and hybrid recommendation systems, with significant potential to improve credit access and risk management across financial institutions.

REFERENCES

[1]. Siddiqi, N. (2017). Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards (2nd ed.). Wiley.

[2]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

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

[4]. Bellamy, R. K. E., et al. (2019). AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. IBM Journal of Research and Development, 63(4/5), 4:1–4:15.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

[5]. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
[6]. 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.
[7]. Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.
[8]. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
[9]. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627–635.
[10]. Project Management Institute. (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide) (7th ed.). PMI Publications