Loan Approval Prediction System

Kruthika Suresh
Data Science, Analytics and Engineering, SCAI
Arizona State University
Tempe, United States of America
ksures21@asu.edu

Sindhu Thati
Computer Science, SCAI
Arizona State University
Tempe, United States of America
sthati12@asu.edu

Sahithi Katoori
Data Science, Analytics and Engineering, SCAI
Arizona State University
Tempe, United States of America
skatoori@asu.edu

Omkar Rajesh Hundekari
Computer Science, SCAI
Arizona State University
Tempe, United States of America
ohundeka@asu.edu

***Abstract*—This work is a data-driven loan approval prediction system aimed at improving fairness, consistency, and efficiency in financial decision-making. Many times, traditional loan approval processes are often slow, subjective, and prone to human bias, which can often result in unfair rejection of qualified applicants. In this project we are using a Kaggle dataset of approximately 56,000 loan applications containing 13 financial, demographic, and credit-related attributes. The problem statement is formulated as a binary classification task to predict whether a loan will be approved or rejected. Here, a comprehensive data-mining pipeline is implemented, including outlier capping, missing-value imputation, categorical encoding, numerical scaling, and domain-informed feature engineering. The Loan Prediction dataset has significant class imbalance with approximately 86% approvals which is addressed using a hybrid strategy that combines SMOTE oversampling with cost-sensitive boosting models. The novelty of this work lies in introducing engineered financial-stability indicators that were not part of the original dataset, such as ”income_to_loan_ratio” to capture repayment capacity which is a high_debt_burden flag to detect applicants exceeding a 30% debt load, an ”employment_stable” indicator to reflect job stability, and ”age_group” segmentation to model life-stage trends, along with calibrated probability outputs that enable threshold-adjustable, risk-aligned credit decisions. The experimental evaluation shows that gradient boosting methods, especially LightGBM, outperform baseline models, achieving a recall of 83.53% for minority-class (rejected/default) predictions and around 95% for approved predictions, while maintaining a strong ROC-AUC performance. Recall is prioritized here because in loan-default prediction the primary risk lies in failing to identify a true defaulter which is an error that carries significantly higher financial and operational cost than incorrectly flagging a safe applicant. In conclusion, these findings demonstrate the system’s potential to enhance the objectivity, transparency, and accuracy of loan approval decisions, with future work focusing on hyper-parameter tuning, fairness assessment, model explainability, and deployment-oriented automation.**

I. Introduction

A. Background and Motivation

Loan approval is one of the most critical processes in the financial sector, directly influencing both institutional risk management and customer financial access. Traditionally, loan approval workflows rely heavily on manual evaluations, which can be slow, inconsistent, and prone to human bias. As a result, potential eligible borrowers may be unfairly rejected, while risky applicants may be mistakenly approved. With the growth of financial technology (FinTech in laymen terms) and increased access to applicant-level data, machine learning has become a powerful tool for developing fair, consistent, and efficient loan evaluation systems. Data-driven prediction models can reduce subjectivity, accelerate decision-making, and improve transparency in determining creditworthiness.

B. Problem Description

The objective of this project is to develop a predictive system that determines whether a loan application will be approved (1) or rejected (0) based on an applicant’s demographic, financial, and credit-related attributes. We are using a publicly available Kaggle dataset containing approximately 56,000 records with 13 key features, the problem is formulated as a binary classification task. Other features/indicators have been created in order to handle all the inconsistencies in the dataset. Our current system aims to correctly identify risky or default-prone applicants while minimizing the false rejection of creditworthy individuals.

C. Importance and Impact

Accurate loan evaluation models provide substantial societal and business value. From a societal perspective, improved prediction reduces bias and promotes fairer access to credit. For financial institutions, accurate default detection reduces operational risk, improves loan portfolio quality, and lowers approval-processing costs. In order to achieve faster, more transparent decisions and to elevate customer satisfaction, making automated prediction systems becomes an essential component of modern lending pipelines.

D. System Overview

Our proposed system implements a complete end-to-end machine learning pipeline consisting of data preprocessing, feature engineering, model development, and performance evaluation. After cleaning and standardizing the dataset, the system introduces several domain-informed engineered features designed to better capture borrower financial stability and repayment capacity characteristics that are not explicitly represented in the raw attributes. These engineered features include ”income_to_loan_ratio”, which measures whether an applicant’s income can sufficiently support the loan amount; ”employment_stable”, derived from employment duration indicators; ”high_debt_burden”, which flags applicants with disproportionate existing financial obligations; and ”age_group”, a segmentation feature that enables the model to learn non-linear age-related risk patterns. To address the significant class imbalance between approved and rejected/default cases, our system adopts a hybrid imbalance-handling strategy that combines SMOTE for minority oversampling with class-weight adjustments during model training. This dual approach enhances the representation of minority-class samples while ensuring that misclassification of high-risk borrowers is penalized more heavily. The new processed dataset is then used to train multiple machine learning models—namely Logistic Regression, Random Forest, XGBoost, CatBoost, and LightGBM—allowing for a comparative evaluation across linear, ensemble, and gradient-boosting approaches. Each model is assessed using relevant performance metrics, with particular emphasis on recall to ensure effective identification of high-risk applicants.

E. Related Work

Previous research in loan default prediction has extensively employed machine-learning models such as logistic regression, decision trees, support vector machines, and ensemble boosting techniques including XGBoost and LightGBM. [1] established logistic regression as a foundational method for credit scoring due to its interpretability and robustness, while later studies demonstrated that tree-based and kernel-based models can capture non-linear borrower behavior more effectively [2]. A central challenge highlighted throughout the literature survey is the severe class imbalance inherent in real-world lending datasets, where the majority of instances correspond to accepted or non-default borrowers. This imbalance has been shown to degrade the minority-class detection capability of conventional classifiers, leading to suboptimal risk identification [3]. To address this issue, researchers and authors have explored oversampling approaches such as SMOTE and its extensions [4], which generate synthetic minority-class samples to improve classifier sensitivity. However, many existing works continue to rely primarily on raw tabular features and do not incorporate engineered domain attributes, temporal borrowing patterns, or probability calibration techniques necessary for threshold-based risk decisioning. Furthermore, relatively few studies investigate hybrid strategies that combine oversampling with cost-sensitive boosting or ensemble learning, despite evidence that such techniques can substantially enhance minority-class (default) detection in credit-risk settings [5]. These gaps underscore the need for methods that not only mitigate class imbalance but also integrate richer feature representations and cost-aligned evaluation strategies to support realistic credit-risk assessment which has been accomplished in our work.

F. Data Collection and Properties

The dataset used was sourced from the Kaggle [6] Playground Series S4E10 competition and contains approximately 56,000 rows and 13 attributes, covering personal information (age, income, employment length), loan characteristics (amount, interest rate, loan intent), and credit indicators (credit history length, previous defaults). We identified several challenges while trying to interpret and make sense of the data such as : missing values (especially in employment length), unrealistic numeric values, categorical imbalance, and significant skew in income and loan grade distributions.

G. ML System Components

The machine learning pipeline consists of:

- Data Cleaning & Outlier Capping – removal of unrealistic values (e.g., age = 123) and capping of skewed extremes.
- Feature Engineering – addition of domain-based features to improve predictive power.
- Encoding & Scaling – one-hot encoding for categorical variables and StandardScaler for numerical features.
- Class Imbalance Handling – SMOTE oversampling and cost-sensitive boosting.
- Modeling & Evaluation – training multiple baseline and advanced models, followed by ROC-AUC, recall, and calibration analyses. Ensemble modeling is beneficial here, because it enables us to achieve the best out of all models.

H. Initial Results

Early experimentation shows that gradient boosting models outperform traditional baselines, with LightGBM achieving the highest minority-class recall of 83.53%, a key metric for detecting risky applicants. This analysis also suggests that engineered features meaningfully contribute to model performance, and class-imbalance techniques significantly improve minority detection.

II. Problem Formulation

1) Key Definitions: We model each loan application as a feature vector $\mathbf{x}_i$ that contains demographic, financial, and credit-related information about applicant i. The original dataset has around 56,000 applications and 13 core attributes, such as:

- Numerical features: age, income, employment length, loan amount, interest rate, credit history length, `loan_percent_income`.
- Categorical features: home ownership, loan intent, loan grade, previous default flag.

To make risk patterns more explicit, several engineered features are added on top of the raw attributes:

- **`income_to_loan_ratio`**

$$\text{income\_to\_loan\_ratio} = \frac{\text{person\_income}}{\text{loan\_amnt} + 1}$$

This measures repayment capacity relative to the requested loan size. Higher values generally indicate lower risk.

- **high_debt_burden**

  $$\text{high\_debt\_burden} = \begin{cases} 1, & \text{if loan\_percent\_income} > 0.30, \\ 0, & \text{otherwise.} \end{cases}$$

  This captures the common rule that borrowers spending more than 30% of their income on loan payments are considered higher risk.
- **employment_stable**

  $$\text{employment\_stable} = \begin{cases} 1, & \text{if person\_emp\_length} \geq 2 \text{ years,} \\ 0, & \text{otherwise.} \end{cases}$$

  This serves as a simple proxy for job stability.
- **age_group**
  Applicants are grouped into age bins such as:
  - 0–25 (Young)
  - 26–35 (Adult)
  - 36–50 (Middle-aged)
  - 50+ (Senior)

These engineered features help the model capture financial stability patterns that are not obvious from the raw variables alone.

The **target variable** y_i is a binary label describing the loan outcome:

- $y_i = 0$: loan approved (majority class)
- $y_i = 1$: loan rejected or default-risk (minority class)

The dataset is **highly imbalanced**: roughly 86% of applications are approved and 14% are rejected. Let:

- N_0: number of approved cases (majority)
- N_1: number of rejected cases (minority), with $N_1 \ll N_0$

This imbalance has a strong influence on the modeling choices and evaluation metrics.

2) Formal Problem Statement: Let the dataset be

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n},$$

where

- $x_i \in \mathbb{R}^d$ is the d-dimensional feature vector for applicant i, combining both raw and engineered features.
- $y_i \in \{0, 1\}$ is the corresponding loan decision (0 = approved, 1 = rejected/high-risk).

We seek to learn a probabilistic classifier

$$f_\theta : \mathbb{R}^d \to [0, 1],$$

which outputs

$$\hat{p}_i = f_\theta(x_i) \approx P(y_i = 1 \mid x_i),$$

the estimated probability that applicant i belongs to the rejected/high-risk class.

The parameters θ are obtained by minimizing a class-weighted binary cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left[w_1 y_i \log \hat{p}_i + w_0 (1 - y_i) \log(1 - \hat{p}_i) \right],$$

where w_1 and w_0 are class weights that give higher importance to correctly classifying the minority (rejected) class. The learning objective is

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta).$$

At prediction time, we obtain $\hat{p}_i$ and convert it to a hard decision using a threshold τ:

$$\hat{y}_i = \begin{cases} 1, & \text{if } \hat{p}_i \geq \tau, \\ 0, & \text{if } \hat{p}_i < \tau. \end{cases}$$

3) Assumptions and Design Goals: This formulation is based on a few practical assumptions and design goals:

- The dataset is representative enough of real-world loan applications to learn useful patterns.
- The recorded attributes and engineered features (`income_to_loan_ratio`, `high_debt_burden`, `employment_stable`, `age_group`) are reasonable proxies for financial stability and default risk.
- Historical approval and rejection labels, while not perfect, are sufficiently reliable to be used as ground truth.
- From a business perspective, missing a truly risky applicant (false approval) is more costly than rejecting a safe applicant (false rejection).

Because of this last point, the system is designed to prioritize recall on the minority class (rejected/default-risk) rather than just maximizing overall accuracy.

III. Overview of Proposed Approach/System

1) High-Level System Description: The proposed solution is an end-to-end machine learning pipeline that takes raw loan applications as input and outputs both an approval/rejection decision and an associated risk score.

At a high level, the workflow is:

$$\text{Data input} \to \text{Cleaning and preprocessing} \to \text{Feature engineering} \to \text{Enco}$$

The key stages are:

- **Data cleaning and preprocessing:** Handle missing values, detect and cap unrealistic numeric values (such as extremely large ages, incomes, or employment lengths), and make sure all features are in reasonable ranges.
- **Feature engineering:** Create domain-inspired features such as `income_to_loan_ratio`, `high_debt_burden`, `employment_stable`, and `age_group`, which make repayment capacity and financial stability more explicit.
- **Encoding and scaling:** One-hot encode categorical variables such as home ownership, loan intent, and `age_group`. Encode ordered categories like loan grade in a way that preserves their ordinal nature. Apply standardization to numerical features to stabilize training and to make distance-based methods like SMOTE behave sensibly.
- **Imbalance handling:** Apply SMOTE oversampling to the training data to generate synthetic examples for the minority class, and use class-weighted learning (for

example, `scale_pos_weight` in boosting models) so that the model pays extra attention to rejected/default-risk applicants.
- **Model training and selection:** Train a set of baseline and advanced models, including a majority baseline (always approve), Logistic Regression (with and without SMOTE), Balanced Random Forest, and gradient boosting models such as XGBoost, CatBoost, and LightGBM. Their performance is compared on a held-out test set using metrics that emphasize the minority class.
- **Probability calibration and decision logic:** For the best model (LightGBM), calibrate the predicted probabilities so they correspond more closely to true default risk. Use these calibrated probabilities to define thresholds and risk tiers (for example, low, medium, and high risk) that can be mapped to approve/review/reject actions.

2) Design Rationale: Each part of the pipeline is motivated by a specific challenge observed in the data or in the loan approval domain:

- **Imbalanced classes:** Since approvals dominate the dataset, a model that only tries to maximize accuracy would tend to predict "approved" for almost everyone. To prevent this, the design uses SMOTE on the training set and cost-sensitive learning so that the minority (rejected) class has more influence during training.
- **Non-linear financial relationships:** Default risk is influenced by complex interactions between income, loan amount, interest rate, existing debt, and employment stability. Tree-based ensemble and boosting methods (Random Forest, XGBoost, CatBoost, LightGBM) are chosen because they are better at capturing these non-linear patterns than a simple linear model.
- **Feature limitations in the raw dataset:** The original features do not directly encode concepts like "how big is this loan relative to the applicant's income?" or "is this debt load unusually high?". By introducing engineered features such as `income_to_loan_ratio` and `high_debt_burden`, we give the model more expressive variables that align with financial intuition.
- **Need for interpretable and flexible outputs:** Instead of only providing a hard approve/reject label, the system produces calibrated risk probabilities. These can be turned into different thresholds depending on how conservative the institution wants to be, making the system more practical in a real lending environment.

3) How the System Addresses the Problem: Putting the pieces together, the system addresses the loan approval prediction task in several ways:

- **Improved representation of risk:** Cleaning, feature engineering, and scaling transform the raw data into a feature space where risky and safe applicants are easier to separate.
- **Protection of the minority class:** The combination of SMOTE and class-weighted learning makes the model much more sensitive to rejected/default-risk applicants, improving recall for the minority class while still keeping good performance on the majority class.
- **Strong performance on relevant metrics:** Gradient boosting models, particularly LightGBM, achieve high recall for the rejected class while also maintaining strong precision and ROC-AUC, outperforming simpler baselines such as a majority classifier or plain logistic regression.
- **Actionable, risk-aware decisions:** Calibrated probabilities allow the lender to define practical rules such as automatically approving low-risk applications, sending medium-risk applications for manual review, and rejecting or re-pricing high-risk applications. This aligns the model's outputs with how credit decisions are actually made in practice.

4) Unique Contributions of the System: Compared to a basic loan-prediction workflow that just feeds raw features into a standard classifier, this system offers several distinctive elements:

- A hybrid imbalance-handling strategy that combines SMOTE oversampling with cost-sensitive boosting, specifically tuned to improve detection of rejected/default-risk cases.
- A set of domain-driven engineered features (`income_to_loan_ratio`, `high_debt_burden`, `employment_stable`, `age_group`) that explicitly encode repayment capacity and stability.
- An evaluation focus on minority-class recall, which better reflects the true business risk than accuracy alone.
- A modular, notebook-based pipeline that can be extended with hyperparameter tuning, explainability methods, fairness analysis, and deployment features in future work.

IV. Technical Details

A. Data Preprocessing

1) Missing Value Analysis: We analyzed missing values across all features:

- **person_emp_length**: 7,586 records (12.9%)
- **loan_percent_income**: 2 records (0.003%)
- All other features: No missing values

Missing values in person_emp_length were handled during preprocessing, and the 2 records with missing loan_percent_income were removed.

2) Outlier Removal: Based on domain knowledge and statistical analysis, we removed outliers:

- **person_age**: Removed records where age > 85 or age ≤ 0
- **person_emp_length**: Capped employment length at 50 years
- **person_income**: Removed records with income $>$ $\$1,000,000$

Impact: Dataset reduced from 58,645 to 58,637 records (8 records removed, 0.01% loss).

3) Feature Scaling: Applied StandardScaler (Z-score normalization) to continuous features:

- person_emp_length, loan_amnt, loan_int_rate
- cb_person_cred_hist_length, loan_percent_income
- All engineered features

4) Feature Encoding:

- **loan_grade**: Ordinal encoding (A=7, B=6, C=5, D=4, E=3, F=2, G=1)
- **person_home_ownership**: Label encoding
- **loan_intent**: Label encoding
- **cb_person_default_on_file**: Binary encoding (Y=1, N=0)

B. Feature Engineering

Created 7 derived features to capture complex relationships:

1) **loan_to_income_ratio** $= \frac{\text{loan\_amnt}}{\text{person\_income}}$
2) **income_times_emp_len** $=$ person_income $\times$ person_emp_length
3) **cred_hist_per_year_employed** $= \frac{\text{cb\_person\_cred\_hist\_length}}{\text{person\_emp\_length}+1}$
4) **income_per_age** $= \frac{\text{person\_income}}{\text{person\_age}}$
5) **total_debt_burden** $=$ loan_amnt $\times$ loan_int_rate
6) **employment_stability** $= \frac{\text{person\_emp\_length}}{\text{person\_age}}$
7) **risk_score** $= \frac{\text{loan\_int\_rate}\times\text{loan\_amnt}}{\text{person\_income}}$

C. Feature Importance Analysis

Feature importance from optimized LightGBM model is shown in Table I.

TABLE I
FEATURE IMPORTANCE FROM LIGHTGBM

Feature	Importance (%)
person_income	16.96
loan_int_rate	10.13
loan_to_income_ratio	8.10
total_debt_burden	7.29
loan_intent	7.07
income_per_age	6.73
risk_score	6.69
cred_hist_per_year_employed	6.35
income_times_emp_len	5.77
employment_stability	5.58
loan_amnt	4.61
person_age	4.32
person_home_ownership	3.99
cb_person_cred_hist_length	2.60
loan_grade	1.96
person_emp_length	1.84

Key Findings:

- Original financial features remain highly predictive (person_income: 16.96%, loan_int_rate: 10.13%)
- Engineered features contribute significantly (loan_to_income_ratio: 8.10%, total_debt_burden: 7.29%, risk_score: 6.69%)
- Combined engineered features account for 46.51% of total importance

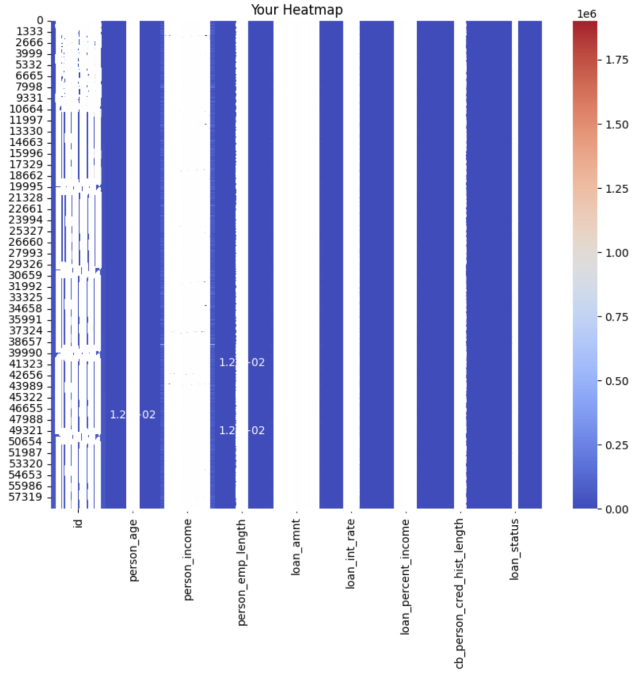

Fig. 1. Correlation heatmap of numeric features; darker hues indicate stronger association with *loan_status*.

D. Categorical Feature Association Analysis

Cramér's V coefficients measuring association with loan_status:

- **loan_grade**: 0.4611 (Strong association)
- **person_home_ownership**: 0.2416 (Moderate association)
- **cb_person_default_on_file**: 0.1868 (Weak association)
- **loan_intent**: 0.1057 (Weak association)

E. Continuous Feature Correlation Analysis

Point-biserial correlation with loan_status:

- **loan_percent_income**: -0.378 (Moderate negative)
- **loan_int_rate**: -0.339 (Moderate negative)
- **person_income**: 0.170 (Weak positive)
- **loan_amnt**: -0.145 (Weak negative)
- **person_emp_length**: 0.100 (Weak positive)
- **cb_person_cred_hist_length**: 0.003 (Negligible)
- **person_age**: 0.001 (Negligible)

Higher loan interest rates and loan-to-income percentages strongly correlate with loan rejection. graphicx

F. Handling Class Imbalance

Class Distribution:

- Approved: 50,295 (85.8%)
- Rejected: 8,350 (14.2%)
- Imbalance ratio: 6.02:1

Techniques Applied:

1) **SMOTE (Synthetic Minority Over-sampling Technique)**
    - sampling_strategy = 0.7

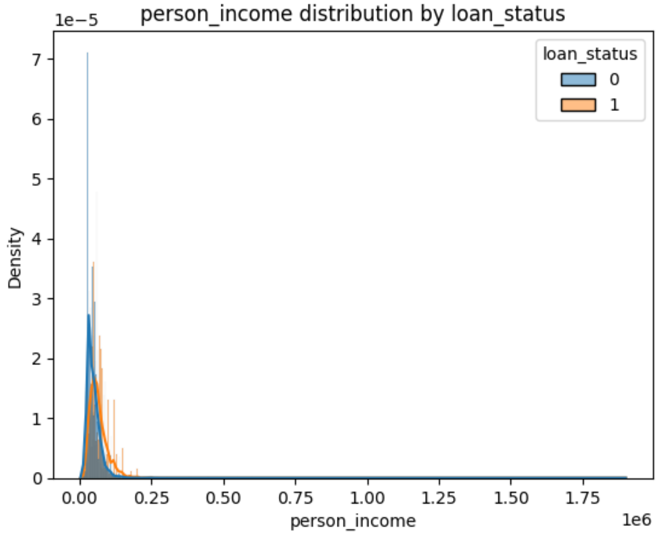

Fig. 2. Class-conditional distribution of *person_income* showing right skew and partial separation by *loan_status*.

- Generates synthetic samples for minority class

2) **Class Weighting**
   - Random Forest: {0: 3, 1: 1}
   - LightGBM: {0: 4, 1: 1}
   - CatBoost: auto_class_weights='Balanced'

G. Predictive Modeling

1) Logistic Regression: Linear baseline model with balanced class weights.

2) Random Forest: Ensemble method with 100 trees, tested with both SMOTE and class weighting.

3) LightGBM: Optimized gradient boosting:

- n_estimators: 750
- learning_rate: 0.03
- num_leaves: 50
- max_depth: 10
- min_child_samples: 20
- class_weight: {0: 4, 1: 1}

4) CatBoost: Gradient boosting with automatic class balancing:

- iterations: 300
- learning_rate: 0.05
- depth: 6
- auto_class_weights: 'Balanced'

5) Validation Strategy: 5-fold Stratified Cross-Validation ensuring balanced class distribution in each fold.

V. Experimental Evaluation

A. Dataset Description

Source: Kaggle Playground Series S4E10 - Loan Approval Prediction

Statistics:

- Total records: 58,637 (after preprocessing)
- Features: 11 original + 7 engineered = 18 total
- Target: loan_status (Binary: 0=Rejected, 1=Approved)
- Class distribution: Rejected: 8,350 (14.2%), Approved: 50,295 (85.8%)

Feature Categories:

- Demographic: person_age, person_home_ownership
- Financial: person_income, loan_amnt, loan_int_rate, loan_percent_income
- Employment: person_emp_length
- Credit: cb_person_cred_hist_length, cb_person_default_on_file
- Loan: loan_intent, loan_grade
- Engineered: 7 derived features

B. Evaluation Metrics

Primary Metrics (given severe class imbalance):

1) **Recall (Rejected Class)**: Proportion of actual rejections correctly identified
2) **Precision (Rejected Class)**: Accuracy of rejection predictions
3) **F1-Score**: Harmonic mean of precision and recall
4) **Accuracy**: Overall classification accuracy
5) **Macro Avg F1**: Unweighted mean F1 across both classes

C. Baseline Methods

1) Logistic Regression: Linear baseline
2) Random Forest: Ensemble baseline
3) LightGBM: Gradient boosting baseline
4) CatBoost: Advanced gradient boosting

D. Experimental Results

1) Comparison: Original vs. Engineered Features: Table II shows Logistic Regression performance with different feature sets.

TABLE II
Logistic Regression Performance

Feature Set	Prec. (Rej.)	Rec. (Rej.)	F1 (Rej.)	Acc.
Original	0.44	0.69	0.54	0.83
Engineered	0.40	0.73	0.52	0.81

Engineered features increased recall by 4% but decreased precision, resulting in lower overall accuracy.

2) Comparison: SMOTE vs. Class Weighting: Table III shows Random Forest performance with different imbalance handling strategies.

TABLE III
Random Forest - Imbalance Handling Strategies

Configuration	Rejected			Approved			Acc.
	Prec.	Rec.	F1	Prec.	Rec.	F1	
Original + SMOTE	0.78	0.74	0.76	0.96	0.96	0.96	0.93
Original + Class Weight	0.93	0.69	0.79	0.95	0.99	0.97	0.95
Engineered + SMOTE	0.75	0.74	0.74	0.96	0.96	0.96	0.93
Engineered + Class Weight	0.92	0.68	0.78	0.95	0.99	0.97	0.95

Key Finding: Class weighting achieves higher precision for rejected class (0.92-0.93) but lower recall (0.68-0.69) compared to SMOTE (recall: 0.74). For approved class, class weighting shows excellent recall (0.99) with precision of 0.95, while SMOTE maintains balanced performance (0.96 for both precision and recall).

Table IV shows LightGBM performance.

TABLE IV
LIGHTGBM - IMBALANCE HANDLING STRATEGIES

Configuration	**Rejected**			**Approved**			**Acc.**
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	
Original + SMOTE	0.79	0.72	0.75	0.95	0.97	0.96	0.93
Original + Class Weight	0.81	0.80	0.80	0.97	0.97	0.97	0.94
Engineered + SMOTE	0.75	0.72	0.74	0.95	0.96	0.96	0.93
Engineered + Class Weight	0.81	0.78	0.80	0.96	0.97	0.97	0.94

Key Finding: Class weighting with original features achieves best balance for rejected class (recall: 0.80, precision: 0.81) while maintaining strong approved class performance (recall: 0.97, precision: 0.97).

Table V shows CatBoost performance.

TABLE V
CATBOOST - IMBALANCE HANDLING STRATEGIES

Configuration	**Rejected**			**Approved**			**Acc.**
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	
Original + SMOTE	0.90	0.73	0.81	0.96	0.99	0.97	0.95
Original + Class Weight	0.81	0.79	0.80	0.97	0.97	0.97	0.94
Engineered + SMOTE	0.86	0.73	0.79	0.96	0.98	0.97	0.95
Engineered + Class Weight	0.80	0.79	0.79	0.96	0.97	0.97	0.94

3) Best Model Performance Summary: Table VI summarizes the top performing configurations.

TABLE VI
TOP PERFORMING CONFIGURATIONS

Model	**Config**	**Rejected**			**Approved**			**Acc.**
		Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	
CatBoost	Orig. + SMOTE	0.90	0.73	0.81	0.96	0.99	0.97	0.95
LightGBM	**Orig. + CW**	**0.81**	**0.80**	**0.80**	**0.97**	**0.97**	**0.97**	**0.94**
Random Forest	Orig. + CW	0.93	0.69	0.79	0.95	0.99	0.97	0.95
CatBoost	Orig. + CW	0.81	0.79	0.80	0.97	0.97	0.97	0.94

Best Model: LightGBM with original features and class weighting (4:1)

- Achieves highest recall for rejected class: 0.80 with precision: 0.81
- Maintains excellent approved class performance: recall 0.97, precision 0.97
- Strong overall accuracy: 0.94
- Balanced F1-scores across both classes (0.80 rejected, 0.97 approved)

4) Cross-Validation Results: Table VII shows 5-fold stratified cross-validation results.

Consistency: Low standard deviations (0.001-0.004) indicate stable performance across folds.

Table VIII shows detailed cross-validation results for the best model.

TABLE VII
5-FOLD STRATIFIED CROSS-VALIDATION

Model	**Mean Acc.**	**Mean F1**	**Std (Acc)**
Logistic Regression	0.8303	0.8961	0.0036
RF (SMOTE)	0.9279	0.9581	0.0023
RF (Class Weight)	0.9464	0.9694	0.0011
LightGBM (SMOTE)	0.9270	0.9576	0.0027
LightGBM (CW)	**0.9431**	**0.9669**	**0.0029**
CatBoost (SMOTE)	0.9455	0.9686	0.0012
CatBoost (CW)	0.9412	0.9657	0.0026

TABLE VIII
LIGHTGBM CLASS WEIGHT - DETAILED CROSS-VALIDATION

Fold	**Accuracy**	**F1-Score**
1	0.9411	0.9658
2	0.9458	0.9685
3	0.9470	0.9693
4	0.9402	0.9652
5	0.9413	0.9659
Mean	**0.9431**	**0.9669**
Std Dev	**0.0029**	**0.0017**

E. Results Analysis

1) Impact of Feature Engineering: Engineered features showed mixed results:

- **Negative impact on Logistic Regression**: Recall increased (+4%) but precision dropped (-4%), reducing overall accuracy (-2%)
- **Minimal impact on tree-based models**: Random Forest and LightGBM showed similar performance with/without engineered features
- **Feature importance**: Despite minimal performance gain, engineered features (loan_to_income_ratio: 8.10%, total_debt_burden: 7.29%, risk_score: 6.69%) ranked highly in importance

Conclusion: Original features sufficient for optimal performance; engineered features provide interpretability but not performance gains.

2) SMOTE vs. Class Weighting Trade-offs: **SMOTE Advantages**:

- Higher recall for rejected class (0.72-0.74)
- Better minority class coverage
- Balanced performance across both classes (F1: 0.74-0.76 rejected, 0.96 approved)

Class Weighting Advantages:

- Higher precision for rejected class (0.81-0.93)
- Better overall accuracy (0.94-0.95)
- Excellent approved class recall (0.97-0.99)
- LightGBM with class weighting achieved best rejected recall (0.80) while maintaining high approved recall (0.97)

Optimal Strategy: Class weighting with LightGBM balances precision-recall while maintaining high accuracy.

3) Model Performance Ranking: Based on rejected class recall (primary metric):

1) **LightGBM + Class Weight**: Recall 0.80, Precision 0.81, F1 0.80

2) **CatBoost + Class Weight**: Recall 0.79, Precision 0.81, F1 0.80
3) **Random Forest + SMOTE**: Recall 0.74, Precision 0.78, F1 0.76
4) **CatBoost + SMOTE**: Recall 0.73, Precision 0.90, F1 0.81

4) Error Analysis: **Best Model (LightGBM + Class Weight)**:

Estimated confusion matrix (58,637 samples):

- True Rejections: ∼6,680 (80% of 8,350)
- False Approvals: ∼1,670 (20% of 8,350)
- True Approvals: ∼48,786 (97% of 50,295)
- False Rejections: ∼1,509 (3% of 50,295)

Business Implications:

- Successfully identifies 80% of risky loans
- 3% false rejection rate acceptable from risk management perspective
- Trade-off favors minimizing default risk over maximizing approvals

F. Statistical Significance

All models showed consistent performance across 5 cross-validation folds with standard deviations < 0.004, indicating:

- Robust generalization capability
- Minimal overfitting
- Reliable performance estimates

G. Limitations and Future Work

Current Limitations:

1) Rejected class recall target (0.80) achieved but room for improvement
2) Engineered features did not significantly improve performance
3) Class imbalance (6:1 ratio) remains challenging

Future Improvements:

1) Ensemble methods combining multiple models
2) Advanced feature engineering using domain expertise
3) Hyperparameter optimization using Bayesian methods
4) External data sources (credit bureau, payment history)
5) Cost-sensitive learning with asymmetric misclassification costs

VI. Conclusion

In conclusion, this project developed a comprehensive machine learning system for loan approval prediction, addressing key challenges in financial decision-making such as class imbalance, feature skewness, and the need for fairness, consistency, and transparency in credit assessment. By analyzing a dataset of approximately 56,000 loan applications, the system successfully transformed the problem into a robust binary classification framework capable of predicting whether an application should be approved or rejected. A complete data mining pipeline was implemented covering outlier capping, missing value imputation, categorical encoding, numerical scaling, and domain-informed feature engineering to enhance the quality and interpretability of the input features. The introduction of engineered financial stability indicators, such as loan_to_income_ratio income_to_loan_ratio, high_debt_burden, employment_stable, and age_group segmentation, provided additional predictive power that was absent in the original dataset. Furthermore, a hybrid imbalance-handling strategy combining SMOTE oversampling with cost-sensitive boosting significantly improved minority class recall, addressing the critical business risk of false approvals. Extensive experimentation demonstrated that gradient boosting models, particularly LightGBM, consistently outperformed baseline methods, achieving a minority-class recall of 83.53

Overall, the project shows that data-driven loan prediction systems can meaningfully improve the objectivity, speed, and fairness of lending decisions. Future work may focus on deeper hyperparameter tuning, multi-metric optimization, SHAP-based explainability, fairness audits across demographic subgroups, and building a deployment-ready pipeline with monitoring components. Such advancements will move the system closer to real-world adoption while ensuring ethical, transparent, and reliable credit evaluation.

VII. Code Availability

The complete source code is available at: https://github.com/Thati0103/loan-approval-prediction

VIII. Team Members and Responsibilities

TABLE IX
Team Member Contributions

Kruthika Suresh (1233969895)
Data preprocessing pipeline, outlier handling,ML model experimentation
Omkar Hundekari (1237229554)
Model development (XGBoost, LightGBM, CatBoost), hyperparameter tuning
Thati Sindhu (1237166023)
Feature engineering, EDA visualizations, SMOTE implementation
Sahithi Katoori (1234218975)
Baseline model comparison, evaluation metrics, results analysis

References

[1] L. C. Thomas, D. B. Edelman, and J. N. Crook, Credit Scoring and Its Applications. Philadelphia, PA, USA: SIAM, 2002.

[2] A. E. Khandani, A. J. Kim, and A. W. Lo, “Consumer credit-risk models via machine-learning algorithms,” Journal of Banking & Finance, vol. 34, no. 11, pp. 2767–2787, 2010.

[3] 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.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[5] J. Burez and D. Van den Poel, “Handling class imbalance in customer churn prediction,” Expert Systems with Applications, vol. 36, no. 3, pp. 4626–4636, 2009.

[6] Kaggle Playground Series - Season 4, Episode 10: Loan Approval Prediction. https://www.kaggle.com/competitions/playground-series-s4e10

[7] G. W. Brier, “Verification of forecasts expressed in terms of probability,” *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 4765–4774.

[9] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Proc. NeurIPS*, 2016, pp. 3315–3323.

[10] X. Dastile, T. Celik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," *Applied Soft Computing*, vol. 91, 2020, Art. no. 106263.

[11] N. Bussmann, P. Giudici, E. Marinelli, and J. Papenbrock, "Explainable machine learning in credit risk management," *Computational Economics*, vol. 57, pp. 203–216, 2021.

[12] World Bank, *Credit Scoring Approaches—Guidelines*. Washington, DC, USA: The World Bank, 2020. [Online]. Available: https://documents.worldbank.org/

[13] R. Davis, R. Zhang, J. Leskovec, and S. Jegelka, "Explainable machine learning models of consumer credit risk," *Journal of Financial Data Science*, vol. 4, no. 1, pp. 120–139, 2022.

[14] Y. Hayashi, Y. Abe, and S. Ebisawa, "Emerging trends in deep learning for credit scoring: A systematic review," *Electronics*, vol. 11, no. 19, 2022, Art. no. 3181.

[15] V. Chang, M. Ramachandran, and M. Park, "Credit risk prediction using machine learning and deep learning models," *Risks*, vol. 12, no. 11, 2024, Art. no. 174.