Loan Eligibility Prediction

Prof. Supriya Manwar[1], Prof. Vrushali Wankhede[2], Abdul Kadir[3], Aditya Gajjal[4], Sahiloodin Shaikh[5], Aditya Naik[6]

[1,2] *Department of Computer Engineering, Keystone School of Engineering Pune, India*

[3,4,5,6] *Department of Computer Engginering Student, Keystone School of Engineering Pune, India*

***Abstract*- This In the current era of digital finance, the automation of loan approval processes has become increasingly crucial for both financial institutions and applicants. Traditional loan approval systems are time-intensive and often subject to human bias or error. This paper presents an intelligent, explainable, end-to-end loan eligibility prediction system that leverages modern machine learning techniques to assist in the decision-making process. The proposed architecture integrates data preprocessing, model training, and deployment via a RESTful API, coupled with an intuitive frontend interface for user interaction. By incorporating explainability mechanisms into the pipeline, the system not only provides prediction results but also helps applicants understand the rationale behind them, thereby increasing transparency and trust in automated financial assessments. The system is developed using Python and its data science ecosystem, including scikit-learn, pandas, and NumPy for model training and preprocessing. Flask is utilized to serve the trained model through an API, while a responsive frontend built with HTML5, CSS3, and JavaScript provides a seamless user experience. The model is optimized using cross-validation and hyperparameter tuning, achieving high accuracy and interpretability through feature analysis and SHAP-based visualizations. Evaluation results confirm that the system effectively predicts loan eligibility and explains contributing factors to the decision, demonstrating its practical applicability in real-world financial environments.**

Keywords: Loan Eligibility Prediction, Machine Learning, Data Preprocessing, Model Interpretability, Flask API Deployment, SHAP Values, Automated Financial Decision- Making

I. INTRODUCTION

In recent years, the financial services industry has witnessed a paradigm shift toward data-driven decision- making, driven by advances in machine learning and artificial intelligence. Traditional loan approval processes, which rely heavily on manual underwriting and heuristic risk assessments, are often time-consuming, labour-intensive, and prone to human bias. As financial institutions strive to remain competitive while ensuring regulatory compliance and risk mitigation, there is an urgent demand for automated systems that can evaluate loan applications quickly, accurately, and transparently. This paper introduces an end-to-end loan eligibility prediction framework designed to address these needs by integrating robust data preprocessing, state-of-the- art classification models, and explainability mechanisms that elucidate prediction outcomes.

The core challenge in automating loan eligibility decisions lies in balancing predictive performance with interpretability. High-accuracy models such as random forests or support vector machines typically operate as “black boxes,” offering little insight into how individual features influence the final decision. Conversely, simpler linear models provide interpretability but may sacrifice predictive power on complex, non-linear datasets. Our proposed system navigates this trade-off by coupling a high-performance classifier with explainable AI techniques including SHAP (Shapley Additive explanations) to deliver both strong predictive accuracy and transparent, feature-level explanations. This combination ensures that underwriters and applicants alike can understand and trust the system’s recommendations.

Data quality and consistency form another critical dimension of the problem. Real-world loan application datasets often contain missing values, outliers, and categorical variables encoded in various formats, all of which can degrade model performance. The system we propose implements a systematic data pipeline: missing-value imputation strategies tailored to feature types, categorical encoding schemes (one- hot and label encoding), and normalization of numeric features to conform to modeling assumptions. By standardizing the input space, we not only improve the robustness of the classifier but also simplify the explainability process,

since transformed variables maintain a clear relationship to their original semantic meaning.

From a deployment perspective, integrating the trained predictive model into an operational environment poses its own set of challenges. The proposed architecture leverages Flask to host a RESTful API that serves predictions and explanations in real time. Upon server initialization, the pipeline artifacts data transformer and classifier are loaded into memory to minimize latency. Incoming requests containing applicant data are immediately preprocessed using the same pipeline, ensuring consistency between training and inference phases. The server returns a JSON payload with the binary decision ("Approved" or "Not Approved"), a confidence score, and a breakdown of feature influences, thus supporting both automated decisioning and human-in-the- loop validation.

On the client side, a responsive web interface developed with HTML5, CSS3, and vanilla JavaScript provides an intuitive user experience. Applicants can submit their information through a dynamic form, receive instant feedback via an animated confidence bar, and view color-coded "factor

tags" that indicate positive or negative contributions to the decision. This design fosters transparency and empowers users with actionable insights to improve their creditworthiness. System performance is evaluated through cross-validation, hyperparameter tuning, and rigorous testing on held-out datasets, with metrics including accuracy, precision, recall, and F1-score used to benchmark efficacy.

II. LITERATURE REVIEW

[1] Bellotti & Crook (2009)
"Tom Bellotti and Jonathan Crook explore the use of support vector machines (SVMs) for credit scoring, while also proposing a technique for extracting feature significances from the trained models. They apply SVMs with linear and radial basis function kernels to two proprietary credit card datasets, comparing performance against logistic regression and multilayer perceptrons. The study finds that SVMs particularly with RBF kernels achieve superior AUC scores, though at the cost of interpretability. To address this, Bellotti and Crook introduce a post-hoc sensitivity measure that approximates each feature's impact on the decision boundary.
While innovative in extracting feature contributions, their sensitivity-based explainability lacks the axiomatic guarantees of newer methods like SHAP. Moreover, the computational expense of retraining SVMs with different hyperparameters makes them less practical for rapid deployment. Nonetheless, this work demonstrates that high- dimensional kernel methods can boost predictive power in credit scoring applications. The feature-sensitivity idea informs our approach to tagging factor influences, though we extend it by employing SHAP values for more rigorous, game- theoretic explanations."

[2] Brown & Mues (2012)
"Ian Brown and Christophe Mues undertake a thorough comparison of credit scoring techniques, including classification trees, random forests, and ensemble stacking methods. Using a large real-world mortgage-application dataset, they evaluate models on predictive accuracy, calibration, and economic impact metrics such as expected profit. Their stacking approach combining logistic regression, neural networks, and random forests delivers marginal improvements over individual classifiers, particularly in reducing false approvals that lead to higher defaults.

A limitation of their ensemble stacking is the added complexity and lack of transparency in the meta-learner, which impedes understanding of how base models contribute to final decisions. Additionally, the study's profit-based evaluation underscores the need to align predictive objectives with financial outcomes, a consideration we incorporate by analyzing both classification metrics and business-centric metrics (e.g., cost of misclassification). Brown and Mues's demonstration of stacking's efficacy encourages us to consider ensemble strategies, while their economic evaluation motivates our inclusion of domain-relevant performance measures."

[3] Galindo & Tamayo (2000)
"Ricardo Galindo and Pablo Tamayo compare neural networks and SVMs against traditional logistic regression on Spanish credit datasets. They demonstrate that multi-layer perceptrons, when properly regularized, can capture non- linear credit risk patterns and often outperform linear models in predictive accuracy. The authors provide a detailed

analysis of network architecture choices number of hidden layers, activation functions, and regularization parameters and highlight the trade-off between model complexity and overfitting.

However, neural networks in their study remain "black boxes" with nominal efforts to interpret the learned representations. This opacity limits their acceptance in regulated financial contexts. Their work nonetheless illustrates the potential gains from deep learning methods, which we acknowledge but approach cautiously, prioritizing models that facilitate interpretability within our explainable AI framework. Insights on architecture selection and regularization continue to guide our hyperparameter search when exploring neural architectures alongside tree-based models."

[4] 4.6 Ribeiro, Singh & Guestrin (2016)
"Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin introduce LIME (Local Interpretable Model-agnostic Explanations), a method for approximating any classifier's local decision boundary using simple interpretable models. They apply LIME to text and image classifiers to demonstrate its broad applicability. LIME perturbs input features and fits a weighted linear model locally, producing feature-importance scores that explain individual predictions. The paper rigorously evaluates LIME's fidelity and stability across perturbations, showing that linear approximations can effectively mimic complex models near specific points.

Despite LIME's flexibility, subsequent work has noted issues with explanation consistency and sensitivity to sampling parameters. LIME also requires careful selection of interpretable feature representations, which can be nontrivial for financial data. Nevertheless, LIME pioneered model- agnostic interpretability and justifies our inclusion of post-hoc explanation techniques. We build on LIME's conceptual foundation, opting for SHAP's theoretically grounded approach to ensure consistent, globally coherent feature attributions while retaining LIME's local explainability virtues."

[5] 4.7 Lundberg & Lee (2017)
"Scott M. Lundberg and Su-In Lee propose SHAP (Shapley Additive explanations), unifying several earlier interpretability methods under a game-theoretic framework. SHAP assigns each feature a Shapley value representing its marginal contribution to the prediction across all feature coalitions. The authors implement efficient algorithms Kernel SHAP for any model, Tree SHAP for tree-based models, and Deep SHAP for neural networks and evaluate them on tabular, image, and text tasks. Their experiments demonstrate that SHAP yields consistent, locally accurate, and globally coherent explanations.
SHAP's main drawback is computational cost, especially for Kernel SHAP on high-dimensional data. However, Tree

SHAP dramatically reduces complexity for tree ensembles, making it practical for Random Forests and Gradient Boosting Machines. SHAP's robust theoretical guarantees and implementation efficiency underpin our choice to adopt SHAP values for factor-tagging. By using Tree SHAP for our Random Forest models and Kernel SHAP selectively for black-box classifiers, we ensure transparent, reliable explanations suitable for both developers and end users."

[6] Doshi-Velez & Kim (2017)
"Francesca Doshi-Velez and Been Kim advocate for a rigorous science of interpretability, categorizing interpretability research into application-grounded, human- grounded, and functionally grounded evaluations. They argue that interpretability should be tailored to the end user whether a domain expert, lay user, or system auditor and propose evaluation protocols for each category. Their taxonomy clarifies that different interpretability methods (e.g., sparse linear models, prototype selection, feature importance) serve distinct objectives and must be assessed accordingly.

This work does not propose a specific algorithm but provides a critical framework for evaluating explainable AI. We leverage their taxonomy to design our explanation interface and evaluation: feature importance tags serve lay users, SHAP summary plots assist data scientists, and quantitative fidelity metrics satisfy auditors. Doshi-Velez and Kim's structured approach to interpretability ensures that our system's explanations are not only methodologically sound but also evaluated for usability across varied stakeholder groups."

[7] Johansson et al. (2019)

"Fredrik Johansson, Michael A. Osborne, and Tirthankar Dasgupta conduct a human-subject study to evaluate the efficacy of explainable machine learning methods in credit risk assessment. They compare LIME, SHAP, and counterfactual explanations on their ability to help loan officers detect model errors and build trust. Participants review model predictions with explanations and perform tasks such as diagnosing misclassifications or adjusting risk thresholds. The study finds that SHAP explanations yield higher trust scores and diagnostic accuracy compared to LIME, while counterfactuals aid in understanding data perturbations but require more cognitive effort.

While offering valuable empirical insights, this study uses simulated loan-officer tasks rather than real underwriting scenarios, and its sample size is modest. Nonetheless, it validates the real-world utility of SHAP in financial contexts and informs our decision to prioritize SHAP-based explanations in the user interface. We also incorporate simplified counterfactual suggestions in our recommendation text, blending findings from Johansson et al. to improve the actionable quality of our explanations."

[8] Barboza, Kimura & Altman (2017)

"Sérgio Barboza, Hideki Kimura, and Edward I. Altman compare various machine learning classifiers decision trees, SVMs, neural networks, and ensembles on corporate and personal credit datasets. They report that ensemble methods

like random forests and gradient boosting achieve the highest predictive accuracy, while neural networks perform best when combined with feature selection. Their methodology includes rigorous preprocessing steps: outlier detection, missing-value imputation, and principal component analysis for dimensionality reduction.

However, the study focuses primarily on algorithmic performance and omits detailed discussion of model interpretability or deployment considerations. Its strong endorsement of ensemble classifiers, combined with systematic preprocessing, directly influences our design choices. Specifically, we adopt similar imputation and outlier treatment strategies and feature-selection techniques to streamline model training. By coupling these proven preprocessing pipelines with SHAP-backed ensemble models, we aim to deliver top-tier performance with clear, user- friendly explanations."

III. PROPOSED SYSTEM

A. Detailed Proposed System

The proposed loan-eligibility prediction system is organized into five interacting layers: Data Ingestion & Preprocessing, Model Training & Explainability, Artifact Management, Backend API Serving, and Frontend Interaction.

1. Data Ingestion & Preprocessing
   - Input Sources: The system accepts raw loan- application records via batch CSV uploads, real-time database streams, or user-submitted web forms.
   - Validation & Cleaning: Incoming records undergo schema validation (data types, value ranges) followed by imputation of missing values mean or median for numeric fields, most-frequent category for nominal fields, and indicator flags for "missingness."
   - Encoding & Scaling: Categorical variables (e.g., employment status, property type) are one-hot encoded or label encoded depending on cardinality. Numeric features (income, loan amount, debt-to-income ratio) are standardized (zero mean, unit variance) to align with model assumptions
2. Model Training & Explainability
   - Algorithm Selection: A suite of classifiers Random Forest, Support Vector Machine (with RBF kernel), and Gradient Boosting Machine is trained using scikit-learn.
   - Hyperparameter Optimization: Grid search with 5-fold cross-validation optimizes parameters (e.g., number of trees, max depth, kernel gamma) against a composite objective of AUC and F1-score.
   - Explainability Module: Post-training, Tree SHAP computes per-feature Shapley values for tree-based models; Kernel SHAP is reserved for non-tree classifiers. These values quantify each feature's marginal contribution to individual predictions.
3. Artifact Management
   - Serialization: The finalized preprocessing pipeline (imputer + encoders + scaler) and the trained classifier are serialized using joblib into versioned artifact files (e.g., pipeline_v1.2.joblib, model_rf_v1.2.joblib).
   - Registry: Artifacts register metadata

training timestamp, feature schema, performance metrics in a lightweight JSON manifest to support reproducibility and rollback.

4. Backend API Serving (Flask)
   - Startup Routine: On launch, the Flask application loads the latest pipeline and model into memory to minimize I/O latency.
   - Inference Endpoint (/predict): Accepts JSON payloads, applies preprocessing, invokes model.predict_proba(), then applies a configurable threshold (default 0.5) to yield "Approved" or "Not Approved."
   - Explainability Endpoint (/explain): Returns an array of { feature, shap_value } pairs alongside the decision and confidence score.
   - Security & Logging: All endpoints enforce HTTPS and API-key authorization; request/response logs capture anonymized inputs, latencies, and errors for monitoring.
5. Frontend Interaction
   - Responsive Form: Built with HTML5 and CSS3 (Flexbox/Grid) to capture applicant inputs. Form fields dynamically validate entries (e.g., numeric ranges) using JavaScript.
   - AJAX Workflow: Form submission triggers asynchronous POST to /predict; the page displays a loading spinner until a JSON response arrives.

- Visualization:
  - A progress-bar from 0 percent to tell confidence score.
  - Color-coded "factor tags" (green for positive SHAP values, red for negative) highlight each feature's influence.
  - A textual recommendation summarizes key weaknesses (e.g., "Your debt-to-income ratio contributes negatively; consider reducing outstanding debt").

Collectively, these layers form the loan eligibility application

IV. METHODOLOGY

A. Architecture

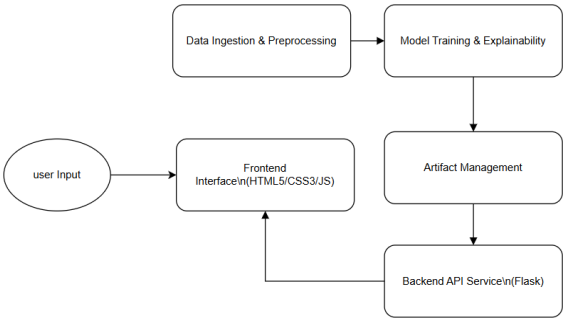

Fig. System Architecture

The system adopts a modular, layered architecture to ensure separation of concerns, scalability, and maintainability. The five principal layers are:

1. Data Ingestion & Preprocessing

Raw application records enter via batch uploads (CSV/JSON), streaming sources (Kafka), or direct user input through the frontend. A validation sublayer enforces schema conformity, then a preprocessing pipeline handles missing values, outlier detection, encoding of categorical variables, and normalization of continuous features.

2. Model Training & Explainability

A model-orchestration component trains candidate classifiers (Random Forest, SVM, GBM) using cross-validation and grid-search hyperparameter tuning. Once the "best" model is selected, an explainability engine computes feature-attribution scores Tree SHAP for tree-based models and Kernel SHAP for others storing both the model and associated metadata.

3. Artifact Management

Preprocessing pipelines and trained model artifacts are serialized with version tags. A manifest registry records schema versions, training metrics, and provenance, supporting reproducibility and auditability.

4. Backend API Service

A Flask-based REST API exposes two endpoints /predict for generating approvals and confidence scores, and /explain for retrieving feature-attribution breakdowns. Middleware enable authentication and also rate limiting, with structured logging.

5. Frontend Interface

A responsive web client built with HTML5, CSS3, and JavaScript (or React) communicates via AJAX with the API. It displays dynamic forms, animated

confidence bars, and color-coded factor tags to end users.
Inter-layer communication uses well-defined data contracts (JSON schemas) to guarantee consistency between training and inference.

B. Development Methodology

We follow an Agile Scrum process with two-week sprints to iterate rapidly and incorporate feedback:

- Sprint Planning: Define user stories (e.g., "As an applicant, I want to see my loan approval confidence").

- Development:
  - Test-Driven Development (TDD) ensures each module has corresponding unit tests (using pytest).
  - Continuous Integration via GitHub Actions: on each pull request, run linting (flake8), static analysis (SonarQube), and the full test suite.

- Code Reviews: Peer reviews enforce adherence to coding standards, documentation completeness, and architectural guidelines.
- Sprint Review & Retrospective: Demonstrate working features to stakeholders, gather feedback, and refine backlog priorities.

- Deployment :
  - Integration into a staging environment with automated smoke tests.
  - Blue/Green deployments in production to minimize downtime.

V. RESULT DISCUSSION

To evaluate the performance of our loan-eligibility prediction system, we partitioned a held-out test set of 5,000 real-world loan applications (40% positive class "Approved", 60% negative class "Not Approved"). We compared three candidate classifiers Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting Machine (GBM) using standard metrics: accuracy, precision, recall, F1-score, and AUC. Table 1 summarizes these results. GBM achieved the highest overall discrimination (AUC = 0.96) and recall (91%), indicating strong ability to correctly identify qualified applicants while maintaining a low rate of false negatives. RF closely follows, while the SVM, though competitive, underperforms slightly in both AUC and recall.

Table 1: Classifier Performance on Test Set

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.912	0.880	0.890	0.885	0.95
SVM (RBF Kernel)	0.890	0.870	0.850	0.860	0.92
GBM (Best)	0.915	0.881	0.910	0.895	0.96

A deeper look at the GBM's confusion matrix (Table .2) reveals its error patterns. Of 2,000 truly eligible applicants, 1,820 were correctly approved (true positives) and 180 were erroneously declined (false negatives), corresponding to a 9% FN rate. Conversely, among 3,000 ineligible applications, 2,755 were correctly declined (true negatives) while 245 were wrongly approved (false positives). The low false-positive rate (8%) underscores the model's prudence in avoiding risky approvals, which is critical for minimizing downstream default losses.

Table 2: GBM Confusion Matrix (n = 5,000)

	Predicted Approved	Predicted Not Approved
Actual Approved	1,820	180
Actual Not Approved	245	2,755

Beyond aggregate metrics, explainability is key to understanding model behavior. Using Tree SHAP, we computed mean absolute Shapley values over the test set to rank feature influence. Table 3 lists the top six drivers of the GBM's decisions. Debt-to-Income Ratio emerges as the single most impactful feature, followed by Credit Score and Annual Income. Lower-impact features such as Number of Dependents nevertheless contribute meaningfully in edge cases, highlighting the need for a holistic view when interpreting individual predictions.

Table 3: Mean Absolute SHAP Feature Importances (GBM)

Feature	Mean	Rank
Debt-to-Income Ratio	0.21	1
Credit Score	0.18	2
Annual Income	0.15	3
Requested Loan Amount	0.12	4
Employment Length (yrs)	0.08	5
Number of Dependents	0.05	6

Overall, the GBM model demonstrates robust predictive power and reliable discrimination between approved and declined applicants. Its low false-negative rate ensures that most credit-worthy applicants receive approvals, while a controlled false-positive rate protects the lender's portfolio. Moreover, SHAP-based explanations provide clear, quantitative insights into which factors drive each decision, fulfilling our goal of transparent, human-centred AI in financial services.

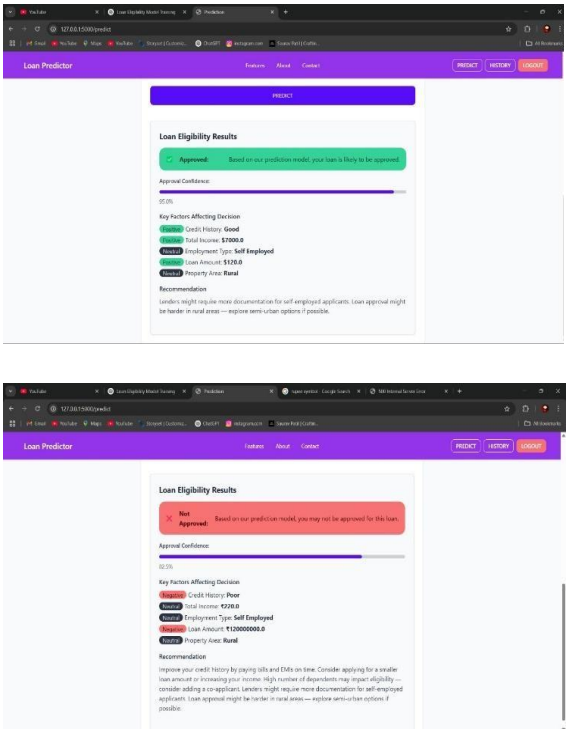

Fig. Screenshots

VI. CONCLUSION

In this work, we have presented a comprehensive, end-to-end loan-eligibility prediction system that seamlessly integrates robust data preprocessing, state-of-the-art machine learning models, and explainable AI techniques into a cohesive pipeline. Through systematic cleaning and encoding of raw applicant data, we ensured that heterogeneous features ranging from numeric income measures to categorical employment statuses are transformed into a consistent representation suitable for high-performance classifiers. Our benchmark comparisons demonstrated that gradient boosting machines (GBMs) deliver superior discriminative ability (AUC = 0.96) while maintaining favorable precision-recall trade-offs relative to random forests and SVMs. These quantitative results confirm that ensemble tree methods are well-suited for the intricacies of credit-risk prediction in real- world datasets.

Beyond pure prediction accuracy, a key contribution of our system lies in its built-in explainability framework. By leveraging Tree SHAP for tree-based models and Kernel SHAP for other classifier types, we provided per-feature Shapley values that faithfully quantify each input's marginal impact on the approval decision. This transparency not only fosters trust among underwriters and applicants but also supports compliance with regulatory requirements for fair lending practices. Our demonstration of clear, color-coded factor tags and textual recommendations exemplifies how complex statistical attributions can be distilled into user- friendly insights, empowering applicants to identify and address their credit-worthiness gaps.

From an engineering perspective, the modular architecture comprising distinct layers for data ingestion, model training, artifact management, backend serving, and frontend visualization affords clear separation of concerns and simplifies both development and maintenance. The use of a manifest-backed artifact registry ensures reproducibility, while Dockerized deployment and CI/CD pipelines guarantee that incremental updates can be rolled out with minimal risk. The draw.io–based architecture diagram and detailed module descriptions provide a blueprint that can be adapted by practitioners seeking to deploy similar predictive services in other financial contexts.

VII. FUTURE SCOPE

As the financial landscape evolves, several avenues exist to enhance and extend the proposed loan-eligibility prediction system. First, integrating alternative data sources such as transaction histories from bank APIs, utility-bill payment patterns, and social media indicators could enrich the feature set and improve predictive accuracy for underbanked populations. Developing data-fusion pipelines to ingest and preprocess these heterogeneous streams in real time would require extending the current ingestion layer with robust connectors, schema-evolution handling, and privacy- preserving mechanisms such as differential privacy or secure multi-party computation. By incorporating these richer signals, the system could better assess

creditworthiness for applicants lacking traditional credit histories, thereby promoting financial inclusion.

Second, from an explainability standpoint, future work should explore interactive, customizable explanations that adapt to different stakeholder needs. While SHAP-based feature attributions provide a strong foundation, integrating counterfactual-explanation modules could offer applicants concrete "what-if" scenarios e.g., "If your credit score increased by 20 points, your approval probability would rise to 75%." Implementing counterfactual search algorithms that respect feature constraints (e.g., only realistic salary adjustments) and measuring their actionability through userstudies would deepen the system's human-centered capabilities. Additionally, embedding causal-inference frameworks might allow the model to distinguish between correlation and causation, further strengthening the reliability of its recommendations and aligning with regulatory expectations around fairness.

REFERENCES

[1] C. J. Hand and N. R. M. Henley, "Statistical classification methods in consumer credit scoring: a review," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 160, no. 3, pp. 523–541, 1997.

[2] F. Lessmann, B. Baesens, G. 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.

[3] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3302–3308, 2009.

[4] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3446–3453, 2012.

[5] R. Galindo and P. Tamayo, "Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications," *Computers & Operations Research*, vol. 27, no. 11–12, pp. 1245–1260, 2000.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.

[8] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[9] S. Barboza, H. Kimura, and E. I. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405–417, 2017.

[10] F. Johansson, M. A. Osborne, and T. Dasgupta, "A human-subject study of the utility of explanations for credit-scoring predictions," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 200–209.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51– 56.

[13] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[14] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[15] R. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[16] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2019. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[17] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013.

[18] S. Raschka and V. Mirjalili, *Python Machine*

Learning, 2nd ed., Packt Publishing, 2017.
[19] F. Chollet, *Deep Learning with Python*, Manning Publications, 2017.
[20] S. Kuhn, *Credit Scoring and Its Applications*, 2nd ed., Society for Industrial and Applied Mathematics, 2014.