Rochester Institute of Technology

B. Thomas Golisano College of

Computing and Information Sciences

Master of Science in Information Technology and Analytics

Project Approval Form

Student Name: **Rohan Parkar**

Project Title: **Evaluating Machine Learning Algorithms to Build Web-Based Predictive Loan Approval System Based on Credit Score**

Project Committee

Name: Prof. Erik Golen

_____

Chair

Name: Prof. Zhiqiang Tao

_____

Committee Member

Table of Contents

Abstract

In the financial sector, the accuracy and efficiency of loan approval processes are critical for minimizing default risk and ensuring that only eligible applicants receive loans. This research project focuses on applying machine learning algorithms to predict loan approval based on credit scores and other relevant factors. The study employed a comprehensive methodology, including data collection, preprocessing, exploratory data analysis, model development, and evaluation. A detailed analysis and comparison of five widely used classification algorithms: Logistic Regression, Extreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP) Classifier, and Support Vector Machine (SVM) was conducted. The results demonstrate that Extreme Gradient Boosting outperforms other models' accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) score. A web-based application was developed to integrate the best-performing model for real-time loan approval predictions using HTML, CSS, JavaScript, and Flask. In the financial sector, the accuracy and efficiency of loan approval processes are critical for minimizing default risk and ensuring that eligible applicants receive loans. This project highlights the potential of machine learning algorithms in transforming traditional credit scoring methods, addressing limitations of subjective assessments, and improving the overall reliability and accuracy of loan approval systems.

Keywords: machine learning, loan approval, credit score, predictive modeling, web-based application.

1. Introduction

Loan approval is one of the most critical functions in the financial sector. It ensures that only the eligible gets the loan and, at the same time, minimizes the risk of default. Traditionally, it depended on the subjective judgment of the loan officer, who relied on a personal interview and professional experience to evaluate an applicant based on his creditworthiness. The Industrial Revolution in the 19th century increased the demand for capital, resulting in increased loan lending. The 20th century saw a dramatic shift with the introduction of credit cards and credit scores, revolutionizing the methods used for loan assessment. Previously, loan approval depended upon examination of income, employment, repayment history, and relation with the financial institutions. While these factors remain relevant, credit scoring has introduced a more systematic and streamlined approach to assessing creditworthiness.

Established between 1950 and 1960 by Bill Fair and Earl Isaac through Fair, Isaac, and Company (FICO), the credit scoring system has become a cornerstone of modern lending practices [6]. The FICO score ranges from 300 to 850, considering the borrower's payment history, outstanding debt, length of credit history, credit diversity, and recent credit inquiries. Enhanced technology and a general broadening of consumer data have already significantly refined credit scoring in assessing loan default risk, making it more effective and accurate across all lending.

The introduction of credit scoring was a massive upgrade in streamlining loan approval and having a systematic approach to moving away from subjective decisions. While equipping the already active loan approval systems with credit scores was a significant upgrade, there was still much scope for improvement. Credit scores indicate only the borrower's creditworthiness. They cannot capture the complexity and nuance of each case. Machine learning algorithms can make the loan approval system more elaborate and trustworthy in distinguishing high-risk and low-risk applicants.

The project primarily focused on evaluating and comparing the five widely used classification machine learning algorithms for loan approval prediction based on their performance. These involve K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Extreme Gradient Boosting, and Multi-Layer Perceptron. Based on the performance of the models, the best-performing algorithm was determined by analyzing the confusion matrix and the ROC AUC curve. A web-based application was developed using the developed model to ease loan approval. The developed application makes the process easy and automatic for both the applicant and the financial institution through its user-friendly interface.

The implemented project tries to improve loan approval by integrating advanced machine learning algorithms, moving beyond conventional and credit score-based methods, and addressing limitations and biases in subjective assessments. The optimized algorithm, integrated into the web application, provides a convenient way to systemize this process for applicants and financial institutions. Although this project encounters data quality and

generalizability limitations, it has prepared a sound foundation for further research and applications into credit risk assessment, making the loan approval system more accurate and reliable.

This report is organized into seven chapters to give a detailed study overview. The second chapter, Related Work, presents related work in the domain of loan approval processes and credit scoring systems and discusses the application domains of machine learning in finance. The third chapter, Methodology, elaborates on the research design, including data collection and preprocessing, implementation, and evaluation of machine learning algorithms. The fourth chapter is about the results, which show the findings of the analysis. It comprises performance comparisons of different models using accuracy, ROC AUC, and other scores. The fifth chapter, Future Works, presents some limitations of this study and indicates future research potentials in the area. The next chapter, Conclusion, summarizes the main findings and their implications, reflecting on the effectiveness of the chosen model. The final chapter seven is about References and lists all sources consulted during the research, thus avoiding plagiarism and guiding further readings. This structure will ensure that the research procedure and its results are well-explained and logical.

2. Related Work

The study by Prasanth et al., "Intelligent Loan Eligibility and Approval System based on Machine Learning using Random Forest Algorithm," aims to improve the loan approval process by adopting the Random Forest Algorithm [14]. The authors demonstrated that machine learning can effectively predict loan repayment likelihood by analyzing customer data. Although the research performs well in loan approval predictions, it has some drawbacks. First, there is a potential bias in the training data. Second, the data contains limited relevant economic factors. The presented project handled these limitations by using a broader data set containing a variety of economic indicators and implementing methods like balanced class weights to reduce biases.

In the research "Loan Approval Prediction using Machine Learning: A Comparative Analysis of Classification Algorithms," Saini et al. evaluated various machine learning algorithms for loan approval prediction [11]. It focuses on Random Forest Classifier, K-Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression. The Random Forest Classifier had an accuracy of 98.04%, which was the best-performing model. The study also mentioned some drawbacks, like overfitting due to a Random Forest model and the comparatively poor performance of the Support Vector Classifier. All these limitations were addressed in the presented project by including adequate data preprocessing steps, such as balancing the dataset against overfitting and hyperparameter tuning across all models to boost their performance.

In the work of Orji et al., "Machine Learning Models for Predicting Bank Loan Eligibility," a good number of machine learning algorithms like Random Forest, Gradient Boost, Decision Tree, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression have been strictly put into consideration for predicting loan eligibility [15]. The result portrayed the efficiency of these models in showing high precision. Random Forest was the best-performing model, with an accuracy of 95.55%. The research focuses on sophisticated preprocessing techniques like SMOTE and extensive EDA in handling class imbalance and model performance improvement [9]. These were applied in the project, improving the models' predictive accuracy and robustness.

The paper "Customer Loan Eligibility Prediction using Machine Learning Algorithms in Banking Sector" by Ch. Naveen Kumar et al. carried out various machine learning techniques for predicting loan approval decisions [16]. Customer data was collected from different banks, and the following machine learning algorithms were implemented: Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor, and Decision Tree with AdaBoost. The results proved that Decision Tree with AdaBoost had the highest accuracy. However, some overfitting issues were encountered due to simple feature selection methods and limited feature engineering. All these shortcomings have been identified and resolved in the presented project by incorporating feature selection techniques. It used the F-value as a scoring method to remove irrelevant features and employed advanced ensemble methods for training and testing datasets, including Extreme Gradient Boosting.

In the research of Sheikh et al., "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," logistic regression was used to predict loan approval [10]. The study showed that logistic regression could correctly classify the defaulter and reliable borrowers. However, the research had limitations, such as a smaller dataset and handling missing values. Also, logistic regression was the only model used for the study that may not capture complex patterns. The presented project alleviates these drawbacks by using a comprehensive dataset, handling missing data using pre-processing techniques, and comparing different classification algorithms to improve the chance of building a model with high predictive accuracy and robustness.

In the research paper "Prediction of the Approval of Bank Loans Using Various Machine Learning Algorithms," Rahman et al. review various machine learning methods in loan approval prediction [12]. The models were built using data from a bank in Dhaka, Bangladesh, which contained 1000 records and 16 features. It used algorithms such as Decision Tree, Gradient Boosting, Random Forest, Naive Bayes, Support Vector Machine, and Logistic Regression. According to the results, the Random Forest algorithm had the best predictability for loan approvals. The significant drawbacks of the study were a lack of feature engineering methods and a comparatively smaller dataset. Additional features, such as the applicant's credit score and financial details, were considered to address these issues, and the models were trained on a broader dataset in the presented project.

Dr. R. Priscilla et al. experimented with different machine learning models to increase the accuracy and reliability of the prediction of loan approval in their paper "Baseline Modeling for Early Prediction of Loan Approval System" [13]. The research used data preprocessing

methods and assessed various classification algorithms, such as gradient boosting, logistic regression, and random forest classifier, through K-Fold cross-validation to identify the most effective model. The results showed that models efficiently recognized qualified loan candidates, reduced the possibility of loan defaults, and offered several advantages to financial institutions and customers.

3. Methodology

3.1 Data Collection

The first step in implementing this project involved gathering a comprehensive dataset to understand the loan approval process. The dataset [3] used in this project is sourced from Kaggle and includes significant features for assessing loan applications. The features include applicant details such as the term of the loan, months of employment experience, homeownership status, number of open credit lines, total inquiries, available bankcard credit, debt-to-income ratio, income verifiability, stated monthly income, loan number, loan amount, monthly installment, interest rate, employment status, average credit score, delinquencies, and the loan outcome (good/bad loan). The dataset consists of both numerical and categorical data. This dataset from Kaggle comprises 16 predictors and one target variable with 55106 data points.

3.2 Data Preprocessing

Preprocessing the dataset is essential to ensure its quality and suitability for the machine learning algorithms chosen for implementation. The following steps were performed to get the data into the necessary shape.

3.2.1 Handling Missing Values

The missing values were treated to maintain the dataset's integrity and usability. Numerical features with missing values were replaced using the mean, while categorical features were dropped. This ensured a complete dataset, reducing the risk of introducing bias or errors due to missing values.

3.2.2 Cleaning Columns in the Dataset

Several columns required cleaning and transformation to ensure consistency for machine learning models. The following transformations were applied:

i. **Employment Status:** The 'EmploymentStatus' column was transformed into a binary and renamed 'IsEmployed'. Employment statuses such as 'Not employed' or 'Retired' were mapped to 0, indicating no employment, while all other employment statuses were mapped to 1. This binary classification simplifies the feature, making processing easier for algorithms. The original 'EmploymentStatus' column was then dropped.

ii. **Homeownership Status:** The column 'IsBorrowerHomeowner' was renamed 'IsHomeowner', and its data type was converted to an integer.

iii. **Average Credit Score:** The 'AverageCreditScore' column was created by taking an average of the 'CreditScoreRangeLower' and 'CreditScoreRangeUpper' columns, rounding up to the nearest integer. The original credit score range columns were then dropped.

iv. **Current Delinquencies:** The 'CurrentDelinquencies' column was converted into a binary 'AnyDelinquencies' column, indicating 1 if there were any delinquencies and 0 if there were none. The original 'CurrentDelinquencies' column was then dropped.

v. **Income Verifiability:** To maintain consistency across the dataset, the 'IncomeVerifiable' column's data type was converted to an integer.

vi. **Total Inquiries:** The 'TotalInquiries' column's data type was converted to an integer for consistency.

vii. **Loan Status:** The 'LoanStatus' column contains multiple categories indicating a loan's status. For the purposes of this project, loans with a status of 'Current' were excluded as they do not provide any information about loan outcomes. A new binary column, 'GoodLoan,' was created to indicate successful completion ('Completed' or 'FinalPaymentInProgress') with 1 and the bad loans with 0. The original 'LoanStatus' column was then dropped.

3.2.3 Renaming Columns

Several columns were renamed for clarity and consistency, making them more intuitive for analysis.

3.2.4 Dropping Unnecessary Columns

The 'LoanNumber' column was dropped because it was just an identifier and was irrelevant to predictive modeling.

3.2.5 Categorical and Numerical Feature Separation

The features were categorized into categorical and numerical groups to apply appropriate processing techniques.

3.2.6 Handling Outliers

Outliers in the numerical features were addressed using the Interquartile Range (IQR) method. Values beyond 1.5 times the IQR from the first (Q1) and third quartiles (Q3) were identified as outliers and replaced with the median value of the respective feature. This step was crucial to reduce the skewness and ensure the robustness of the model.

3.2.7 Standardizing Numerical Features

Numerical features were standardized using feature scaling from Scikit-learn to ensure they were on a similar scale. This step was necessary to improve the performance of distance-based algorithms like KNN and ensure that features contributed equally to the model.

The preprocessing steps ensured the dataset was clean, consistent, and appropriately scaled for input into machine learning models. Handling missing values, transforming and renaming columns, addressing outliers, and standardizing numerical features were essential to preparing the dataset for model training and evaluation.

3.3 Data Exploration

Data exploration is a crucial phase in any data science project. It involves understanding the structure, content, and relationships within the dataset. This section provides a detailed examination of the dataset used in this project, focusing on key variables, their distributions, and their correlations with the target variable, GoodLoan.

3.3.1 Overview of the Dataset

The dataset consists of 55,106 entries and 16 columns. The columns encompass a mix of numerical and categorical variables, each contributing unique information relevant to the loan approval process. Below is a summary of the dataset:

Feature	Type	Description
Term	Numerical	Loan term in months.
MonthsOfEmployementExperience	Numerical	Number of months of employment experience.
IsHomeowner	Categorical	Whether the applicant is a homeowner. (1: Yes, 0: No)
OpenCreditLines	Numerical	Number of open credit lines.
TotalInquiries	Numerical	Total number of credit inquiries.
AvailableBankcardCredit	Numerical	Available bankcard credit.
DebtToIncomeRatio	Numerical	Debt to income ratio.
IncomeVerifiable	Categorical	Whether the income is verifiable (1: Yes, 0: No).
StatedMonthlyIncome	Numerical	Stated monthly income of the applicant.
LoanNumber	Numerical	Unique identifier for each loan.
LoanAmount	Numerical	Amount of the loan.
MonthlyInstallment	Numerical	Monthly installment amount.
InterestRate	Numerical	Interest rate of the loan.
IsEmployed	Categorical	Whether the applicant is currently employed. (1: Yes, 0: No)
AverageCreditScore	Numerical	Average credit score of the applicant.
AnyDelinquencies	Categorical	Whether there are any delinquencies. (1: Yes, 0: No)
GoodLoan	Categorical	Whether the loan is considered good. (1: Good, 0: Bad)

Table 1. List of Features

The target variable, GoodLoan, is a binary classification indicating whether a loan is considered good (1) or not (0). This binary classification is essential for the predictive modeling of loan approvals.

3.3.2 Key Variables and Their Relationships with GoodLoan

During the exploration phase, several variables showed notable correlations with GoodLoan. Understanding these relationships helps in selecting and engineering features for machine learning models.

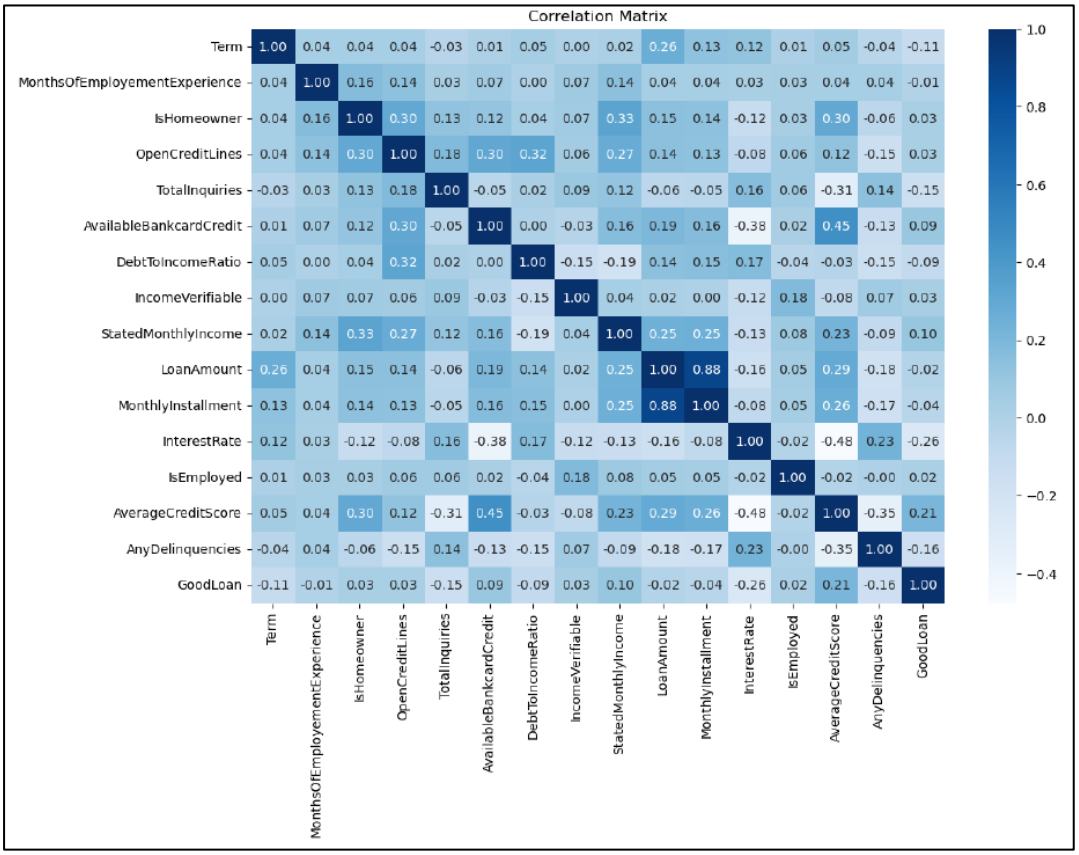

Fig 1. Correlation Matrix

i. **InterestRate:**
   - Correlation with GoodLoan: -0.26
   - Insight: Lower interest rates are associated with a higher likelihood of a good loan.

ii. **AverageCreditScore:**
   - Correlation with GoodLoan: 0.21
   - Insight: Higher credit scores positively correlate with the likelihood of a good loan. This means a higher credit score is associated with an increased likelihood of a good loan.

iii. **AnyDelinquencies:**
   - Correlation with GoodLoan: -0.16
   - Insight: The presence of delinquencies is negatively correlated with GoodLoan. Fewer delinquencies increase the likelihood of a loan being classified as good.

iv. **TotalInquiries:**
   - Correlation with GoodLoan: -0.15

- Insight: Fewer credit inquiries are weakly associated with a higher likelihood of a good loan. This relationship implies that fewer inquiries may reflect more cautious borrowing behavior.

v. **Term:**
   - Correlation with GoodLoan: -0.11
   - Insight: Shorter loan terms are slightly more likely to be good loans. This weak negative correlation suggests that shorter repayment periods could reduce the risk of default.

vi. **StatedMonthlyIncome:**
   - Correlation with GoodLoan: 0.10
   - Insight: Higher stated monthly incomes are weakly positively correlated with good loans. Borrowers with higher incomes might be better positioned to manage loan repayments.

vii. **DebtToIncomeRatio:**
   - Correlation with GoodLoan: -0.09
   - Insight: A lower debt-to-income ratio is associated with a higher likelihood of a good loan. This weak negative correlation indicates that borrowers with lower relative debt burdens are more likely to repay their loans successfully.

viii. **AvailableBankcardCredit:**
   - Correlation with GoodLoan: 0.09
   - Insight: Higher available bankcard credit limits are weakly positively correlated with good loans. This suggests that borrowers with more available credit are less likely to default.

3.4 Implementation of Machine Learning Algorithms

The dataset was split into training, validation, and testing sets. Initially, the data was divided into a training and testing set with a 70-30 split. Then, the training set was divided into a training and validation set with a 75-25 split. A method to select the most relevant features was applied with the ANOVA F-value as the scoring function, and only the selected 12 features were used in training the model [4].

Each algorithm was applied forward stepwise selection to find the most predictive feature subset using the validation set. Logistic Regression was the first algorithm implemented, with hyperparameter tuning conducted through grid search, optimizing for the regularization parameter C. The model demonstrated a balanced performance with the selected features, yielding cross-validation and testing accuracies thoroughly evaluated through precision, recall, F1 score, and ROC-AUC metrics.

Next, the Extreme Gradient Boosting (XGBoost) model was implemented. The class weights were assigned using the scale_pos_weight parameter to handle class imbalances for extreme gradient boost and logistic regression models [9]. Feature selection and hyperparameter tuning were again employed, focusing on the number of estimators, learning rate, and tree depth. The XGBoost model exhibited strong predictive performance, especially in handling complex data relationships.

The Random Under Sampler addressed the class imbalance for the remaining models [9]. The optimal parameters for KNN, including the number of neighbors, distance metric, and weighting scheme, were determined via grid search cross-validation. The model was assessed for its precision, recall, F1 score, and ROC-AUC, highlighting its capability in classification tasks.

The Multilayer Perceptron (MLP) classifier, a type of neural network, was also trained and tested. Feature selection was performed through forward stepwise selection and hyperparameter tuning for the neural network architecture, activation functions, and learning rate schedule. The MLP demonstrated good generalization capabilities, with high validation accuracy and a solid ROC-AUC score.

Finally, the Support Vector Machine (SVM) model was implemented. Using RandomizedSearchCV, parameters such as the regularization parameter, kernel type, and gamma values were optimized. The SVM provided a competitive performance with well-balanced metrics across the models.

Each algorithm's performance was rigorously evaluated using cross-validation scores, training and testing accuracies, and detailed classification reports. Confusion matrices and ROC curves were plotted to interpret the results visually, ensuring a comprehensive analysis of each model's effectiveness in the predictive task.

3.5 Handling Challenges: Class Imbalance

While developing the machine learning models, a significant challenge related to class imbalance in the dataset was encountered. The dataset was heavily skewed towards the majority class, with 70% of the data belonging to the "loan approved" class and only 30% to the "loan rejected" class. This imbalance posed a risk of the models being biased towards the majority class, potentially leading to poor predictive performance for the minority class.

The following methods [7] [9] were used to counter this challenge:

i. **Class Weights Adjustment:** For algorithms like Logistic Regression and Extreme Gradient Boosting (XGBoost), the class weights were adjusted to penalize misclassifications of the minority class heavily. This technique helped the models to pay more attention to the minority class, improving their performance for the minority classes.

ii. **Under-sampling:** For the remaining algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Multilayer Perceptron (MLP), the majority class was under-sampled to create a more balanced training set. This method reduced the size of the majority class to match the minority class, thereby mitigating the bias towards the majority class.

iii. **Evaluation Metrics:** The metrics sensitive to class imbalance, such as Precision, Recall, F1-Score, and ROC-AUC Score, were focused rather than relying only on accuracy. This allowed a better understanding of the model's performance on the majority and minority classes.

These techniques significantly improved the model performance for the minority class. Initially, the accuracy achieved was high, but the model did a poor job handling the minor classes. Introducing the class imbalance handling methods reduced the overall accuracy of the models but provided a balanced performance for majority and minority classes. The recall for the "loan rejected" class improved across all models, with XGBoost achieving a balanced performance with a f1-score of 0.69 and an overall accuracy of 64.09%. This challenge highlighted the importance of addressing the class imbalance in predictive modeling. It reinforced the value of combining data resampling, algorithmic adjustments, and appropriate evaluation metrics to achieve robust model performance.

3.6 Evaluation of Algorithm Performance

The accuracies, F1 scores, and ROC curves were compared to evaluate the performance of the machine learning algorithms implemented. The models tested included Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Extreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP). The analysis included training, validation, testing accuracies, precision, recall, and evaluation of AUC-ROC metrics [2].

i. Model Accuracy Comparison

Each model's accuracy was evaluated on the training, validation, and test datasets. Logistic Regression, SVM, KNN, XGBoost, and MLP each demonstrated varying degrees of performance:

- **Training Accuracy:** This metric measures how well each model fits the training data. High training accuracy generally indicates that the model has learned the training data well but could also suggest overfitting if there's a significant drop in validation accuracy.

- **Validation Accuracy:** This measures the model's performance on unseen data used during training for hyperparameter tuning. It helps to understand the model's ability to generalize to new data.

- **Test Accuracy:** This metric assesses the final model's performance on completely unseen data, providing an unbiased evaluation of how well the model will perform in real-world scenarios.

ii. F1 Scores

The F1 score is a harmonic mean of precision and recall, which can used to assess the balance between the two metrics. It is beneficial for evaluating models on imbalanced datasets. Each model's F1 score was calculated, showing their effectiveness in correctly predicting positive and negative classes.

iii. ROC Curves

The ROC (Receiver Operating Characteristic) curve is a graphical representation of a model's ability to distinguish between classes. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The Area Under the Curve (AUC) provides a single scalar value to summarize the model's performance across all thresholds. Higher AUC values indicate better model performance.

3.7 Development of a Web-Based Application

In the final development phase, the best-performing model, XGBoost, was integrated into a web application. This integration aimed to create a seamless and efficient user experience for customers applying for loans. The web application was built using HTML, CSS, and JavaScript for the front end to create a visually appealing and user-friendly interface. The Flask framework was used on the back end due to its simplicity, flexibility, and compatibility with Python when integrating the machine learning model [1] [5].

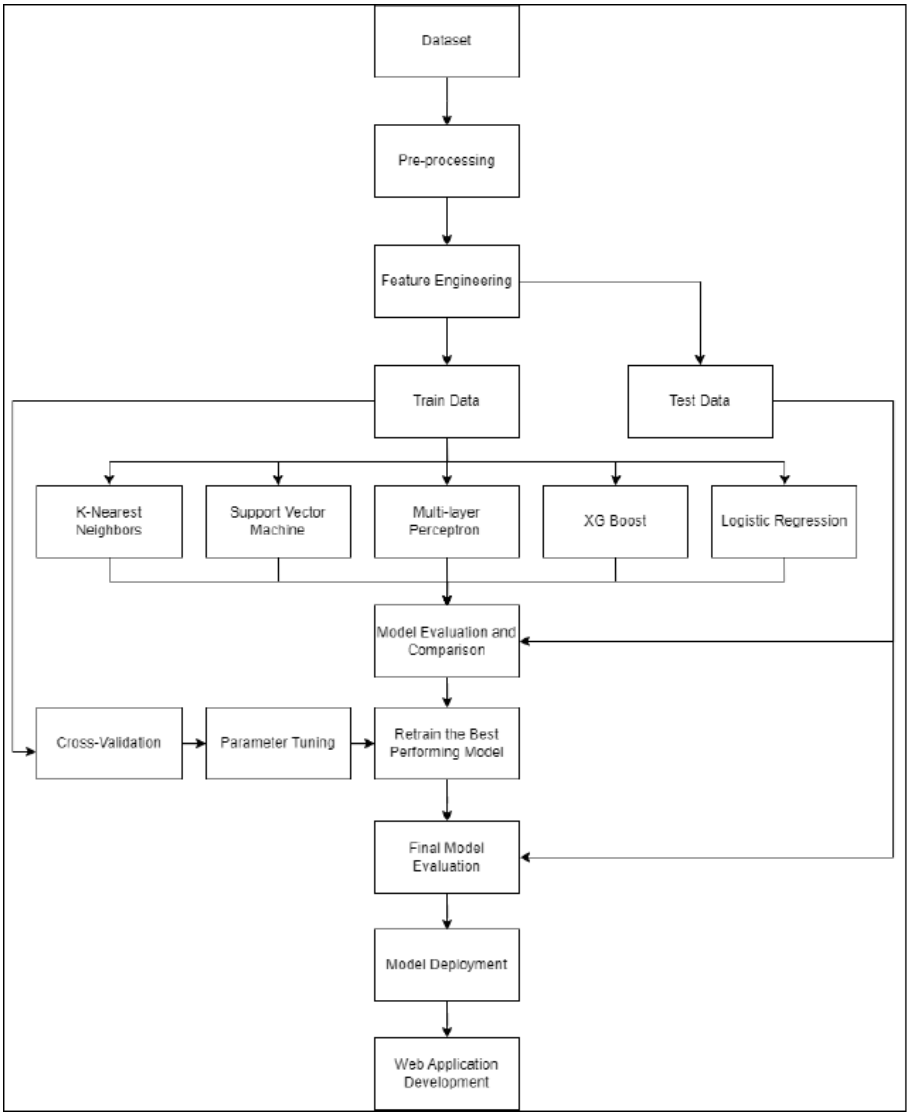

Fig 2. Project Workflow

4. Results

4.1 Model Iterations and Comparisons

This section highlights the iterative process of developing models, demonstrating how methods were used to enhance model performance, including feature selection, addressing class imbalance, and tuning parameters.

4.1.1 Iteration 1: Base Model Performance

The base models were trained with default settings without addressing class imbalance or optimizing parameters. These are utilized as a baseline to measure the improvements attained in subsequent iterations.

	Model	Train Accuracy	Validation Accuracy	Test Accuracy	ROC-AUC Score	F1-Score
0	Logistic Regression	0.686692	0.691414	0.686910	0.585068	0.653066
1	Support Vector Machine	0.667819	0.670780	0.668461	0.500000	0.535632
2	K-Nearest Neighbour	0.770135	0.661655	0.660477	0.588286	0.647727
3	Extreme Gradient Boosting	0.822503	0.701058	0.701549	0.622455	0.683769
4	Multilayer Perceptron	0.702212	0.691000	0.689693	0.632485	0.683292

Fig 3. Metrics Overview: Base Model

Discussion: The base models, particularly Extreme Gradient Boosting, showed strong accuracy. However, the ROC-AUC and F1 scores indicated that the models struggled with classifying the minority class correctly. This highlighted the need to address class imbalance and optimize features and tune parameters to improve the model's ability to generalize.

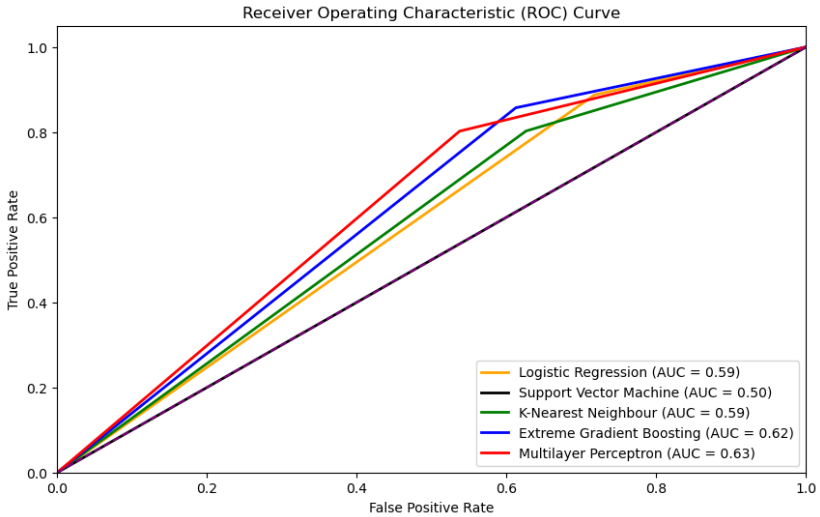

Fig 4. ROC Curve: Base Model

4.1.2 Iteration 2: Applying Class Imbalance Measures

Techniques like class weighting and under sampling were applied to handle class imbalance, aiming to improve the model's sensitivity to the minority class.

	Model	Train Accuracy	Validation Accuracy	Test Accuracy	ROC-AUC Score	F1-Score
0	Logistic Regression	0.575838	0.575487	0.577486	0.641112	0.581043
1	Support Vector Machine	0.644438	0.624989	0.609418	0.640421	0.619262
2	K-Nearest Neighbour	0.747773	0.605390	0.604609	0.606686	0.615418
3	Extreme Gradient Boosting	0.795817	0.665388	0.658541	0.655485	0.667411
4	Multilayer Perceptron	0.667677	0.624354	0.616585	0.651983	0.625839

Fig 5. Metrics Overview: Applying Class Imbalance Measures

Discussion: After addressing the class imbalance, there was a noticeable improvement in the ROC-AUC and F1 scores across most models, indicating better handling of the minority class. The Logistic Regression model, for example, showed an improved ROC-AUC score from 0.585068 to 0.641112. While these measures led to a drop in overall accuracy, this trade-off was expected and acceptable, given the focus on improving the model's ability to classify both classes correctly. The improvement in F1 scores suggests a better balance between precision and recall, especially in the minority class.

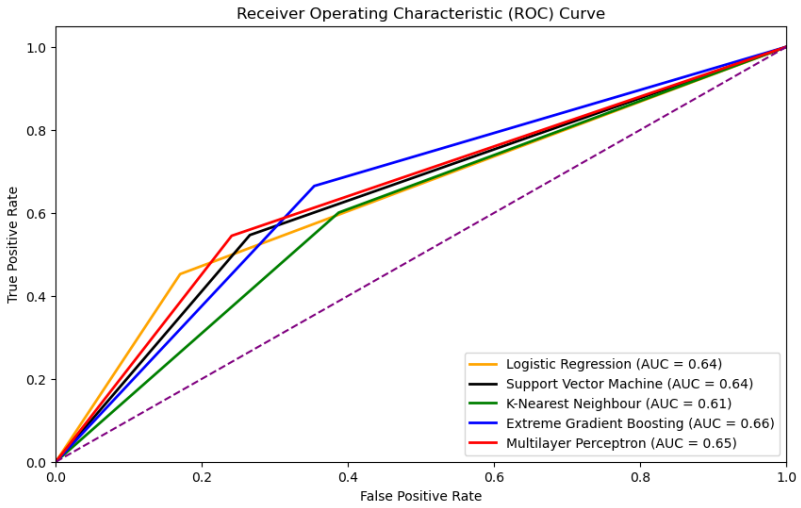

Fig 6. ROC Curve: Applying Class Imbalance Measures

4.1.3 Iteration 3: Feature Selection and Parameter Tuning with Grid Search CV

Feature selection was applied to remove irrelevant or redundant features, followed by parameter tuning using Grid Search CV to find the optimal hyperparameters for each model.

	Model	Train Accuracy	Validation Accuracy	Test Accuracy	ROC-AUC Score	F1-Score
0	Logistic Regression	0.575216	0.574663	0.577426	0.640791	0.581063
1	Support Vector Machine	0.663406	0.650363	0.619942	0.645385	0.630119
2	K-Nearest Neighbour	0.692018	0.626665	0.613047	0.633693	0.623644
3	Extreme Gradient Boosting	0.652886	0.639993	0.640878	0.663424	0.651274
4	Multilayer Perceptron	0.662579	0.649674	0.633370	0.646228	0.643601

Fig 7. Metrics Overview: After Feature Selection and Parameter Tuning

Discussion: The application of feature selection and parameter tuning led to more stable performance metrics, with a slight increase in F1 scores for most models, indicating that the models became more robust in handling the data. For example, the Multilayer Perceptron model improved F1-score from 0.625839 to 0.643601. Feature selection helped reduce model complexity, potentially lowering the risk of overfitting, while parameter tuning optimized the models' predictive performance. The ROC-AUC scores also reflect a more balanced model performance, suggesting that the models are better at distinguishing between classes.

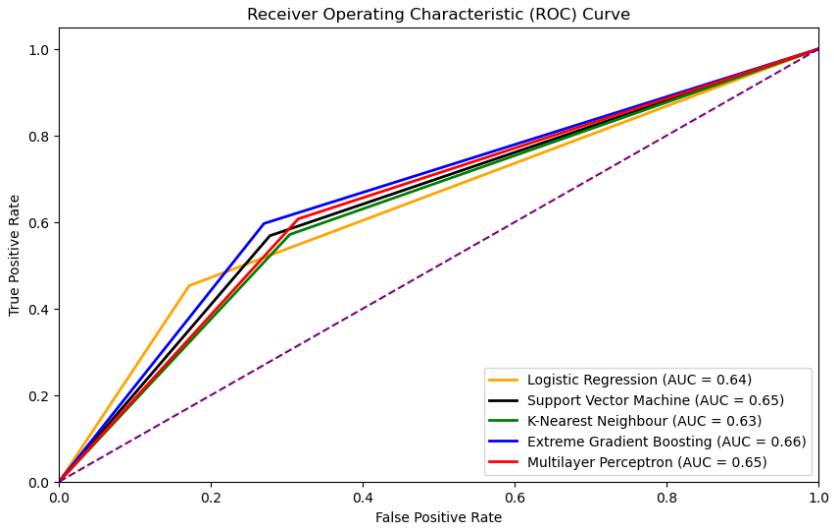

Fig 8. ROC Curve: Feature Selection and Parameter Tuning with Grid Search CV

4.2 Final Model Results

In this project, multiple machine learning models were developed to predict loan approval decisions accurately. The models included Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Extreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP) Classifier. Below are the detailed results for each model based on performance metrics such as accuracy, ROC-AUC score, and F1- Score.

4.2.1 Logistic Regression

Accuracy: 57.74%

ROC-AUC Score: 0.6408

F1-Score: 0.5811

Classification Report: The model showed a precision of 0.7051, a recall of 0.5774, and a weighted average F1 score of 0.58. It performed better in classifying the majority class (class 0) with a recall of 0.83 but struggled with the minority class (class 1) with a recall of 0.45.

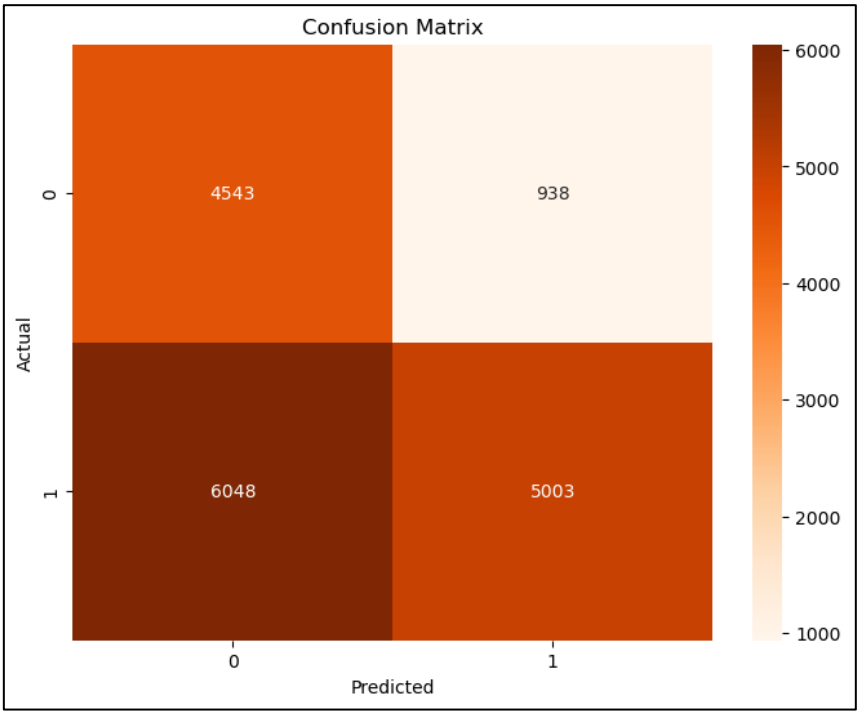

Fig 9. Logistic Regression: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.43      0.83      0.57      5481
           1       0.84      0.45      0.59     11051

    accuracy                           0.58     16532
   macro avg       0.64      0.64      0.58     16532
weighted avg       0.71      0.58      0.58     16532

Precision: 0.7051338751472839
Recall: 0.577425598838616
F1 Score: 0.5810628014428235

The accuracy of the Logistic Regression Model is: 57.74 % .

The ROC-AUC Score of the Logistic Regression Model is: 0.6407912785179315 .
```

Fig 10. Logistic Regression: Classification Report

4.2.2 Extreme Gradient Boosting (XGBoost)

Accuracy: 64.09%

ROC-AUC Score: 0.6634

F1-Score: 0.6513.

Classification Report: XGBoost had a precision of 0.7029, a recall of 0.6409, and a weighted average F1 score of 0.65. The model provided balanced performance across both classes, with a recall of 0.73 for class 0 and 0.60 for class 1.

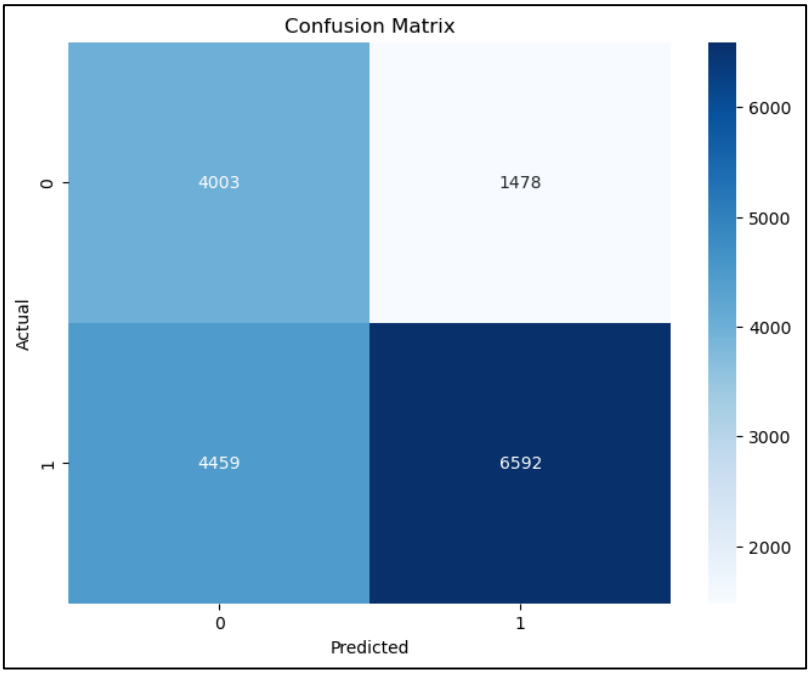

Fig 11. Extreme Gradient Boosting: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.73      0.57      5481
           1       0.82      0.60      0.69     11051

    accuracy                           0.64     16532
   macro avg       0.64      0.66      0.63     16532
weighted avg       0.70      0.64      0.65     16532

Precision: 0.7028706412690978
Recall: 0.6408782966368256
F1 Score: 0.6512743583745236

The accuracy of the Extreme Gradient Boosting Model is: 64.09 % .

The ROC-AUC Score of the Extreme Gradient Boosting (XGBoost) is: 0.6634241410232973 .
```

Fig 12. Extreme Gradient Boosting: Classification Report

4.2.2 K-Nearest Neighbour (KNN)

Accuracy: 61.3%

ROC-AUC Score: 0.6337

F1-Score: 0.6236

Classification Report: The KNN model showed a precision of 0.6751, a recall of 0.6130, and a weighted average score of 0.62. It had a recall of 0.70 for class 0 and 0.57 for class 1, indicating a reasonable balance in performance.

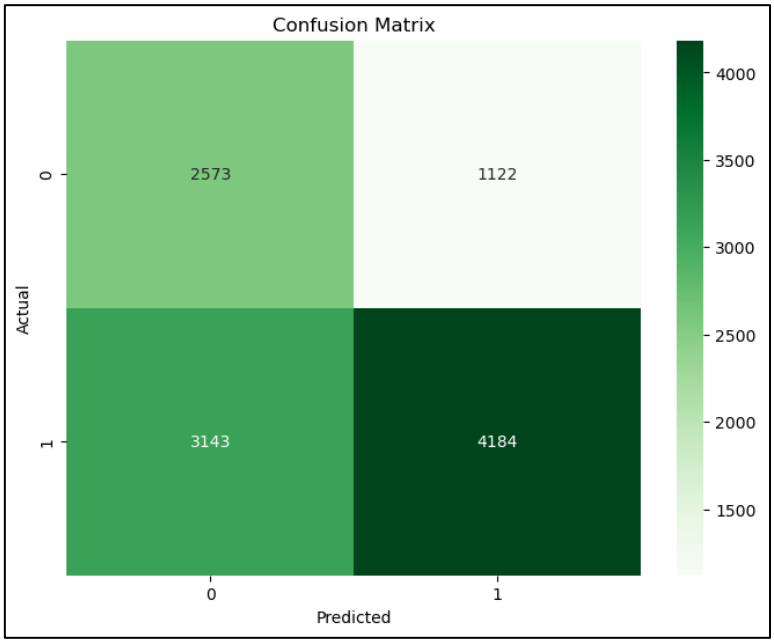

Fig 13. K-Nearest Neighbour: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.70      0.55      3695
           1       0.79      0.57      0.66      7327

    accuracy                           0.61     11022
   macro avg       0.62      0.63      0.60     11022
weighted avg       0.68      0.61      0.62     11022

Precision: 0.6750960859798549
Recall: 0.6130466340047178
F1 Score: 0.6236435188070928

The accuracy of the K - Nearest Neighbour Model is: 61.3 % .

The ROC-AUC Score of the K - Nearest Neighbour Model is: 0.6336925191697419 .
```

Fig 14. K-Nearest Neighbour: Classification Report

4.2.3 Multilayer Perceptron (MLP) Classifier

Accuracy: 63.34%

ROC-AUC Score: 0.6462

F1-Score: 0.6436

Classification Report: The MLP Classifier had a precision of 0.6839, a recall of 0.6334, and a weighted average F1 score of 0.64. It performed well in classifying both classes, with a recall of 0.69 for class 0 and 0.61 for class 1.

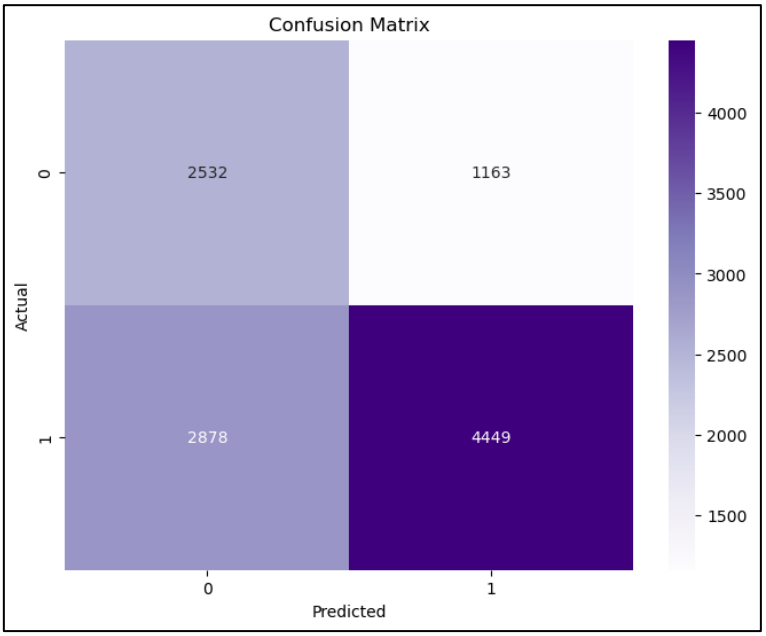

Fig 15. Multilayer Perceptron: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.69      0.56      3695
           1       0.79      0.61      0.69      7327

    accuracy                           0.63     11022
   macro avg       0.63      0.65      0.62     11022
weighted avg       0.68      0.63      0.64     11022

Precision: 0.6838990016433403
Recall: 0.6333696243875885
F1 Score: 0.6436009997482628

The accuracy of the Multilayer Perceptron (MLP) Classifier Model is: 63.34 % .

The ROC-AUC Score of the Multilayer Perceptron (MLP) Classifier Model is: 0.6462282809258506 .
```

Fig 16. Multilayer Perceptron: Classification Report

4.2.4 Support Vector Machine (SVM)

Accuracy: 61.99%

ROC-AUC Score: 0.6454

F1-Score: 0.6301

Classification Report: The SVM model showed a precision of 0.6868, a recall of 0.6199, and a weighted average score of 0.63. It had a recall of 0.72 for class 0 and 0.57 for class 1, demonstrating a good balance between the two classes.

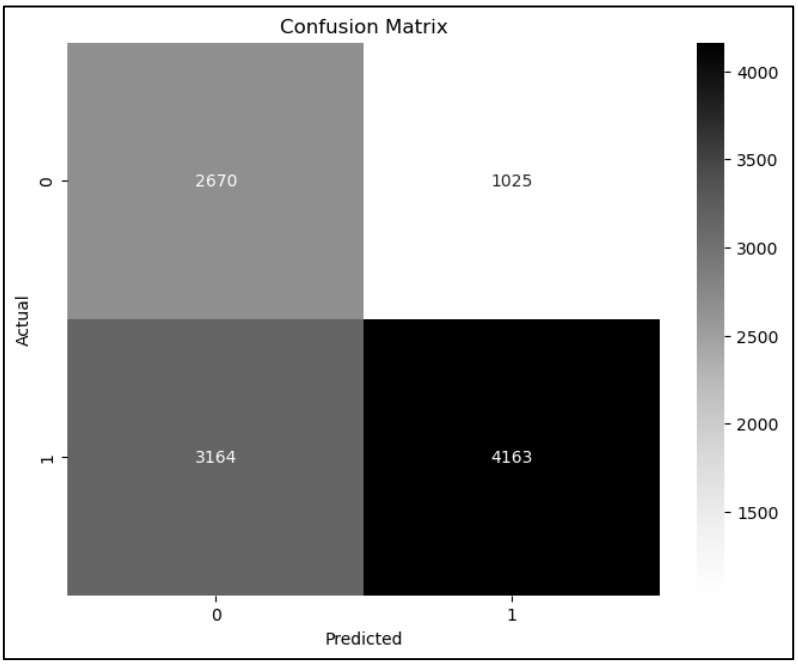

Fig 17. Support Vector Machine: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.46      0.72      0.56      3695
           1       0.80      0.57      0.67      7327

    accuracy                           0.62     11022
   macro avg       0.63      0.65      0.61     11022
weighted avg       0.69      0.62      0.63     11022

Precision: 0.6868495709927309
Recall: 0.6199419343131918
F1 Score: 0.6301194637007504

The accuracy of the Support Vector Machine Model is: 61.99 % .

The ROC-AUC Score of the Support Vector Machine Model is: 0.6453853090862888 .
```

Fig 18. Support Vector Machine: Classification Report

4.3 Comparison Summary for Final Models.

Best Performance: The Extreme Gradient Boosting (XGBoost) model outperformed the other models with the highest ROC-AUC score of 0.6634 and accuracy of 64.09%.

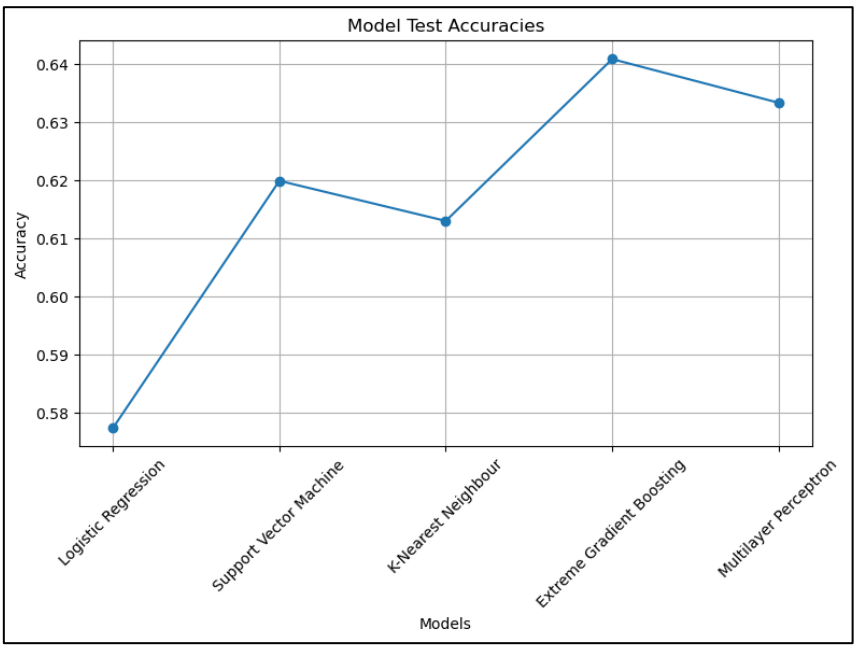

Fig 19. Final Models: Test Accuracies

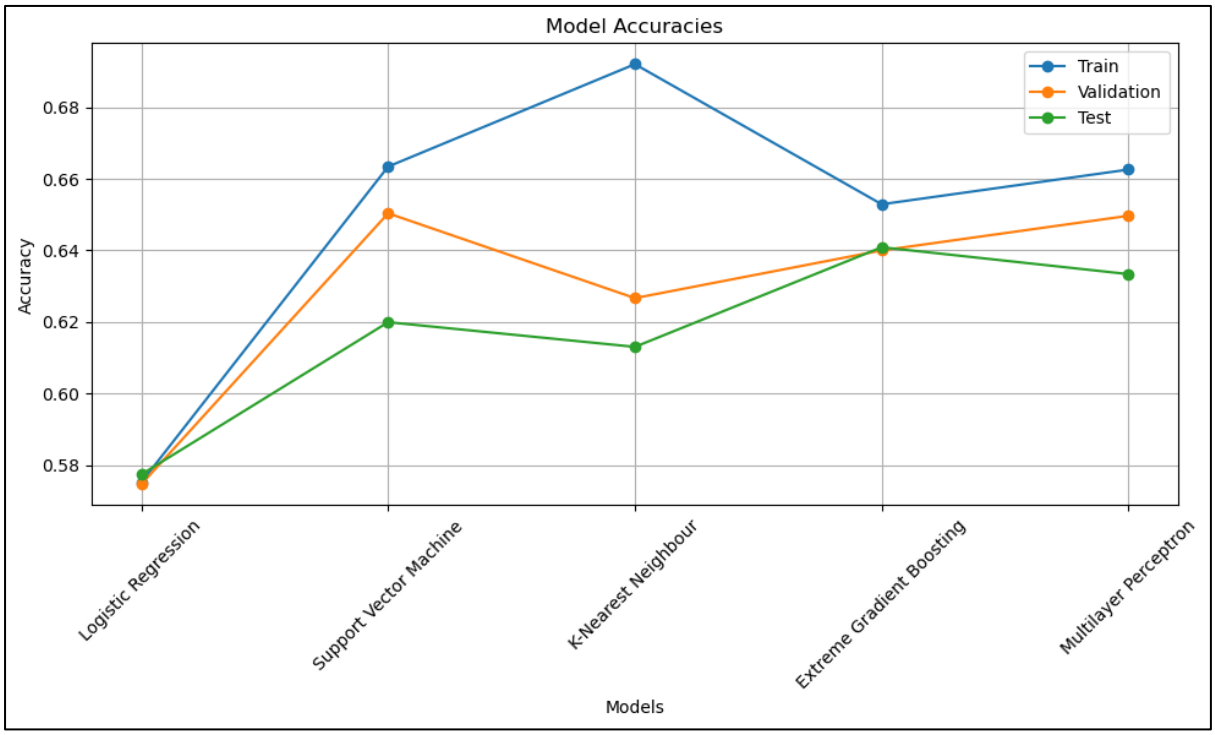

Fig 20. Final Models: Train, Validation, and Test Accuracy Comparison

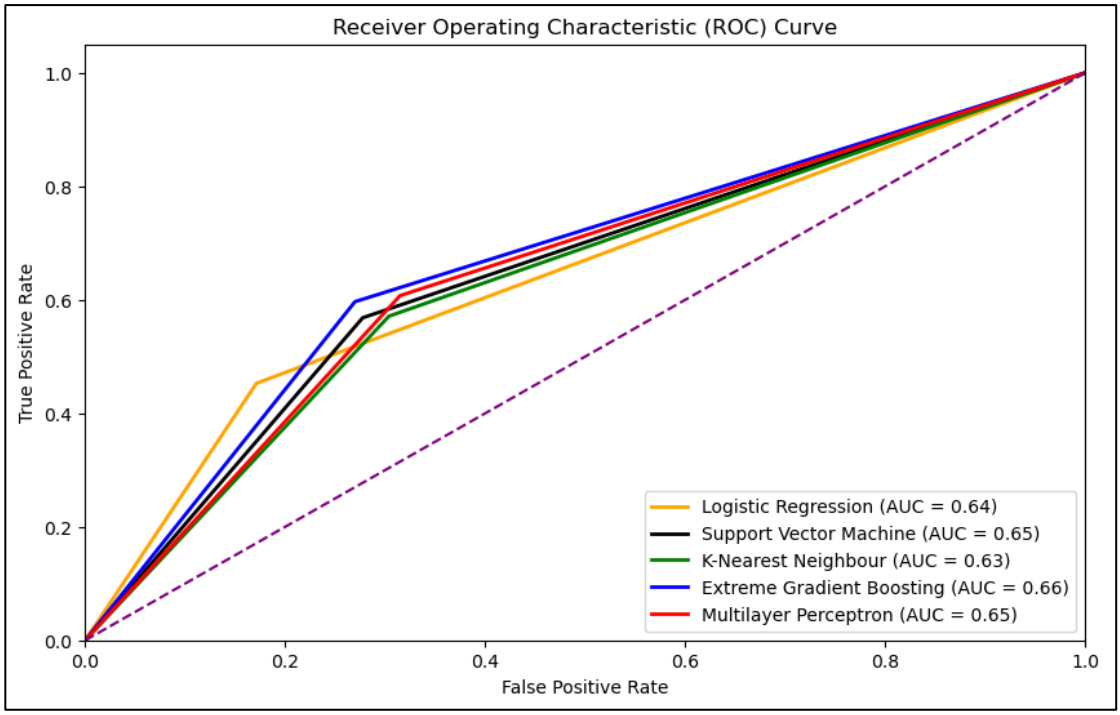

Fig 21. Final Models: Receiver Operating Characteristics (ROC) Curve Comparison

Model	Train Accuracy	Validation Accuracy	Test Accuracy	ROC-AUC Score	F1-Score
Logistic Regression	0.575216	0.574663	0.577426	0.640791	0.581063
Support Vector Machine	0.663406	0.650363	0.619942	0.645385	0.630119
K-Nearest Neighbour	0.692018	0.626665	0.613047	0.633693	0.623644
Extreme Gradient Boosting	0.652886	0.639993	0.640878	0.663424	0.651274
Multilayer Perceptron	0.662579	0.649674	0.633370	0.646228	0.643601

Table 2. Final Models: Metrics Comparison

4.4 Overall Evaluation

All models showed a trade-off between precision and recall, with XGBoost providing the most balanced performance as shown in figure 21. The metrics indicate that handling the class imbalance and improving the model's recall for the minority class was crucial. XGBoost's superior performance can be attributed to its ability to effectively handle the imbalanced dataset and its robustness in learning complex patterns.

In conclusion, the Extreme Gradient Boosting (XGBoost) model was selected for integration into the web-based application due to its better performance metrics as shown in table 2, which provide a reliable and efficient solution for loan approval prediction.

5. Web Application Loan Approval System

5.1 Web Application Overview and Functionality

5.1.1 Overview

The web application is designed to streamline the loan approval process by leveraging machine learning. It offers a user-friendly interface where applicants can input their financial and personal information. The application aims to provide quick and accurate loan approval decisions based on the data provided by users.

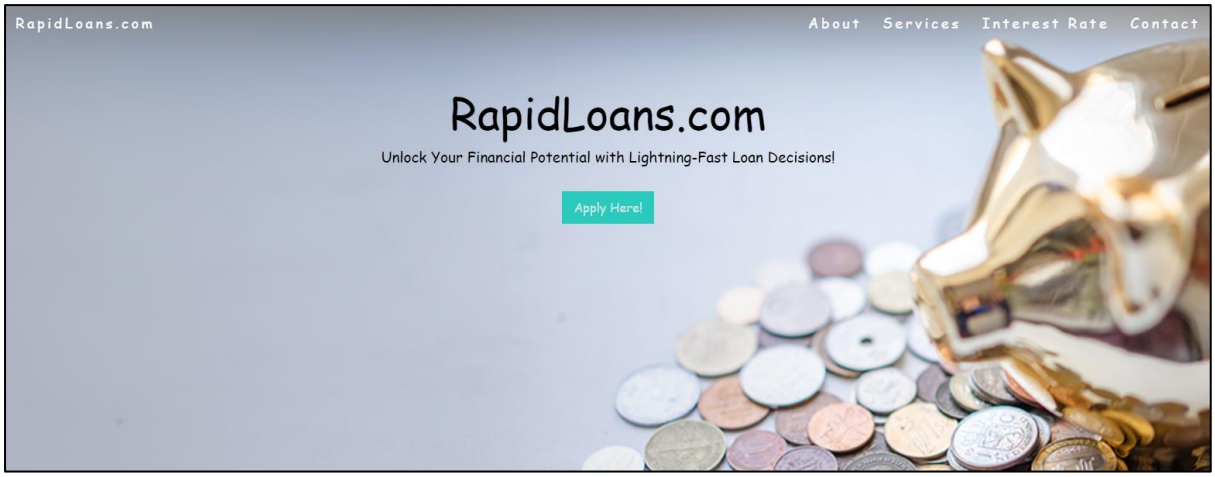

Fig 22. Web Application – Home Page

5.1.2 Functionality

i. **Loan Application Form:** Users fill out a detailed form with essential customer information, loan details, and financial data. This form includes fields for credit score, income, loan amount requested, employment status, and other relevant details (refer to figure 23).

ii. **Initial Validation:** Upon submission, the application checks the inputs against predefined minimum requirements to ensure that all necessary information is provided and meets basic criteria.

iii. **Predictive Model Integration:** Valid inputs are then passed to an Extreme Gradient Boost model for further analysis. The model processes the data and generates an output based on the likelihood of loan approval.

iv. **Decision Output:** The results are presented to the user in real-time, indicating whether the loan is approved or rejected (refer to figures and 25 and 27). The interface is designed to be intuitive and easy to navigate, ensuring a straightforward loan application process for users.

The application's real-time decision-making capability ensures that users receive immediate feedback on their loan applications, enhancing the user experience and providing quick insights into their loan eligibility.

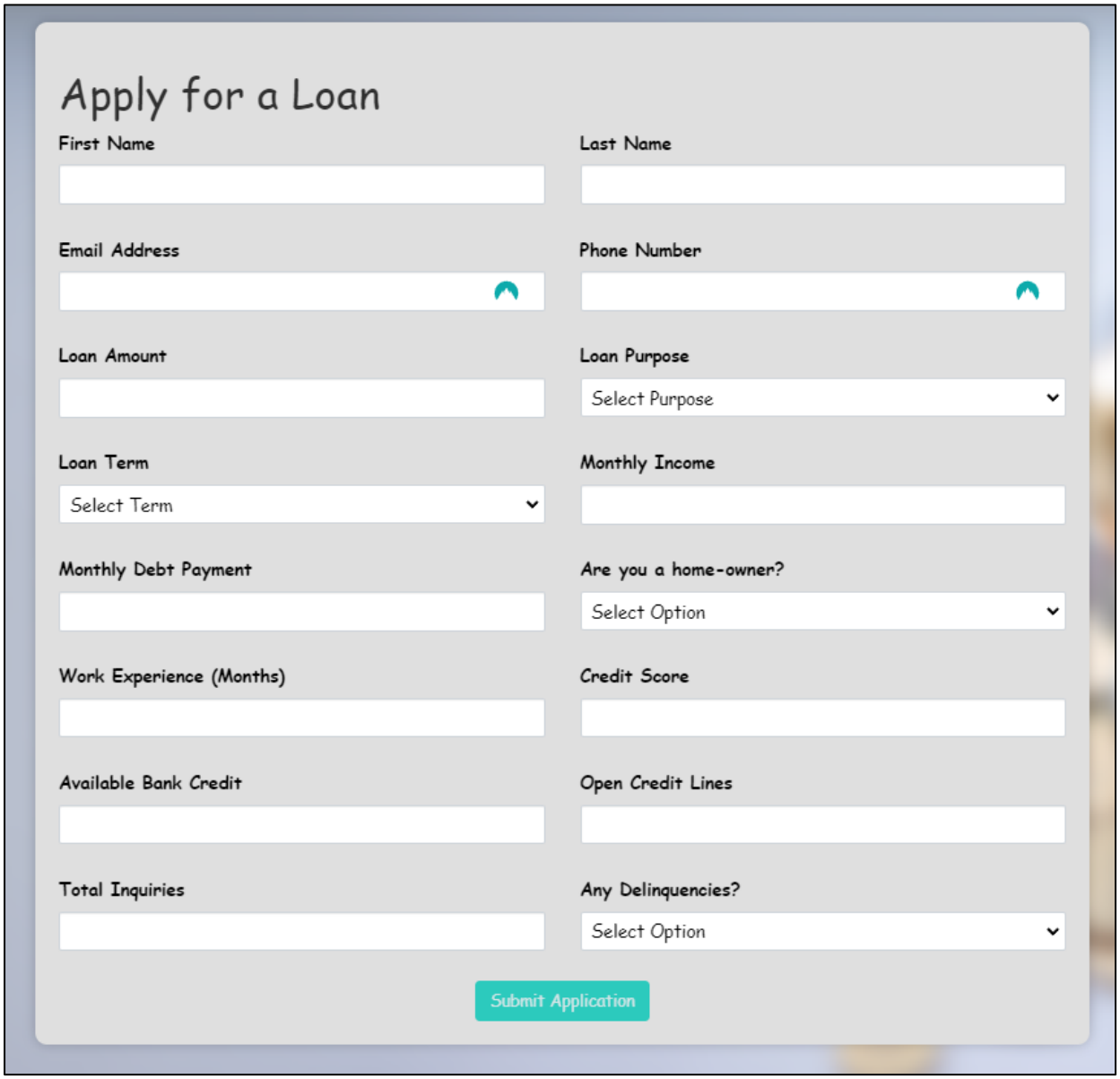

Fig 23. Web Application – Loan Form

5.2 Examples of Loan Approval/Denial

5.2.1 Approved Loan Example

- Interest Rate: 12%
- Monthly Installment: $664.29
- Average Credit Score: 750
- Term: 36 months
- Any Delinquencies: 0
- Is Homeowner: Yes
- Debt-to-Income Ratio: 0.10
- Stated Monthly Income: $5000
- Open Credit Lines: 6

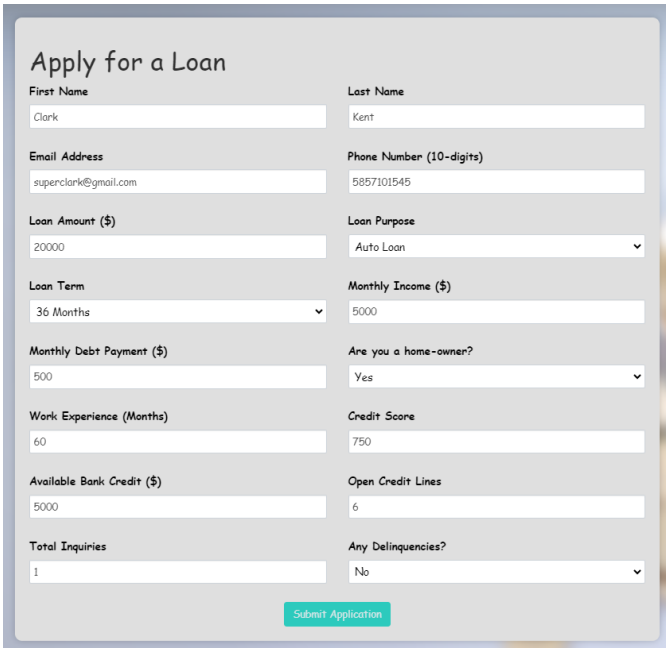

Fig 24. Loan form – Approved

In this example, the loan is approved because the borrower demonstrates a strong credit profile A lower interest rate, along with a credit score indicates that the borrower is less risky. The monthly payment is reasonable compared to the borrower's income and the debt-to-income ratio is low showing that the borrower is not overly indebted. The lack of missed payments and the fact that the borrower owns a home also boost their credibility. These factors together result in a decision to approve the loan.

Fig 25. Web Application – Loan Approved

5.2.2 Declined Loan Example

- Interest Rate: 15%
- Monthly Installment: $2,773.23
- Average Credit Score: 650
- Term: 36 months
- Any Delinquencies: 0
- Is Homeowner: No
- Debt-to-Income Ratio: 0%
- Stated Monthly Income: $1000
- Open Credit Lines: 3

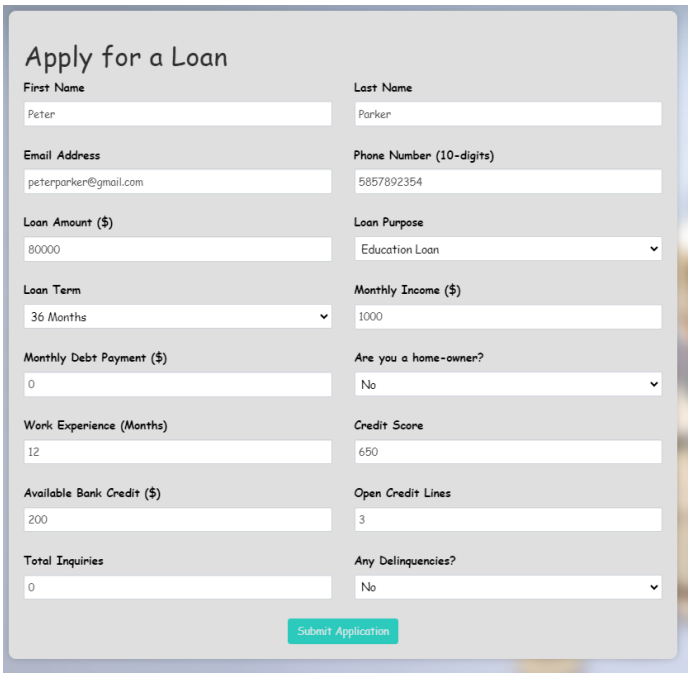

Fig 26. Loan form - Declined

On the other hand, this loan application is declined due to several risk factors. The higher interest rate and lower credit score suggest a higher likelihood of default. The monthly installment is high compared to the borrower's stated income, raising concerns about the borrower's ability to make payments. Moreover, the borrower does not own a home, indicating less financial stability. Although there are no delinquencies, the combination of a lower income and fewer open credit lines contributes to the decision to decline the loan.

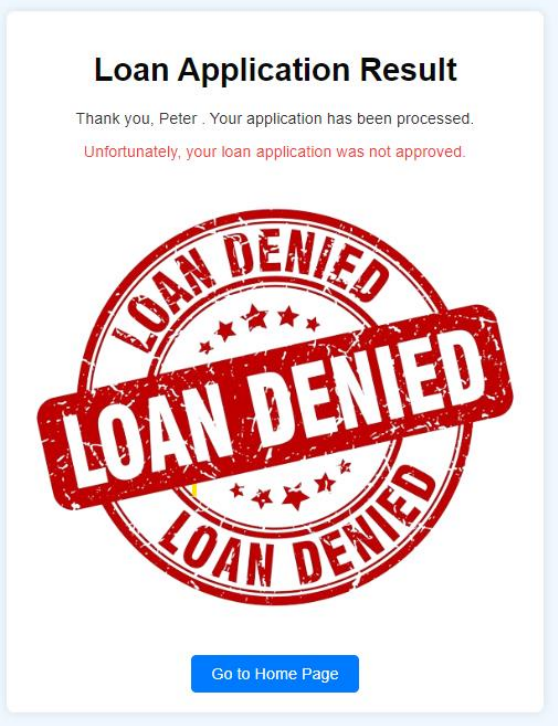

Fig 27. Web Application – Loan Denied

6. Future Work

Although the present project has successfully developed a robust loan approval prediction model, several points need improvement to make the system more efficient and effective.

1. Model Ensemble and Stacking

There is a possibility of implementing ensemble methods beyond those currently used for better prediction accuracy. The Stacked Ensemble approach combines the strengths of multiple models to avoid individual model weaknesses, aiding in more robust prediction.

2. Hyperparameter Optimization

While initial tuning of hyper-parameters was done, more extensive methods of optimization could be employed using Bayesian optimization or Genetic Algorithms to cover a large space of hyper-parameters.

3. Real-Time Data Integration

Integrating real-time data streams into the predictive model could improve its relevance and accuracy. Incorporating recent financial transaction data, market conditions, and customer activities could make the predictions more dynamic and responsive to current conditions.

4. User Interface and Experience

Additional user interface designs and functions can be added to the web application. For instance, providing the user with a visualization of their current loan eligibility status and recommending seeking better approval chances can increase the user's engagement and satisfaction.

5. Expanding the Dataset

Expanding the dataset by collaborating with more financial institutions could help generalize the model across different demographics and economic conditions. A more diverse dataset would allow the model to learn from a broader range of examples, improving its robustness.

Addressing these areas in future iterations will make the loan approval prediction system more accurate, user-friendly, and applicable to various scenarios. This ongoing development will help create a more reliable and efficient tool for loan approval processes, ultimately benefiting both financial institutions and customers.

7. Conclusion

This project aimed to develop a robust and efficient machine-learning model for predicting loan approval decisions. Various algorithms, including Logistic Regression, Support Vector Machine, K-Nearest Neighbour, Extreme Gradient Boosting, and Multilayer Perceptron, were trained, tested, and evaluated to identify the most effective model. The extensive preprocessing steps, including handling missing values, encoding categorical variables, and addressing class imbalance through techniques like class weighting and under-sampling, ensured the integrity and quality of the data. Among the models, Extreme Gradient Boosting (XGBoost) emerged as the top performer, achieving the highest accuracy and ROC-AUC score as shown in fig 16 and fig 18 respectively.

This model was then integrated into a web-based application using the Flask framework, providing a user-friendly interface for real-time loan approval predictions. The application enables users to input their data and receive immediate feedback on their loan application status, enhancing the decision-making process for both customers and financial institutions. The project encountered challenges, particularly with handling imbalanced data. This challenge was addressed through various techniques, ensuring reliable model performance and including comprehensive evaluation metrics such as precision, recall, F1 score, and ROC-AUC score.

In conclusion, this project has laid the groundwork for an efficient loan approval prediction system. Integrating a high-performing machine learning model with an intuitive web application marks a significant step towards automating and improving the loan approval process. With further enhancements and refinements, such a system may add great value to financial institutions, driving more informed and timely loan approval decisions.

8. References

[1] Nachiketa Hebbar, Deploy ML Model On Webpage|Python(Flask)| Forest Fire Prevention Using AI, (Dec. 24, 2019). Accessed: Aug. 02, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=Pc8WdnIdXZg

[2] "Classification Metrics using Sklearn," GeeksforGeeks. Accessed: Aug. 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/sklearn-classification-metrics/

[3] "Prosper Loan Data." Accessed: Aug. 02, 2024. [Online]. Available: https://www.kaggle.com/datasets/henryokam/prosper-loan-data

[4] C. Y. Wijaya, "5 Feature Selection Method from Scikit-Learn you should know," Medium. Accessed: Aug. 02, 2024. [Online]. Available: https://towardsdatascience.com/5-feature-selection-method-from-scikit-learn-you-should-know-ed4d116e4172

[5] H. Vyas, "Deploy a machine learning model using flask," Medium. Accessed: Aug. 02, 2024. [Online]. Available: https://towardsdatascience.com/deploy-a-machine-learning-model-using-flask-da580f84e60c

[6] Rob_Kaufman, "The History of the FICO® Score." Accessed: Aug. 02, 2024. [Online]. Available: https://www.myfico.com/credit-education/blog/history-of-the-fico-score

[7] U. Lal, "Mastering Loan Default Prediction: Tackling Imbalanced Datasets for Effective Risk Assessment," Geek Culture. Accessed: Aug. 02, 2024. [Online]. Available: https://medium.com/geekculture/mastering-loan-default-prediction-tackling-imbalanced-datasets-for-effective-risk-assessment-8e8dfb2084d0

[8] K. Kumari, "Loan Prediction Problem From Scratch to End," Analytics Vidhya. Accessed: Aug. 02, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/05/loan-prediction-problem-from-scratch-to-end/

[9] guest_blog, "10 Techniques to Solve Imbalanced Classes in Machine Learning (Updated 2024)," Analytics Vidhya. Accessed: Aug. 02, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/

[10] M. A. Sheikh, A. K. Goel, and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Jul. 2020, pp. 490–494. doi: 10.1109/ICESC48915.2020.9155614.

[11] P. S. Saini, A. Bhatnagar, and L. Rani, "Loan Approval Prediction using Machine Learning: A Comparative Analysis of Classification Algorithms," in 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), May 2023, pp. 1821–1826. doi: 10.1109/ICACITE57410.2023.10182799.

[12] A. T. Rahman, M. R. H. Purno, and S. A. Mim, “Prediction of the Approval of Bank Loans Using Various Machine Learning Algorithms,” in 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), Jul. 2023, pp. 272–277. doi: 10.1109/AIC57670.2023.10263880.

[13] R. Priscilla, T. Siva, M. Karthi, K. Vijayakumar, and R. Gangadharan, “Baseline Modeling for Early Prediction of Loan Approval System,” in 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Jan. 2023, pp. 1–7. doi: 10.1109/ICECONF57129.2023.10083650.

[14] C. Prasanth, R. P. Kumar, A. Rangesh, N. Sasmitha, and D. B, “Intelligent Loan Eligibility and Approval System based on Random Forest Algorithm using Machine Learning,” in 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Mar. 2023, pp. 84–88. doi: 10.1109/ICIDCA56705.2023.10100225.

[15] Ugochukwu. E. Orji, Chikodili. H. Ugwuishiwu, Joseph. C. N. Nguemaleu, and Peace. N. Ugwuanyi, “Machine Learning Models for Predicting Bank Loan Eligibility,” in 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), Apr. 2022, pp. 1–5. doi: 10.1109/NIGERCON54645.2022.9803172.

[16] Ch. Naveen Kumar, D. Keerthana, M. Kavitha, and M. Kalyani, “Customer Loan Eligibility Prediction using Machine Learning Algorithms in Banking Sector,” in 2022 7th International Conference on Communication and Electronics Systems (ICCES), Jun. 2022, pp. 1007–1012. doi: 10.1109/ICCES54183.2022.9835725.