Loan Approval Prediction Using Machine Learning

Meredith A. Hughes
Data Science Program, Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
meredith.hughes@stthomas.edu

Sruthi Vemavarapu
Data Science Program, Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
vema3735@stthomas.edu

Massara Terfassa
Data Science Program, Graduate Programs in Software
University of St. Thomas
St. Paul, MN, USA
terf0002@stthomas.edu

Abstract - This project investigates loan approval prediction using supervised machine learning techniques applied to a structured dataset of 4,269 loan applications. The dataset includes financial attributes, credit information, and basic demographic indicators. After initial exploration, the data was cleaned, encoded, and transformed through log scaling of skewed financial variables, standardization of numeric features, and the creation of engineered indicators.

Three classification models were developed and evaluated: Logistic Regression, Random Forest, and XGBoost. All models were trained using a consistent preprocessing pipeline and assessed through a stratified 70/30 train-test split and five-fold cross-validation on the training set. Performance was measured using accuracy, precision, recall, F1-score, and the area under the ROC curve.

The results show that both ensemble models significantly outperformed the linear baseline. Random Forest and XGBoost achieved test accuracies above 98% and ROC-AUC values near 1.00, with XGBoost showing slightly higher recall and F1-score. These findings indicate that tree-based ensemble methods are highly effective for loan approval prediction in this dataset and capture important nonlinear relationships among the features

Keywords - loan approval, credit scoring, classification, logistic regression, random forest, XGBoost, ROC-AUC, F1-score.

I. Introduction

Determining whether a loan application should be approved or rejected is a major task in consumer finance. Banks and lending institutions make these decisions every day, and inaccurate assessments can lead to financial losses or unfair treatment of customers. Machine learning methods are widely used in this area because they can detect patterns in applicant information that may not be obvious through traditional manual review.

The goal of this project is to build and evaluate three machine learning models that predict loan approval status using historical application data. The dataset includes financial attributes such as income, loan amount requested, and asset values, as well as categorical variables such as education level and employment type. The target variable indicates whether each past loan application was approved or rejected.

This work follows a typical supervised learning workflow. The raw dataset is analyzed and prepared through a series of cleaning and preprocessing steps before model development. Three classification models are developed and compared: Logistic Regression, Random Forest, and XGBoost. These models were chosen because they represent different levels of model complexity and are commonly used in credit scoring and loan prediction research.

The purpose of this project is to evaluate which model performs best on the dataset, how preprocessing decisions influence performance, and which features appear most informative for predicting loan approvals. The results provide insight into how machine learning techniques can support data-driven decision-making in financial settings.

II. Related Work

Machine learning methods have been widely applied to credit scoring and loan approval prediction. Traditional lending systems often rely on Logistic Regression because of its interpretability and long history of use in financial risk assessment [1]. Prior studies show that Logistic Regression performs well when relationships between predictors and outcomes are mostly linear and when interactions among features are limited.

More recent research has explored tree-based models such as decision trees, Random Forests, and gradient boosting for forecasting loan performance and credit default. Comparative studies indicate that ensemble techniques frequently outperform traditional statistical models in accuracy and generalization, especially on structured financial datasets [2]. These methods are effective at capturing nonlinear relationships and interactions that linear models may not represent well.

Boosting algorithms such as XGBoost have gained significant attention due to their strong predictive performance, robustness, and efficiency across a variety of credit risk datasets. Prior work consistently reports that boosting approaches often achieve higher accuracy than both linear models and bagging ensembles [3].

Overall, the literature supports the model choices in this project. Logistic Regression provides a meaningful baseline for comparison, while Random Forest and XGBoost offer more flexible nonlinear alternatives for evaluating loan approval prediction.

III. DATASET DESCRIPTION

The dataset used in this project contains 4,269 loan applications and 13 original variables. Each row represents a single applicant and includes information about income, loan amount, asset values, credit score, and basic demographic characteristics. The target variable, *loan_status*, indicates whether the applicant's loan was approved or rejected.

The dataset includes a combination of numeric and categorical fields. Financial variables such as *income_annum*, *loan_amount*, and the different asset value fields are stored as numeric features. The categorical attributes include *education* and *self_employed*, which describe the applicant's education level and employment type. All variables in the dataset are complete, and no missing values were present.

A summary of the original variables is provided in **Table I.** The dataset contains a higher proportion of approved applications compared to rejected ones, and this class distribution is shown in **Table II**. This imbalance is addressed later through stratified splitting and stratified cross-validation to preserve the approval ratio during model training.

TABLE I. SUMMARY OF ORIGINAL DATASET FEATURES

Feature	Type	Description
loan_id	Identifier	Unique ID assigned to each loan application.
no_of_dependents	Numeric	Number of individuals financially supported by the applicant.
education	Categorical	Applicant's education level ("Graduate" or "Not Graduate").
self_employed	Categorical	Employment type indicating whether the applicant is self-employed (Yes / No)
income_annum	Numeric	Annual income of the applicant
loan_amount	Numeric	Requested loan amount.
loan_term	Numeric	Duration of the loan
cibil_score	Numeric	Applicant's credit score.
residential_assets_value	Numeric	Value of residential property
commercial_assets_value	Numeric	Value of commercial property
luxury_assets_value	Numeric	Value of luxury assets owned.
bank_asset_value	Numeric	Total value of bank assets
loan_status	Target	Original loan decision ("Approved" or "Rejected").

TABLE II. CLASS DISTRIBUTION OF LOAN APPLICATIONS

Loan Status	Count	Proportion
Approved (1)	2656	0.6222
Rejected (0)	1613	0.3778

A. Correlation Analysis

A correlation heatmap was generated to examine relationships among the numeric variables in the dataset. Several patterns were observed:

Income and loan amount are strongly correlated (approximately 0.93), which is expected because applicants with higher income typically qualify for larger loan amounts.

The various asset value fields (residential, commercial, luxury, and bank assets) show moderate correlations with one another, reflecting their shared financial characteristics.

cibil_score does not strongly correlate with any other numeric variable, indicating that it provides distinct predictive information rather than overlapping with income or asset-based features.

no_of_dependents shows almost no correlation with any other variable, suggesting that it captures a separate dimension of applicant characteristics.

Overall, the numeric features exhibit a mixture of weak and moderate correlations, suggesting the presence of nonlinear and interacting relationships in the financial data.

These correlation patterns help explain later modeling results, where tree-based algorithms such as Random Forest and XGBoost performed especially well due to their ability to capture nonlinear feature interactions. The relationships among the numeric variables are summarized in the correlation heatmap shown in **Figure 1.**

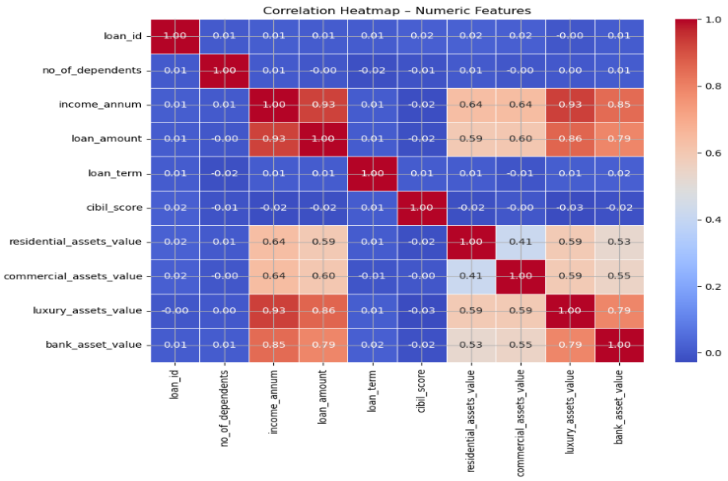

Fig. 1. Correlation Heatmap of Numeric Features

IV. DATA PREPARATION

This section describes how the raw dataset was transformed into a set of features suitable for machine learning models. The main steps included encoding categorical variables, constructing a numeric target variable, handling invalid values, and building a preprocessing pipeline used across all models.

A. Target Construction and Class Balance

The original target column, *loan_status*, was stored as text with values "Approved" and "Rejected." This variable was converted into a numeric target named *loan_status_numeric*, where 1 represents approved applications and 0 represents rejected ones. After this conversion, the class balance was reviewed. Approved loans accounted for approximately 62

percent of the dataset, while rejected loans made up about 38 percent. Because the classes are not perfectly balanced, stratified splitting and stratified cross-validation were used to preserve this ratio during model training and evaluation.

B. Categorical encodings

Two categorical predictors were present in the dataset: *education* and *self_employed*. These variables were encoded using simple binary transformations.

- Education → education_graduate
  The *education* column was converted into a binary feature named *education_graduate*, where 1 indicates "Graduate" and 0 indicates "Not Graduate."
- self_employed → self_employed_yes
  The *self_employed* column was encoded using one-hot encoding with *drop_first = True*, which produced a single indicator variable named *self_employed_yes*, where 1 represents self-employed applicants.

After encoding, the original education and self_employed columns were removed, and only the engineered variables were retained for modeling.

C. Handling negative residential asset values

During exploratory checks, some applications were found to contain a negative value for *residential_assets_value* (-100000). Because asset values should not be negative, these entries were treated as data quality issues. A binary indicator named *residential_assets_value_was_negative* was created to mark rows that originally contained negative values. The negative entries in *residential_assets_value* were then replaced with zero. After applying this correction, the minimum residential asset value in the dataset is 0, and the flag variable preserves information about which rows were affected.

D. Feature set and preprocessing pipeline

After the encoding and cleaning steps, the final feature set used for modeling included the following groups:

- **Money-related variables:**
  income_annum, loan_amount, residential_assets_value, commercial_assets_value, luxury_assets_value, bank_asset_value
- **Other numeric variables:**
  no_of_dependents, loan_term, cibil_score
- **Binary engineered variables:**
  education_graduate, self_employed_yes, residential_assets_value_was_negative

These twelve features were processed using a preprocessing pipeline. The money-related variables were first log transformed (with clipping applied to prevent undefined values) and then standardized using StandardScaler. The remaining numeric variables were standardized without log transformation, and the binary variables were passed through without scaling. A ColumnTransformer was used to ensure that these transformations were applied consistently across all model pipelines.

The resulting feature matrix X had shape (4269, 12), and the target vector y had length 4269. These prepared inputs were used for the train/test split and the model training procedures described in the following sections. A summary of the main data preparation steps is provided in **Table III.**

TABLE III. SUMMARY OF DATA PREPARATION STEPS

Step	Variables	Description
Target encoding	loan_status→loan_status_numeric	Converted the original loan decision ("Approved" / "Rejected") into a numeric target where 1 = approved and 0 = rejected.
Categorical encoding (education)	Education→education_graduate	Created a binary feature indicating whether the applicant is a graduate (1) or not (0), then dropped the original education column.
Categorical encoding (employment)	self_employed→self_employed_yes	Applied one-hot encoding with drop_first=True to create a single indicator for self-employed applicants, then dropped the original self_employed column.
Negative asset handling	residential_assets_value, residential_assets_value_was_negative	Flagged rows with negative residential asset values, stored this in residential_assets_value_was_negative, and replaced negative values in residential_assets_value with 0.
Log transformation of money fields	income_annum, loan_amount, residential_assets_value, commercial_assets_value, luxury_assets_value, bank_asset_value	Applied a clipped log1p transformation to reduce skewness in money-related variables before scaling.
Standardization of numeric fields	income_annum, loan_amount, residential_assets_value, commercial_assets_value, luxury_assets_value, bank_asset_value, no_of_dependents, loan_term, cibil_score	Standardized all numeric features using StandardScaler inside the preprocessing pipeline.
Binary features passthrough	education_graduate, self_employed_yes, residential_assets_value_was_negative	Kept binary engineered features as-is (no scaling) and passed them directly through the preprocessing pipeline.

E. Dimensionality Reduction (PCA Analysis)

As part of the dimensionality reduction requirement, Principal Component Analysis (PCA) was applied to the nine numeric features in the dataset. The purpose of this analysis was

to determine whether a lower-dimensional representation could retain most of the variance while still supporting accurate classification.

Before applying PCA, all numeric predictors were standardized using the preprocessing steps described earlier. PCA was then fitted on the training set. The first principal component (PC1) explained approximately 49.7 percent of the variance, and the first two components together accounted for about 61 percent. The cumulative explained-variance curve indicated that at least six components would be required to retain 95 percent of the variance, suggesting that the information in the dataset is distributed across several financial variables rather than concentrated in a small number of dominant components.

A two-dimensional visualization using PC1 and PC2 revealed substantial overlap between approved and rejected applications. The classes do not separate cleanly in the PCA-transformed space, showing that a low-dimensional representation is insufficient for capturing the structure needed for accurate prediction.

This comparison evaluates two models: one trained on the full set of standardized numeric features and another trained on only the first two principal components. The performance comparison is summarized in **Table IV**.

TABLE IV. PERFORMACE OF LOGISTIC REGRESSION WITH AND WITHOUT PCA

Model	Accuracy	F1 Score	ROC-AUC
Full numeric feature set	0.918	0.934	0.973
PCA(2 components)	0.607	0.735	0.631

The PCA-based model performed substantially worse across all evaluation metrics. This result indicates that dimensionality reduction does not benefit this dataset. Because the financial attributes each provide distinct and complementary predictive information, reducing the feature space removes important signal. Tree-based models such as Random Forest and XGBoost also do not benefit from PCA, as they naturally handle correlated and nonlinear features.

For these reasons, PCA is included in this project as an exploratory dimensionality-reduction analysis, but the final modeling pipeline uses the full feature set without PCA.

Figure 2 Cumulative explained variance by PCA components.
Figure 3 Training data projected onto the first two PCA components.

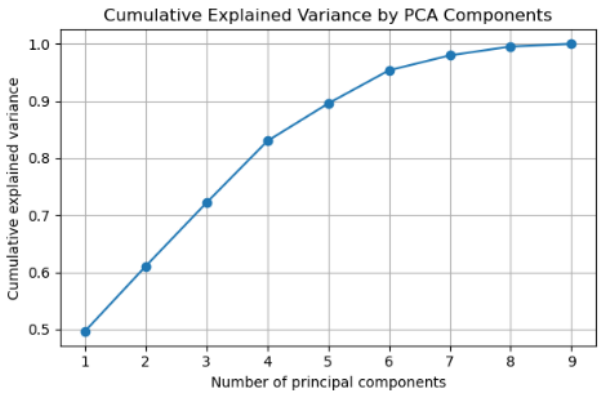

Fig. 2. Cumulative Explained Variance by PCA Components.

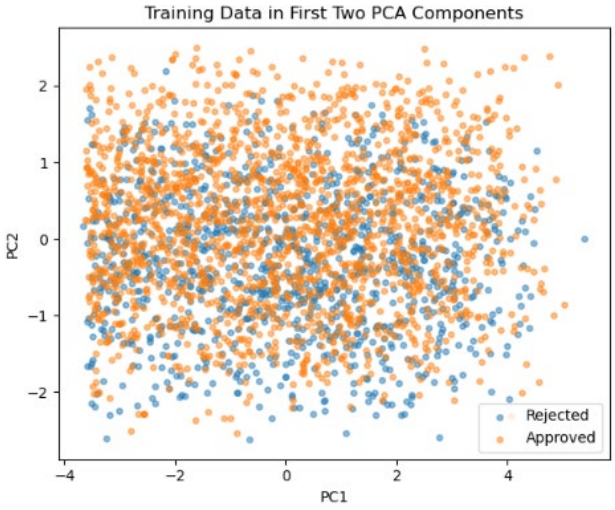

Fig. 3. Training data projected onto the first two PCA components.

V. MODELS AND METHODS

This section describes the modeling approach used in the project, including the preprocessing pipeline, the three classification models, and the evaluation procedures. All models were developed using a consistent workflow to ensure that their performance could be compared fairly.

A. Overall Modeling Framework

All models were implemented using the scikit-learn pipeline structure. The pipeline first applies the preprocessing steps described earlier, including log transformations for skewed financial variables, standardization of numeric features, and passthrough of binary indicators, and then fits the selected classifier. Using a unified pipeline ensures that identical transformations are applied during both training and testing, which prevents data leakage and keeps the workflow organized and reproducible.

The dataset was divided using a stratified 70/30 train-test split to preserve the original class distribution. Model performance was further assessed using stratified 5-fold cross-validation on the training set. This provides a more reliable estimate of generalization performance and helps reduce variance that may arise from relying on a single train-test split.

B. Logistic Regression

Logistic Regression was chosen as the baseline model. It is a commonly used method in credit scoring because it is straightforward to interpret and performs well when relationships between variables and the target are mostly linear. The model was trained using the preprocessed features, with the liblinear solver and a maximum of 1000 iterations to ensure convergence. Logistic Regression helps establish a reference point for evaluating the benefits of more flexible, nonlinear models.

C. Random Forest

Random Forest is an ensemble learning method that combines the predictions of many decision trees. It performs well on structured tabular datasets and naturally captures nonlinear relationships and interactions among features. In this project, the model was configured with 300 trees and no maximum depth, allowing each tree to grow until all leaves are

pure or until the minimum split criteria are met. Because the method averages predictions across many trees, Random Forest is generally stable and resistant to overfitting. These characteristics make it a strong candidate for credit-related prediction tasks.

D. XGBoost

XGBoost is a gradient boosting model that builds trees sequentially, with each new tree attempting to correct the errors made by the previous ones. It is widely used for financial and other tabular datasets because of its strong predictive performance and efficient training process. In this project, the model was configured with 300 trees, a learning rate of 0.05, a maximum depth of 4, and subsampling applied to both rows and features to improve generalization. XGBoost is able to capture complex nonlinear relationships and interactions, making it a strong competitor to Random Forest for loan approval prediction.

E. Evaluation Procedure

All models were evaluated using the same set of metrics to allow direct comparison. The metrics included accuracy, precision, recall, F1-score, and the area under the ROC curve (ROC-AUC). These measures capture different aspects of model performance and are commonly used in binary classification tasks. Cross-validation scores were computed using stratified 5-fold splits on the training data, and final performance was assessed using the held-out test set. This approach helps ensure that the results are not overly influenced by a single train-test split.

A summary of the key configuration settings for each model is provided in **Table V.**

TABLE V. KEY MODEL CONFIGURATION SETTINGS

Model	Main Settings
Logistic Regression	Solver: liblinear; Maximum iterations: 1000; Random state: 42
Random Forest	Number of trees (n_estimators): 300; Maximum depth: None; Minimum samples split: 2; Minimum samples leaf: 1; Random state: 42
XGBoost Classifier	Number of trees (n_estimators): 300; Learning rate: 0.05; Maximum depth: 4; Subsample: 0.9; Column sample by tree (colsample_bytree): 0.9; Objective: binary:logistic; Tree method: hist; Random state: 42

VI. RESULTS

This section presents the performance of the three machine learning models on both cross-validation and the held-out test set. The goal is to compare how well Logistic Regression, Random Forest, and XGBoost predict loan approval status after applying the preprocessing steps described earlier.

A. Train/Test Split Overview

The dataset was divided into training and test sets using a stratified 70/30 split to maintain the original proportion of approved and rejected applications. The training set contained 2,988 records, and the test set contained 1,281 records. The proportion of approved loans (class label 1) remained consistent across all partitions, with a ratio of 0.62 in the overall dataset, the training set, and the test set.

Maintaining the same class ratio in each partition ensures that the evaluation is representative of the full dataset and prevents the models from becoming biased toward the majority class. This stratified split provides a reliable foundation for the cross-validation and test-set evaluations that follow.

B. Logistic Regression Performance

Logistic Regression served as the baseline model for this project. On the test set, the model achieved an accuracy of 0.91, a precision of 0.92, a recall of 0.94, an F1-score of 0.93, and a ROC-AUC of 0.97. These results show that even a linear classifier can capture meaningful patterns in the dataset.

Figure 4 shows the confusion matrix for Logistic Regression. The model correctly classified most approved and rejected applications, with 422 true negatives and 747 true positives. However, it produced 62 false negatives and 50 false positives. Although these error levels are reasonable for a linear model, they are noticeably higher than those of the ensemble methods evaluated later.

Figure 5 presents the ROC curve for Logistic Regression. The area under the curve is approximately 0.97, demonstrating strong separation between the positive and negative classes. This indicates that Logistic Regression is effective at ranking applications by approval likelihood, even though its overall accuracy remains lower than that of the tree-based models.

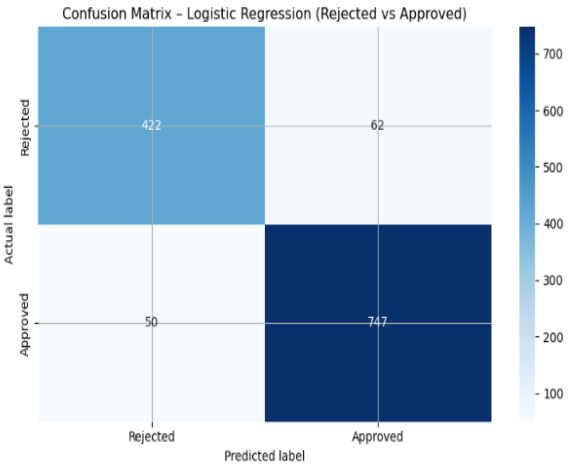

Fig. 4. Confusion Matrix: Logistic Regression (Rejected Vx Approved)

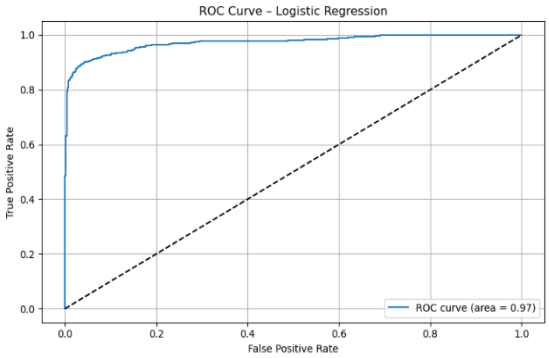

Fig. 5. ROC Curve: Logistic Regression

C. *Random Forest Performance*

The Random Forest model showed a substantial improvement over the Logistic Regression baseline. On the test set, it achieved an accuracy of 0.99, a precision of 0.99, a recall of 0.99, an F1-score of 0.99, and a ROC-AUC of 0.99. These results indicate that Random Forest is highly effective at capturing the nonlinear relationships and feature interactions present in the dataset.

Figure 6 presents the confusion matrix for the Random Forest model. The model correctly classified nearly all approved and rejected applications, with only 9 false negatives and 9 false positives. This demonstrates the stability and reliability of the ensemble approach.

Figure 7 shows the ROC curve for the Random Forest model. The curve is positioned very close to the top-left corner of the plot, indicating excellent separation between the two classes. The resulting ROC-AUC is approximately **1.00**, confirming that Random Forest performs extremely well in ranking loan applications by approval likelihood.

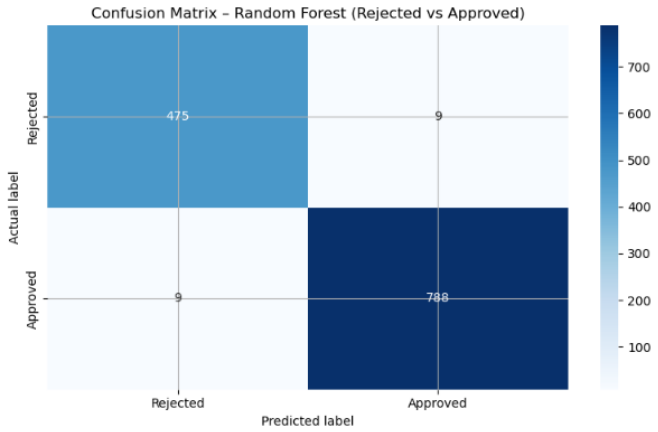

Fig. 6. Confusion Matrix: Random Forest (Rejected Vx Approved)

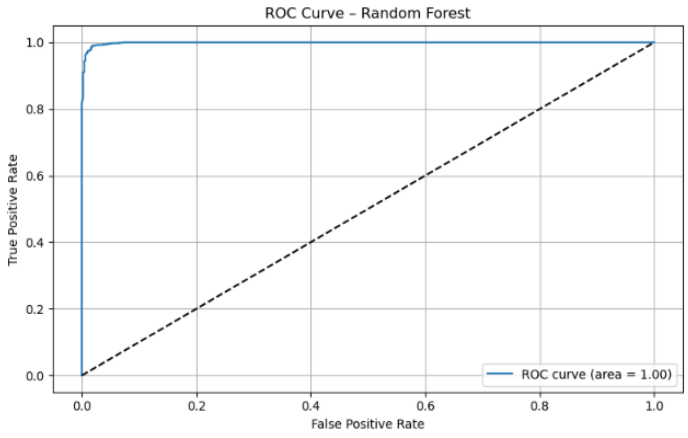

Fig. 7. ROC Curve: Random Forest

D. *XGBoost Performance*

XGBoost achieved the strongest overall performance among the three models. On the test set, it produced an accuracy of 0.99, a precision of 0.99, a recall of 0.99, an F1-score of 0.99, and a ROC-AUC of approximately 1.00. These results closely match those of the Random Forest model and confirm that XGBoost is highly effective for this prediction task.

Figure 8 shows the confusion matrix for XGBoost. The model made very few misclassifications, correctly predicting nearly all approved and rejected applications. It produced only 8 false negatives and 9 false positives, demonstrating its ability to capture complex structure in the data, including nonlinear patterns and subtle interactions among financial variables.

Figure 9 displays the ROC curve for XGBoost. The curve lies extremely close to the top-left corner of the plot, indicating a high true-positive rate and a low false-positive rate across all thresholds. The resulting ROC-AUC, close to 1.00, confirms that XGBoost performs exceptionally well in ranking applicants by their likelihood of loan approval.

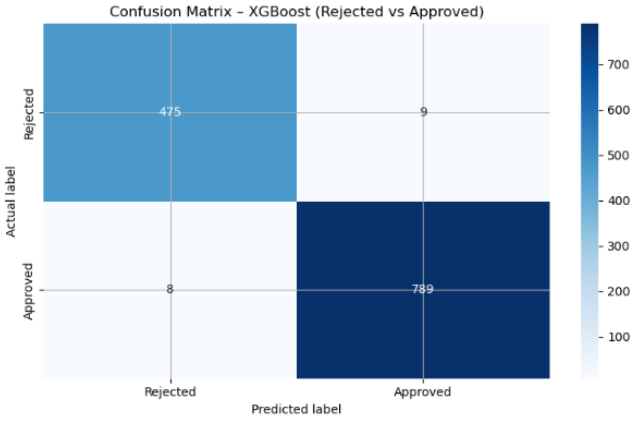

Fig. 8. Confusion Matrix: XGBoost (Rejected vs. Approved)

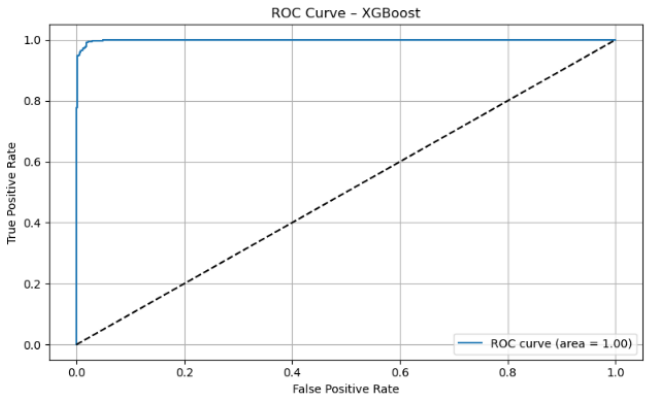

Fig. 9. ROC Curve: XGBoost

E. *Model Comparison*

To evaluate the overall performance of the three classifiers, both cross-validation metrics and held-out test-set results were examined. Table VI presents the stratified 5-fold cross-validation results for the three models, including the mean and standard deviation for accuracy, precision, recall, F1-score, and ROC-AUC.

TABLE VI. CROSS-VALIDATION PERFORMANCE OF LOGISTIC REGRESSION, RANDOM FOREST, AND XGBOOST

Model	Accuracy (mean ± std)	Precision (mean ± std)	Recall (mean ± std)	F1 (mean ± std)	ROC-AUC (mean ± std)
Logistic Regression	0.9110 ± 0.0069	0.9273 ± 0.0121	0.9301 ± 0.0144	0.9286 ± 0.0056	0.9671 ± 0.0057
Random Forest	0.9772 ± 0.0063	0.9797 ± 0.0058	0.9839 ± 0.0057	0.9817 ± 0.0051	0.9958 ± 0.0030
XGBoost	0.9803 ± 0.0055	0.9813 ± 0.0081	0.9871 ± 0.0058	0.9842 ± 0.0043	0.9974 ± 0.0013

The cross-validation results show that the ensemble models clearly outperform the linear baseline. Random Forest and XGBoost achieved higher average performance across all metrics, with XGBoost showing the strongest results overall. These findings indicate that the dataset contains nonlinear patterns and feature interactions that are better captured by tree-based methods than by Logistic Regression.

Table VII summarizes the performance of each classifier on the held-out test set. Logistic Regression produced strong results, with accuracy above 0.91 and a ROC-AUC of 0.973, demonstrating that it captures meaningful structure in the data. However, it also produced more misclassifications than the ensemble models.

Random Forest and XGBoost achieved nearly identical performance on the test set, both surpassing 0.98 in accuracy, precision, recall, and F1-score. Their ROC-AUC values were close to 1.00, indicating excellent separation between approved and rejected applications. XGBoost showed slightly higher recall and F1-score, suggesting a small advantage in correctly identifying approved applications while maintaining low error rates

TABLE VII. TEST SET PERFORMANCE OF LOGISTIC REGRESSION, RANDOM FOREST, AND XGBOOST

Model	Accuracy	Precision	Recall	F1	ROC_AUC
Logistic Regression	0.913	0.923	0.937	0.930	0.973
Random Forest	0.986	0.989	0.989	0.989	0.999
XGBoost	0.987	0.989	0.990	0.989	0.999

Figure 10 presents a combined ROC comparison plot for all three models. The ROC curves for Random Forest and XGBoost lie almost on top of each other and remain close to the top-left corner of the plot, illustrating their exceptional ranking ability. The ROC curve for Logistic Regression remains strong but clearly separates below the ensemble models, which is consistent with both the cross-validation and test-set results.

These comparisons demonstrate that ensemble methods are much better suited for this type of structured financial dataset. Their ability to model nonlinear interactions results in significantly improved predictive performance, while the linear baseline provides a useful reference point.

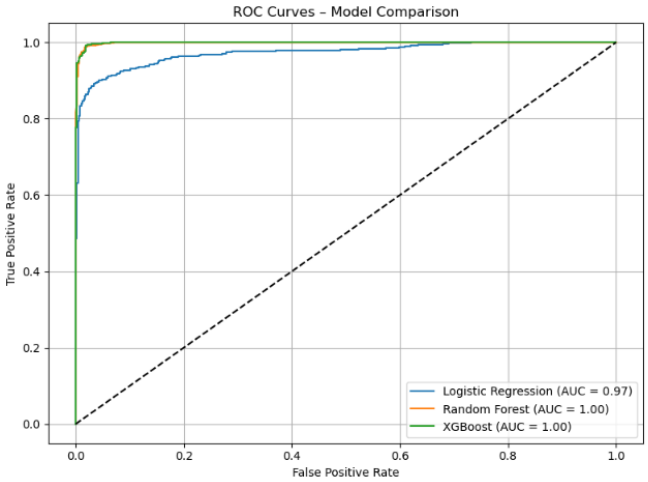

Fig. 10. ROC Curves for Logistic Regression, Random Forest, and XGBoost.

F. *Feature Importance (Random Forest & XGBoost)*

Feature importance was examined to understand which predictors contributed most to the decisions made by the ensemble models. As shown in **Figure 11**, Random Forest identified *cibil_score* as the most influential feature, which is consistent with lending practices where credit score is a major determinant of loan approval. The *loan_term* variable ranked second in Random Forest, indicating that the duration of the loan carries meaningful predictive value.

The XGBoost importance plot in **Figure 12** displays a similar pattern, with *cibil_score* dominating and *loan_term* again appearing as the second-most influential predictor. Financial variables such as *income_annum*, *loan_amount*, and the various asset value fields provided moderate contributions. In contrast, engineered features such as *education_graduate*, *self_employed_yes*, and the *residential_assets_value_was_negative* flag had relatively small influence on prediction outcomes.

Together, the importance trends shown in **Figure 11** and **Figure 12** help explain the strong performance of the ensemble approaches. Random Forest and XGBoost are able to capture nonlinear interactions and focus attention on the most relevant financial attributes, giving them a significant advantage over Logistic Regression

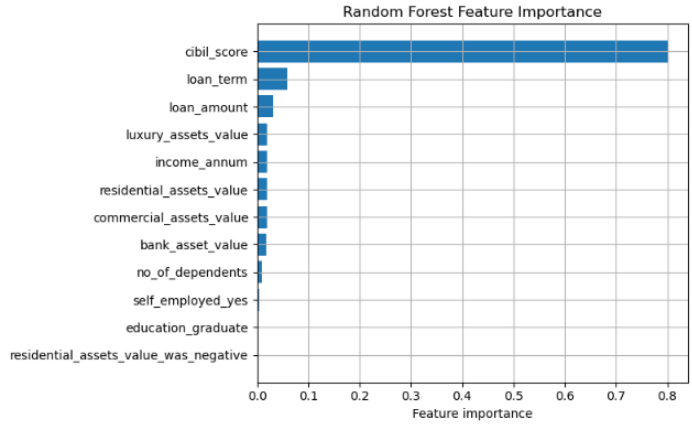

Fig. 11. Random Forest feature importance plot.

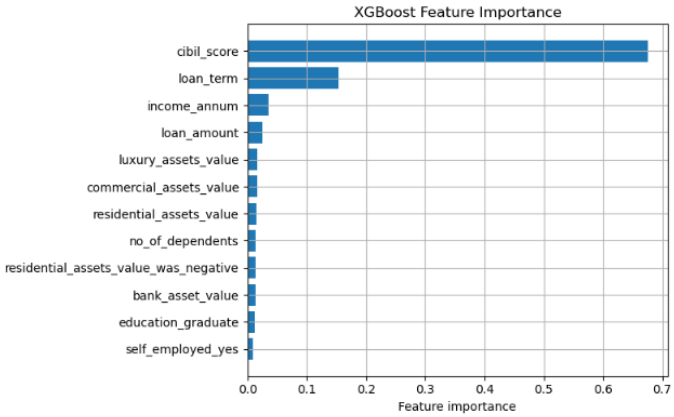

Fig. 12. XGBoost feature importance plot.

VII. Discussion

The results of this project show clear differences in performance between the Logistic Regression baseline and the two ensemble models. Logistic Regression produced solid performance with a test accuracy of approximately 0.91 and a ROC-AUC of about 0.97, indicating that several relationships in the dataset follow patterns that can be captured by a linear decision boundary. However, the confusion matrix and evaluation metrics show that Logistic Regression performed less effectively than the tree-based models in correctly classifying some loan outcomes, especially in cases where the underlying relationships may be nonlinear or involve interactions among features.

Random Forest and XGBoost achieved significantly higher performance across all evaluation measures. Both models reached accuracy levels above 0.98 and produced ROC-AUC values near 0.999, demonstrating almost perfect separation between approved and rejected applications. Their ability to model nonlinear structures, capture feature interactions, and adapt to variations in scale likely contributed to their strong predictive performance. XGBoost showed a slight advantage in recall and F1-score, indicating that it was particularly effective at correctly identifying approved applications while maintaining a low number of false predictions.

The dimensionality reduction experiment using PCA further supports these findings. Although PCA can simplify high-dimensional datasets, the first two principal components in this case captured only about 61 percent of the total variance. The scatter plot showed substantial overlap between the two classes, and Logistic Regression trained on PCA components performed far worse than the full-feature model, with accuracy dropping to 0.61 and ROC-AUC to 0.63. This indicates that important predictive information is lost when compressing the feature space. The PCA analysis reinforces that the dataset contains meaningful nonlinear patterns that are better captured by ensemble models than by linear methods or PCA-based dimensionality reduction.

Methods such as LDA and Kernel PCA were not applied because the dataset is not high dimensional, and the ensemble models already handle nonlinear feature relationships effectively, making additional dimensionality reduction unnecessary.

VIII. Conclusion

This project explored the use of machine learning techniques to predict loan approval decisions based on applicant financial and credit-related information. After preparing the dataset through feature engineering, log transformations, and scaling, three models were developed and evaluated using a consistent pipeline with stratified cross-validation. The comparison highlights the differences between a traditional linear model and more flexible ensemble approaches.

Overall, Random Forest and XGBoost outperformed Logistic Regression by a considerable margin. Logistic Regression achieved a test accuracy of roughly 0.91 and a ROC-AUC of about 0.97, while the ensemble models achieved accuracy above 0.98 and ROC-AUC values close to 0.999. These results indicate that nonlinear relationships and interactions among financial features play an important role in predicting loan approval outcomes. The PCA analysis also showed that reducing the feature space led to a significant loss in performance, confirming that the predictive information in this dataset is not easily captured by a low-dimensional representation.

The findings suggest that ensemble techniques are highly effective options for organizations seeking to improve the accuracy of credit-related decision systems. Future work could include integrating additional applicant information, exploring more advanced boosting methods, incorporating domain-specific financial risk metrics, or evaluating model fairness across demographic groups.

References

[1] J. Hand and W. Henley, “Statistical classification methods in consumer credit scoring,” Journal of the Royal Statistical Society: Series A (Statistics in Society), vol. 160, no. 3, pp. 523–541, 1997.

[2] S. Lessmann, B. Baesens, H. Seow, and L. Thomas, “Benchmarking state-of-the-art classification algorithms for credit scoring,” European Journal of Operational Research, vol. 247, no. 1, pp. 124–136, 2015.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[4] A. Sharma, “Loan approval prediction dataset,” Kaggle, 2021. [Online]. Available:https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset. Accessed: Oct. 30, 2025.