Credit Risk Assessment

Manan Parikh, Kris Patel, Aditya Shah
MSDS Program, Rutgers University, New Brunswick, NJ, USA
{mpp150, ksp177, as5069}@scarletmail.rutgers.edu

***Abstract*—Credit risk assessment plays a crucial role in loan management for financial institutions. Traditional credit scoring methods often fail to detect complex patterns in borrower data. In this project, we explore both various machine learning approaches for predicting loan defaulters using a comprehensive dataset. We apply models such as Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, and Support Vector Machines, and K-Means Clustering along with PCA. Extensive EDA and feature engineering techniques are employed to enhance model performance. Evaluation is based on accuracy, precision, recall, F1-score, and ROC-AUC. The findings highlight the trade-offs of interpretability and performance among various methods and propose future directions for enhancement.**

I. Introduction

Credit risk assessment is essential for financial institutions to ensure repayment capacity and reduce bad debt. Conventional methods based on fixed thresholds or simple linear models often fail to capture nonlinear relationships and interactions. Machine learning techniques enable pattern discovery and probabilistic classification using historical borrower data.

This study applies data mining techniques to a credit risk classification problem using a comprehensive simulated dataset. We evaluate the performance of various different models and interpret the importance of various features influencing default.

II. Dataset and Exploratory Data Analysis

The dataset used in this project is a simulated credit risk dataset consisting of 32,581 observations, each corresponding to a loan applicant. It includes a wide range of financial, demographic, and behavioral attributes relevant to creditworthiness, such as annual income, home ownership status, loan purpose, interest rate, employment length, loan amount, loan grade, and credit history length. The binary target variable, $loan\_status$, indicates whether an applicant defaulted (1) or is not defaulted (0).

The dataset is moderately imbalanced, with a default rate of approximately 21.66%, which is realistic for credit risk modeling scenarios. Recognizing this imbalance early on helped inform our modeling strategy, including the use of evaluation metrics beyond accuracy.

Exploratory Data Analysis (EDA) provided critical insights that guided feature selection and model design. Key findings included:

- **Home Ownership:** A large proportion of defaulters were renters. We inferred that they had not much investments , they could be just starting their career.
- **Loan Intent:** Medical and education loans had the highest default rates, while home improvement and debt consolidation loans were relatively safer.
- **Interest Rate Distribution:** A clear upward skew in interest rates was observed among defaulters, indicating that interest rate is not only a pricing mechanism but also a risk proxy.
- **Income vs. Loan Amount:** Scatter plots revealed that lower-income applicants requesting larger loans were more likely to default. Logically, this makes sense, if you do not have enough income to pay your debts, you will probably default.

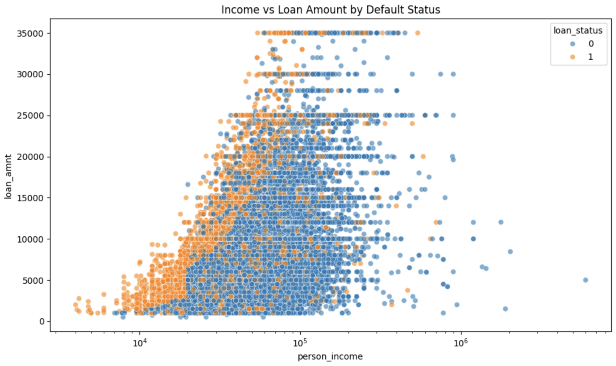

Fig. 1: income vs loan_status

- **Correlation Matrix:** Strong positive correlation was found between loan amount and percent income. A negative correlation between credit history length and default status suggested that longer histories are associated with better repayment behavior.

Overall, EDA validated our assumptions and helped in identifying high-impact variables and relationships, which were then encoded into features and used in the supervised models. The findings also supported risk profiling, which is essential in credit underwriting tasks.

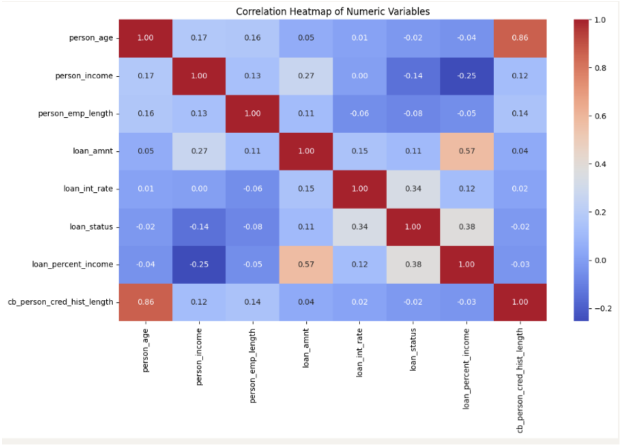

Fig. 2: Correlation Matrix

The Correlation matrix shows relationship between the variables and how strong or weak is a particular relationship between two covariates. For example there is a strong relationship between *loan_status* and *loan_int_rate*

III. Feature Engineering and Preprocessing

Effective feature engineering and pre-processing are critical to enhancing the predictive power and generalizability of machine learning models. In our project, we applied a structured pipeline to clean, transform, and enrich the dataset for credit risk modeling.

Null values handling: Missing data was handled with care to preserve statistical integrity. For numerical attributes such as income and interest rate, we applied median imputation to minimize the influence of outliers. For categorical features like home ownership and loan intent, we used maximum occurrences approach, replacing missing values with the most frequent category.

Transformation: Several variables exhibited right-skewed distributions, particularly income and loan amount. To normalize these and reduce variance, we used log transformation , which improved model stability and performance during training.

Encoding: To make categorical features usable by ML algorithms, we applied one-hot encoding to nominal variables (e.g., loan intent, home ownership) and label encoding for ordinal variables (e.g., loan grade). This preserved hierarchical relationships and prevented the model from assuming arbitrary order where none existed.

Engineered Features: We introduced domain-specific features based on expert knowledge:

- *Debt-to-Income Ratio:* This ratio was calculated as loan amount divided by annual income, providing insight into the applicant's repayment capacity.
- *Income* × *Interest Rate:* An interaction term capturing the joint effect of borrowing cost and financial strength. This improved the detection of high-risk profiles.

These engineered variables significantly enhanced model interpretability and predictive performance. The preprocessing pipeline ensured consistent data quality and formed the foundation for robust modeling in subsequent stages.

IV. Modeling Methodologies

A. Logistic Regression

Logistic Regression is a probabilistic linear classifier that models the log-odds of the probability of default as a linear function of the input variables:

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \qquad (1)$$

In our project, Logistic Regression served as a baseline due to its simplicity and interpretability. We used features such as loan percent income, interest rate, and credit history length. The model provided insight into how changes in predictors affected the likelihood of default. Although it had lower recall compared to ensemble methods, its transparent nature made it ideal for understanding risk indicators.

B. Decision Trees and Random Forest

We started with a Decision Tree model to better capture non-linear interactions between variables, something logistic regression often struggles with. A decision tree builds simple, interpretable if-else rules to segment the dataset. For example, the tree may first split applicants based on income and then on whether they rent or own a home. This structure allowed us to understand threshold effects, such as a critical cutoff in the loan percent income ratio that clearly separated defaulters from non-defaulters.

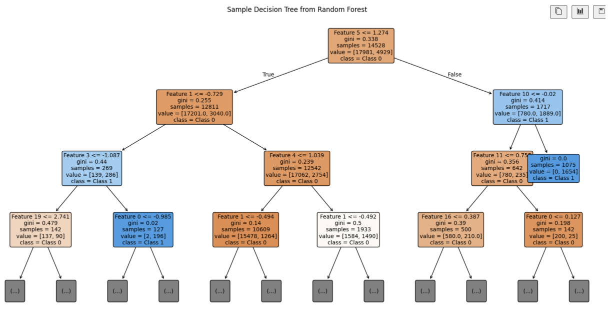

Fig. 3: Decision Tree

While the decision tree was useful for explanation, we encountered overfitting, especially when the tree depth grew large. This caused the model to perform well on the training data but poorly on unseen test data. Despite this, it gave us actionable insights into variables that influence credit decisions, such as person home ownership and employment length.

To overcome these limitations, we applied a Random Forest model. This ensemble technique builds multiple trees using bootstrapped subsets of the data and aggregates their predictions using majority voting. The benefit is reduced variance and better generalization. Random Forest also ranks features by importance, helping us identify loan percent income, credit score, and loan amount as the top predictors of default risk.

Compared to other models, Random Forest was the most robust in handling class imbalance and noise. It achieved the highest ROC-AUC in our experiments, confirming its strength in capturing complex patterns without overfitting. This model became our primary tool for making data-driven lending decisions.

C. Gradient Boosting and AdaBoost

Gradient Boosting builds models step by step, where each new model tries to correct the errors made by the previous one. Rather than adjusting weights like AdaBoost, Gradient Boosting minimizes a differentiable loss function using the gradient which reflects how wrong the model's predictions are. It fits new decision trees to these residual errors. This makes Gradient Boosting highly effective in modeling complex relationships and noisy, real-world data.

In our project, Gradient Boosting was applied after transforming and balancing the data. We observed that it handled mixed-risk borrower segments well and captured nuanced patterns between borrower income, loan amount, and repayment behavior. We tuned key hyperparameters like learning rate, number of estimators, and maximum tree depth. To prevent overfitting, we implemented early stopping based on validation performance. Gradient Boosting showed excellent generalization and ranked second in overall ROC-AUC performance.

AdaBoost (Adaptive Boosting), in contrast, focuses explicitly on mistakes made by earlier models. After each weak learner is trained, AdaBoost increases the weights of misclassified examples, forcing subsequent models to focus more on the difficult cases. All the learners then vote to determine the final classification, but models with better performance get more influence. This mechanism makes AdaBoost faster and simpler than Gradient Boosting, but more sensitive to outliers.

In our experiments, AdaBoost performed well on cleaner subsets of data and demonstrated competitive accuracy and precision. However, it was slightly less robust to noise compared to Gradient Boosting. Both models, being boosting algorithms, contributed valuable ensemble diversity and reinforced the strength of our modeling suite.

D. Support Vector Machines (SVM)

Support Vector Machine (SVM) is a powerful and flexible supervised learning algorithm used for classification tasks. The core idea is to find the hyperplane that best separates the data into different classes while maximizing the margin, the distance between the hyperplane and the nearest data points. In cases where the data are not linearly separable, kernel functions, such as a polynomial or radial basis function (RBF), can project the data into higher-dimensional space to enable separation.

In our credit risk assessment task, we used Principal Component Analysis (PCA) to reduce the dimensionality of the dataset after applying SVM. We used PCA to visualize our predicted graph. We had a total of 13 dimensions , we have tha reduced down to 3

One of the most insightful aspects of this approach was visualizing the data in three dimensions. The resulting graph shows data points colored by the predicted probability of default: red for higher risk, blue for lower risk. A white margin zone appears where the model is least certain these are soft margins, which allow for some misclassification and enhance generalization.

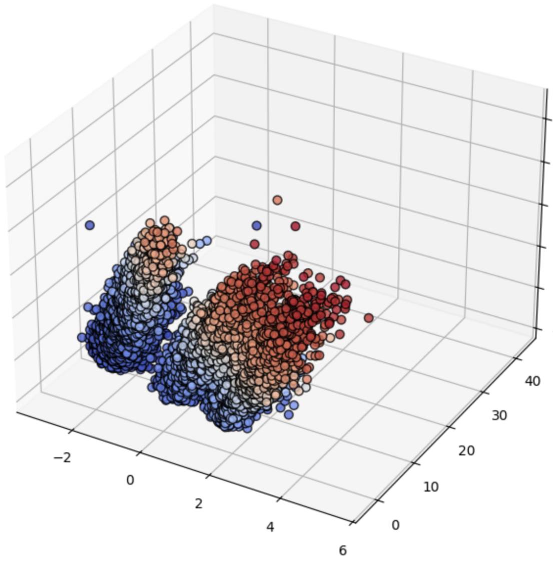

Fig. 4: Support vector machine

This visualization demonstrated the SVM's effectiveness in carving out two well-separated regions in the feature space. Borderline applicants fall close to the margin, providing a natural confidence interval. To convert SVM outputs into probabilities for ROC analysis, we employed Platt scaling, which fits a sigmoid function to the model's decision values. Overall, SVM offered a robust, margin-based classifier with good predictive performance and valuable visual interpretability.

E. Principal Component Analysis (PCA)

Principal Component Analysis, or PCA, is a widely used technique for dimensionality reduction that preserves the directions of highest variance in the data while transforming it into a new set of uncorrelated variables called principal components. This transformation is particularly useful for visualization, noise reduction, and improving model performance.

In our project, PCA played a critical role in simplifying the high-dimensional feature space. We projected the original variables into a lower-dimensional subspace consisting of the top 2 or 3 principal components. These components retained much of the essential structure of the data and enabled us to better visualize patterns and groupings. Specifically, the first three components captured approximately 27.4% of the total variance, broken down as follows: PC1 explained 10.52%, PC2 explained 9.42%, and PC3 accounted for 7.45%.

While PCA is inherently a lossy technique, in this case the reduction still preserved sufficient information to identify meaningful clusters and separate default risk profiles. This transformation proved especially useful in conjunction with Support Vector Machines and K-Means clustering. The 3D PCA projection allowed us to see where high-risk applicants concentrated and how clearly the classifier separated different segments.

Additionally, PCA helped reduce model training time and computational overhead by compressing irrelevant or redundant dimensions. It also improved interpretability when showcasing model behavior to stakeholders through visuals. Despite being unsupervised, PCA contributed directly to both performance and clarity in our machine learning pipeline.

F. K-Means Clustering

K-Means Clustering is an unsupervised learning algorithm used to partition data into a predefined number of clusters (k). It minimizes the within-cluster variance by iteratively assigning points to the nearest cluster centroid and then recalculating those centroids. The goal is to group similar instances together, which can reveal underlying structures in the data.

In our credit risk analysis, K-Means was applied after reducing the dataset with PCA, ensuring that the clustering was not affected by multicollinearity or high-dimensional noise. Our understanding of the two Principal Components were:

- If PC1 has high values for $loan\_amount$, income, and $credit\_score$, it might represent financial strength.
- If PC2 has high loadings on $home\_ownership$ and $employment\_length$, it might capture stability.

We selected $k = 3$ based on silhouette scores and domain knowledge, aiming to distinguish between low-risk, medium-risk, and high-risk borrowers.

The resulting clusters revealed meaningful groupings:

- **Cluster 0:** Represented applicants with lower loan-to-income ratios, longer credit histories, and generally lower interest rates. These profiles aligned with low-risk borrowers.
- **Cluster 1:** Contained applicants with more variability mixed credit histories and mid-range loan values. This group showed moderate risk and could benefit from targeted policies.
- **Cluster 2:** Dominated by high-risk profiles high loan amounts relative to income, short employment lengths, and higher interest rates.

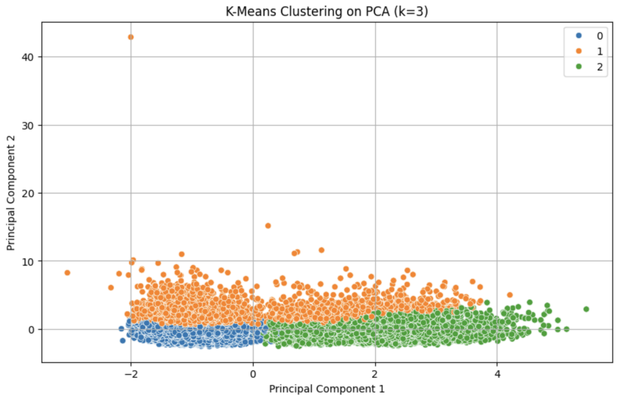

Fig. 5: K-means clustering

The clustering results provided an interpretable segmentation that stakeholders could use to differentiate marketing strategies or underwriting rules. Visualizations in PCA space made the results easy to interpret and communicate.

K-Means added an exploratory layer to our credit risk pipeline by uncovering natural groupings not bound by the training labels, enhancing both model validation and business insight.

V. Evaluation Metrics and Results

TABLE I: Model Comparison

Model	Accuracy	Precision	Recall	F1	ROC-AUC
LogReg	0.86	0.76	0.54	0.63	0.87
DecTree	0.88	0.73	0.75	0.74	0.84
RandForest	**0.93**	**0.95**	0.72	**0.81**	**0.93**
SVM	0.91	0.92	0.62	0.74	0.90
GradBoost	0.92	0.92	0.69	0.79	0.92
AdaBoost	0.88	0.80	0.62	0.70	0.90

Logistic Regression, while interpretable, struggled with recall, indicating it missed many actual defaulters. This is typical for linear models in complex data. Decision Tree improved recall but slightly underperformed in AUC, revealing its tendency to overfit.

Random Forest delivered the best all-round performance, with the highest precision and ROC-AUC, confirming its ability to capture complex interactions and remain stable across samples. SVM was also strong, showing high precision and decent recall. Its margin-based nature gave it a good balance, though not quite at the level of Random Forest.

Gradient Boosting closely followed, showing competitive metrics across all evaluation points, particularly in recall and AUC, which highlights its iterative focus on reducing residual errors. AdaBoost performed well on cleaner subsets but was slightly more sensitive to noise.

Visual Validation: The ROC curve illustrates the trade-off between sensitivity and specificity. Curves closer to the top-left represent better performance. This is because the AUC then comes close to 1. Confusion matrices helped us analyze misclassifications, especially the false negatives (undetected defaulters), which are costly in credit risk settings.

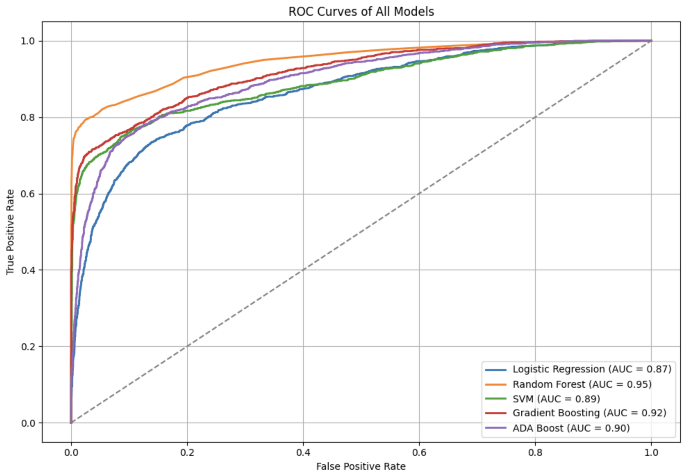

Fig. 6: ROC Curve

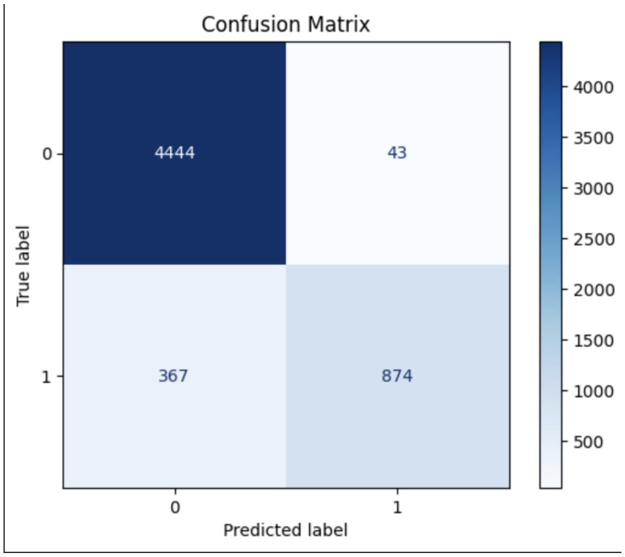

Fig. 7: Confusion Matrix

VI. Conclusion and Future Work

In this project, we explored a range of machine learning models for predicting credit default risk, applying them to a comprehensive dataset with engineered features and robust evaluation. Among all models tested, Random Forest delivered the most consistent and highest-performing results, striking an optimal balance between precision, recall, and overall discriminative ability. Gradient Boosting also performed competitively, especially in capturing complex borrower profiles, while Logistic Regression remained valuable for its interpretability. SVM and AdaBoost contributed with solid precision and recall, showing their utility in nuanced decision boundaries and clean data segments, respectively.

Beyond individual model performance, techniques like PCA and K-Means helped reveal underlying structure in the data and supported model interpretability through dimensionality reduction and segmentation.

For future work, we propose integrating transactional and behavioral data, which could offer deeper context to borrower risk assessment. Advanced resampling techniques like SMOTE can help address class imbalance more effectively. We also aim to explore deep learning models and ensemble stacking approaches for further performance gains. Finally, enhancing explainability through SHAP values or LIME could increase stakeholder trust in model recommendations, making these techniques more deployable in high-stakes lending environments.

References

[1] L. Breiman, ”Random forests,” Machine learning, 2001.

[2] J. Friedman, ”Greedy function approximation: A gradient boosting machine,” 2001.

[3] I. Guyon and A. Elisseeff, ”An introduction to variable and feature selection,” JMLR, 2003.