Predictive Modeling for Loan Approval

Minnu V B

Roll No: 33

Reg.No: KTE23MCA-2033

Guided By : Prof. Swapna K J

Department of Computer Applications

Rajiv Gandhi Institute Of Technology, Kottayam

March 1, 2025

Index

Introduction

- A predictive loan approval model automates loan decisions by analyzing applicant data.
- It addresses class imbalance and bias, ensuring faster, more accurate, and fair loan approvals.
- SMOTENC balances datasets by generating synthetic minority class data.
- Machine learning algorithms like Logistic Regression, Decision Trees, Random Forest, and XGBoost are used here, with evaluation based on Accuracy, F1 Score, and AUC-ROC.
- SHAP is used to interpret model predictions, ensuring transparency and trust by identifying key factors influencing loan decisions.

Scope

- Development of a robust predictive model for loan approval using machine learning algorithms.
- Implementation of techniques such as SMOTENC to handle class imbalance, ensuring fair predictions across all applicant categories.
- Improvement of decision-making efficiency by automating loan approval processes, reducing processing time and human errors.

Relevance

- Focus on fairness and inclusivity by mitigating biases related to sensitive attributes like gender, ethnicity, or socioeconomic status.
- Alignment with ethical AI practices, ensuring transparency in model decisions, fostering trust with applicants and financial institutions.

REQUIREMENT ANALYSIS

Existing System

Table 1: Literature Review

Sl No.	Title	Author(s)	Method	Key Findings
1	Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning	Han, Hui; Wang, Wen-Yuan; Mao, Bing-Huan (2005)	Borderline-SMOTE	Introduced an over-sampling technique to handle imbalanced datasets for improved model learning.
2	Credit Scoring and its Applications	Thomas, Lyn; Edelman, David; Crook, Jonathan (2002)	Credit Scoring Techniques	Provided an overview of credit scoring methods and their applications in financial decision-making.
3	Credit Risk Measurement: Developments Over the Last 20 Years	Altman, E. I.; Saunders, A. (1994)	Credit Risk Measurement	Discussed the evolution of credit risk assessment methods and their relevance to banking.

Proposed System

- The system uses machine learning algorithms like Logistic Regression, Decision Trees, Random Forests and XGBoost to predict loan outcomes.
- SMOTENC addresses class imbalance, ensuring fair representation of both approved and rejected applications.
- GridSearchCV tunes hyperparameters to optimize model performance.
- Evaluation metrics like accuracy, F1 score, and AUC-ROC assess the system's effectiveness.
- The system is scalable, handling large datasets and complex interactions for real-world financial environments.

S/W & H/W requirement

- **Hardware:** Processor: Intel Core i5 10th Gen or higher; RAM: At least 8 GB; Storage: 256 GB SSD or higher; Operating System: Windows 10, Linux, or macOS.
- **Software:**
  - Programming Language: Python 3.8 or higher
  - Libraries: Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn
  - Development Tools: Jupyter Notebook, Visual Studio Code, or PyCharm
- **Datasets:** Publicly available datasets from Kaggle.
- **Machine Learning Techniques:**
  - Data Balancing: SMOTENC
  - Algorithms: Logistic Regression, Decision Trees, Random Forests
  - Optimization: GridSearchCV for hyperparameter tuning
- **Evaluation Metrics:** Accuracy, F1 Score, AUC-ROC

Development Methodology

- **Data Collection:**
  - Data collection from Kaggle.
- **Data Preprocessing:**
  - Missing values were imputed to ensure data completeness.
  - Nominal and continuous features were encoded to prepare the dataset for model training.
  - Class imbalance was addressed using SMOTENC.
- **Model Training and Evaluation:**
  - Machine learning algorithms such as Logistic Regression, Decision Trees, Random Forests and XGBoost were used to train predictive models.
  - GridSearchCV was employed for hyperparameter tuning to identify the optimal configuration for each algorithm.
  - Evaluated using metrics such as accuracy, F1 score, and AUC-ROC.

Development Methodology

Deployment:

- The predictive model was integrated into loan processing systems for real-time decision-making.
- The system was designed to handle high volumes of applications efficiently and scalably.

Continuous Monitoring and Improvement:

- System logs and performance metrics were continuously monitored to detect anomalies or performance degradation.
- Periodic retraining was conducted using updated data to adapt to changing market trends and behaviors.

Design

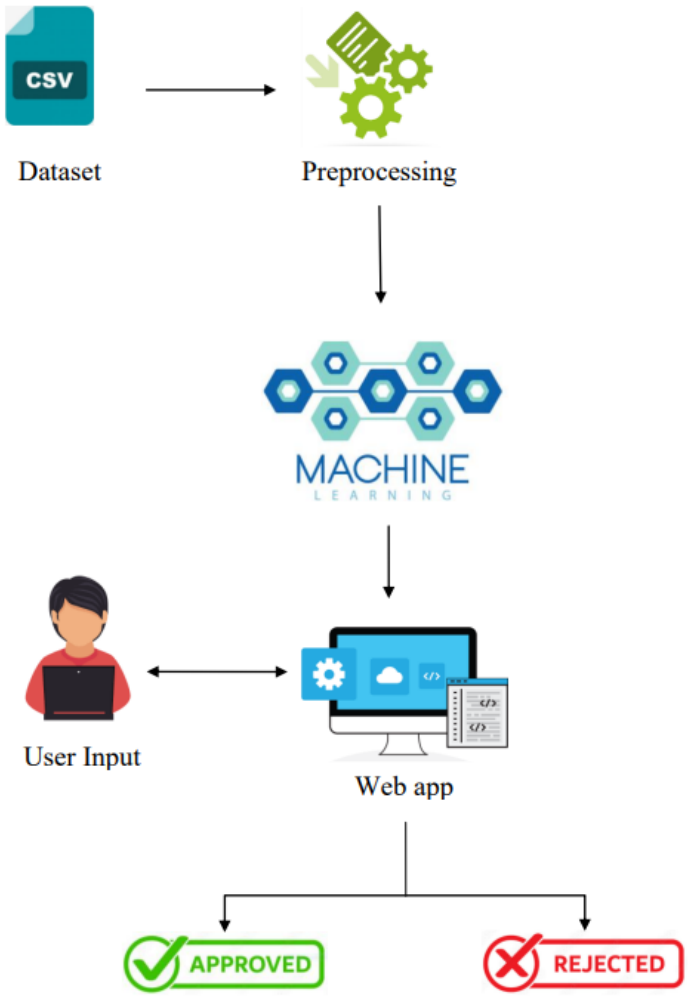


Figure 1: System Architecture

Design

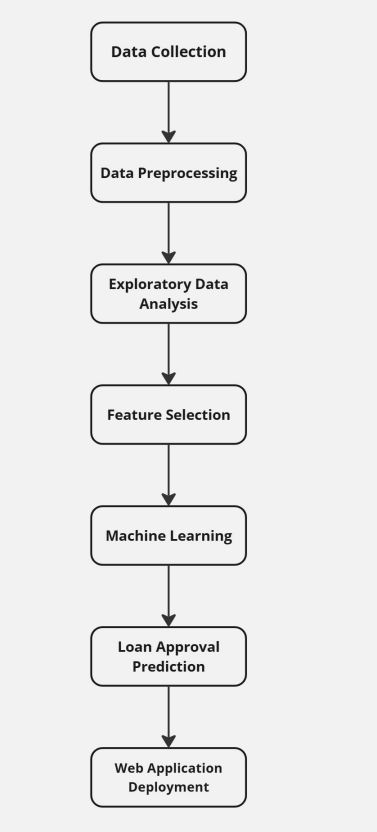


Figure 2: Work Flow

Implementation Details

- **Programming Languages:**
  - Python was used for its extensive support in data analysis, machine learning, and visualization tools.
- **Libraries and Frameworks:**
  - **Scikit-learn:** Used for building and training the predictive model (e.g., Logistic Regression, Random Forest,Decision Tree).
  - **Pandas and NumPy:** Used for efficient data manipulation and preprocessing.
  - **Matplotlib and Seaborn:** Used for data visualization and exploratory data analysis.

Implementation Details

- **Data Handling:**
  - Missing values were handled using.
  - Data was normalized and encoded to ensure compatibility with the machine learning model.
- **Model Architecture:**
  - Several machine learning models were tested, including Logistic Regression, Decision Trees, Random Forests and XGBoost.
  - The model with the best performance (e.g., XGBoost) was chosen for final deployment.
- **Model Training:**
  - The dataset was split into training and testing sets using an 80/20 ratio.
  - Grid Search with cross-validation was employed to optimize hyperparameters .
  - Class imbalance was addressed using SMOTE.

Implementation Details

- **Testing Setup:**
  - Evaluated model performance using metrics such as Accuracy, F1-score, and AUC-ROC.
  - Tested the model on unseen loan application data to verify its ability to generalize.
- **Visualization of Results:**
  - Plotted ROC curves to evaluate the trade-off between True Positive Rate and False Positive Rate.

Current Status of Work

- Collected and preprocessed the loan dataset, including handling noisy and missing data.
- Used SMOTENC to balance the class distribution in the training data before model training.
- Performed hyperparameter tuning using GridSearchCV to optimize model performance.
- Applied SHAP (SHapley Additive exPlanations) to interpret model predictions and ensure transparency.
- Trained the Logistic Regression model, Decision Tree model, Random Forests and XGBoost.

Results

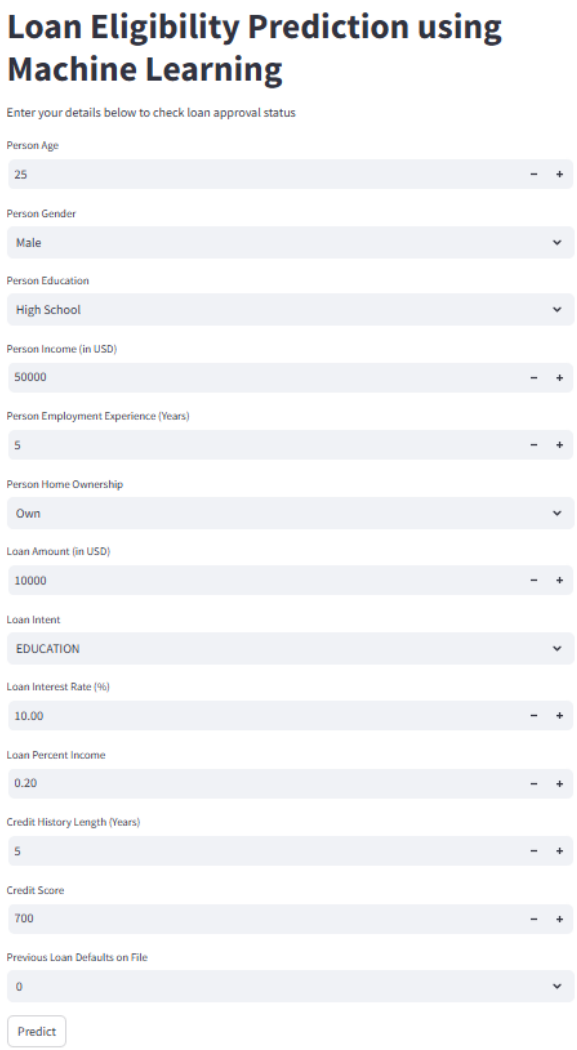


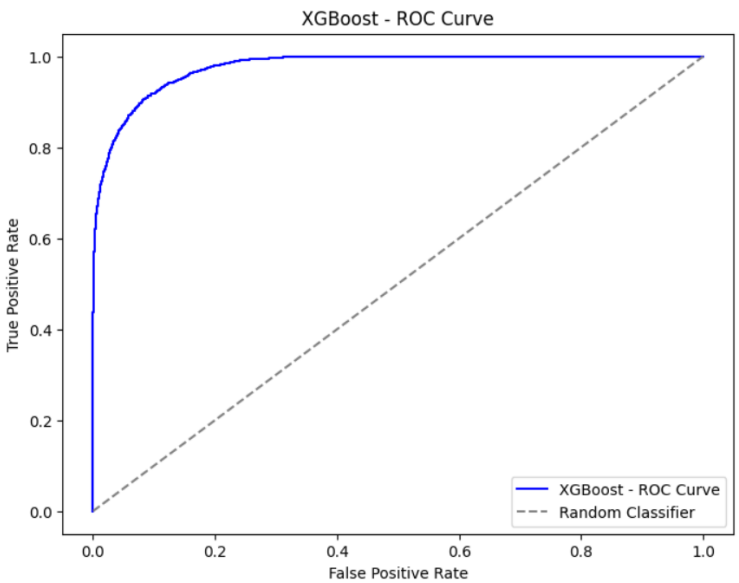
Figure 3: Userinterface

Analysis of Results

Table 2: Evaluation Metrics for Predictive Loan Approval Models

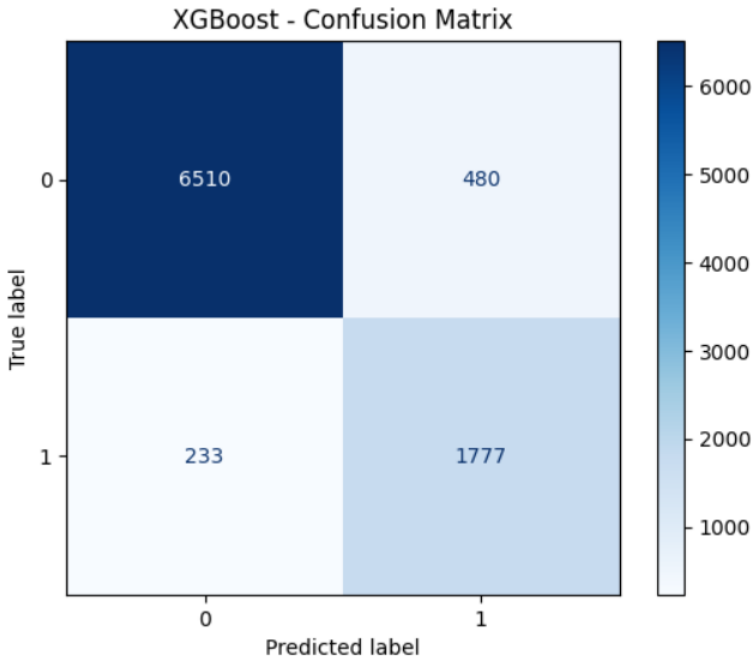
Model	**Accuracy**	**F1 Score**	**AUC-ROC**
Logistic Regression	0.8573	0.7376	0.9491
Decision Tree	0.8898	0.7376	0.9491
Random Forest	0.9123	0.8159	0.9720
XGBoost	0.9208	0.8329	0.9768

Analysis of Results



- The ROC Curve illustrates the performance of the XG-Boost model in predicting loan approvals.
- The curve stays near the top-left corner, indicating high true positive rate (TPR) and low false positive rate (FPR).
- The AUC value is close to 1, demonstrating excellent classification ability and strong model performance.

Analysis of Results



- The model correctly classified 6,510 negative cases (True Negatives) and 1,777 positive cases (True Positives).
- Misclassifications: 480 False Positives (incorrectly predicted approvals) and 233 False Negatives (missed approvals).
- The high count of correct predictions reflects strong model performance with minimal classification errors.

Pending Works

- Loan Rejection Explanation: It will explain why a loan application was rejected, improving transparency.
- Loan Amount Prediction: Extending the model to predict the possible loan amount an applicant can receive based on their details.

Project Plan

Table 3: Project Plan

Task	**Status**	**Remarks**
Find Problem	Completed	
Data Collection and Preprocessing	Completed	
Initial Model Training	Completed	
Model refinement and validation	Completed	
Development of front end	Completed	
Model Integration	Yet to start	Planning to complete by March 15th 2025
Final adjustments and validation	Yet to start	Planning to complete by March 30th 2025
Fair Report (Draft)	Yet to start	Planning to complete by April 4th 2025
Fair Report (Final)	Yet to start	Planning to complete by April 9th 2025

Conclusion

- Successfully trained multiple models, including Logistic Regression, Decision Tree, Random Forests and XGBoost , using the collected loan dataset.
- Future enhancements include adding a Loan Rejection Explanation module to provide reasons for loan denials, improving trust and user understanding.
- Expanding functionality to include Loan Amount Prediction, allowing applicants to estimate the possible loan they can receive based on their details.
- The system could be deployed as a web or mobile application, enabling real-time loan approval assessments for financial institutions and applicants.

Git History Screenshot

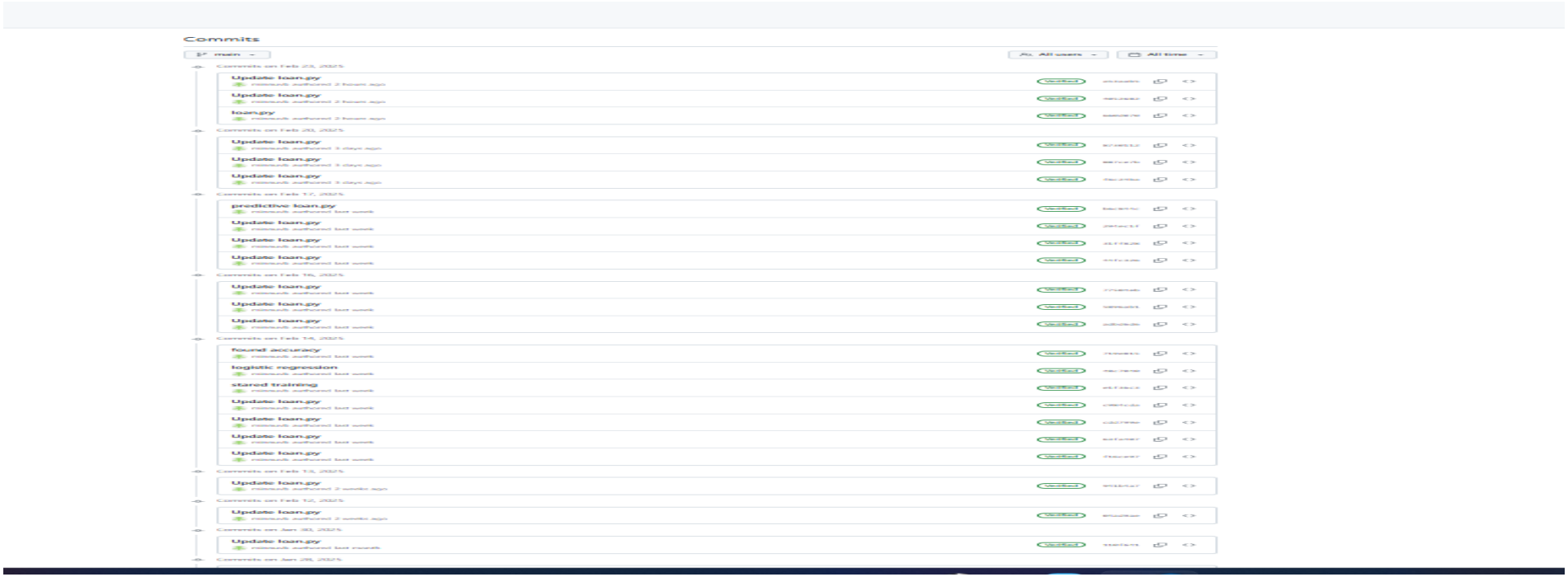


Figure 4: Git history

Bibliography

[1] Lyn Thomas, David Edelman, and Jonathan Crook. *Credit Scoring and its Applications*. 2002.

[2] E. I. Altman and A. Saunders. "Credit risk measurement: Developments over the last 20 years". In: *Journal of Banking Finance* 21.11-12 (1994), pp. 1721–1742. DOI: 10.1016/S0378-4266(97)00036-8.

[3] Si Tse Shi et al. "Machine learning-driven credit risk: a systemic review". In: *Neural Computing and Applications* 34 (2022). DOI: 10.1007/s00521-022-07472-2.

[4] GeeksforGeeks. *Loan Approval Prediction Using Machine Learning*. https://www.geeksforgeeks.org/loan-approval-prediction-using-machine-learning/. n.d.

Thank you!