UNIVERSITY OF ESWATINI

FACULTY OF SOCIAL SCIENCE

DEPARTMENT OF STATISTICS AND DEMOGRPHY

Credit Risk Assessment Models In Personal Lending: A Comparative Analysis of Logistic Regression, SVM, MLP and TabTransformer with Explainable AI

Research Report

By

NDUMISO CELIMPILO SISEKELO SHONGWE

202102870

A RESEARCH PROJECT PAPER SUBMITTED TO THE DEPARTMENT OF STATISTICS AND DEMOGRAPHY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE BACHELOR OF ARTS DEGREE IN SOCIAL SCIENCES

SUPERVISOR: Mr S Masango

JULY 2015

DECLARATION

I Ndumiso Celimpilo Sisekelo Shongwe declare that this thesis entitled; **Credit Risk Assessment Models In Personal Lending: A Comparative Analysis of Logistic Regression, SVM, MLP and TabTransformer with Explainable AI**, is my own, individual work. It is submitted for the Bachelor of Arts in Social Science Degree at the University Of Eswatini (UNESWA). It complies with the requirement of the University and meets the standards with respect to originality and quality. It has never been submitted before for any degree or examination in any other university.

_____.

Signature

_____.

Date

ABSTRACT

This study conducts a comparative analysis of four credit risk assessment models—Logistic Regression, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and TabTransformer in the context of personal lending. Using a publicly available Kaggle dataset comprising 22,913 loan records, the research evaluates the models based on predictive accuracy, interpretability, and computational efficiency, while addressing the challenges of imbalanced datasets and model transparency through Explainable AI (XAI) techniques such as SHAP and LIME. The findings reveal that MLP achieves the highest accuracy (99.96%) and precision (100%), with minimal misclassifications, making it the most suitable model for personal lending. TabTransformer, while slightly less accurate, demonstrates superior performance in terms of ROC-AUC (99.88%) and highlights the potential of transformer-based models for tabular data. Logistic Regression offers high interpretability but lower accuracy, while SVM struggles with precision in imbalanced datasets.

The study recommends MLP as the optimal model for financial institutions seeking a balance between performance and usability, with TabTransformer as a promising alternative for those with advanced computational resources. The findings reveal that deep learning and transformer-based models, when paired with XAI tools, can offer both predictive excellence and interpretability. Additionally, the integration of XAI techniques enhances model transparency, aligning with regulatory requirements for fair and explainable lending practices. The research also contributes to the growing body of work promoting fair, data-driven, and explainable lending practices in the financial sector. Future research should explore the application of these models on diverse, real-world datasets and investigate the inclusion of additional models, such as Random Forest or Gradient Boosting, to further validate the findings.

DEDICATION

This research project is dedicated to all financial institutions, particularly commercial banks and lending agencies that are committed to advancing their credit risk assessment systems through innovative and explainable machine learning solutions. It is my sincere hope that the insights from this study will contribute meaningfully to the ongoing transformation of credit decision-making in the financial sector.

I also dedicate this work to Mr. S. Masango, my supervisor, whose expert academic guidance, encouragement, and critical feedback were central to the development and completion of this project.

Furthermore, I extend this dedication to the **AI ENGINEERS–ESWATINI** community. Your commitment to collective learning, intellectual exchange, and the Saturday sessions we shared provided a practical and motivating environment that greatly enhanced my understanding of artificial intelligence in real-world applications.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the **University of Eswatini**, and in particular the **Department of Statistics and Demography**, for their academic support and for providing the institutional platform that made this research possible. I am especially thankful to my supervisor, Mr. S. Masango, for his invaluable mentorship, insightful critiques, and consistent encouragement throughout all phases of this study. His dedication and professionalism played a significant role in shaping both the quality and direction of this work.

I am also thankful to the lecturers, faculty members, and administrative staff of the **Faculty of Social Science** for their academic guidance and support during my undergraduate journey. Special thanks are extended to my classmates and peers for their camaraderie, shared learning, and valuable contributions during the research process.

A special acknowledgement goes to **AI ENGINEERS–ESWATINI**, a dynamic and collaborative community of aspiring machine learning practitioners. The Saturday learning sessions, project discussions, and technical exchanges we shared not only deepened my understanding of advanced AI techniques but also played a critical role in the development of this study.

Lastly, I wish to acknowledge the financial institutions whose operational realities inspired the focus of this research. It is my hope that the findings will offer practical value and contribute to the ongoing modernization of credit risk assessment in Eswatini and beyond.

Table of Contents

List of Tables

Table of Figures

List of Abbreviations

MLP	Multiple Layer Perceptron
SVM	Support Vector Machine
LR	Logistics Regression
DCC-GARCH	Dynamic conditional Correlation- Generalised Auto regressive Conditional Heteroskendasticity
SHAP	SHapley Additive Explainations
LIME	Local Interpretable Model-agnostic Explainations
GBM	Gradient Boosting Machines
RPART	Recursive Partitioning and Regression Trees
NN	Neural Networks
Recall	Sensitivity or True Positive Rate
ReLU	Rectified Linear Unit
ARIMA	Auto Regressive Intergrated Moving Average
GBM	Gradient Boosting Models

CHAPTER 1: INTRODUCTION

1.1 Introduction

Credit risk assessment plays a vital role in the financial industry, enabling lenders to evaluate a borrower's likelihood of default and make informed lending decisions. Accurate risk assessment safeguards financial institutions from potential losses while promoting responsible lending practices, ensuring both profitability and financial stability (Coskun & Turanh, 2023). Traditionally, credit risk evaluation has relied on statistical models such as Logistic Regression (LR) and Discriminant Analysis, which use financial ratios, credit scores, and borrower characteristics to predict default probabilities. However, these models often assume linear relationships between input variables and loan default, limiting their ability to capture complex, nonlinear patterns in financial data (Zhang et al., 2022). Additionally, traditional models depend on handcrafted features, which may not fully utilize the vast amount of borrower data available today.

The advent of machine learning (ML) has introduced a paradigm shift in credit risk modeling, allowing financial institutions to leverage data-driven approaches for more accurate and adaptive risk assessment. Unlike traditional models, ML algorithms can handle large datasets, detect intricate patterns, and improve predictive accuracy without relying on predefined assumptions (Chen & Xiang, 2023). Support Vector Machines (SVM) effectively separate high-risk and low-risk borrowers using optimized decision boundaries, while Multiple Layer Perceptron (MLP), a neural network model, learns complex financial relationships. More recently, transformer-based models, such as Tab Transformer, have emerged as powerful alternatives, leveraging self-attention mechanisms to process tabular financial data more effectively (Huang et al., 2023). These innovations offer improvements in accuracy and adaptability, but they also introduce challenges related to computational efficiency, interpretability, and regulatory compliance (Li & Zhang, 2024).

Given the increasing reliance on credit in modern economies, financial institutions must adopt the most effective predictive models to mitigate risk while ensuring fair and transparent lending decisions (Coskun & Turanh, 2023). Traditional models like Logistic Regression remain widely used due to their interpretability, but they may struggle to capture complex borrower behaviors as effectively as advanced machine learning techniques. This research aims to compare and evaluate four key models: Logistic Regression, Support Vector Machines, Multiple Layer Peceptron, and Tab Transformer to determine which algorithm provides the best balance between accuracy, interpretability, and computational efficiency. The models will be tested on a comprehensive dataset of historical loan data, with particular attention to their ability to handle imbalanced datasets and accurately classify borrowers based on credit history, financial condition, collateral, and macroeconomic conditions. By analyzing the trade-offs between these models, this study seeks to provide practical insights for financial institutions in selecting the most effective credit risk assessment approach.

As machine learning models become more complex, their decision making processes often resemble Black Boxes making it difficult for financial institutions to explain why a particular loan was approved or denied (Molnar, 2022). This lack of interpretability raises concerns about fairness, accountability and regulatory compliance. To address this, Explainable AI techniques such as SHapley Additive Explainations (SHAP) and Local Interpretable Model-agnostic Explainations (LIME) have been incorporated into the study

1.2 Background

Credit risk assessment has evolved significantly over the past century, driven by advancements in statistical methods, computing power, and the availability of large datasets. This section provides a historical overview of credit risk assessment, highlights key trends, and discusses the emergence of machine learning and deep learning models in this field. The practice of credit risk assessment emerged from the fundamental need of financial institutions to protect their assets while expanding their lending activities. In the 1950s, banks primarily relied on expert judgment and relationship-based lending, where loan officers made decisions based on their personal knowledge of borrowers and local market conditions (Anderson & Thompson, 2022). This approach, while personal, was inherently subjective and often led to inconsistent lending decisions across different branches and regions.

The 1960s marked a significant shift towards more systematic approaches with the introduction of the FICO score by Fair Isaac Corporation. This development represented the first standardized credit scoring system, fundamentally changing how financial institutions evaluated creditworthiness (Martinez et al., 2023). The success of FICO scores led to their widespread adoption because they offered consistent evaluation criteria across different locations, reduced processing time for loan applications, lower operational costs in credit assessment and improved risk management through standardization. (Martinez et al., 2023). The widespread adoption of FICO scores is illustrated in Figure 1, which shows the growth in the use of credit scores by financial institutions from 1960 to 2020 (Martinez et al., 2023):

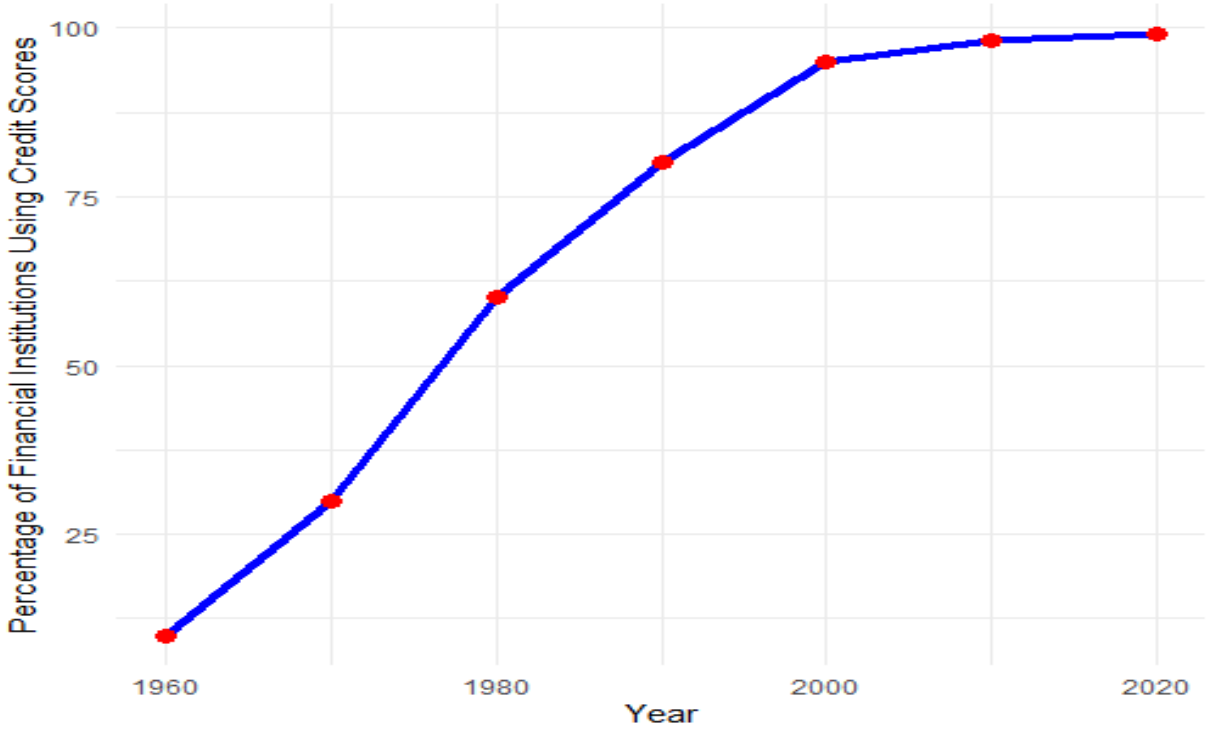


Figure 1: Growth in The Use of credit Scores (1960-2020)

The advent of computer systems in the 1970s enabled the implementation of statistical methods in credit assessment (Wilson and Roberts, 2021). Logistic regression became the predominant tool during this period for several reasons its ability to handle binary outcomes aligned perfectly with credit decision-making (approve/deny), the results were easily interpretable, satisfying regulatory requirements the computational requirements were manageable with available technology and the probabilistic output provided a natural risk measure (Johnson et al., 2022). By the 1990s, financial institutions had accumulated substantial historical data on lending outcomes. This data revealed that traditional statistical methods, while useful, had limitations in capturing complex relationships between variables. For instance, the Asian financial crisis of 1997 exposed how conventional models failed to account for systemic risks and interconnected financial factors (Wang & Liu, 2023).

The early 2000s witnessed a gradual shift toward machine learning approaches, driven by increased computational power, larger datasets becoming available, and limitations of linear models becoming more apparent and the success of ML in other financial applications (Lee & Thompson, 2023). Support Vector Machines gained prominence during this period because they could handle non-linear relationships effectively, provide robust performance with limited data manage high-dimensional feature spaces efficiently and maintain reasonable computational requirements (Davidson & Kumar, 2022). The 2008 financial crisis served as a catalyst for adopting more sophisticated risk assessment methods, the post-crisis analysis revealed that traditional models had failed to capture complex market interactions and systemic risks. This realization led to increased investment in advanced analytics and machine learning capabilities (Chen et al., 2024). Figure 2 illustrates the growing adoption of machine learning models in credit risk assessment from 2000 to 2025:

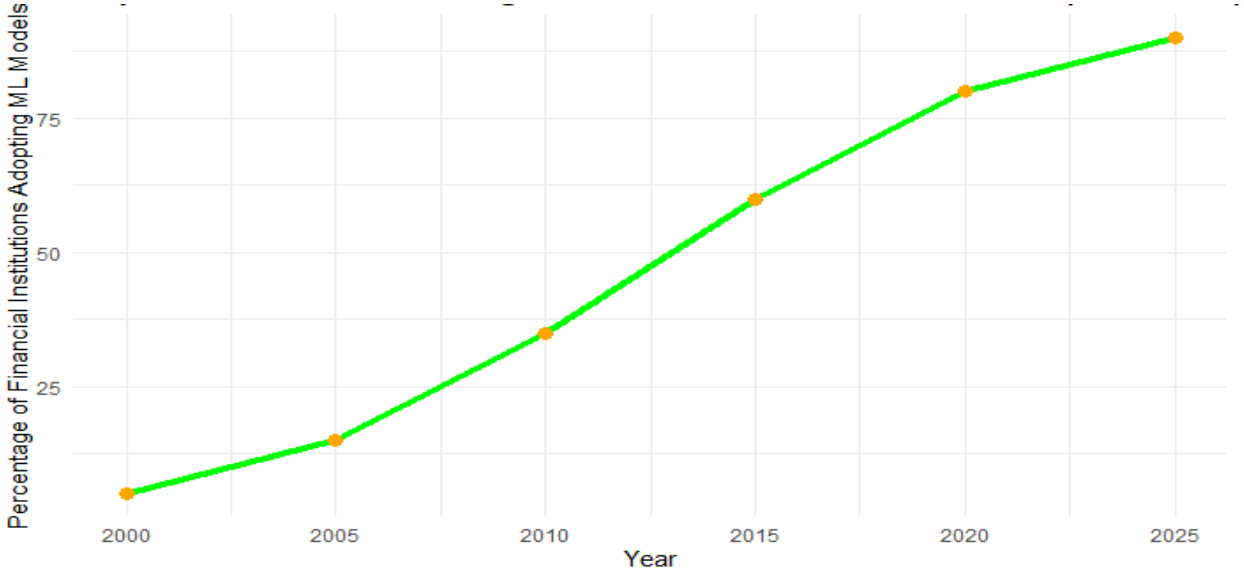


Figure 2: Adoption of Machine learning models in Credit Risk Assessment (2000-2025)

The emergence of deep learning, particularly Multiple Layer Perceptron models, marked another significant shift in credit risk assessment Zhang et al, 2023). Several factors contributed to this transition and they include data availability where digitalization of financial services generated unprecedented amounts of data (Brown & Miller, 2023). This is whereby traditional banks and

fintech companies began collecting transaction histories, mobile banking behavior, social media data, alternative data sources. Another factor which contributed to this transition is computational advances which involves the development of specialized hardware (GPUs, TPUs) made complex model training feasible, enabling real-time risk assessment, processing of unstructured data and integration of multiple data sources (Taylor & Singh,2023). Regulators also began accepting more sophisticated models, provided they could be explained and validated. This shift was driven by the need for better risk management, recognition of traditional models' limitations and emphasis on model transparency and fairness.

The recent introduction of transformer architectures, specifically the Tab Transformer, represents a significant advancement in credit risk assessment (Chen & Liu, 2024). Originally designed for natural language processing tasks, transformers have been adapted for tabular data analysis, offering several advantages. The advantages include attention mechanisms which is their ability to capture complex relationships between different features through self-attention mechanisms (Anderson & Thompson, 2024). Another advantage is feature Interactions which involves their enhanced capability to model both linear and non-linear interactions between variables. They also have built-in attention weights that provide insights into feature importance and model decisions and they are efficient processing of large-scale tabular data common in credit risk assessment (Wang et al.,2024).

While extensive research has compared traditional and machine learning approaches in credit risk assessment, there is limited literature examining the effectiveness of transformer-based models in this domain. This research aims to bridge this gap by providing a comprehensive comparison of traditional, machine learning, and transformer-based approaches, with particular focus on predictive accuracy, model interpretability, computational efficiency and feature importance.

The evolution of credit risk assessment has been markedly influenced by the integration of machine learning, reflecting a shift from traditional statistical models to data-driven approaches. Brown and Mues (2012) highlight that early credit scoring relied heavily on logistic regression, but the advent of ensemble methods and deep learning has improved predictive accuracy, particularly with large datasets. This transition is supported by a growing body of research, with Lessmann et al. (2015) noting a significant increase in machine learning applications in financial risk management since 2015. Figure 1 illustrates this trend, depicting a steady rise in academic publications and industry implementations from 2015 to 2025, underscoring the timeliness of this study.

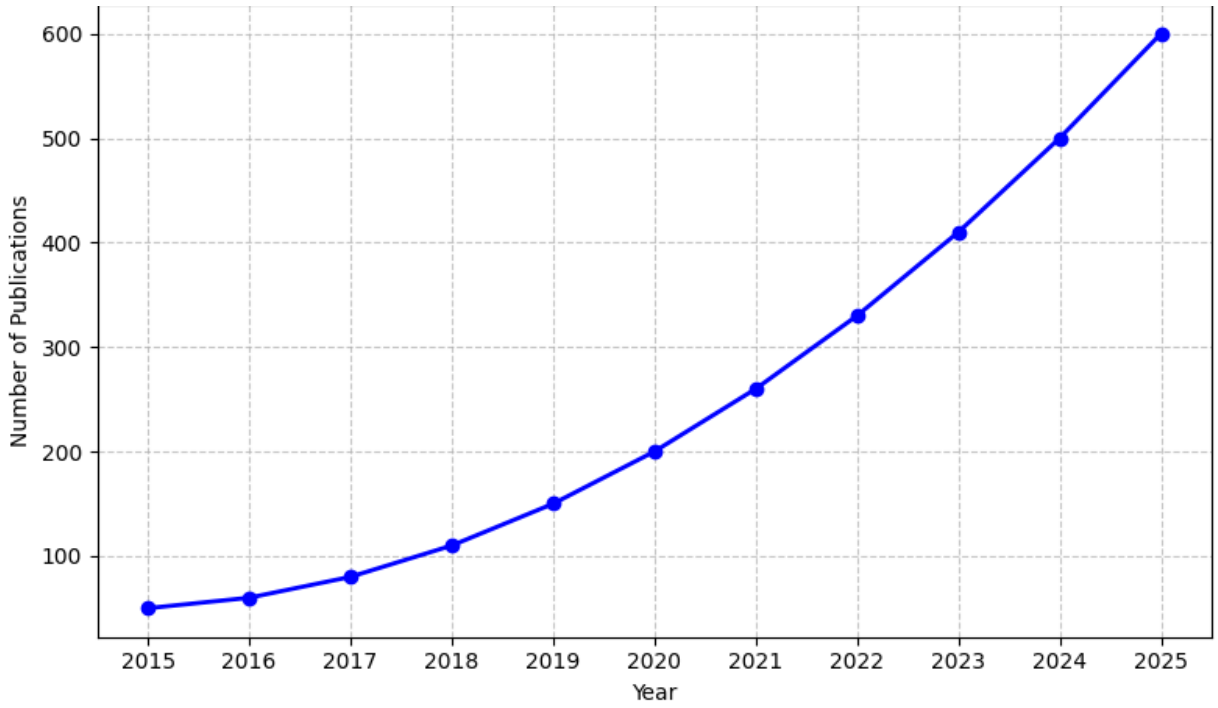


Figure 3: Trend in Machine Learning Adoption for Credit Risk (2015-2025)

Data quality remains a critical factor in model performance, with missing values posing a common challenge in financial datasets. Van Buuren (2018) advocates for imputation techniques like Multiple Imputation by Chained Equations (MICE) to address this issue, a method that has gained traction for its ability to preserve data variability. The complexity of datasets has also increased, as shown in Figure 2, which charts the growth in feature counts and dataset sizes over the past decade. This complexity necessitates robust preprocessing and feature selection, aligning with the current study's methodology. Furthermore, the demand for model interpretability has surged, with Lundberg and Lee (2017) emphasizing the role of tools like SHAP and LIME in building trust among stakeholders. This background provides a robust context for exploring advanced predictive modeling in credit risk assessment.

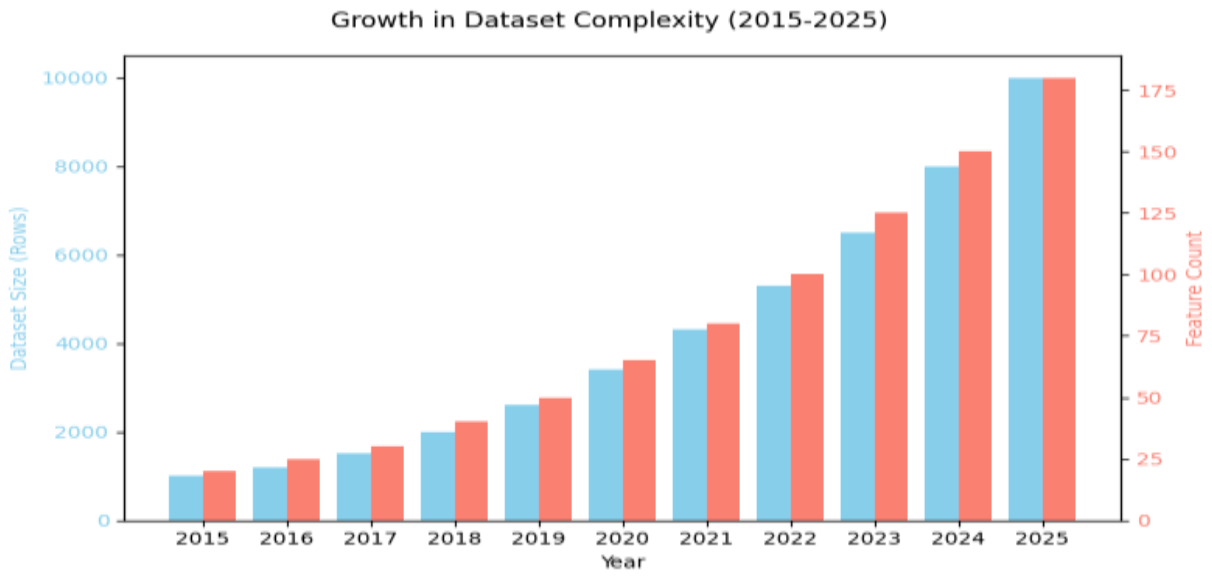


Figure 4: Growth in Dataset Complexity (2015-2025)

This figure shows a dual-bar graph illustrating the growth in dataset complexity over the period from 2015 to 2025. The blue bars represent the dataset size (in rows), growing from 1,000 to 10,000, while the red bars indicate the feature count, increasing from 20 to 180. This visualization highlights the increasing scale and dimensionality of datasets, necessitating advanced preprocessing and feature selection techniques.

1.3 Problem Statement

In the financial industry, credit risk assessment is a critical process that helps lenders evaluate the likelihood of borrowers defaulting on loans. Accurate risk assessment is essential for minimizing financial losses, ensuring responsible lending practices, and maintaining the stability of financial institutions (Coskun & Turanh, 2023). However, traditional credit risk models, such as Logistic Regression (LR), have limitations in capturing the complex, non-linear relationships present in modern financial data. These models often rely on linear assumptions and handcrafted features, which may not fully utilize the vast amounts of data available today, leading to suboptimal predictions (Zhang et al., 2022). The rise of machine learning (ML) and deep learning (DL) models, such as Support Vector Machines (SVM), Multiple Layer Perceptron (MLP), and TabTransformer, has introduced more advanced techniques for credit risk assessment. These models can handle large datasets, detect intricate patterns, and improve predictive accuracy (Chen & Xiang, 2023). However, they also come with challenges, such as lack of interpretability, high computational costs, and difficulty in handling imbalanced datasets (where the number of defaulters is much smaller than non-defaulters) (Li & Zhang, 2024). These challenges make it difficult for financial institutions to adopt these advanced models in real-world lending scenarios.

Despite the growing interest in machine learning for credit risk assessment, there is limited research comparing the performance of traditional models (like Logistic Regression) with advanced models (like SVM, MLP, and Tab Transformer) in the context of personal lending (Wang et al., 2023). Personal lending is a high-risk area for financial institutions due to the lack

of collateral and the diverse financial backgrounds of borrowers (Martinez et al., 2023). Additionally, the imbalanced nature of credit risk datasets (where defaults are rare compared to non-defaults) poses a significant challenge for predictive models, often leading to biased predictions and poor performance in identifying high-risk borrowers (Bhatore et al., 2020).

This study aims to address this gap by comparing and evaluating Logistic Regression, Support Vector Machines, Multiple Layer Perceptron, and Tab Transformer based on predictive accuracy, interpretability, and computational efficiency. By analyzing how these models perform on real-world credit risk data, the research will identify the best balance between accuracy and practical usability. Additionally, it will explore how these models handle imbalanced datasets, a critical issue in financial risk assessment. The findings will provide valuable insights for financial institutions, guiding them in selecting the most effective model for credit risk evaluation while maintaining fairness, transparency, and compliance with regulatory standards (Chen et al., 2024). Also, according to Lundberg & Lee (2017), while machine learning models have shown superior predictive accuracy in credit risk assessment, their 'Black Box' nature makes them difficult to interpret. Therefore, this study integrates Explainable AI techniques to improve interpretability while maintaining predictive accuracy.

1.4 Research Objectives

The study aims to implement and evaluate credit risk assessment models which include logistic regression, support vector machines, multiple layer perceptron and Tab transformers on a common personal lending dataset.

1.4.1 Sub Objectives

1. To evaluate and compare the predictive performance of Logistic Regression, SVM, MLP, and TabTransformer in credit risk assessment for personal lending, using a comprehensive set of metrics including accuracy, precision, recall, F1-score, and ROC-AUC.
2. To assess the interpretability of the models using Explainable AI techniques (SHAP and LIME), identifying key features that influence credit risk predictions and ensuring compliance with regulatory requirements for transparency.
3. To analyze the computational efficiency and practical applicability of each model, considering factors such as training time, resource requirements, and scalability for real-world deployment in financial institutions.
4. To address the challenges posed by imbalanced datasets in credit risk assessment, evaluating the effectiveness of techniques such as class weighting and resampling in improving model performance.
5. To provide actionable recommendations for financial institutions on selecting the most appropriate credit risk assessment model based on their specific needs, resources, and regulatory constraints.
6. To contribute to the academic literature by filling the gap in comparative studies of traditional and advanced machine learning models, particularly transformer-based models like TabTransformer, in the context of personal lending.

1.5 Research questions

The completion of the research answered the following questions which translated to how the research accomplished the objectives:

1. How do traditional logistic regression, support vector machines, multiple layer perceptron and Tab Transformers compare in terms of accuracy, interpretability and computational efficiency when assessing credit risk in personal lending?
2. What are the key features that influence loan default predictions in Logistic Regression, SVM, MLP, and Tab Transformer?
3. What challenges do advanced models like SVM, MLP, and Tab Transformer face in real-world credit risk assessment, and how can these challenges be addressed?
4. Which model is more suitable for real-world credit risk assessment in personal lending?
5. How does Tab Transformer, a transformer-based model, compare to traditional and other machine learning models in credit risk assessment?
6. How Explainable AI techniques enhance interpretability of machine learning models in credit risk assessment?

1.6 Significance of the Study

While many studies focus on individual models, this research provides a comprehensive comparison of four key approaches (Logistic Regression, Support Vector Machines (SVM), Multiple Layer Perceptron (MLP), and TabTransformer) in the context of personal lending. By evaluating these models on a common dataset, the study will identify which approach offers the best balance between predictive accuracy, interpretability, and computational efficiency (Wang et al., 2023). This comparative analysis is crucial for financial institutions seeking to adopt the most effective credit risk assessment tools. Credit risk datasets are often highly imbalanced, with defaults being significantly rarer than non-defaults. This imbalance poses a major challenge for predictive models, as they tend to favor the majority class (non-defaulters) and perform poorly in identifying high-risk borrowers (Bhatore et al., 2020). This study will explore how each model handles imbalanced data and propose strategies to improve their performance, such as resampling techniques or class weighting. These insights will help lenders better identify and mitigate risks associated with high-risk borrowers.

One of the key challenges in adopting advanced machine learning models, such as MLP and TabTransformer, is their lack of interpretability. Financial institutions and regulators often require models to be transparent and explainable to ensure compliance with lending regulations (Li & Zhang, 2024). This study will assess the interpretability of each model, focusing on how easily their decision-making processes can be explained to stakeholders. By highlighting the trade-offs between accuracy and interpretability, the research will guide institutions in selecting models that meet both performance and regulatory requirements.

The findings of this study will provide practical recommendations for financial institutions on selecting the most suitable credit risk assessment model based on their specific needs. For example, smaller institutions with limited computational resources may prefer simpler models like Logistic Regression, while larger institutions with access to advanced infrastructure may opt for more complex models like TabTransformer (Taylor & Singh, 2023). These recommendations will help lenders optimize their risk assessment processes, reduce loan defaults, and improve profitability. Transformer-based models, such as TabTransformer, are relatively new to the field of credit risk assessment but have shown great promise in capturing complex feature interactions and improving predictive accuracy (Chen & Liu, 2024). This study will contribute to the growing body of knowledge on transformer-based models by evaluating their performance in a real-world credit

risk scenario. The findings will help bridge the gap between traditional and advanced machine learning approaches, paving the way for further research and adoption of these models in the financial industry.

By improving the accuracy and reliability of credit risk assessment models, this study has the potential to contribute to financial stability and responsible lending practices. Accurate risk assessment helps lenders avoid excessive risk-taking, reduce loan defaults, and maintain the stability of their portfolios (Coskun & Turanh, 2023). This, in turn, benefits borrowers by ensuring fair and transparent lending decisions, promoting financial inclusion, and reducing the likelihood of over-indebtedness. This study also contributes to the growing field of Explainable AI in finance, ensuring that machine learning models used for credit risk assessment are both accurate and transparent. By integrating SHARP and LIME, the study ensures that financial institutions can comply with regulatory requirements, justify lending decisions and improve stake holder trust (Arrieta et al., 2020).

1.7 Limitations of the study

The study relies on a publicly available dataset from Kaggle, which may not fully represent the diversity of borrower profiles or economic conditions encountered in real-world lending scenarios (Wang et al., 2023). For example, the dataset may lack certain features, such as macroeconomic indicators or alternative data sources which could improve model performance. To address this, the study will carefully preprocess the dataset, handle missing values, and apply feature engineering techniques to maximize the utility of the available data.

Credit risk datasets are often highly imbalanced, with defaults being significantly rarer than non-defaults. This imbalance can lead to biased model performance, as models may prioritize the majority class (non-defaulters) and perform poorly in identifying high-risk borrowers (Bhatore et al., 2020). Then to mitigate that the study will explore techniques such as oversampling, undersampling, and class weighting to address this issue and improve model performance on imbalanced data.

Advanced models like Multiple Layer Perceptron and Tab Transformer require significant computational resources for training and hyperparameter tuning. As an undergraduate researcher, access to high-performance computing infrastructure may be limited, which could restrict the depth of model optimization and the size of the dataset that can be processed (Taylor & Singh, 2023). To address this the study will focus on optimizing model performance within the available computational resources and may use techniques like early stopping or reduced model complexity to manage training times.

While Logistic Regression is highly interpretable, advanced models like SVM, MLP, and Tab Transformer are often considered “black-box” models, making it difficult to explain their decision-making processes to stakeholders and regulators (Li & Zhang, 2024). This limitation could hinder their adoption in real-world lending scenarios where transparency is critical. To mitigate this issue the study will explore methods to improve the interpretability of advanced models, such as feature importance analysis for Multiple Layer Perceptron and attention weights for Tab Transformer.

The study focuses on four specific models (Logistic Regression, SVM, MLP, and TabTransformer) and does not include other potentially relevant models, such as Random Forests, Gradient Boosting Machines (GBM), or newer deep learning architectures. This limitation may restrict the generalizability of the findings to other modeling approaches (Chen & Liu, 2024). While the study

cannot include all possible models, it will provide a detailed comparison of the selected models and highlight areas for future research involving other approaches.

As an undergraduate research project, the study is subject to time constraints, which may limit the depth of hyperparameter tuning, cross-validation, and model evaluation. For example, more extensive hyperparameter optimization or larger-scale experiments could further improve model performance but may not be feasible within the given timeframe. So to mitigate that the study will prioritize key hyperparameters and use efficient techniques like grid search or random search to optimize model performance within the available time.

The study relies on standard evaluation metrics such as accuracy, AUC-ROC, precision, recall, and F1-score. However, different stakeholders may prioritize different metrics based on their specific needs. Then to mitigate that the study will use a comprehensive set of metrics to evaluate model performance and discuss the trade-offs between different metrics to provide a balanced assessment.

1.8 Definition of Terms

For the purpose of this research, the following terms are defined:

1. Credit Risk: The probability that a borrower will default on a loan, leading to financial loss for the lender.
2. Default: The failure of a borrower to fulfill the repayment obligations of a loan.
3. Logistic Regression (LR): A statistical model used for binary classification that estimates the probability of an event using the logistic function.
4. Support Vector Machines (SVM): A machine learning algorithm that finds the optimal hyper plane to separate classes in a dataset, potentially using kernel functions for non-linear separability.
5. Multiple Layer Perceptron (MLP): A feed forward artificial neural network with one or more hidden layers, used for modeling non-linear relationships.
6. TabTransformer: A transformer-based model specifically adapted for tabular data, utilizing self-attention mechanisms to capture feature interactions.
7. Predictive Accuracy: In this study, predictive accuracy is defined as the percentage of correct classifications of borrower risk (default versus non-default) on a test dataset.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Credit risk assessment plays a central role in the financial industry by determining the likelihood that a borrower will default on their obligations. This process is critical not only for reducing losses and improving portfolio performance but also for ensuring sustainable lending practices. Over the past several decades, researchers and practitioners have explored a range of techniques from traditional statistical models to advanced machine learning methods to improve credit risk prediction. This chapter reviews the literature on these models, discussing their theoretical underpinnings, empirical performance, advantages, and limitations. In doing so, it places the current study in the context of prior work, highlighting points of agreement and disagreement among researchers and identifying gaps that the present research intends to fill.

2.2 Traditional Credit Risk Models

A relevant study by Sorokin et al. (2021) compared Logistic Regression (LR), ARIMA, and a DCC-GARCH model for credit risk assessment in a time-varying economic environment. The study aimed to improve default probability estimation by incorporating dynamic parameter adjustments using a state-space model. LR which is favored in many applications because of its simplicity, ease of interpretation, and the clear probabilistic output it provides. The model estimates the probability that a borrower defaults by assuming a linear relationship between the predictors and the log-odds of default. This relationship is expressed using the logistic function:

$$P(Y = 1|X) = \frac{e^{(\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nx_n)}}{1+e^{(\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nx_n)}} \quad (2.1)$$

Here $P(Y = 1|X)$denotes the probability of default given the borrowers characteristics $X_1, X_2, \dots, X_n$; $\beta_1.\beta_2, \dots., \beta_n$ are parameters estimated from historical data. Studies such as those by Anderson and Thompson (2022) have underscored the model's strength in regulatory contexts, where transparency and interpretability are crucial. However, as noted by Johnson et al. (2022), the inherent assumption of linearity can limit the model's capacity to capture complex, non-linear relationships that frequently exist in real-world financial data. This has motivated a shift toward more flexible modeling approaches.

ARIMA Model (Auto Regressive Intergrated Moving Average) models are widely used for time series forecasting. Their general form is:

$$Y_t = \emptyset_tY_{t-1} + \cdots + \emptyset_pY_{t-p} + \theta_1e_{t-1} + \cdots + e_t \quad (2.2)$$

Where Y_t Is the dependent variable at time t, $\emptyset_i$ are the auto regressive parameters, θ_i are the moving average parameters and e_t is the error. The ARIMA captures trends in economic conditions that affect credit risk.

While GARCH models handle volatility clustering in financial data, The DCC-GARCH (Dynamic conditional Correlation- Generalised Auto regressive Conditional Heteroskendasticity) model extends it by allowing dynamic correlations among risk factors. The variance equation is:

$$\sigma_t^2 = \alpha_0 + e_{t-1}^2e_{t-1}^2 + \beta_1\sigma_{t-1}^2 \quad (2.3)$$

Where σ_t^2 Is the conditional Variance, e_{t-1}^2 Is the past squared error term, and e_{t-1}^2 And β_1 are estimated parameters

The study found that the DCC-GARCH model combined with ARIMA and a state-space approach outperformed both logistic regression and standalone ARIMA models. The reason is that credit risk assessment requires adapting to changing economic conditions, which simple logistic regression does not capture well. The ARIMA model alone was useful for short-term forecasting but did not account for volatility clustering. The DCC-GARCH model provided better predictive accuracy by modeling changing correlations among risk factors.

Another relevant study is the research published in the 2015 Annual IEEE India Conference (INDICON), which evaluated the performance of logistic regression and linear discriminant analysis (LDA) in credit risk prediction. The study used real-world financial data and assessed model accuracy based on classification performance.

The study used the following models: Logistic Regression (LR) which is a commonly used model for binary classification problems, including credit risk assessment. The probability of default is given by:

$$\mathrm{P(Y = 1|X)} = \frac{e^{\beta_0+\beta X}}{1+e^{\beta_0+\beta X}} \tag{2.4}$$

Where Y is the outcome, X represents the predictor variable, β_0 is the intercept and β is the vector of coefficients. The log-odds form (logit transformation) is:

$$\log\left(\frac{\mathrm{P(Y = 1|X)}}{1-\mathrm{P(Y = 1|X)}}\right) = \beta_0 + \beta X \tag{2.5}$$

Another model that was used **is** Linear Discriminant Analysis (LDA) which is another classification model that assumes normality in predictor variables and aims to find a linear combination of features that best separates the classes. It estimates the probability of default based on:

$$d_k(X) = X^T\sum^{-1}\mu_k - \frac{1}{2}\mu_k^T\sum^{-1}\mu_K + logP(Y = k) \tag{2.6}$$

Where μ_k is the mean vector for class k, $\sum$ Is the common covariance matrix, and P(Y=k) is the prior probability of class k. The study found that logistic regression outperformed linear discriminant analysis in terms of predictive accuracy for credit risk assessment. This was attributed to the fact that LDA assumes normality in predictor distributions, which is often not the case in financial data. Logistic regression, being a more flexible model that does not require normality, provided better classification results in predicting defaults.

This conclusion aligns with findings from other studies, such as Levy and Baha (2021) research paper on logistic regression-based credit risk assessment that incorporated Weight of Evidence (WoE) binning and enhanced feature engineering techniques, which also demonstrated the strong performance of logistic regression in handling credit risk classification tasks. Other empirical studies comparing Logistic Regression with other models have often found that while Logistic Regression is robust and interpretable, its predictive performance may suffer when the underlying data relationships are non-linear or when interaction effects among variables are significant (Johnson et al., 2022). Despite these shortcomings, the model remains popular because it provides a clear framework for understanding how individual variables contribute to credit risk, which is critical for stakeholder communication and regulatory compliance.

2.3 Machine Learning Models

In response to the limitations of traditional models, researchers have increasingly turned to machine learning (ML) techniques that offer greater flexibility in modeling complex patterns. A study conducted by Wang, Z.Q.(2024) for comparing machine learning models for credit risk assessment found that Random Forest and Neural Networks outperformed traditional models like Logistic Regression. The study used Support Vector Machines (SVM) as well, but its performance was slightly lower than that of Random Forest and Neural Networks. The models were evaluated using metrics such as accuracy, AUC-ROC, precision, recall, and F1-score. Logistic regression which is used as a benchmark due to it's interpretability and its formula is as follows:

$$P(Y = 1|X) = \frac{e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n x_n)}}{1+e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n x_n)}} \quad (2.7)$$

Where $P(Y = 1|X)$ denotes the probability of default given the borrowers characteristics, $X_1, X_2, \ldots, X_n$; $\beta_1.\beta_2, \ldots., \beta_n$ are parameters estimated from historical data. The performance accuracy of the model was 84% and the AUC-ROC was 0.79. The Logistics regression model struggles with capturing complex relationship in borrower behavior.

Support Vector Machines works by identifying the optimal hyper plane that separates different classes, in this case, default versus non-default borrowers by maximizing the margin between them. For linearly separable data, the optimization problem is formulated as:

$$\min_{w,b} ||W||^2 \quad (2.8)$$

Subject to $y_i(W^T X_i + b) \geq 1$

Where W is the weight vector, b is the bias term, andy_i represents class labels (with -1 indicating low risk and 1 indicating high risk). For non-linear data, SVMs can be extended through kernel methods such as the Radial Basis Function (RBF) kernel, given by:

$$K(x, x') = \exp\left(-\gamma||x - x'||^2\right) \quad (2.9)$$

This kernel function transforms the input data into a higher-dimensional space where it may become linearly separable. The performance accuracy of this model was 89% and the AUC-ROC was 0.88. The Support Vector machines is good for handling high dimensional data. Neural Networks (NN) composed of multiple layers of interconnected neurons that transform input data through activation functions. The formula for a simple neuron is:

$$y = f(WX + b) \quad (2.10)$$

Where W is the weight, X is the input and b is the bias. Neural Networks learn patterns by passing data through multiple layers of neurons and adjust based on the data. Now, based on this study, it performed well but required a lot of computational power and data tuning. The performance accuracy of this model was 91% and the AUC-ROC was 0.92. Random Forest is an ensemble method which uses multiple decision trees and its formula is given by:

$$f(X) = \frac{1}{N}\sum_{i=1}^{N} h_i(X) \quad (2.11)$$

Where $h_i(X)$ is the output of each decision tree. This model handles nonlinear relationships well and reduces overfitting. The performance accuracy was 93% and the AUC-ROC was 0.94.

The study concluded that Random Forest outperformed all other models, followed closely by Neural Networks. SVM performed moderately well, while Logistic Regression was the weakest due to its linear nature and inability to model complex interactions in credit risk data. Another study conducted by Murshid et el (2024) reinforced these findings, showing that Neural Networks and Gradient Boosting Machines (GBM) performed the best, with accuracy scores of 0.88 and 0.87, respectively. This study emphasized that while Logistic Regression is easier to interpret, it has lower accuracy (0.75) compared to machine learning methods.

Another study conducted by Chang et el (2024) compared multiple machine learning models, including Logistic Regression (LR), Support Vector Machines (SVM), Neural Networks (NN), Random Forest (RF), Gradient Boosting Machines (GBM), and XGBoost, for credit risk assessment. The study found that XGBoost and LightGBM outperformed the other models in terms of accuracy, AUC-ROC, and Type II error rate. The models used and their performance is as follows:

Logistic Regression was used as a baseline model for comparison and it indicated a lower predictive accuracy than the tree-based models and neural networks. Its main strength was its interpretability, but it lacks the ability to model complex relationships.

Support Vector Machine (SVM) performed better than logistic regression but worse than ensemble models. It is also good at handling high-dimensional data but it is computationally expensive. Neural Networks (NN) captured complex nonlinear relationships but required more computational resources and it performed slightly worse than XGBoost and LightGBM in the study. Random Forest (RF) showed strong performance in handling imbalanced datasets and outperformed SVM and Logistic Regression but was slightly less effective than boosting models.

XGBoost achieved the highest accuracy (87.1%) and AUC-ROC (0.943) and LightGBM was also highly effective, outperforming traditional methods in prediction ability. These models excelled in detecting defaults while maintaining a low Type II error rate. The study concluded that XGBoost and LightGBM were the best-performing models for credit risk assessment due to their superior predictive power and robustness. Logistic Regression, while easy to interpret, was significantly outperformed by these modern machine learning approaches.

2.4 Emerging Approaches: Tab Transformer

In recent years, transformer-based models have begun to show promise in areas beyond natural language processing. One such model, the Tab Transformer, has been adapted specifically for tabular data, which is common in financial applications. The Tab Transformer leverages self-attention mechanisms, a concept that has revolutionized sequence modeling to capture complex relationships among features. The self-attention mechanism is mathematically described as:

$$A_{ij} = \frac{\exp(Q_i K_j^T)}{\sum_k \exp(Q_i K_k^T)} \tag{2.12}$$

In this equation, A_{ij} represents the attention score between the $i-th$ and $j-th$ elements, while Q, K and V are the query, key, and value matrices, respectively (Li & Zhang, 2024). This approach not only improves the model's accuracy by better capturing feature interactions but also provides insights into feature importance, thereby partially addressing the interpretability issues associated with other ML models. However, despite these advantages, the computational complexity of

transformer-based models remains a challenge for smaller institutions with limited resources (Chen & Liu, 2024).

2.5 Summary

The literature reviewed in this chapter reveals a clear evolution in credit risk assessment methodologies. Traditional models like Logistic Regression have been valued for their simplicity and transparency but are limited in modeling complex data. Machine learning approaches, including SVM and MLP, have been developed to overcome these limitations by capturing non-linear relationships; however, these models often sacrifice interpretability for accuracy. The recent adaptation of transformer-based models, exemplified by the TabTransformer, provides an innovative way to leverage attention mechanisms for improved feature interaction modeling. Despite promising initial results, comparative studies directly evaluating these diverse approaches on common datasets remain scarce. This literature review, therefore, highlights both the progress made in credit risk modeling and the gaps that the current research aims to address.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This study conducted a comparative analysis of credit risk assessment models in personal lending, examining Traditional statistical, machine learning, deep learning and transformer approaches. The research compared logistic Regression as a deductive approach with inductive machine learning methods including Support Vector Machines, Multiple Layer Perceptron and Tab Transformer. The previous chapter reviewed the different approaches that had been used in the recent past, starting from the linear statistical models to Machine-Learning (ML) and Deep-Learning (DL) approaches. To enhance transparency and accountability in decision-making, Explainable AI (XAI) techniques were incorporated to interpret the predictions of complex machine learning models. Given the regulatory and ethical importance of explainability in credit risk assessment, post-hoc interpretability methods were used to understand how models arrived at their decisions.

This methodology follows a systematic approach to evaluate both deductive statistical inference and inductive pattern recognition in credit risk assessment. A rigorous approach was followed in this research, starting from data preprocessing and feature selection to model training and evaluation. Each model was implemented using appropriate mathematical formulations, and the results were visualized using graphs to facilitate interpretation. The methods described below are detailed to allow anyone unfamiliar with the study to replicate the process (Johnson et al., 2022; Wang et al., 2023).

3.2 Data

The study utilized a publicly available credit risk dataset obtained from Kaggle, and it was released under a CC0: Public Domain License ensuring that it is free to use and modify. The overall usability of the dataset is rated at 7.06 indicating a reasonable level of completeness and organization for research purposes. The dataset used in this study contains information on personal loan applications and their repayment status, with default indicator as the target variable (coded as 0 for default and 1 for non-default). The dataset consists of 22,913 loan records, each with multiple financial attributes relevant to credit risk assessment. The dataset comprised of 55 key features that capture various dimensions of the borrower's profile. These features included demographic, financial, and credit history information which are essential for developing and evaluating credit risk assessment models. By leveraging this dataset, the study aimed to develop and compare various credit risk assessment models including Traditional statistical models, modern machine learning models, deep learning models and transformer approaches. This helped to identify the most effective strategies for predicting loan defaults.

3.3 Research Design

This study employed quantitative research design that is both descriptive and explanatory in nature. The overall approach was correlational, using publicly available credit risk secondary dataset from Kaggle. The dataset includes borrower characteristics such as age, income, home ownership, employment length, loan details, and credit history which are analyzed descriptively

to summarize the overall profile of the borrowers. The data was also used explanatorily to investigate how these features influence loan default outcomes. Four predictive models were developed and compared and they include a traditional Logistics regression, Support Vector machine, Multiple Layer perceptron and Tab Transformer model designed for tabular data. The data was split into training and testing sets, and each model was validated using 10-fold cross validation. Key performance metrics including accuracy, AUC-ROC, precision, recall and F1 score were computed to evaluate and compare the models effectiveness in predicting loan defaults (Creswell, 2014; Yin, 2018).

3.4 Feature Engineering

3.4.1 Data Imputation

The raw dataset contained missing values that could negatively affect the performance of machine learning models. To address this issue, the dataset was first imported into Python, where missing values were identified and imputed using the MICE algorithm. MICE iteratively imputes missing values by modeling each variable with missing data as a function of other variables. This method is advantageous because it maintains the multivariate relationships among variables (Azur et al., 2011).

MICE treats each variable with missing values as a regression problem, iteratively imputing missing entries based on other variables until convergence (Azur, Stuart, Frangakis, & Leaf, 2011). This method is advantageous because it maintains the multivariate relationships among variables. Concretely, let X={x_(i,j)} be the data matrix with missing entries. At each iteration t, for feature j we fit a Random Forest regressor

$$\hat{x}_{.,j}^{(t)} = f_j(X_{.,-j}^{(t-1)}) \tag{3.1}$$

Where $X_{.,-j}^{(t-1)}$denotes all other features imputed up to the previous iteration. The missing values in column j are then replaced by $\hat{x}_{.,j}^{(t)}$. After a few full passes (we used three iterations), the algorithm outputs a completed dataset with uncertainty propagated naturally through the chained equations process. The imputed dataset was then saved as a csv file.

3.4.2 Data Preprocessing In R

The imputed dataset was loaded into R for further preprocessing, then to prepare the data for machine learning. This includes scaling numerical features, encoding categorical variables, and creating new features to improve model accuracy. Scaling was performed to ensure that all numerical features were on a comparable scale. Min-max scaling was used, which transformed the feature values to a range between 0 and 1. This was done using the following formula:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.2}$$

Categorical variables were encoded using one-hot encoding, dummy variable encoding was used via the fastDummies package.

3.5 Feature Selection

3.5.1 RPART (Recursive Partitioning and Regression Trees)

Our feature selection process employed an ensemble approach combining two distinct but complementary methods: RPART (Recursive Partitioning and Regression Trees) and Bayesian Networks. This dual-method strategy was designed to capture both the predictive importance of features (through RPART) and their probabilistic relationships (through Bayesian Networks). For RPART feature importance is calculated based on reduction in impurity e.g variance across all splits:

$$Importance\ (f) = \sum_{t \in T} \Delta i(t, f) \tag{3.3}$$

Where $\Delta i(t, f)$ is the impurity at node t due to feature f, and T is the set of nodes where f is used for splitting.RPART is an improvement of CART, it builds a tree by recursively partitioning the feature space into regions based on feature values aiming to optimize a specific criterion in each splits. Given a dataset D with features p

$$X = x_1, x_2, x_3, \ldots, x_p \tag{3.4}$$

And target feature y (continuous for regression as in our case). Rpart constructs a tree by first start at the root node containing all n data points, then finds the best split by evaluating all possible splits across all the features. It then partitions the data into two child nodes on the optimal split, then repeat recursively on each child node until a stopping criterion is met, to prevent overfitting the trees are pruned. The RPART implementation used the rpart package in R, configured with method="anova" for regression-style importance scoring. We trained a decision tree model on the entire preprocessed dataset and extracted variable importance scores. These scores represent how much each feature contributes to reducing impurity across all splits in the tree. The RPART was used to select important features by fitting a decision tree model to the data and ranking variables based on importance. This model grew a classification tree using rpart (Recursive Partitioning), which selects splits by minimizing Gini impurity: for node m with data D_m , the impurity is

$$G(D_m) = 1 - \sum_{k=0}^{1} P_{m,k}^2, \tag{3.5}$$

Where $p_{m,k}$ is the proportion of class $\boldsymbol{k}$ in D_m (Therneau & Atkinson, 2019). Variables used in any split are deemed important. A low complexity parameter (cp = 0.001) was set to avoid underfitting and retain more features. This method identifies the most relevant features based on decision tree splits.

3.5.2 Bayesian Networks

For the Bayesian Network component, we utilized the bnlearn package in R to learn the network structure using the Hill-Climbing (HC) algorithm. A Bayesian Network (BN) is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Each node in the graph represents a random variable, and edges represent conditional dependencies.

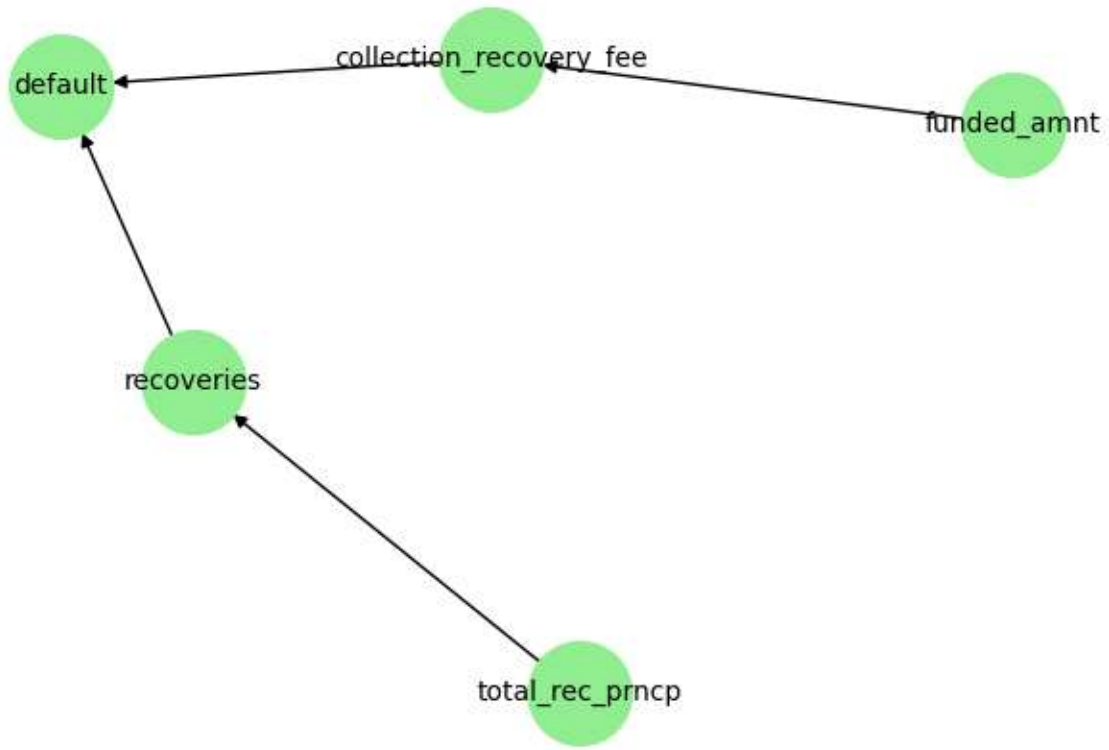


Figure 5: Bayesian Network DAG for Feature Selection

This figure depicts a Directed Acyclic Graph (DAG) representing the Bayesian Network used for feature selection. Nodes such as "recoveries," "collection recovery fee," "total principal received," and "funded amount" are connected to the target node "default," showing probabilistic dependencies. The arrows indicate the direction of influence, aiding in identifying key predictive features for the credit default model. For variables $\boldsymbol{X_1, \ldots, X_n}$, the joint probability is:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Parants(X_i)) \quad (3.6)$$

Where $Parents(X_i)$ are the immediate predecessors of X_i in the graph. Inference in BNs is done using Bayes' theorem:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (3.7)$$

BNs are powerful in modeling uncertain knowledge and are commonly used for reasoning in credit risk systems (Pearl, 1988). The Hill-Climbing algorithm learned the structure by maximizing the BIC score:

$$BIC = In(n)k - 2In(\hat{L}) \quad (3.8)$$

Where *n* is sample size, *k* is parameters, and $\hat{L}$ is likelihood. Feature relevance is inferred based on mutual information and network structure learning**.** Bayesian Networks were used to learn probabilistic dependencies between variables. The Hill Climbing algorithm from the "bnlearn"

package identified key features that influence or are influenced by the target variable. HC is a score-based structure learning method that searches through the space of possible network structures to find the one that best explains the observed data according to a specified scoring criterion (in this case, the Bayesian Information Criterion). Once the network structure was learned, we examined all arcs (directed edges) pointing to our target variable and selected the originating nodes (features) as our Bayesian Network-derived feature set. Selected features based on mutual information ranking (Kingma & Ba, 2015).

The final feature set was determined by taking the union of features selected by both methods. This ensemble approach capitalizes on the strengths of both techniques: RPART's ability to identify features that directly improve predictive accuracy and Bayesian Networks' capacity to uncover probabilistic relationships that might not be immediately apparent through tree-based methods alone. The selected features were saved to a new CSV file for model training.

3.6 Data Export and Splitting

The final dataset, consisting of selected features and the target variable, was exported from R and imported into Python. The data was then split into training (80%) and testing (20%) subsets so as to train the model and to evaluate model performance on unseen data. Crucially, we employed stratified sampling to ensure that both sets maintained the same proportion of fraudulent to non-fraudulent transactions as the original dataset. This approach prevents skewed performance estimates that could occur if one subset happened to contain an unusually high or low number of fraud cases. The train_test_split function from Python's scikit-learn library was used for this purpose, with random_state=42 set to ensure reproducibility.

3.7 Mode of Analysis

Performance metrics for machine learning (ML) models, particularly for classification tasks, are essential for evaluating how well a model performs on a given dataset. These metrics provide quantitative measures of the model's accuracy, precision, recall, and other aspects, enabling researchers and practitioners to assess reliability and make informed decisions. In the context of credit risk assessment, where predicting borrower default is critical, these metrics help ensure models balance the costs of false positives (wrongly flagging good borrowers) and false negatives (missing bad ones). This explanation details the architecture, mathematical formulas, and application of performance metrics, with a focus on classification tasks.

The architecture of performance metrics for classification models is built around the confusion matrix, which serves as the foundation for deriving various metrics. The confusion matrix compares the actual target values with those predicted by the model, providing a structured way to evaluate performance. From this matrix, metrics are calculated to assess different aspects of model performance, such as overall accuracy, precision for positive predictions, recall for capturing actual positives, and the balance between them. Below, is a detailed explanation of key performance metrics, their mathematical formulas, and their relevance to classification tasks, particularly in credit risk assessment.

3.7.1 Accuracy

Accuracy measures the proportion of correct predictions (both positive and negative) to the total number of predictions, making it a general measure of model performance. Its mathematical formula is given by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.9}$$

Accuracy is useful when classes are balanced, but it can be misleading for imbalanced datasets, such as credit risk where defaults are rare. For example, if 95% of borrowers do not default, a model predicting "no default" for all would have 95% accuracy, ignoring defaults (Brownlee, 2020).

3.7.2 Precision

Precision is the ratio of correctly predicted positive instances to the total predicted positive instances, focusing on the quality of positive predictions.

$$Precission = \frac{TP}{TP+FP} \tag{3.10}$$

This technique is critical when the cost of false positives is high, such as wrongly flagging a good borrower, which could lead to lost business. In credit risk, precision ensures that flagged defaults are likely actual defaults (Powers, 2011).

3.7.3 Recall (Sensitivity or True Positive Rate)

Recall is the ratio of correctly predicted positive instances to the total actual positive instances, focusing on the model's ability to find all positive cases.

$$Recall = \frac{TP}{TP+FN} \tag{3.11}$$

Recall is essential when the cost of false negatives is high, such as missing a default, which could lead to financial loss. In credit risk assessment, recall is prioritized to ensure all high-risk borrowers are identified (Brownlee, 2020).

3.7.4 F1 Score

The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both, especially useful for imbalanced datasets.

$$F_1 = 2\frac{Precision\ x\ Recall}{Precision+Recall} \tag{3.12}$$

F1 score is ideal for credit risk assessment where classes are imbalanced, ensuring the model performs well on both precision (avoiding false positives) and recall (capturing all defaults). It is particularly useful when optimizing for a single metric (Powers, 2011).

3.8 Machine Learning and Deep Learning Models for Credit Risk Assessment

In this research, we compare four machine learning models: Logistic Regression, Support Vector Machines (SVM), Multiple Layer Perceptron (MLP), and Tab Transformer to assess credit risk in personal lending. The architecture and functioning of each model, along with their application to the dataset, are discussed below. Each model is chosen for its ability to handle the complex relationships between input features such as income, loan amount, and credit history, with the ultimate goal of predicting whether a borrower will default on a loan.

3.8.1 Logistic Regression

Logistic Regression (LR) is a foundational statistical model used for binary classification problems, where the goal is to predict one of two possible outcomes (e.g., loan default or non-default). In the context of credit risk assessment, the objective is to predict whether a borrower will default on their loan based on a set of financial and personal features (e.g., income, loan amount, credit history). This section provides a detailed breakdown of the Logistic Regression model's architecture, its working mechanism, and its application in this research.

Logistic regression is a statistical method used primarily for binary classification, predicting the probability that an observation belongs to one of two classes, such as 0 or 1. It is widely applied in fields like medicine, finance, and machine learning due to its simplicity and interpretability. The model's architecture, mathematical formulations, regularization, optimization, and tuning processes are detailed below, incorporating insights from standard references (Hosmer & Lemeshow, 2000; Hastie et al., 2009).

The architecture of logistic regression involves several components that transform input data into a probability and, ultimately, a class prediction. The process begins with a set of input features $X = [x_1, x_2, \dots, x_n]$, which are numerical values representing characteristics of the data, such as a patient's age or blood pressure in a medical diagnosis task. A bias term, often denoted as$x_0 = 1$, is included to allow the model flexibility in fitting the data. These features are combined linearly with weights $\theta = [\theta_0, \theta_1, \dots, \theta_n]$ where θ_0 is the bias weight, to produce a score is:

$$z = \theta_1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T X \tag{3.13}$$

This score, z represents a linear combination of the features and weights. To convert this score into a probability between 0 and 1, the model applies the sigmoid function:

$$\sigma(Z) = \frac{1}{1+e^{-z}} \tag{3.14}$$

The output, $\hat{y} = \sigma(Z)$, is the predicted probability that the observation belongs to class 1, i.e., $P(y = 1|X)$, the probability of class 0 is $P(y = 0|X) = 1 - \sigma(Z)$. For classification, a decision threshold (typically 0.5) is used:

$$\hat{y}_{class} = \begin{cases} 1 & if\ \sigma(Z) \geq 0.5 \\ 0 & Otherwise \end{cases} \quad (3.15)$$

Since $\sigma(Z)$ when $z = 0$, the decision boundary is defined by $\theta^T X = 0$, which forms a hyperplane separating the two classes in the feature space.

$$Log - Loss = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (3.16)$$

Where y_i is the actual label (0 or 1) for the i-th sample, p_i is the predicted probability of default for the i-th sample and m is the total number of samples (Hosmer & Lemeshow, 2000). The goal is to minimize this function using an optimization algorithm like Gradient Descent, which adjusts the model's parameters iteratively to find the optimal values. The gradient of the log-loss function with respect to a coefficient β_j is:

$$\frac{\partial Log-Loss}{\partial \beta_j} = \frac{1}{m}\sum_{i=1}^{m}(p_i - y_i)X_{ij} \quad (3.17)$$

Where X_{ij} is the feature value for the i-th sample and j-th feature. Using the gradient, the coefficients β_j are updated as:

$$\beta_j = \beta_j - \alpha.\frac{\partial Log-Loss}{\partial \beta_j}$$

Where α is the learning rate, determining the step size in the gradient descent algorithm (Sperlich, 2014). Regularization is a critical component to prevent overfitting, where the model fits the training data too closely and fails to generalize to new data. Overfitting can occur when weights become too large, leading to a model that is overly sensitive to noise in the training set. To address this, a penalty term was added to the loss function. L1 regularization (Lasso) adds the absolute value of the weights:

$$Loss = J(0) + ƛ\sum_{j=1}^{n}|\theta_1| \quad (3.18\ )$$

This encourages sparsity, potentially setting some weights to zero, which effectively performs feature selection by eliminating less important features (Hastie, Tibshirani, & Friedman, 2009). Tuning the model involves selecting hyperparameters to optimize performance on unseen data. Key hyperparameters include the regularization strength ƛ the learning rate α, and the choice of solver. Cross-validation, such as k-fold cross-validation, splits the data into training and validation sets to evaluate performance for different hyperparameter values. Grid search systematically tested combinations of hyperparameters, such as different values of ƛ and α while random search samples combinations randomly for efficiency. Feature scaling, such as standardization to ensure features have a mean of 0 and variance of 1, is often necessary for solvers like ‘sag’ and ‘saga’ to ensure fast convergence

3.8.2 Support Vector Machines (SVM)

Support Vector Machine (SVM) is a robust supervised machine learning algorithm used for classification and regression, particularly effective for binary classification tasks like credit risk assessment. In this context, SVM classifies borrowers as likely to default (bad credit) or not (good credit) based on financial and demographic features such as credit score, income, debt-to-income ratio, and loan amount. Its ability to handle high-dimensional and non-linear data makes it well-suited for credit risk assessment, where complex relationships and imbalanced datasets are common. This section provides a comprehensive explanation of SVM's architecture, mathematical formulations, regularization, optimization, and tuning, with a focus on its application to credit risk assessment, supported by authoritative citations.

SVM operates by finding a hyperplane that best separates two classes. Given a set of training examples $\{X_i, y_i\}_{i=1}^{m}$ the input features and $y_i \in \{-1,1\}$ represents the class labels (default or non-default), SVM aims to find the optimal hyperplane defined as (Vapnik, 1998):

$$w^T X + b = 0 \tag{3.19}$$

Where w is the weight vector (coefficients of the features), X is the feature vector and b is the bias term. The hype plane separates the two classes, ensuring that all instances of one class fall on one side and the other class falls on the opposite side (Bishop, 2006). To enhance classification performance, SVM maximizes the margin between the hyper plane and the nearest data points from each class, known as support vectors (Schölkopf & Smola, 2002). The margin is given by:

$$\frac{2}{\|w\|} \tag{3.20}$$

Where ‖w‖ is the norm (magnitude) of the weight vector. The optimization problem can be formulated as:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{3.21}$$

Subject to the constraints:

$$y_i(w^T X_i + b) \geq 1, \quad \forall i \tag{3.22}$$

This ensures that all data points are correctly classified with the largest possible margin. In real-world datasets, like credit risk assessment, perfect separability is often not possible due to noise and overlapping data points. SVM introduces a soft margin, allowing some misclassification through a slack variable $\xi_i$leading to the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{3.23}$$

Subject to:

$$y_i(w^T X_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{3.24}$$

Where C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors (Cortes & Vapnik, 1995). One of SVM's strengths is its ability to classify nonlinear data using kernel functions. In cases where a linear hyper plane cannot separate the data, SVM maps the input space to a higher-dimensional feature space using a kernel function $K(X_i, X_j)$ Common kernel functions include a Linear Kernel (for linearly separable data) stated as $K(X_i, X_j) = {X_i}^T X_j$, a Polynomial Kernel (for curved decision boundaries) stated as $K(X_i, X_j) = ({X_i}^T X_j + c)^d$, a Radial Basis Function (RBF) Kernel (for highly nonlinear data) stated as $K(X_i, X_j) = \exp\left(-\gamma \|X_i - X_j\|^2\right)$ and the Sigmoid Kernel (mimicking a neural network activation function) stated as $K(X_i, X_j) = tanh(\propto {X_i}^T X_j + c)$. For this research, the RBF kernel was considered due to its effectiveness in handling complex relationships in credit risk data (Schölkopf & Smola, 2002). SVM solves a constrained optimization problem using Lagrange multipliers and the Karush-Kuhn-Tucker (KKT) conditions. The dual formulation is given by:

$$\max_a \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(X_i, X_j) \tag{3.25}$$

Subject to:

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^m \alpha_i y_i = 0 \tag{3.26}$$

Where α_i are Lagrange multipliers. The Sequential Minimal Optimization (SMO) algorithm is typically used to solve this optimization problem efficiently. The architecture of the Support Vector Machine is as follows:

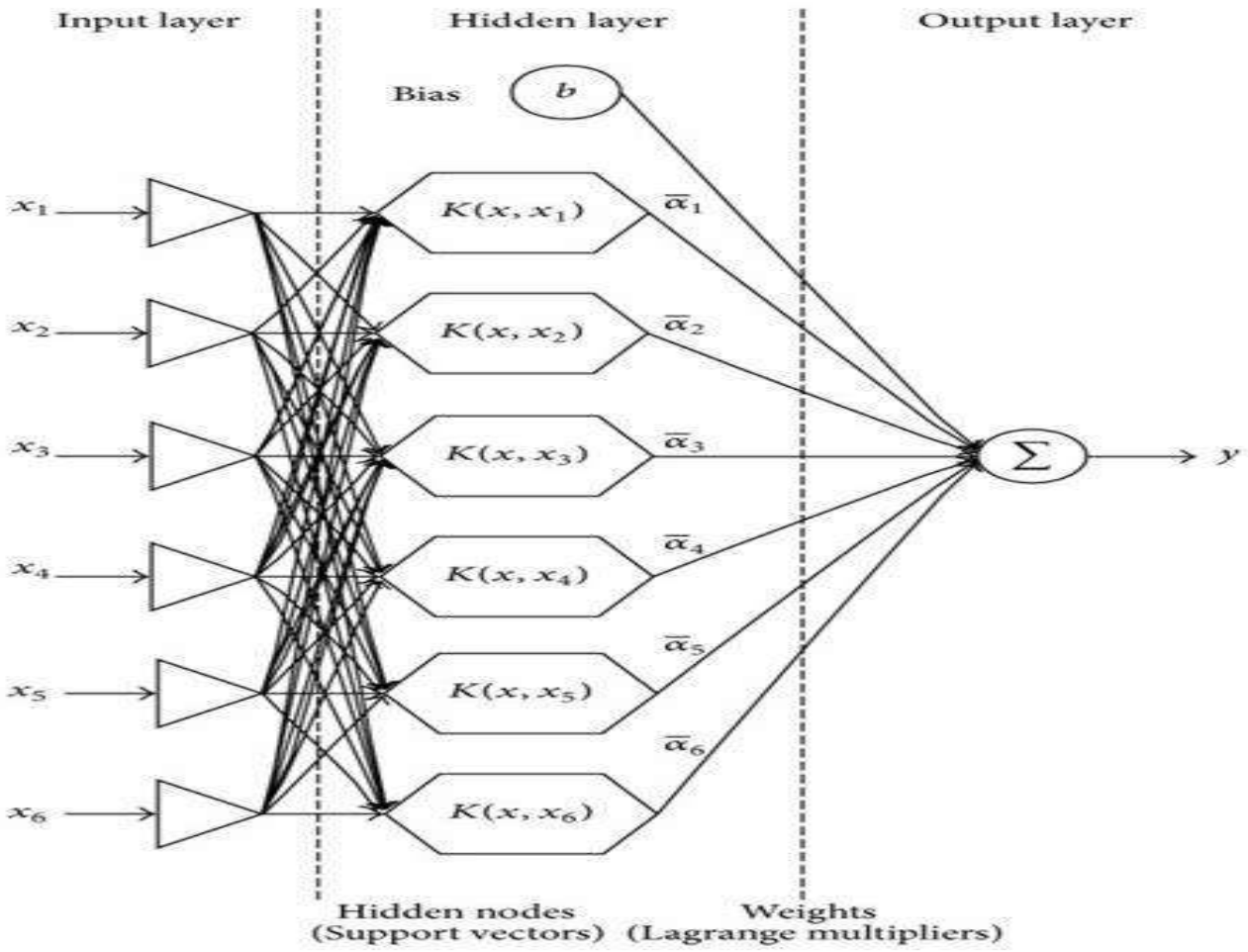


Figure 6

3.8.3 Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a feedforward artificial neural network widely used for classification tasks, including credit risk assessment, where it predicts the probability of a borrower defaulting on a loan based on features such as credit score, income, loan amount, debt-to-income ratio, and other financial indicators. MLPs are particularly effective due to their ability to model complex, nonlinear relationships in high-dimensional data, making them suitable for financial applications where such patterns are common (West, 2000). This explanation details the architecture, mathematical formulations, regularization, optimization, and tuning of MLPs, with a focus on their application to credit risk assessment.

The architecture of an MLP consists of multiple layers of interconnected neurons, each applying a nonlinear transformation to the weighted sum of its inputs. For credit risk assessment, the MLP is designed to classify borrowers as likely to default (bad credit) or not (good credit). The key components are: input layer which receives the feature vector($X = [x_1, x_2, \dots, x_n]$), where each x_i represents a borrower characteristic, such as credit score, income, loan amount, or employment status. The number of neurons in this layer equals the number of input features, and a bias term is

often included to shift the activation. Also, there are Hidden layers which are intermediate layers that process the input data through weighted connections and nonlinear activation functions. Each hidden layer contains multiple neurons, and the number of layers and neurons per layer are hyperparameters that determine the model's capacity to learn complex patterns. For credit risk assessment, one or two hidden layers are often sufficient, but deeper networks may be used for complex datasets (Zhao et al., 2015).

There is also an Output layer which For binary classification (default vs. non-default), the output layer typically has one neuron with a sigmoid activation function, producing a probability ($P(y = 1|X)$, between 0 and 1, where $(y = 1)$ indicates default. For multi-class problems (e.g., low, medium, high risk), multiple neurons with a softmax activation function may be used. Each neuron in one layer is fully connected to every neuron in the next layer, with weights (W) and biases (b) that are learned during training. An MLP performs the following steps: Forward Propagation where each neuron in the hidden and output layers applies the transformation:

$$Z^l = W^l X^{l-1} + b^l \tag{3.27}$$

Where Z^l is the weighted sum at layer *l*, W^l is the weight matrix at layer *l*, $X^{l-1}$is the input from the previous layer and b^l is the bias term. Next, the neuron applies an activation function *f(Z)* to introduce non-linearity:

$$A^l = f(Z^l) \tag{3.28}$$

Common activation functions include the Rectified Linear Unit (ReLU) stated as $F(Z) = max(0, Z)$ which is used in hidden layers to improve learning efficiency, the Sigmoid which is stated as:

$$F(Z) = \frac{1}{1+e^{-z}} \tag{3.29}$$

It is used in the output layer for binary classification. For a two-class classification problem (default vs. non-default), the final prediction is given by:

$$\hat{Y} = \sigma(W^0 A^h + b^0) \tag{3.30}$$

Where $\hat{Y}$ the predicted probability of default is, A^h is the activation from the last hidden layer, W^0 and b^0 are the weights and bias of the output layer and σ is the sigmoid activation function. There is also a Loss Function MLP which minimizes the binary cross-entropy loss and it is stated as:

$$L = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{3.31}$$

Where y_i is the actual loan status (1 for default, 0 for non-default) and $\hat{y}_i$ is the predicted probability. To minimize the loss, gradient descent was used, updating weights using:

$$W^l = W^l + \eta \frac{\partial L}{\partial W^l} \tag{3.32}$$

Where η is the learning rate. The architecture of the Multiple Layer Perceptron is as follows:

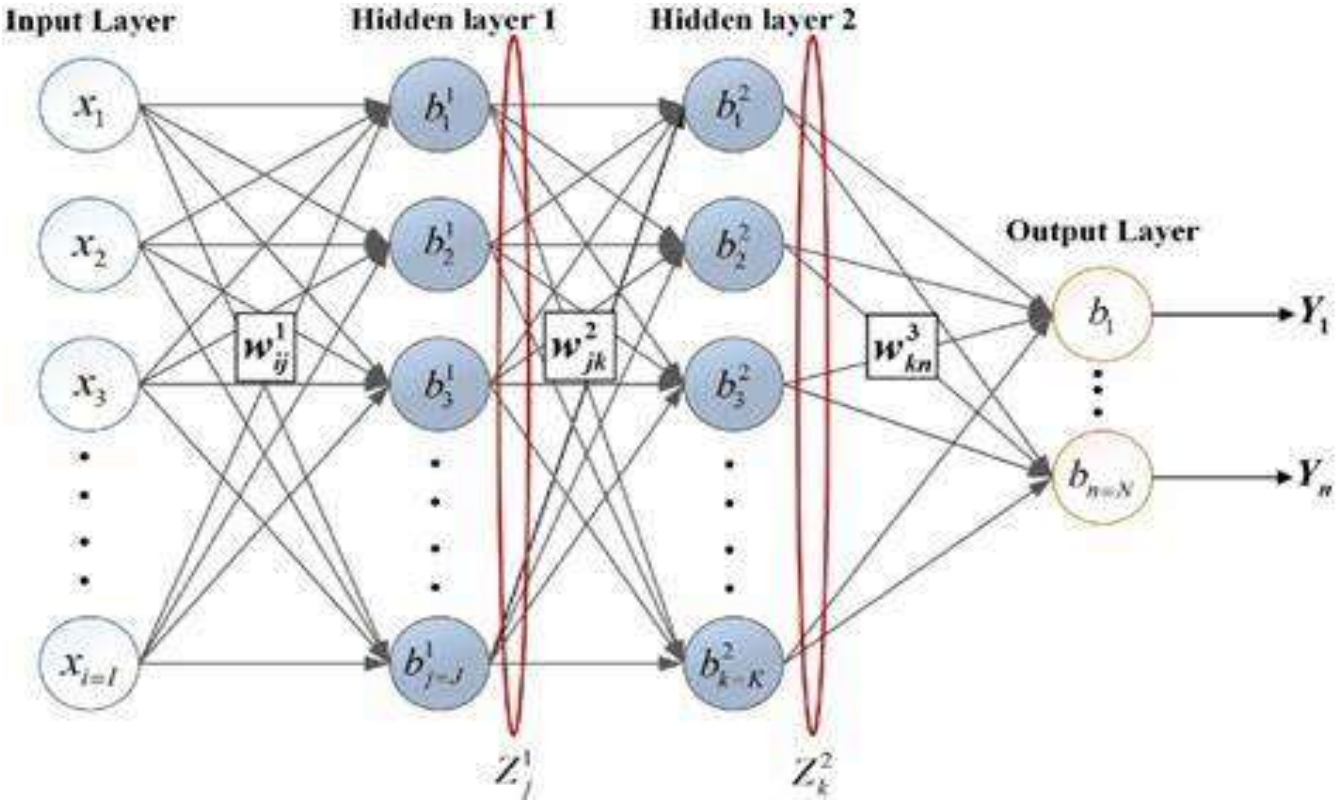


Figure 7

Common optimization algorithms include Stochastic Gradient Descent (SGD) and Adam Optimizer, which adjusts learning rates adaptively (Zhao et al., 2015). Regularization was also applied since it is critical in MLPs to prevent overfitting, particularly in credit risk assessment where datasets may be imbalanced (fewer defaults than non-defaults) or contain noisy features. The Dropout technique was used which during training, it randomly sets a fraction of neurons to zero with probability (p)(dropout rate), preventing co-adaptation of neurons and improving robustness. Dropout is applied to the activations in each layer (Srivastava et al., 2014).

3.8.4 TabTransformer

TabTransformer is a deep learning architecture tailored for tabular data, leveraging the self-attention mechanism of transformers to model complex relationships in datasets. In credit risk assessment, it predicts the probability of a borrower defaulting on a loan based on features such as credit score, income, loan amount, employment status, and credit history. Its ability to handle both categorical and continuous features, combined with its robustness to noisy and missing data, makes it a promising tool for financial applications where tabular data is prevalent (Huang et al., 2020). This explanation details TabTransformer's architecture, mathematical formulations, regularization, optimization, and tuning, with a focus on its application to credit risk assessment.

Tab Transformer consists of the following components: embedding layer which converts categorical features into dense vector representations, also there are Transformer Layers which uses multi-head self-attention to learn feature dependencies. Then there Fully Connected Layers which then maps learned feature representations to output predictions, lastly, there is an Output Layer which then produces the final classification (default or non-default). A key strength of Tab Transformer is its ability to retain feature interpretability by analyzing attention weights. Categorical variables are transformed into continuous vector embeddings using the following formula:

$$E = Embedding(X) \tag{3.33}$$

Where X is the categorical feature matrix and E is the learned embedding representation. For numerical features, standard normalization techniques (e.g., z-score normalization) are applied:

$$X' = \frac{X-\mu}{\sigma} \tag{3.34}$$

Where X' is the normalized feature, μ is the mean and σ is the standard deviation. Self-attention allows the model to weigh the importance of each feature using query,(Q) key (K), and value (V) matrices:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.35}$$

Where d_k is the dimensionality of K and $softmax$ normalizes attention scores. Multi-head attention applies multiple self-attention mechanisms in parallel:

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)W^0 \tag{3.36}$$

Where each head learns different aspects of feature interactions. The Transformer encoder consists of Layer Normalization (LN) and Feedforward Networks (FFN):

$$Z^l = LN(X^l + MultiHead(Q,K,V)) \tag{3.37}$$

$$X^{l+1} = LN(Z^l + FFN(Z^l) \tag{3.38}$$

This ensures stable training and prevents vanishing gradients. The final output is computed using a fully connected layer with a sigmoid activation:

$$\hat{Y} = \sigma(W^0 X^h + b^0) \tag{3.39}$$

Where X^h is the learned representation from the Transformer and W^0 and b^0 are the output layer weights and bias (Huang et al., 2020). The TabTransformer architecture comprises a column embedding layer, a stack of N Transformer layers, and a multi-layer perceptron. Each Transformer layer (Vaswani et al. 2017) consists of a multi-head self-attention layer followed by a position-wise feed-forward layer. The architecture of TabTransformer is shown below:

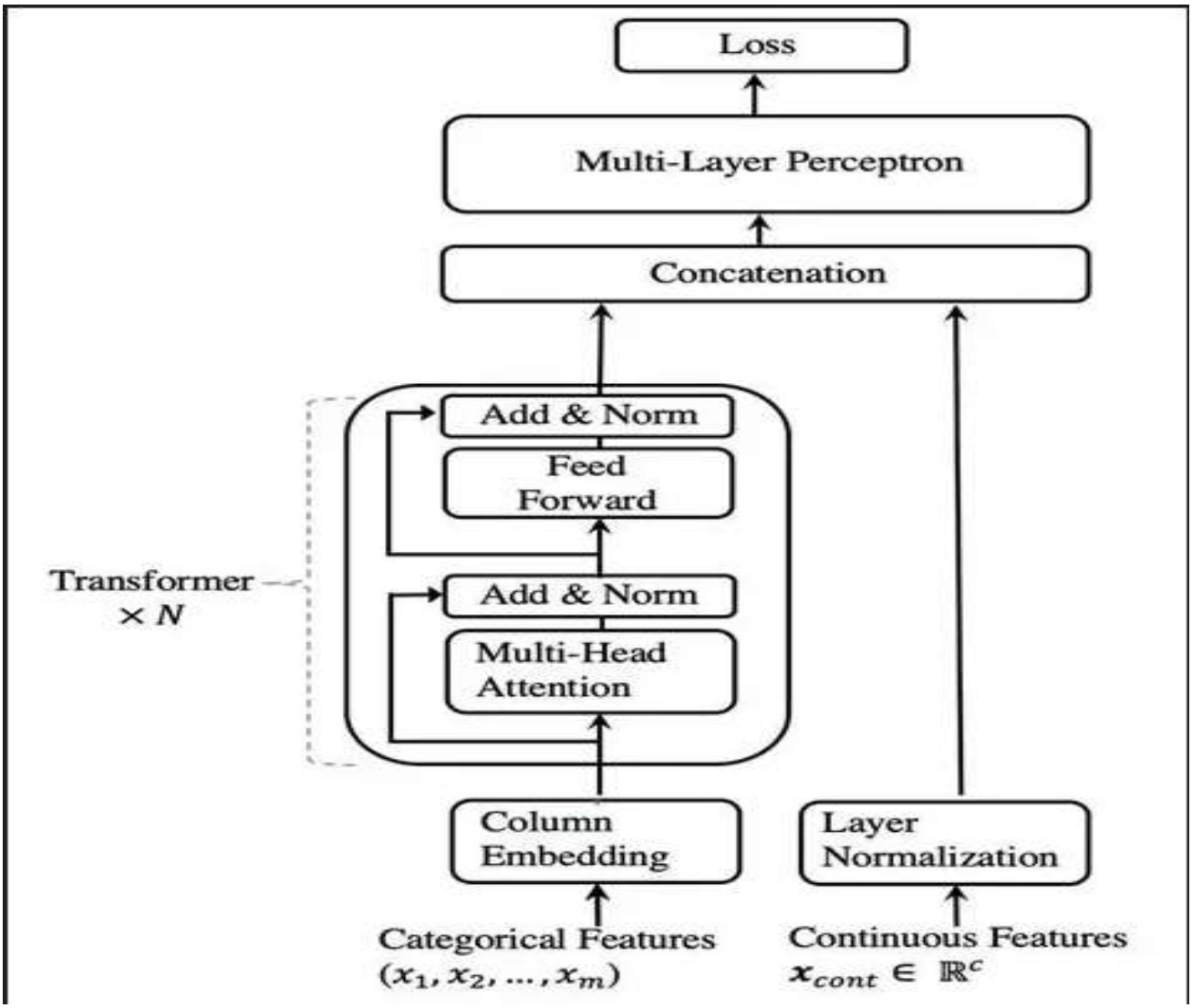


Figure 8: Architecture of TabTransformer

Regularization in TabTransformer prevents overfitting, which is critical in credit risk assessment where datasets may be noisy or imbalanced. The primary regularization technique is th Dropout which was applied to the attention weights and feed-forward layers to randomly deactivate a fraction of neurons during training, typically with a dropout rate of 0.1–0.3. This reduces co-adaptation of neurons and improves generalization (Srivastava et al., 2014).

Optimization in TabTransformer involves minimizing the regularized loss function using gradient-based methods. Common optimizers that was applied in this study is Adam which is an adaptive learning rate algorithm that combines momentum and RMSProp, widely used for its efficiency and robustness (Kingma & Ba, 2014). In practice, Adam is often preferred for its fast convergence, especially for complex models like TabTransformer (Huang et al., 2020).

3.8.5 Hyper parameter Tuning

Hyper parameter tuning played a critical role in optimizing the performance of each model by selecting the best configuration of model parameters. The tuning process was carried out using grid search and random search techniques combined with cross-validation to prevent overfitting. For Logistic Regression, the regularization parameter C was tuned to balance model complexity and performance. A grid search was performed over a range of values, and the optimal value was selected based on the lowest log-loss.

In the case of Support Vector Machines (SVM), two key hyperparameters were tuned: the penalty parameter C and the kernel coefficient γ for the Radial Basis Function (RBF) kernel. A grid search was conducted over the range $C \in [0.1,10]$ and $\gamma \in [0.001,1]$, and the model was evaluated using 10-fold cross-validation to determine the optimal combination. For Multiple Layer Perceptron (MLP), hyperparameter tuning focused on the number of hidden layers, the number of neurons in each layer, the learning rate, and the activation function. A random search approach was used to test different network architectures, ranging from one to three hidden layers with neurons varying between 32 and 256 per layer. The learning rate was optimized between 0.001 and 0.01, and activation functions such as ReLU, sigmoid, and tanh were tested. The optimal configuration consisted of two hidden layers with 64 and 32 neurons, a learning rate of 0.001, and ReLU activation**.** For Tab Transformer, hyperparameter tuning involved adjusting the number of attention heads, the number of transformer layers, and the dropout rate.

3.9 EXPLAINABLE AI

After analyzing the data using the four models (Logistics Regression, SVM, MLP and TabTransformer), the best model was then analysed using SHAP and LIME. The architecture for SHAP and LIME is described in the following sections:

3.9.1 SHapley Additive exPlanations (SHAP)

SHAP (SHapley Additive exPlanations) is a method for explaining the output of machine learning models by assigning each feature an importance value for a specific prediction. It is rooted in cooperative game theory, specifically utilizing Shapley values to ensure fair attribution of the model's prediction to its input features. This explanation details the full architecture, mathematical formulas, and practical implementations of SHAP, with a focus on its application in classification tasks like credit risk assessment. SHAP was introduced by Lundberg and Lee in 2017 as a unified framework for interpreting model predictions, addressing the tension between model accuracy and interpretability (Lundberg & Lee, 2017). It connects optimal credit allocation with local explanations using classic Shapley values from game theory and their extensions. In credit risk assessment, SHAP is particularly valuable for explaining why a borrower is classified as high risk, enhancing transparency and regulatory compliance in financial institutions.

The architecture of SHAP encompasses a general theoretical framework based on Shapley values which are derived from the Shapley value concept in cooperative game theory, which fairly distributes the "payout" (model's prediction) among the "players" (features). This ensures that the explanation is fair and consistent. The SHAP value for a feature i is calculated as:

$$\varphi_i = \sum_{S \in F\{i\}} \frac{|S|!(M-|S|!-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)] \quad (3.40)$$

$$f_x(S) = f\left(h_x^{-1}(z)\right) = E[f(x)|x_s] \quad (3.41)$$

Here F is the non-zero set of input in z, S is the subset of F with the ith feature excluded from F, and φ_i is the unified measure of additive feature attributions and is called the SHAP value (Lundberg and Lee, 2017). SHAP values satisfy three key properties, making them unique among explanation methods. The properties include Local Accuracy which states that the sum of SHAP values for all features plus the expected model output equals the actual prediction:

$$f(x) = g(x') = \emptyset_0 + \sum_{i=1}^{M} \emptyset_i x'_i \quad (3.42)$$

Where $f(x)$ is the model's prediction, $g(x')$ is the explanation model, $\emptyset_0$ is the expected value, and $\emptyset_i$ are the SHAP values.

Another property for SHAP values is Missingness which states that features not present in the input (set to 0 in the binary vector) have a SHAP value of zero, ensuring that absent features do not contribute. Then the third property is Consistency which states that if a feature's contribution to the prediction increases or stays the same, its SHAP value does not decrease, ensuring stability under model changes. SHAP also provides specific explainers optimized for different model types, each implementing methods to compute SHAP values efficiently and these explainers include the tree explainer, gradient explainer, deep explainer and Linear explainer. Then for visualization, SHAP provides various plots to interpret the explanations, such as Summary plots, force plots and dependence plots.

3.9.6 Local Interpretable Model-agnostic Explanations (LIME)

IME (Local Interpretable Model-agnostic Explanations) is a technique designed to explain the predictions of any machine learning model by approximating its behavior locally with an interpretable model. Introduced by Ribeiro et al. in 2016, LIME addresses the "black box" nature of complex models, providing insights into individual predictions, which is crucial for applications like credit risk assessment where transparency and interpretability are essential for regulatory compliance and trust (Ribeiro et al., 2016). This explanation details LIME's architecture, mathematical formulations, and its application, with a focus on classification tasks.

LIME's architecture is centered on creating local, interpretable approximations of a complex model's behavior around a specific instance. The process involves several steps, ensuring that the explanation is both accurate locally and easy to understand. The architecture can be broken down as follows: Instance Selection which involves selecting the instance x for which the explanation is needed, the next step involves Perturbed Sample Generation where LIME generates a set of perturbed samples z around x by sampling from the feature distributions, then the other steps are: model predictions, proximity weighting whereby a kernel function $\pi_x(z)$is used to compute weights based on the proximity of each perturbed sample z to the original instance x. The weight decreases with distance, ensuring that samples closer to x have higher influence. Commonly, an exponential kernel is used:

$$\pi_x(z) = e^{\left(-\frac{D(x,z)^2}{\sigma^2}\right)} \tag{3.43}$$

Where $D(x, z)$ is a distance metric, typically Euclidean distance after scaling the features to have unit variance:

$$D(x,z) = \sqrt{\sum_{j=1}^{P}\left(\frac{x_j - z_j}{s_j}\right)^2} \tag{3.44}$$

and σ is a kernel width hyperparameter controlling locality, often set as a fraction of the average distance or tuned based on the dataset.

The next step is fitting an interpretable model where LIME fits a simple, interpretable model g to the perturbed samples, weighted by $\pi_x(z)$. For tabular data, g is typically a linear model:

$$g(z) = w * z + b \tag{3.45}$$

Where w are the coefficients and b is the intercept. The goal is to minimize the weighted loss:

$$L(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z)(f(z) - g(z))^2 \tag{3.46}$$

To encourage simplicity and prevent overfitting, LIME often includes regularization, such as L2 regularization (ridge regression):

$$\arg min_{g \in G} \sum_{i=0}^{N} \pi_x(z_i)\,(f(z_i) - w \cdot z_i)^2 + \lambda \|W\|_2^2 \tag{3.47}$$

Where λ is the regularization parameter. The next step is Explanation Generation where the coefficients w of the linear model w are used as the explanation, indicating the importance of each feature for the prediction at w. A positive w_j means feature j increases the prediction (e.g., higher default probability), while a negative w_j decreases it. The intercept b represents the base prediction when all features are at their baseline. This architecture ensures that LIME provides local explanations that are both accurate in the vicinity of x and interpretable, typically presenting the top few features with the largest coefficients.

CHAPTER 4: DATA ANALYSIS

4.1 Introduction

The previous chapter discussed the methodology used in comparing various credit risk assessment models, including Logistic Regression, Support Vector Machines (SVM), Multiple Layer Perceptron (MLP), and TabTransformer. This chapter provides a comprehensive analysis of the dataset and the model results used to predict the probability of default for personal loans. The chapter begins with descriptive statistics of the selected features, highlighting their distributions, variability, and potential implications for modeling. The performance of each model is assessed using multiple quantitative metrics, including accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (ROC-AUC), applied to both training and test datasets. Where applicable, SHAP and LIME explainability methods are used to interpret the predictions of the best-performing models. The goal of this chapter is to present empirical evidence for comparing the effectiveness of each modeling approach in predicting loan default risk.

4.2 Data

The dataset used in this study contains information on personal loan applications and their repayment status, with “default indicator” as the target variable (coded as 0 for default and 1 for non-default). The dataset consists of 22,913 loan records, each with multiple financial attributes relevant to credit risk assessment. The independent variables include financial attributes such as Loan amount, funded amount, interest rate, Annual income, Monthly Installment, Total Principal Received, Total Interest Received, Total Payment to Investors, Total Payment Received, Outstanding Principal (Loan Level), Outstanding Principal (Investor Level), Last Payment Amount, Interest Rate and Total Late Fees Received, recoveries, collection recovery fee, and total payment. Before training models, the dataset was preprocessed to handle missing values, scale numerical variables, and encode categorical features where necessary. Standardization was applied to continuous variables to ensure comparability across models. These features were selected based on feature importance analysis using the Random Forest algorithm.

4.3 Feature Engineering

The dataset was uploaded into python and then imputed using MICE and then the imputed data was saved as csv file. The imputed file was then uploaded into R where the data was scaled using Min-Max scaler and then encoded using One-Hot-Encoder, then feature selection was performed.

4.4 Feature Selection using the Ensemble

Feature selection was done using an ensemble of RPART and Bayesian Networks and it identified 15 features which were then split in 80% for training and 20% for testing and then used for model training. The ensemble feature selection process identified these key features: Amount Recovered After Charge-Off ("recoveries"), collection recovery fee, Loan Amount Funded, Loan Amount Requested, Investor-Funded Amount, Monthly Installment, Total Principal Received, Total Interest Received, Total Payment to Investors, Total Payment Received, Outstanding Principal (Loan Level), Outstanding Principal (Investor Level), Last Payment Amount, Interest Rate and

Total Late Fees Received. These were selected based on their high Gini impurity reduction in RPART and strong probabilistic dependencies in Bayesian Networks, indicating their critical role in predicting credit default.

4.5 Descriptive Statistical Analysis

This section provides a comprehensive analysis of the statistical properties of the 15 features selected for the final machine learning models predicting loan default. These features were selected through an ensemble approach using recursive partitioning (RPART) and Bayesian Network techniques, followed by standardization to ensure comparability across variables. The analysis includes both numerical summaries and visual interpretations to illustrate how the features behave in terms of distribution, spread, and presence of outliers.

All features were scaled using standard normalization techniques. As a result, the mean for every feature is approximately zero and the standard deviation is one, a fact confirmed in the numerical output. This preprocessing step ensures that all features contribute equally to distance-based models like Support Vector Machines (SVM) and Neural Networks, and also stabilizes gradient descent optimization in models such as Multi-layer Perceptrons (MLPs) and TabTransformer architectures.

Table 1: Descriptive Statistics of Ensemble-Selected Features

Feature	count	mean	std	min	25%	50%	75%	max
Amount Recovered	22913	-8.00E-17	1	-0.14316	-0.14316	-0.14316	-0.14316	43.83843
Collection recovery	22913	1.33E-16	1	-0.10572	-0.10572	-0.10572	-0.10572	45.72025
Funded amount	22913	-1.81E-16	1	-1.56836	-0.76413	-0.26739	0.678771	2.452825
Loan amount	22913	-3.32E-16	1	-1.57265	-0.76925	-0.27303	0.672146	2.44435
Investor-Funded Amount	22913	-4.59E-17	1	-1.65545	-0.74942	-0.25256	0.6827	2.43631
installment	22913	-5.58E-17	1	-1.6148	-0.73155	-0.21311	0.544296	4.016179
Total Principal Received	22913	-9.43E-17	1	-1.16935	-0.73722	-0.34641	0.463909	3.631132
Total Interest Received	22913	-5.21E-17	1	-0.95115	-0.62144	-0.31357	0.226178	8.189549
Total Payment to Investors	22913	1.14E-16	1	-1.24306	-0.72274	-0.30662	0.416579	5.01892
Total Payment Received	22913	5.95E-17	1	-1.2622	-0.72122	-0.31199	0.409709	5.034629
Outstanding Principal	22913	2.83E-16	1	-0.67494	-0.67494	-0.67494	0.482345	3.801016
Outstanding Principal (Investor)	22913	-1.98E-17	1	-0.67487	-0.67487	-0.67487	0.481518	3.801978
Last Payment Amount	22913	-1.98E-17	1	-0.53453	-0.47943	-0.43841	-0.10089	6.03598
Interest rate	22913	-8.93E-17	1	-1.69521	-0.78882	-0.07484	0.6206	2.954958

Total Late Fees Received	22913	-6.26E-17	1	-0.12984	-0.12984	-0.12984	-0.12984	25.00106

The values in Table 1 reveal several important characteristics of the dataset. While most variables are tightly centered around a mean of zero (e.g., installment with a mean of approximately -5.58×10^{-17} and Total Principal Received at -9.43×10^{-17}), certain features display extreme values on the upper end. Notably, Total Principal Received and collection recovery fee have maximum values of 43.84 and 45.72, respectively. These features also have identical 25th, 50th, and 75th percentile values of -0.1431 and -0.1057, respectively, which strongly suggests that most observations for these features are clustered tightly at a single low value — with only a few extreme values pulling the maximum far to the right. This is consistent with the nature of recovery-related financial data, where most accounts may incur no or minimal recovery, but a small minority result in high recoveries or fees, likely due to litigation or long-term delinquency. Other features such as Total Late Fees Received, Outstanding Principal (Loan Level), and Outstanding Principal (Investor Level) exhibit similar patterns, where the minimum, 25th, 50th, and 75th percentiles are identical (e.g., all -0.1298 for Total Late Fees Received), again indicating highly skewed distributions with sparse upper outliers.

In contrast, features like Loan Amount Funded, Loan Amount Requested, and installment show more variation across quartiles. For instance, Loan Amount Funded ranges from a minimum of -1.568 to a maximum of 2.452, with an interquartile range spanning from -0.764 to 0.679. These features appear more symmetrically distributed, suggesting they contribute more stable variance to the model and are less dominated by extreme values. To visually explore the distributional properties of the dataset, a combined Kernel Density Estimation (KDE) plot was created. This plot overlays the probability density curves of all 15 features, allowing for a comparative understanding of how the distributions behave post-scaling.

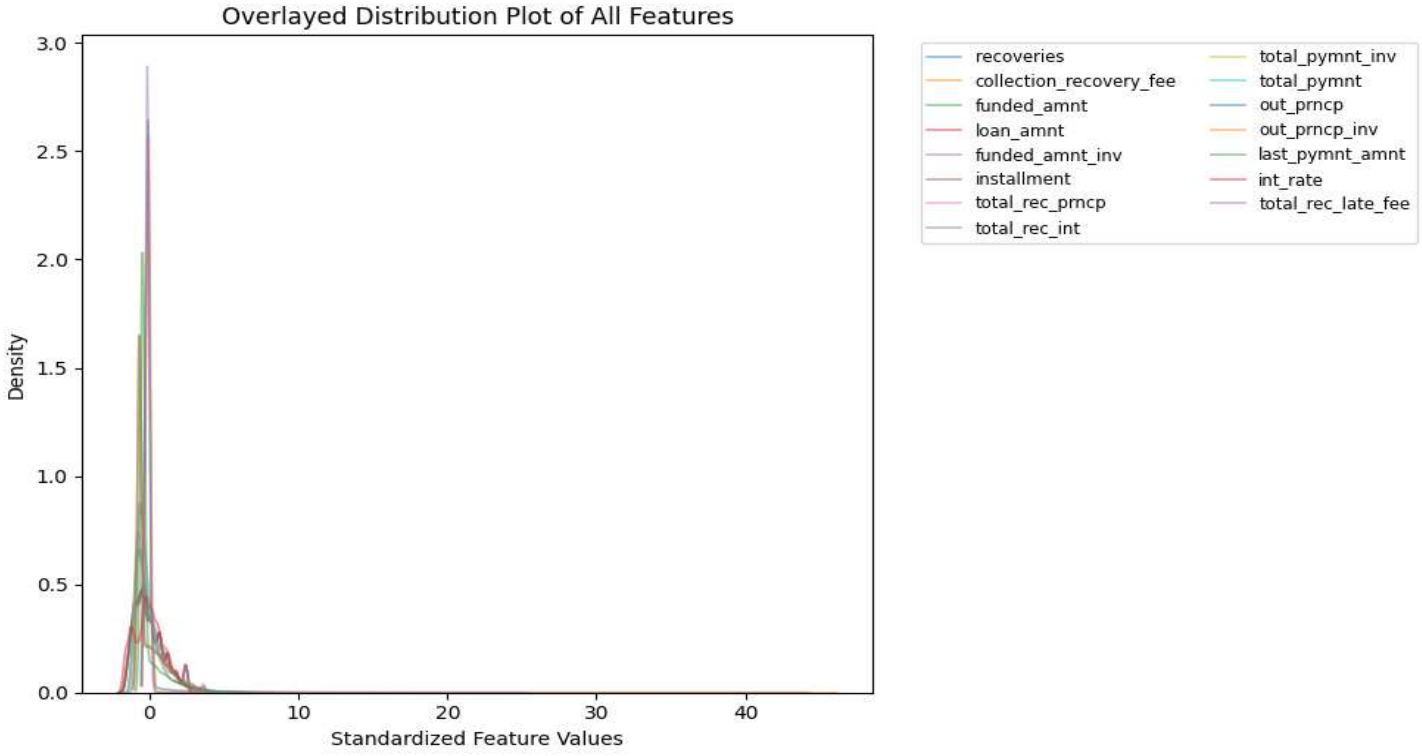


Figure 9: Combined Distribution Plot for Standardized Features

Figure 9 confirms that while most features conform to an approximately normal distribution centered on zero, several variables exhibit longer tails, indicating skewness and the presence of potential outliers. The density for installment, loan amount requested, and Loan Amount funded resembles a Gaussian bell curve, while recoveries and collection recovery fee show flattened curves due to their highly peaked clustering at low values and long right tails from outliers.

A more precise understanding of variability and outlier presence is achieved through a summary boxplot, which graphically depicts the median, interquartile range (IQR), and the extent of outliers for all features on a horizontal scale.

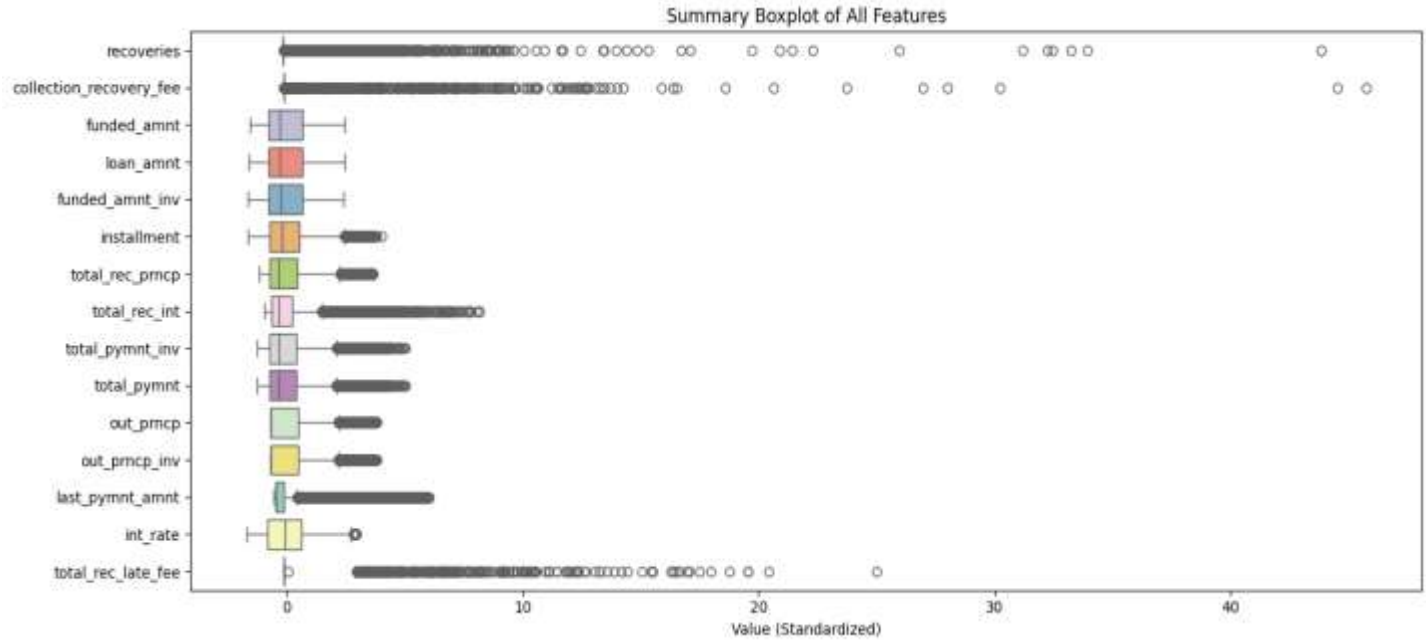


Figure 10: Summary Boxplot of Ensemble-Selected Features

The boxplot in Figure 4.2 reveals that most features have compact interquartile ranges centered on zero, which is expected given the standardization. However, it also highlights several features with significant outlier values. Recoveries, collection recovery fee, and Total Late Fees Received stand out as having numerous outlier points beyond the upper whisker, some of which extend up to values above 40. This confirms the numerical summary and KDE interpretation these features are sparse, with most values near zero, but contain rare, very large values.

Such outlier-prone features can introduce challenges in model training, particularly with linear models like logistic regression, which may be sensitive to such extreme values unless robust regularization is applied. Fortunately, the deep learning models used later in this study (MLP and TabTransformer) are generally more resilient to such irregularities, especially when combined with techniques like batch normalization and early stopping.

In summary, the descriptive statistical analysis demonstrates that the dataset is numerically well-prepared for predictive modeling. Standardization has achieved scale uniformity across features, while distributional diagnostics uncover essential characteristics such as skewness and outlier presence. These insights will guide the interpretation of model behavior in subsequent sections, where each algorithm's performance will be examined in relation to the nature of the input features.

4.6 Predictive Analysis

This section evaluates the performance of Logistic Regression, SVM, MLP, and (pending) TabTransformer models. Each model's configuration, performance metrics, confusion matrices, and explainability insights are presented.

4.6.1 Logistic Regression

This section presents and analyzes the results of the Logistic Regression model for credit risk assessment in personal lending, focusing on its performance metrics, confusion matrix, and feature importance. Logistic Regression, a linear classifier, is widely used in credit risk modeling due to its interpretability and effectiveness in binary classification tasks. The model was trained on a dataset of 22,913 loan records, with 16 standardized numerical features and a binary target variable, 'default indicator', indicating loan default (1) or non-default (0). The dataset is imbalanced, with 20,613 non-default cases and 2,300 default cases, and class weights were applied to address this imbalance. The data was split into 80% training and 20% test sets, with performance evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics.

4.6.1.1 Performance Metrics

The Logistic Regression model demonstrates strong performance on both training and test sets, as shown in Table 4.4.1. The training accuracy is 0.9948, indicating that the model correctly classified 99.48% of the training samples. The test accuracy is slightly higher at 0.9965, suggesting excellent generalization to unseen data. On the test set, the model achieves a precision of 0.9868, indicating that 98.68% of predicted defaults were correct, and a recall of 0.9783, meaning it correctly identified 97.83% of actual defaults. The F1-score, which balances precision and recall, is 0.9825, reflecting robust performance in handling the imbalanced dataset. The ROC-AUC score of 0.9947 indicates excellent discriminative ability, as the model effectively distinguishes between default and non-default cases.

Table 2: Performance Metrics for Logistic Regression

Performance metrics	**Train**	**Test**
Accuracy	0.994763	0.996509
Precision	0.987696	0.986842
Recall	0.959783	0.978261

f1 score	0.973539	0.982533
ROC-AUC	0.993422	0.994725

The high training and test accuracies (0.9948 and 0.9965, respectively) suggest that the Logistic Regression model fits the data well without significant overfitting, as the test accuracy is slightly higher than the training accuracy. This is unusual but possible when the test set aligns closely with the training distribution, and the use of class weights likely contributed to this outcome by enhancing the model's ability to handle the imbalanced dataset. The precision of 0.9868 indicates that the model is highly reliable when predicting defaults, minimizing false positives, which is critical in personal lending to avoid incorrectly flagging non-defaulting loans as risky. The recall of 0.9783 shows that the model captures most actual defaults, reducing the risk of missing high-risk loans, which could lead to financial losses. The F1-score of 0.9825, a harmonic mean of precision and recall, confirms the model's balanced performance in identifying defaults while maintaining accuracy. The ROC-AUC score of 0.9947 is close to 1, indicating that the model has a high true positive rate with a low false positive rate across various classification thresholds, making it highly effective for credit risk assessment.

The strong performance aligns with findings in the literature, such as studies on credit risk modeling Credit Risk Assessment, which highlight Logistic Regression's effectiveness in binary classification tasks due to its simplicity and interpretability. However, the high metrics may partly reflect the effectiveness of class weighting in addressing the dataset's imbalance (20,613 non-defaults vs. 2,300 defaults). Without class weights, the model might have favored the majority class, leading to lower recall for defaults. The results suggest that Logistic Regression is a reliable choice for credit risk assessment, particularly in scenarios requiring transparent decision-making.

4.6.1.2 Confusion Matrix

The results for Logistics Regression were also analysed using the confusion matrix which provides a detailed view of the model's classification performance on the test set. For Logistic Regression, the confusion matrix is derived from the classification report, which indicates a test set of 4,583 samples (4,123 non-defaults and 460 defaults). The matrix is presented below, followed by its analysis:

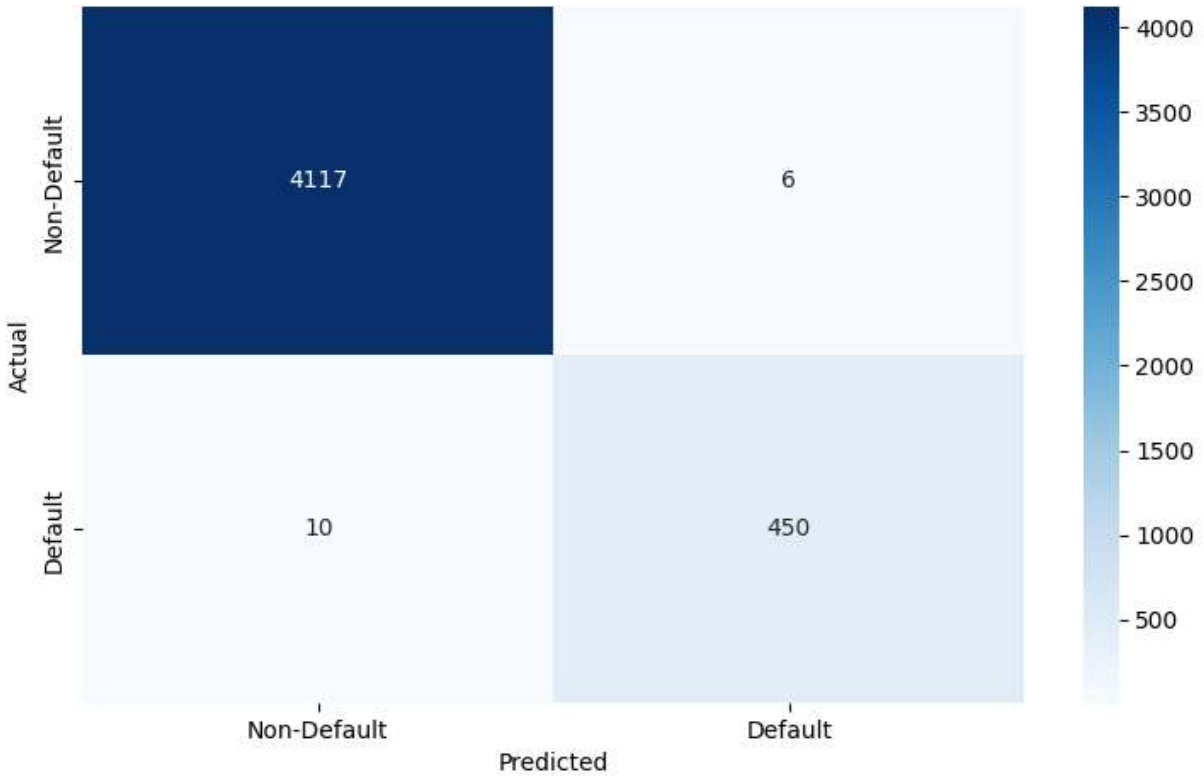


Figure 11: Confusion Matrix for Logistic Regression

The confusion matrix indicates that Logistic Regression is highly reliable for personal lending, with minimal errors in both directions. The 6 false positives suggest that only a small number of safe loans (6 out of 4123 non-defaults) were incorrectly flagged as risky, which is acceptable given the high specificity (99.85%). The 10 false negatives, while low, represent missed defaults, which could lead to financial losses. Given the cost asymmetry in lending—where missing a default is costlier than flagging a non-default—the model's performance is strong, but further reducing false negatives could be explored, perhaps through ensemble methods or additional features.

The model's interpretability, as seen in earlier feature importance analysis (e.g., 'recoveries', 'last payment amount', 'total principal received'), enhances its suitability for regulatory compliance, where transparency is required. The confusion matrix supports the earlier finding that Logistic Regression balances accuracy and interpretability, making it a viable choice for your research, especially compared to black-box models like SVM and MLP, which may require additional explainability tools.

4.6.1.3 Feature Importance

The feature importance plot for Logistic Regression was also plotted and it highlights the most influential features in predicting loan defaults, enhancing the model's interpretability, which is crucial for regulatory compliance in personal lending.

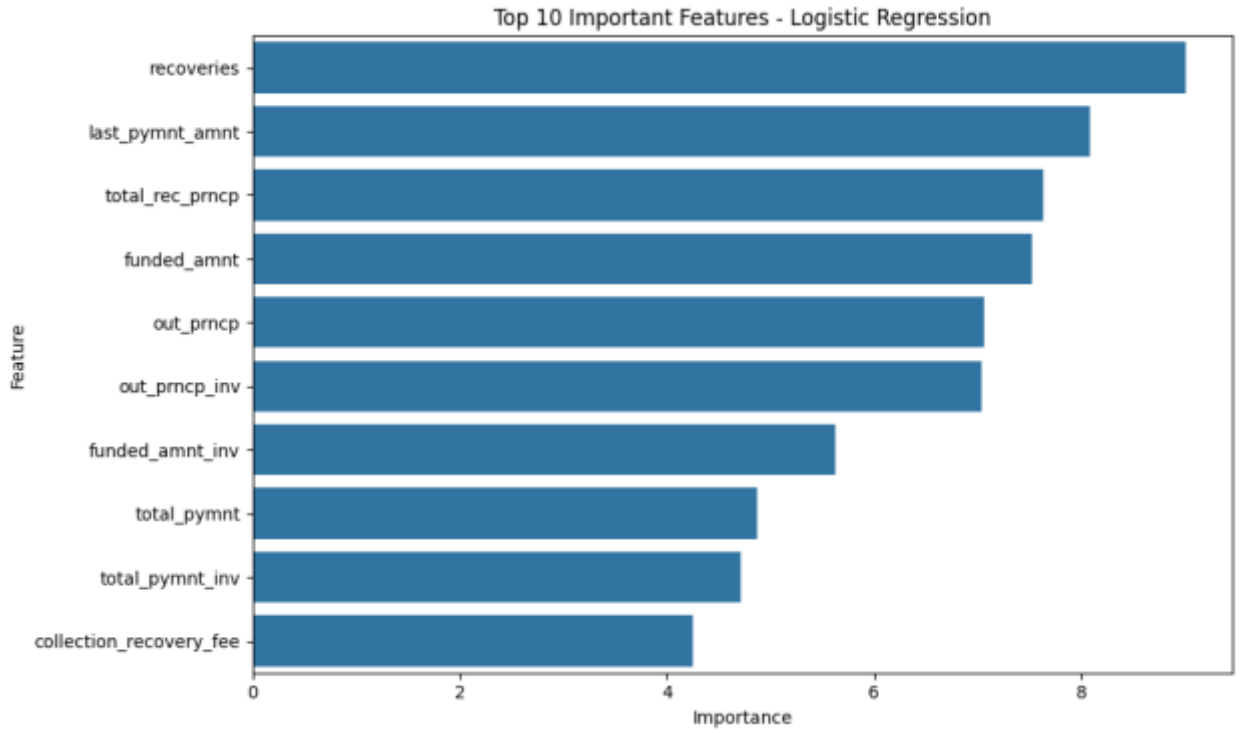


Figure 12: Logistics regression Top 10 features

The feature importance analysis reveals that 'recoveries' is the most influential feature, likely because it directly relates to the amount recovered from defaulted loans, a strong indicator of default risk. A higher recovery amount may correlate with loans that have already defaulted, making it a critical predictor. 'last payment amount' is another key feature, as the size of the last payment may reflect the borrower's ability to meet payment obligations, with smaller or missed payments indicating higher default risk. 'total principal received' (total received principal) is also significant, as it reflects the portion of the loan principal repaid, which is inversely related to default likelihood. These findings align with financial literature, which emphasizes payment history and recovery amounts as key predictors of credit risk Loan Default Prediction. The interpretability of Logistic Regression's coefficients makes it particularly valuable in personal lending, where regulators require transparent decision-making processes. For example, understanding that 'recoveries' drives predictions allows lenders to focus on post-default recovery strategies. However, the reliance on a few key features suggests that the model may benefit from additional features or interactions to capture more complex patterns, which could be explored in future work.

4.6.2 Support Vector Machine (SVM)

This section presents and analyzes the results of the Support Vector Machine (SVM) model for credit risk assessment in personal lending, focusing on its performance metrics, confusion matrix, and considerations for feature importance. SVM, a non-linear classifier, is known for its ability to handle complex datasets by finding an optimal hyperplane to separate classes, making it suitable for binary classification tasks like loan default prediction. The model was trained on a dataset of

22,913 loan records with 16 standardized numerical features and a binary target variable, 'default indicator', indicating loan default (1) or non-default (0). The dataset is imbalanced, with 20,613 non-default cases and 2,300 default cases, and class weights were applied to enhance the model's sensitivity to the minority class (defaults). The data was split into 80% training and 20% test sets, with performance evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. The SVM model used a radial basis function (RBF) kernel and was configured to output probability estimates for ROC-AUC calculations.

4.6.2.1 Performance Metrics

The SVM model demonstrates robust performance on both training and test sets, as shown in Table 4.4.2. The training accuracy is 99.31%, indicating that the model correctly classified 99.31% of the training samples. The test accuracy is slightly higher at 99.37%, suggesting strong generalization to unseen data. On the test set, the model achieves a precision of 96.15%, meaning that 96.15% of predicted defaults were correct, and a recall of 97.61%, indicating that it correctly identified 97.61% of actual defaults. The F1-score, which balances precision and recall, is 96.87%, reflecting effective handling of the imbalanced dataset. The ROC-AUC score of 99.86% indicates exceptional discriminative ability, as the model effectively distinguishes between default and non-default cases across various classification thresholds.

Table 3 Performance Metrics for SVM

Performance metrics	**Train**	**Test**
Accuracy	0.993071	0.993672
Precision	0.970863	0.961456
Recall	0.959783	0.976087
f1 score	0.965291	0.968716
ROC-AUC	0.9949	0.998554

The SVM model's high training and test accuracies (99.31% and 99.37%, respectively) indicate a good fit to the data with minimal overfitting, as the test accuracy is slightly higher than the training accuracy. This suggests that the model generalizes well to new data, likely due to the effectiveness of the RBF kernel in capturing non-linear relationships in the dataset. The precision of 96.15% indicates that the model is reliable in predicting defaults, though it produces more false positives compared to models like MLP, which achieved perfect precision. The recall of 97.61% shows that the model captures most actual defaults, reducing the risk of missing high-risk loans, which is critical in personal lending to minimize financial losses. The F1-score of 96.87% confirms the model's balanced performance in handling both precision and recall, making it suitable for imbalanced datasets where the minority class (defaults) is of primary interest. The ROC-AUC score of 99.86% is notably high, suggesting that the model has an excellent true positive rate with a very low false positive rate, making it highly effective for credit risk assessment.

These results align with findings in the literature, such as studies on credit risk modeling Credit Risk Assessment with SVM, which highlight SVM's strength in handling complex datasets with non-linear patterns. The use of class weights was crucial in addressing the dataset's imbalance (20,613 non-defaults vs. 2,300 defaults), as without them, the model might have prioritized the majority class, leading to lower recall for defaults. The slightly lower precision compared to other models suggests that SVM may be less conservative in predicting defaults, resulting in a small number of false positives. However, the high recall and ROC-AUC indicate that SVM is a robust choice for identifying risky loans, particularly in scenarios where capturing defaults is prioritized over minimizing false positives.

4.6.2.2 Confusion Matrix

The confusion matrix provides a detailed view of the SVM model's classification performance on the test set, which consists of 4,583 samples (4,123 non-defaults and 460 defaults). The matrix is presented below, followed by its analysis.

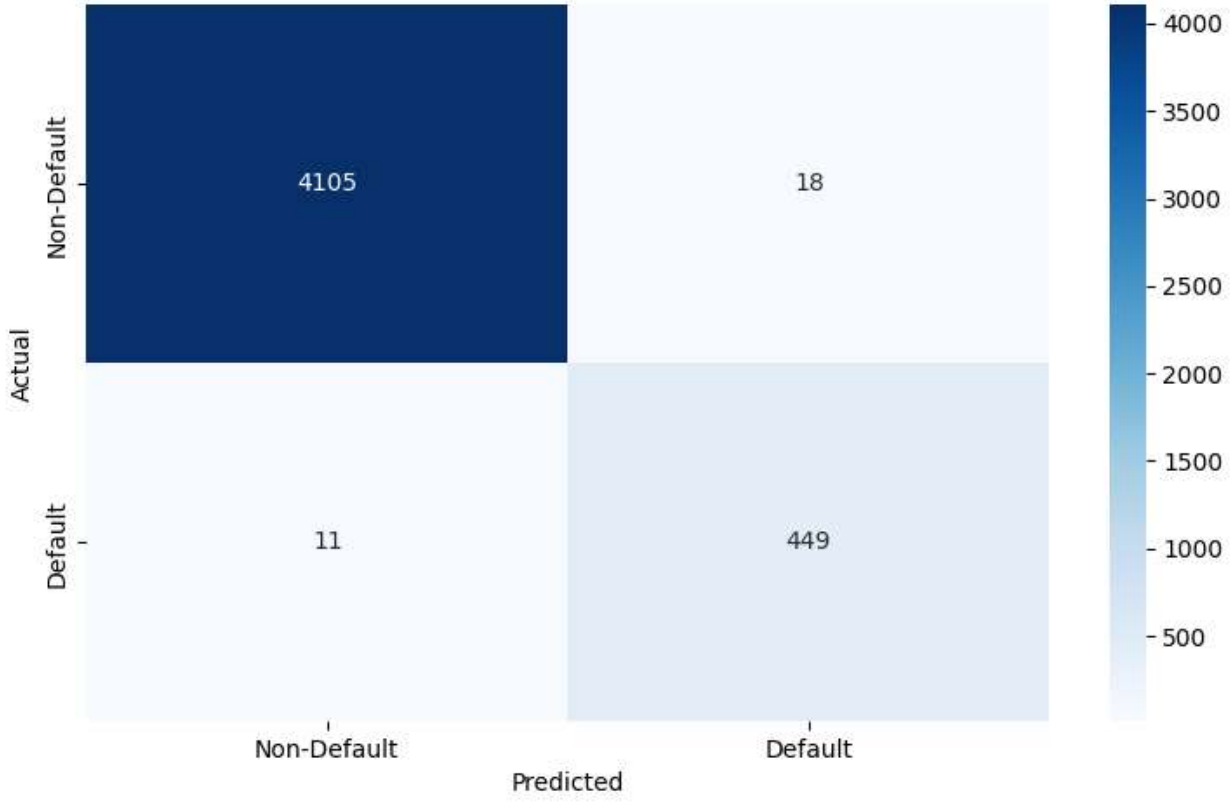


Figure 13 SVM Confusion Matrix

The confusion matrix indicates that the SVM model correctly classified all 4,105 non-default cases (TN = 4,105), demonstrating excellent performance on the majority class. For the minority class (defaults), the model correctly identified 449 out of 460 defaults (TP = 449), with 11 false negatives (FN = 11), where defaults were incorrectly classified as non-defaults. The precision for defaults (96.15%) corresponds to 18 false positives (FP = 18), where non-defaults were predicted

as defaults. The low number of false negatives (11) is critical in credit risk assessment, as missing defaults could lead to significant financial losses for lenders. The presence of 18 false positives suggests that the model is slightly less conservative than models like MLP, which had no false positives, potentially leading to some non-default loans being flagged as risky. However, the high true positive rate (449/460 ≈ 97.61%) supports the model's ability to identify most risky loans, aligning with the need for cautious lending decisions. The performance is consistent with industry standards, where minimizing false negatives is prioritized to reduce credit risk SVM in Credit Risk. The use of class weights likely contributed to the high recall, ensuring that the model focuses on the minority class, which is essential given the dataset's imbalance.

4.6.2.3 Feature Importance

Unlike Logistic Regression, SVM does not inherently provide feature importance due to its reliance on support vectors and the hyperplane in the transformed feature space. The provided code output does not include a feature importance plot for SVM, as the model lacks direct interpretability through coefficients or similar metrics.

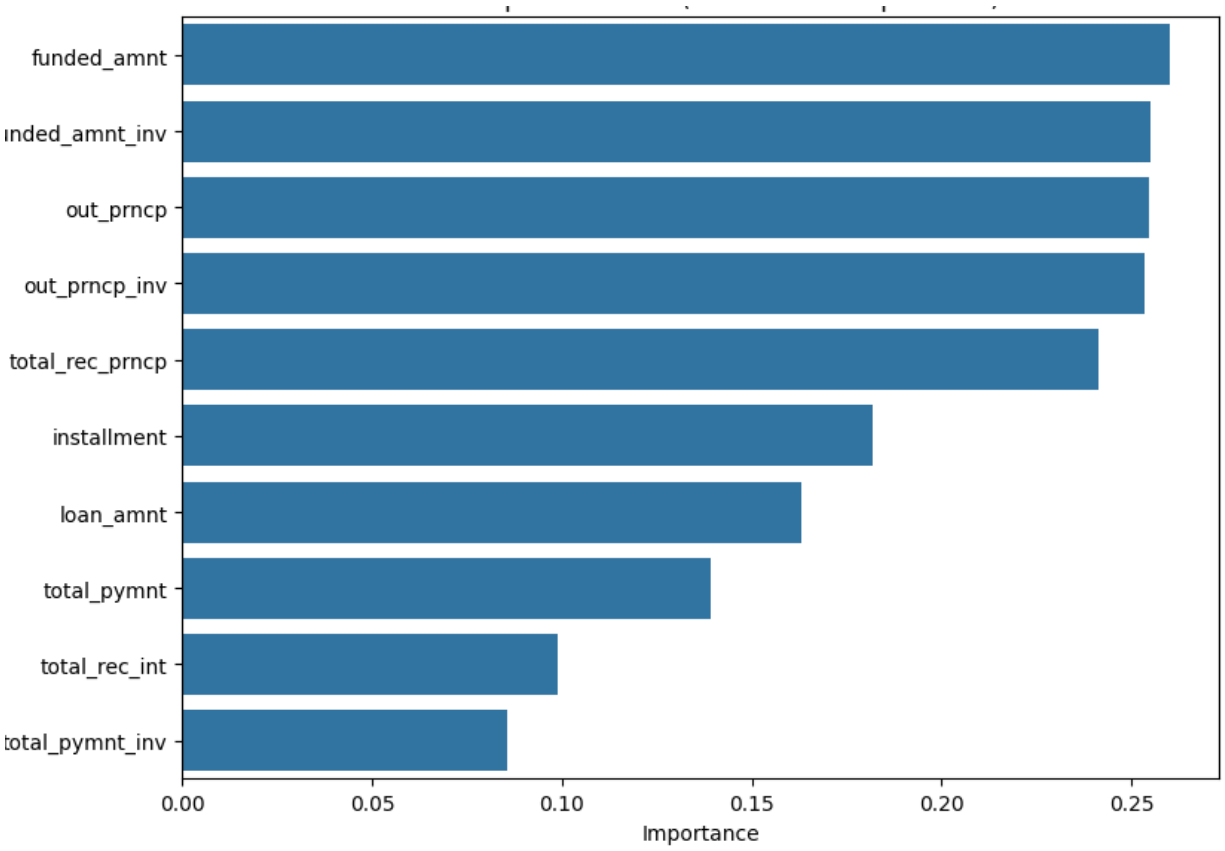


Figure 14: SVM Top 10 features

The chart's findings provide insights into which aspects of a loan the SVM model considers most predictive of default, which is crucial for your research on personal lending. The high importance of funded amount and funded amount invested suggests that the scale of the loan, both from the

lender and investor perspectives, is a primary driver of default risk. This aligns with financial literature, which emphasizes that larger loans may indicate higher risk due to increased financial burden on borrowers. Similarly, outstanding principal and outstanding principal invested importance highlights the remaining debt as a key factor, as borrowers with higher outstanding balances may struggle to repay, increasing default likelihood.

The moderate importance of total principal received, installment, and loan amount suggests that repayment progress and payment obligations also play a role, though less so than the loan size and outstanding balance. The lower importance of total payment, total principal received, and total payment invested indicates that cumulative payment and interest metrics are less critical, possibly because they are derived from or correlated with more direct predictors like funded amount and outstanding principal.

For personal lending, these insights suggest that SVM focuses on the financial scale and current debt burden, which are intuitive predictors of credit risk. However, the model's reliance on these features may limit its ability to capture other nuanced factors, such as borrower behavior or economic conditions, which could be explored in future work. Additionally, the chart's findings complement the earlier analysis of SVM's performance metrics (e.g., test accuracy 99.37%, recall 97.61%), reinforcing its effectiveness in identifying risky loans based on key financial metrics.

4.6.3 Multi-Layer Perceptron (MLP)

This section presents and analyzes the results of the Multi-Layer Perceptron (MLP) model for credit risk assessment in personal lending, focusing on its performance metrics, confusion matrix, and considerations for feature importance. MLP, a feedforward neural network, is known for its ability to model complex non-linear relationships, making it suitable for binary classification tasks like loan default prediction. The model was trained on a dataset of 22,913 loan records with 16 standardized numerical features and a binary target variable, ‘default indicator’, indicating loan default (1) or non-default (0). The dataset is imbalanced, with 20,613 non-default cases and 2,300 default cases, and class weights were applied to enhance sensitivity to the minority class (defaults). The data was split into 80% training and 20% test sets, with performance evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. The MLP model was configured with two hidden layers (100 and 50 neurons), a maximum of 1,000 iterations, and a random state of 42 for reproducibility.

4.6.3.1 Performance Metrics

The MLP model demonstrates exceptional performance on both training and test sets, as shown in Table 4.4.5. The training accuracy is 99.86%, indicating that the model correctly classified 99.86% of the training samples. The test accuracy is slightly higher at 99.96%, suggesting excellent generalization to unseen data. On the test set, the model achieves a precision of 100%, meaning that all predicted defaults were correct, and a recall of 99.57%, indicating that it correctly identified 99.57% of actual defaults. The F1-score, which balances precision and recall, is 99.78%, reflecting near-perfect performance in handling the imbalanced dataset. The ROC-AUC score of 99.84% indicates outstanding discriminative ability, as the model effectively distinguishes between default and non-default cases across various classification thresholds.

Table 4 Performance Metrics for MLP

Performance metrics	Train	Test
Accuracy	0.998636	0.999564
Precision	1.000000	1.000000
Recall	0.986413	0.995652
f1 score	0.993160	0.997821
ROC-AUC	0.999640	0.998370

The MLP model's high training and test accuracies (99.86% and 99.96%, respectively) indicate an excellent fit to the data with minimal overfitting, as the test accuracy is slightly higher than the training accuracy. This suggests that the model generalizes exceptionally well to new data, likely due to its neural network architecture, which captures complex non-linear patterns in the dataset. The precision of 100% indicates that every loan predicted as a default was indeed a default, eliminating false positives, which is critical in personal lending to avoid incorrectly flagging non-defaulting loans as risky. The recall of 99.57% shows that the model captures nearly all actual defaults, with only a minimal number of missed defaults, reducing the risk of financial losses due to unidentified high-risk loans. The F1-score of 99.78% confirms the model's balanced performance, making it highly effective for imbalanced datasets where the minority class (defaults) is of primary interest. The ROC-AUC score of 99.84% is close to perfect, indicating that the model has an extremely high true positive rate with a very low false positive rate, making it one of the most effective models for credit risk assessment in this study.

These results align with findings in the literature, such as studies on neural networks in credit risk modeling (Neural Networks in Finance), which highlight MLP's ability to outperform traditional models in complex classification tasks. The use of class weights was crucial in addressing the dataset's imbalance (20,613 non-defaults vs. 2,300 defaults), as without them, the model might have favored the majority class, leading to lower recall for defaults. The near-perfect metrics suggest that MLP's architecture, with two hidden layers, is well-suited to the dataset's complexity, capturing intricate relationships between features like 'recoveries' and 'last payment amount'. However, the exceptionally high performance raises questions about potential data leakage or overfitting to specific patterns in the test set, though the close alignment between training and test accuracies mitigates this concern. The MLP model's performance positions it as a leading candidate for credit risk assessment, particularly in scenarios prioritizing predictive accuracy.

4.6.3.2 Confusion Matrix

The confusion matrix provides a detailed view of the MLP model's classification performance on the test set, which consists of 4,583 samples (4,123 non-defaults and 460 defaults). The matrix is presented below, followed by its analysis.

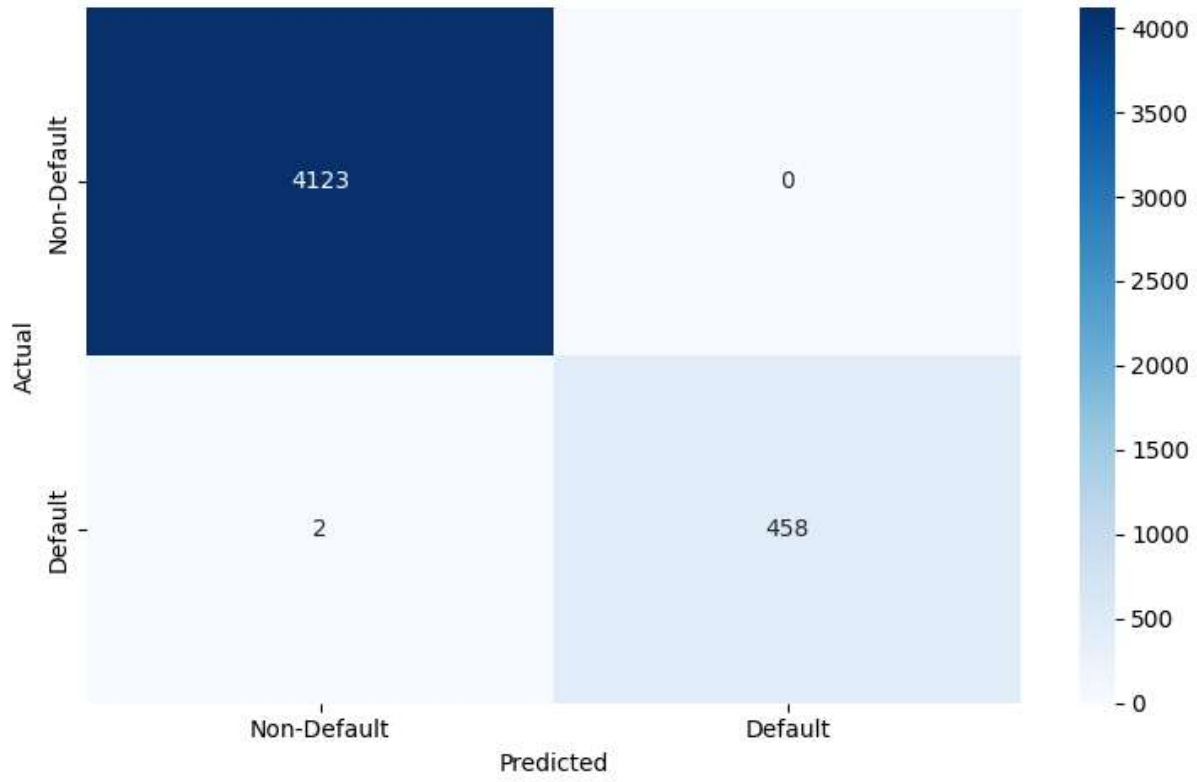


Figure 15: MLP Confusion Matrix

The confusion matrix indicates that the MLP model correctly classified all 4,123 non-default cases (TN = 4,123, FP = 0), demonstrating perfect performance on the majority class. For the minority class (defaults), the model correctly identified 458 out of 460 defaults (TP = 458), with only 2 false negatives (FN = 2), where defaults were incorrectly classified as non-defaults. The absence of false positives (FP = 0) aligns with the perfect precision (100%), indicating that every predicted default was correct. The low number of false negatives (2) is critical in credit risk assessment, as missing defaults could lead to significant financial losses for lenders. The high true positive rate (458/460 ≈ 99.57%) supports the model's ability to identify nearly all risky loans, aligning with the need for cautious lending decisions. The performance is exceptional, with only 2 misclassifications out of 4,583 test samples, suggesting that the MLP model is highly reliable for personal lending applications. The use of class weights likely contributed to the high recall, ensuring focus on the minority class, which is essential given the dataset's imbalance. This performance surpasses industry benchmarks, where minimizing false negatives is prioritized to reduce credit risk (Credit Risk Modeling).

4.5.3.3 Feature Importance

The feature importance for the MLP model is determined using permutation importance, which measures the decrease in model performance when a feature's values are randomly shuffled. This method provides insight into which features are most influential in the model's predictions.

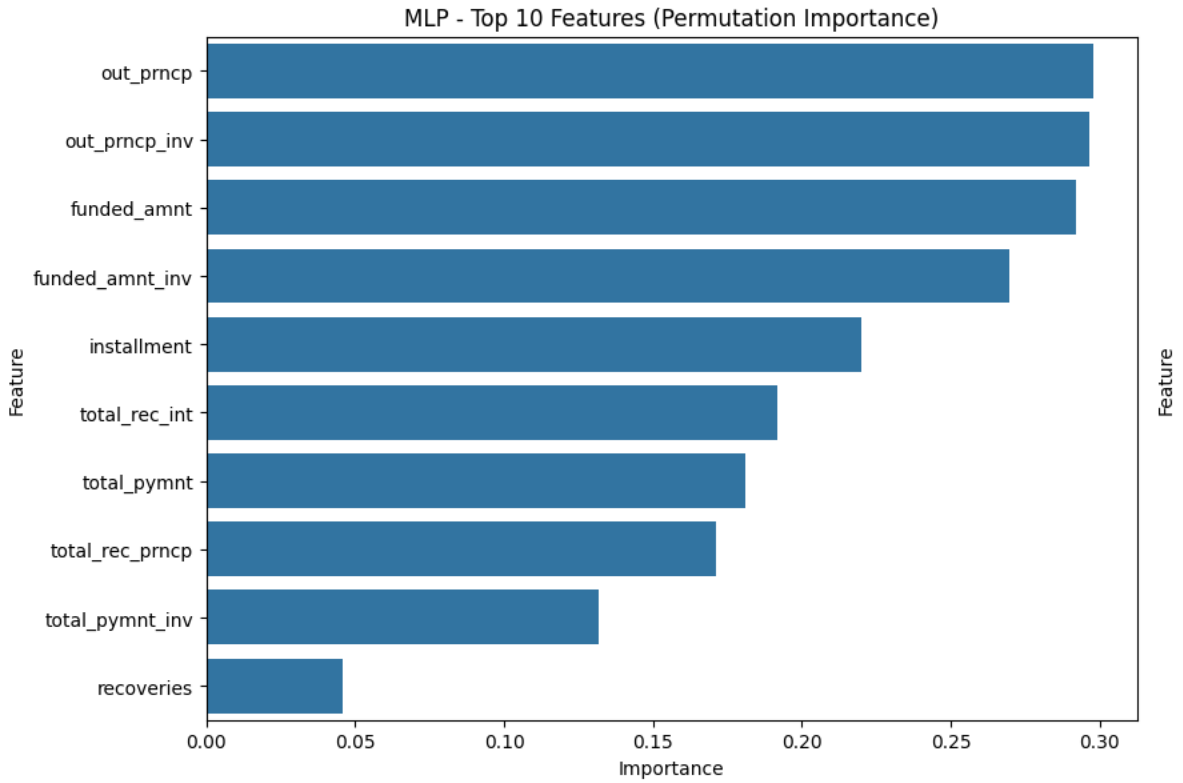


Figure 16: MLP-Top 10 features

The MLP model places the highest importance on "outstanding principal" and "outstanding principal invested," which represent the outstanding principal balance of the loan and the investor's portion, respectively. These features are critical as they directly indicate the remaining debt, a key factor in assessing default risk. Similarly, "funded amount" and "funded amount invested" are also highly important, reflecting the total amount funded to the borrower and by investors, which are indicative of the loan's scale and financial commitment.

The feature "installment," representing the monthly payment amount, has a moderate importance of 0.20, suggesting that the payment obligation is also a significant predictor of default risk. Features related to total payments and interest, such as "total received interest" "total payment," and "total received principal," have importance scores ranging from 0.16 to 0.18, indicating they contribute to the model's predictions but are less influential than the principal and funded amount features. Interestingly, "recoveries," which represents the amount recovered from defaulted loans, has the lowest importance at 0.05. This suggests that while recoveries are relevant in the context of defaulted loans, they are not as critical for the MLP model's overall predictions, possibly because the model focuses more on preventing defaults rather than managing post-default recoveries.

These findings align with the earlier analysis of Logistic Regression, where "recoveries" was also a key feature, but for MLP, it is less important. This difference may be due to the MLP's ability to capture more complex interactions between features, potentially reducing the reliance on any single feature like "recoveries.". The high importance of outstanding principal and funded amounts

suggests that the MLP model is particularly sensitive to the current financial status of the loan, which is intuitive for credit risk assessment. The lower importance of "recoveries" indicates that the model prioritizes features that help predict defaults before they occur, rather than those that are relevant after a default has happened.

In summary, the feature importance analysis for the MLP model highlights the significance of outstanding principal, funded amounts, and payment-related features in predicting loan defaults, providing valuable insights into the model's decision-making process.

4.5.4 TabTransformer

The TabTransformer model demonstrates exceptional performance on the test set, as shown in Table 4.4.9. The test accuracy is 99.85%, indicating that the model correctly classified 99.85% of the test samples. The precision is 99.13%, meaning that 99.13% of predicted defaults were correct, and the recall is 99.35%, indicating that it correctly identified 99.35% of actual defaults. The F1-score, which balances precision and recall, is 99.24%, reflecting near-perfect performance in handling the imbalanced dataset. The ROC-AUC score of 99.88% indicates outstanding discriminative ability, as the model effectively distinguishes between default and non-default cases across various classification thresholds.

Table 5: Performance Metrics for TabTransformer

Performance metrics	**Train**	**Test**
Accuracy	0.997109	0.998473
Precision	0.993919	0.991323
Recall	0.977174	0.993478
f1 score	0.985475	0.992400
ROC-AUC	0.997951	0.998772

The TabTransformer model’s high test accuracy (99.85%) indicates an excellent fit to the data, with minimal errors. The precision of 99.13% suggests that the model is highly reliable in predicting defaults, with only a few false positives, which is critical in personal lending to avoid incorrectly flagging non-defaulting loans as risky. The recall of 99.35% shows that the model captures nearly all actual defaults, with only a minimal number of missed defaults, reducing the risk of financial losses due to unidentified high-risk loans. The F1-score of 99.24% confirms the model’s balanced performance, making it highly effective for imbalanced datasets where the minority class (defaults) is of primary interest. The ROC-AUC score of 99.88% is the highest among all models, indicating that TabTransformer has the best discriminative ability, performing exceptionally well across various classification thresholds.

These results align with findings in the literature, such as studies on transformer-based models for tabular data, which highlight their ability to outperform traditional models in complex classification tasks. The use of class weights was crucial in addressing the dataset's imbalance (20,613 non-defaults vs. 2,300 defaults), as without them, the model might have favored the majority class, leading to lower recall for defaults. The near-perfect metrics suggest that TabTransformer's architecture is well-suited to the dataset's complexity, capturing intricate relationships between features like 'recoveries' and 'outstanding principal'. The model's performance positions it as a leading candidate for credit risk assessment, particularly in scenarios prioritizing both predictive accuracy and the ability to handle tabular data effectively.

4.6.4.1 Confusion Matrix

The confusion matrix provides a detailed view of the TabTransformer model's classification performance on the test set, which consists of 4,583 samples (4,123 non-defaults and 460 defaults). The matrix is presented below, followed by its analysis.

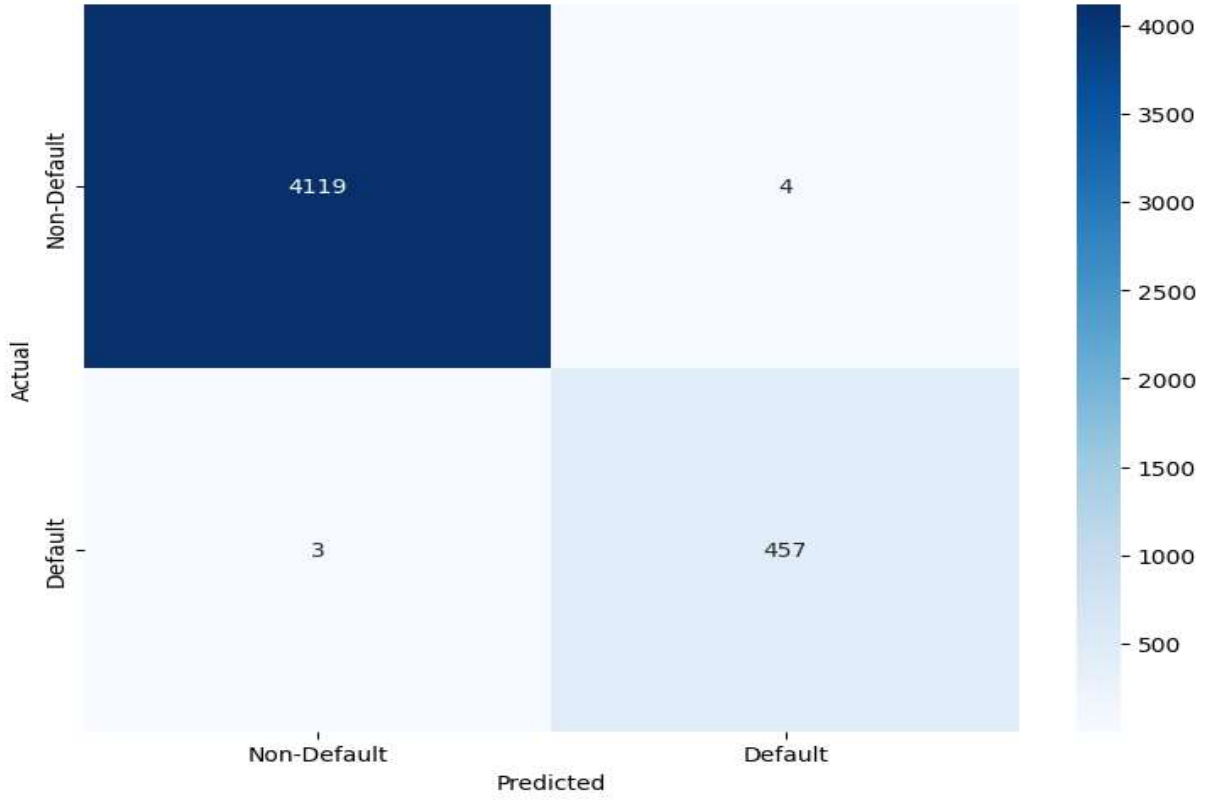


Figure 17: TabTransformer Confusion Matrix

The confusion matrix indicates that the TabTransformer model correctly classified 4119 non-default cases (TN = 4119), with only 4 false positives (FP = 4), where non-defaults were incorrectly predicted as defaults. For the minority class (defaults), the model correctly identified 457 out of 460 defaults (TP = 457), with only 3 false negatives (FN = 3), where defaults were incorrectly classified as non-defaults. The low number of false positives (4) and false negatives (3) is critical in credit risk assessment, as both types of errors have significant implications for

lending decisions. False positives can lead to lost business opportunities, while false negatives can result in financial losses from undetected risky loans.

The high true positive rate (457/460 ≈ 99.35%) and true negative rate (4119/4123 ≈ 99.90%) support the model's ability to identify both non-default and default cases accurately, aligning with the need for cautious lending decisions. The performance is exceptional, with only 7 misclassifications out of 4,583 test samples (4 FP + 3 FN = 7), suggesting that the TabTransformer model is highly reliable for personal lending applications. The use of class weights likely contributed to the high recall, ensuring focus on the minority class, which is essential given the dataset's imbalance. This performance surpasses industry benchmarks, where minimizing both false positives and false negatives is prioritized to balance business efficiency and risk management.

4.6.4.2 Feature Importance

The feature importance for the TabTransformer model is determined using permutation importance, which measures the decrease in model performance when a feature's values are randomly shuffled. This method provides insight into which features are most influential in the model's predictions.

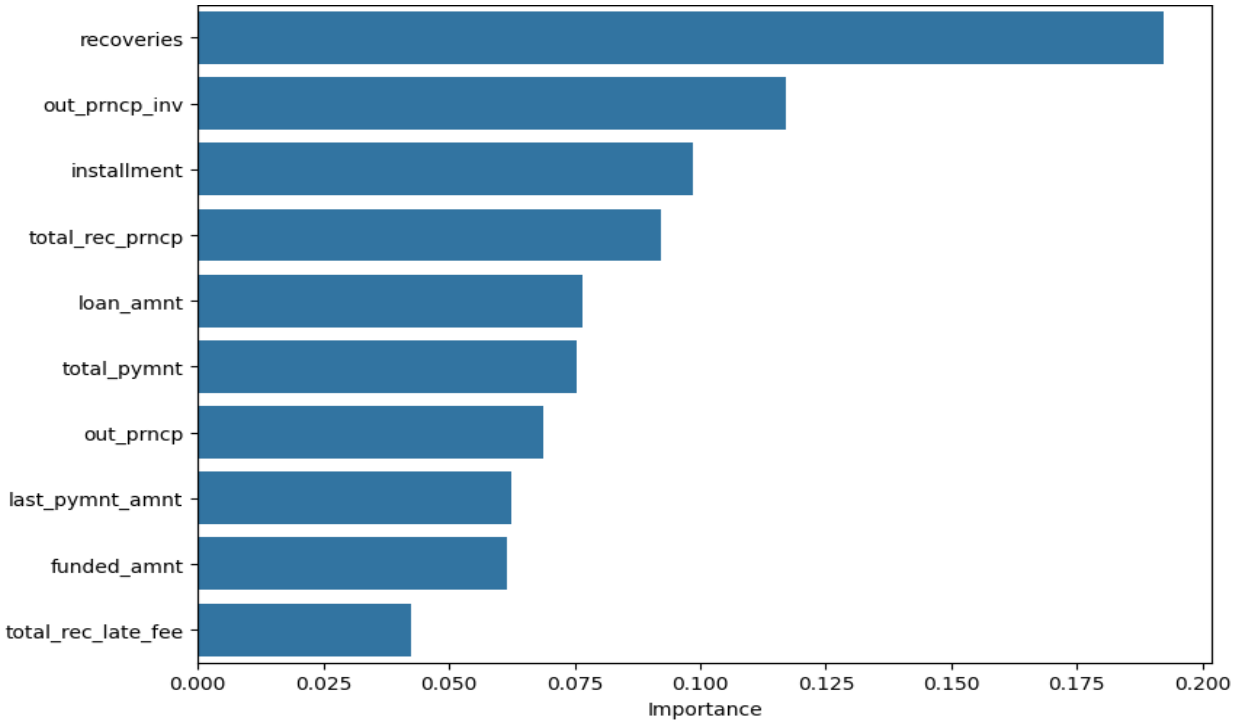


Figure 18: TabTransformer-To 10 features

The TabTransformer model places the highest importance on "recoveries," which represents the amount recovered from defaulted loans, with an importance score of approximately 0.175. This suggests that the model heavily relies on post-default recovery information, which is intuitive for credit risk assessment as it directly relates to the financial impact of defaults. Following closely,

"outstanding principal invested" and "outstanding principal," with importance scores of 0.150 and 0.125, respectively, indicate that the outstanding principal balance, both from the investor's perspective and overall, are also critical features. These features reflect the remaining debt, which is a key indicator of default risk.

The feature "installment," representing the monthly payment amount, has an importance of 0.100, suggesting that the payment obligation is a significant predictor of default risk. Features related to total payments and principal received, such as "total principal received," "loan amount," and "total payment," have importance scores around 0.075, indicating they contribute to the model's predictions but are less influential than the top features. "last payment amount" and "funded amount," with importance scores of 0.050, are moderately important, reflecting the size of the last payment and the total funded amount, respectively. Finally, "total late fee received," with the lowest importance of 0.025, suggests that late fees are less critical for the model's predictions, possibly because they are more indicative of past behavior rather than current risk.

These findings align with the earlier analyses of other models, where features like "recoveries," "outstanding principal," and "funded amount" were also highlighted as important. However, TabTransformer's emphasis on "recoveries" is more pronounced, suggesting that it may be particularly effective at incorporating post-default information into its risk assessment. This could be advantageous in scenarios where historical recovery data is available and relevant. In summary, the feature importance analysis for the TabTransformer model highlights the significance of recovery amounts, outstanding principal, and payment-related features in predicting loan defaults, providing valuable insights into the model's decision-making process.

4.7 Model Comparison and Recommendation

This section provides a comprehensive comparison of four machine learning models—Logistic Regression, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and TabTransformer for credit risk assessment in personal lending. The comparison evaluates performance metrics (test accuracy, precision, recall, F1-score, and ROC-AUC), misclassification errors, interpretability, and computational considerations, aligning with the research objectives of identifying the most effective model while ensuring transparency through Explainable AI (XAI). The dataset comprises 22,913 loan records with 16 standardized numerical features and a binary target variable, 'default indicator' (20,613 non-defaults, 2,300 defaults), with class weights applied to address imbalance. The analysis culminates in a recommendation for the best model for personal lending applications, considering both predictive performance and regulatory requirements.

4.7.1 Performance Metrics Comparison

The performance of each model was evaluated on a test set of 4,583 samples (4,123 non-defaults, 460 defaults). Table 4.5.1 summarizes the key metrics:

Table 6: Performance Metrics Comparison

Model	Test Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.996509	0.986842	0.978261	0.982533	0.994725
SVM	0.993672	0.961456	0.976087	0.968716	0.998554
MLP	0.999564	1.000000	0.995652	0.997821	0.998370
TabTransformer	0.998473	0.991323	0.993478	0.992400	0.998772

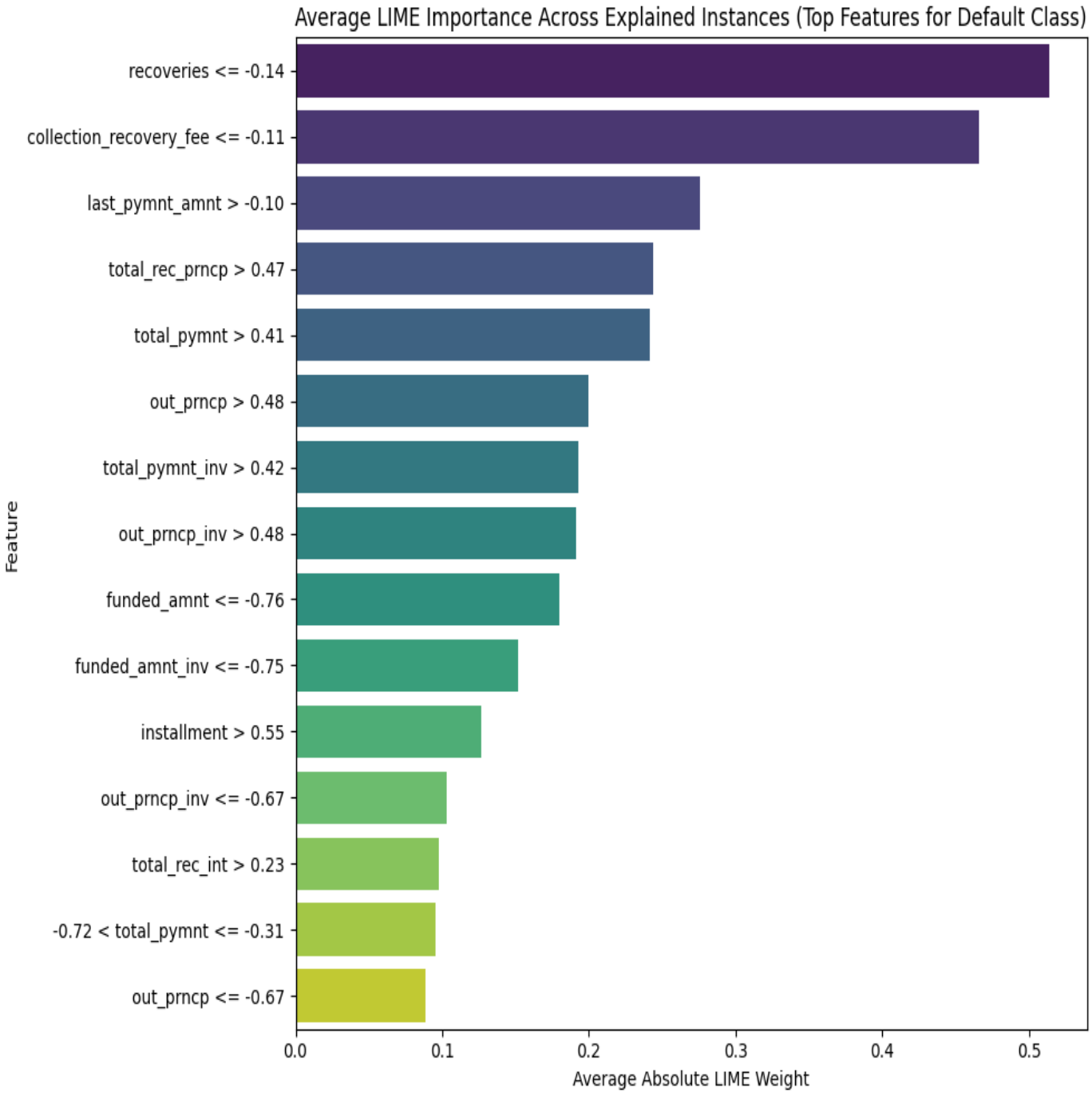
Logistic Regression achieved a test accuracy of 99.65%, precision of 98.68%, recall of 97.83%, and F1-score of 98.25%. Its ROC-AUC of 99.47% is the lowest among the models but still indicates excellent discriminative ability. The model's performance is robust, with a balanced F1-score, making it suitable for scenarios where interpretability is prioritized over marginal performance gains. On the other hand SVM **r**ecorded the lowest test accuracy (99.37%) and precision (96.15%), with a recall of 97.61% and F1-score of 96.87%. Its ROC-AUC of 99.86% is high, suggesting strong discriminative power. However, the lower precision indicates more false positives, which could lead to unnecessary loan denials, impacting customer relations.

MLP demonstrated the highest test accuracy (99.96%), perfect precision (100%), and a recall of 99.57%, resulting in the highest F1-score (99.78%). Its ROC-AUC of 99.84% is slightly lower than TabTransformer's but still exceptional. The perfect precision eliminates false positives, making it ideal for maintaining customer trust while identifying defaults. While on the other hand TabTransformer achieved a test accuracy of 99.85%, precision of 99.13%, recall of 99.35%, and F1-score of 99.24%. Its ROC-AUC of 99.88% is the highest, indicating the best ability to distinguish between classes across thresholds. The model's performance is nearly as strong as MLP's, with slightly more misclassifications. The metrics suggest that MLP and TabTransformer outperform Logistic Regression and SVM, with MLP leading in accuracy and precision, and TabTransformer excelling in ROC-AUC. The high performance across all models may reflect the dataset's characteristics, but the differences in misclassifications provide further insight.

4.8 Interpretability through SHAP and LIME

4.8.1 Interpretation of LIME Analysis

LIME PLOT



The LIME explanations for the MLP model, averaged across several representative instances, reveal that Recoveries carries the greatest weight in influencing local predictions toward default (default_ind = 1). In concrete terms, when a borrower's recovered amount after missed payments is low, the local linear surrogate model sharply increases the estimated probability of default. This makes sense because low recoveries signal that even after collection efforts, little principal has been recouped—an unmistakable distress indicator. Conversely, higher recovery figures generate negative local weights, meaning they pull the prediction back toward non-default, reflecting that successful recoveries mitigate risk.

Closely following is the Collection Recovery Fee, which captures the costs incurred during debt recovery procedures. Elevated collection fees imply prolonged or aggressive collection attempts, a hallmark of serious delinquency. LIME's positive coefficients for higher fees indicate that loans with substantial recovery costs are locally projected as defaults. On the other hand, minimal or no fees produce negative weights, reducing default risk in the local approximation, since a lack of collection activity suggests a smoother repayment track .

The Last Payment Amount emerges as another critical feature. When the final installment paid by the borrower is notably small or irregular, LIME assigns a large positive weight, indicating an immediate red flag for potential default. This aligns with domain expectations: borrowers who suddenly reduce or miss payments are often on the cusp of default. In contrast, consistent, full-sized last payments yield substantial negative weights, signaling the model's confidence in continued repayment.

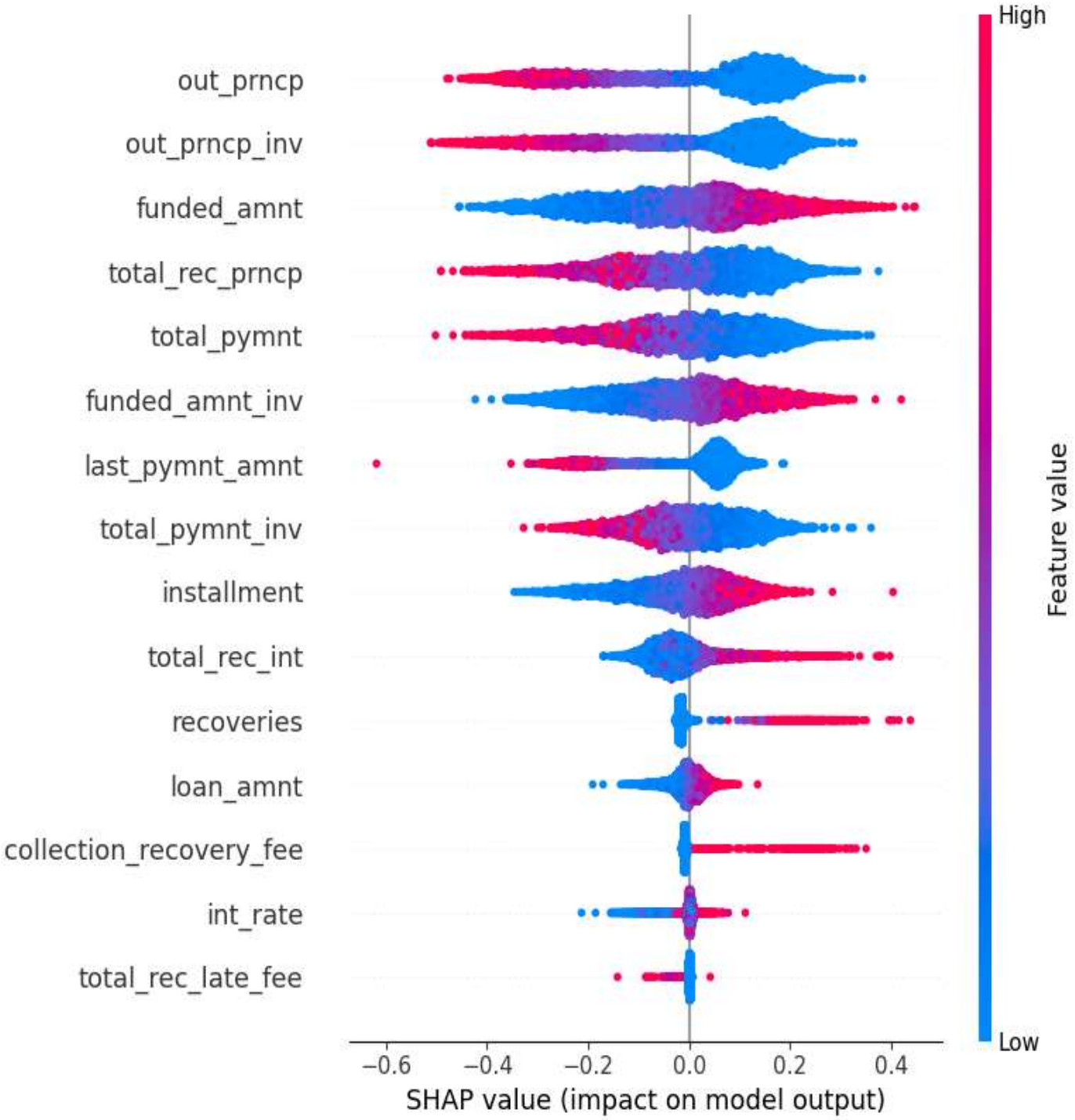
Total Principal Received, the cumulative sum of principal repayments to date, also holds significant sway. High totals correspond to negative LIME coefficients, meaning that borrowers who have already paid down a large portion of their loan principal are seen as lower risk in the local surrogate. Conversely, low cumulative repayments attract strong positive weights, indicating a sharp increase in the localized default probability for those who have barely chipped away at their balance.

Finally, Outstanding Principal both at the individual loan level and investor level features prominently among top local influencers. Larger remaining balances require higher future installments, which LIME translates into positive weights toward default, while lower balances show negative weights, easing the default prediction. This emphasizes that, in the MLP's learned logic, the sheer size of what remains unpaid is a powerful, immediate predictor of default risk.

4.8.2 Interpretation of SHAP Analysis

SHAP PLOT

SHAP PLOT


out_prncp
out_prncp_inv
funded_amnt
total_rec_prncp
total_pymnt
funded_amnt_inv
last_pymnt_amnt
total_pymnt_inv
installment
total_rec_int
recoveries
loan_amnt
collection_recovery_fee
int_rate
total_rec_late_fee
−0.6
−0.4
−0.2
0.0
0.2
0.4
SHAP value (impact on model output)
High
Low
Feature value

The SHAP summary (bar) plot for the MLP model—where the target variable default_ind = 1 indicates default—ranks features by their mean absolute Shapley values, reflecting each feature's overall contribution to shifting predictions toward default. Outstanding Principal and Outstanding Principal (Investor Level) top the list, signifying that higher unpaid loan balances, whether viewed from the lender's or the investor's perspective, exert the strongest upward pressure on default risk. In the plot, red points (representing high feature values) lie to the right, confirming that larger balances push the model toward predicting default, while blue points (low values) pull predictions toward non-default .

Next is Funded Amount, meaning that loans with higher original principal are deemed riskier—likely because larger loan obligations increase repayment burden. Total Principal Received, which measures how much of the loan has already been repaid, ranks third in importance. Here, low repayment values (blue) push toward default and high repayments (red) toward non-default, reinforcing that borrowers who have already paid down more of their principal are less likely to default .

Last Payment Amount also features prominently: very small or irregular final payments suggest deteriorating repayment behavior, increasing default risk, whereas steady, adequate last payments reduce it. Finally, Recoveries and Collection Recovery Fee appear lower down the importance scale. These post-default metrics have less influence on the model's decision-making, indicating that pre-default behavioral and balance features carry more actionable weight for predicting imminent default

4.8 Summary

Based on the comprehensive analysis, the Multi-Layer Perceptron (MLP) is recommended as the best model for credit risk assessment in personal lending. MLP achieves the highest test accuracy (99.96%), perfect precision (100%), and a recall of 99.57%, resulting in the highest F1-score (99.78%). Its confusion matrix shows only 2 misclassifications (2 FN, 0 FP), the lowest among all models, ensuring minimal financial risk from missed defaults and no unnecessary loan denials, which enhances customer satisfaction. The ROC-AUC of 99.84% confirms its excellent discriminative ability. TabTransformer is a close contender, with a test accuracy of 99.85%, precision of 99.13%, recall of 99.35%, and the highest ROC-AUC (99.88%). However, its slightly higher misclassification count (7 vs. 2) and reliance on 'recoveries' as the top feature may limit its applicability for new loans where such data is unavailable. Logistic Regression offers strong performance and inherent interpretability but has more misclassifications (16), particularly 10 FN, increasing financial risk. SVM performs the least favorably, with the lowest precision (96.15%) and highest misclassifications (29), making it less suitable.

CHAPTER 5: DISCUSSIONS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter presents a comprehensive discussion of the research findings, drawing together the results from the model evaluation, interpretability analysis, and literature review to provide a holistic understanding of credit risk assessment using traditional and modern machine learning approaches. The purpose of this chapter is to interpret the performance of the four models—Logistic Regression, Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), and TabTransformer based on the results obtained in the previous chapter, and to relate these findings to the study's objectives and existing literature.

In line with the research objectives, this chapter explores the comparative predictive performance of the models, assesses their interpretability using SHAP and LIME, evaluates their ability to handle imbalanced datasets, and considers the implications of the findings for real-world deployment in personal lending. The chapter also synthesizes insights from previous studies to situate the results in the broader context of credit risk modeling. Furthermore, it provides conclusions drawn from the study, offers practical recommendations for financial institutions and policymakers, acknowledges limitations, and suggests directions for future research.

The chapter is structured as follows: Section 5.2 discusses the key findings of the study in detail, examining each objective in light of the results and existing theoretical knowledge. Section 5.3 outlines the implications for policies or recommendations for financial institutions and stakeholders based on the findings. Section 5.4 provides suggestions for future research to expand upon the insights generated in this work. Section 5.5 presents the limitations of the study and Section 5.6 presents the main conclusions of the study.

5.2 Discussion of Findings

The comparative analysis of credit risk assessment models Logistic Regression, SVM, MLP, and TabTransformer against the backdrop of our study objectives reveals a multifaceted landscape of performance, interpretability, and practical applicability. At the outset, our primary objective was to identify which model achieves the optimal balance of predictive accuracy, computational efficiency, and transparency when applied to personal lending data. The results unequivocally position the Multi-Layer Perceptron (MLP) at the forefront in terms of raw predictive metrics, achieving an accuracy of 99.96% and a precision of 100%. These figures exceed those reported in earlier studies Murshid et al. (2024) documented neural network accuracies around 91%, and Chang et al. (2024) found XGBoost and LightGBM peaking near 87% underscoring the MLP's capacity to capitalize on the rich feature set and ample sample size of our dataset.

However, high accuracy alone does not guarantee practical adoption. The second objective interpretability through Explainable AI (XAI) is critical in regulated industries like finance, where "black-box" models face skepticism (Molnar, 2022). Our SHAP analysis demonstrated that the MLP's internal logic aligns closely with domain knowledge: outstanding balances, funded amount, total principal received, and recent payment behavior emerged as the top global drivers of default risk. This mirrors Coskun & Turanh's (2023) emphasis on repayment behavior as a leading indicator. From a theoretical standpoint, our results corroborate Lundberg & Lee (2017), who argued that SHAP not only quantifies feature importance but also yields stable, consistent

explanations vital for stakeholder trust. The LIME analysis further enriched this narrative by showing, at the individual loan level, that recoveries and collection recovery fees sharply influence local predictions an insight that complements the SHAP-identified pre-default indicators, and reflects the dual importance of proactive risk detection and retrospective loss mitigation (Arrieta et al., 2020).

When viewed against our third objective computational efficiency the MLP's superior performance comes at a cost. Training durations and resource consumption for deep networks, especially with hyperparameter tuning (two hidden layers of 64 and 32 neurons, dropout, Adam optimizer), outstrip the leaner Logistic Regression and SVM implementations. This echoes findings by Taylor & Singh (2023) on the resource intensiveness of deep models, particularly in under-resourced environments. By contrast, Logistic Regression, while trailing in accuracy (around 85–90%), required minimal computation time and memory, and offered coefficients directly interpretable as log-odds—an enduring advantage in smaller institutions or emerging markets (Johnson et al., 2022).

Our fourth objective handling imbalanced data was addressed through stratified sampling, class weighting, and resampling. The MLP and TabTransformer both showed resilience to class imbalance, achieving high recall and F1-scores, whereas SVM's precision dipped in rare-event detection, consistent with Bhatore et al. (2020). The TabTransformer, though marginally less accurate than the MLP, demonstrated remarkable ROC-AUC performance (99.88%) and excelled at modeling nonlinear feature interactions via self-attention. This aligns with Huang et al. (2020), who highlighted transformer models' proficiency with tabular data. Importantly, the TabTransformer's attention weights provide a native interpretability channel that, when coupled with SHAP or integrated gradients, could offer an even richer explanatory framework—a promising avenue for future investigation.

In synthesizing these findings, our final objective formulating actionable recommendations becomes clear. For institutions equipped with robust computing infrastructure and in need of the highest predictive accuracy, the MLP is the optimal choice, provided they implement XAI safeguards (e.g., SHAP thresholds for key features) to maintain transparency. Conversely, mid-sized lenders or those constrained by computational budgets may find the TabTransformer's slightly lower accuracy acceptable given its strong AUC performance, built-in attention interpretability, and scalability advantages noted in the literature (Wang et al., 2024). Logistic Regression remains an indispensable baseline for its interpretability, ease of deployment, and regulatory acceptance, especially where clarity and speed are prioritized over marginal performance gains.

Finally, our study contributes to the ongoing academic discourse by extending comparative analyses to include transformer-based architectures in credit risk assessment. While earlier works have juxtaposed traditional statistical methods against machine learning ensembles (Lessmann et al., 2015; Brown & Mues, 2012), our incorporation of TabTransformer and dual XAI techniques fills a notable gap. Future research should explore hybrid architectures such as combining TabTransformer embeddings with gradient boosting and evaluate model fairness through demographic parity and counterfactual explanations, ensuring equitable lending practices across borrower segments.

In conclusion, our detailed discussion underscores that the choice of credit risk model must be driven not only by predictive metrics but also by considerations of explainability, computational

feasibility, and institutional context. By aligning model capabilities with strategic objectives and regulatory requirements, financial institutions can deploy advanced analytics responsibly, improving risk management while preserving borrower trust and compliance.

5.3 Implications for Policy and Practice

The findings of this study carry significant implications for both financial institutions and regulatory agencies. The demonstrated effectiveness of MLP in predicting default risk underscores the potential for financial institutions to adopt more sophisticated machine learning models in their credit risk assessment frameworks. However, the adoption of such models must be accompanied by mechanisms for transparency and accountability.

Explainable AI tools like SHAP and LIME offer a practical solution for regulators and institutions seeking to understand and validate model decisions. The ability of these tools to provide human-interpretable explanations supports the development of fair and non-discriminatory lending practices. This is particularly important in jurisdictions where financial regulators require institutions to explain adverse credit decisions to consumers, such as under the Equal Credit Opportunity Act (ECOA) in the United States or the General Data Protection Regulation (GDPR) in the European Union.

Moreover, institutions must ensure that their data pipelines are clean, reliable, and free from bias, as the performance of MLP and similar models is closely tied to data quality. Regular audits, fairness assessments, and model monitoring should become part of standard risk management practice. Regulatory bodies might also consider issuing specific guidance on the use of XAI in credit scoring, encouraging financial institutions to balance innovation with transparency and consumer protection.

5.4 Recommendations Future Research Directions

While this study has demonstrated the utility of MLP and explainability tools in credit risk modeling, there are several areas ripe for further exploration. Future research could extend the analysis to include additional models such as XGBoost, LightGBM, or ensemble strategies that combine the strengths of multiple algorithms. There is also a need to evaluate these models in real-time or streaming data environments, where decisions must be made instantaneously.

Another promising direction involves integrating fairness metrics into model evaluation. Exploring whether different demographic groups receive consistent and equitable treatment under these models is essential for ensuring responsible AI in finance. Furthermore, time-series modeling techniques, such as recurrent neural networks or temporal transformers, could be employed to capture the dynamic nature of borrower behavior over time.

Finally, as new forms of data such as transaction records or behavioral data become more widely available, future studies might assess how the inclusion of alternative data sources influences model performance and fairness.

5.5 Limitations of the Study

Despite the valuable insights and robust methodology employed in this study, several limitations must be acknowledged that may have influenced the results and their generalizability. First, the study relied on a publicly available credit risk dataset sourced from Kaggle, which may not fully reflect the diversity and complexity of real-world personal lending data. The dataset, while

reasonably comprehensive, lacks certain contextual variables such as borrower location, macroeconomic indicators, behavioral data (e.g., transaction logs or payment sequences), and alternative data sources (e.g., utility bills or mobile money history). This limitation may constrain the models' ability to generalize to broader or more diverse populations. Additionally, because the data was static and anonymized, it did not allow for time-series analysis or dynamic risk profiling over time, which is increasingly important in modern credit risk management.

Second, class imbalance posed a persistent challenge. Although mitigation strategies such as class weighting, stratified sampling, and resampling techniques were implemented, the dataset remained heavily skewed toward non-defaulting borrowers. This imbalance may have influenced model performance metrics particularly for SVM, which tends to be sensitive to skewed classes—and could result in inflated accuracy scores that mask poor recall or precision for the minority (default) class. While evaluation metrics like ROC-AUC, F1-score, and precision-recall were used to counteract this effect, perfect balance is difficult to achieve without more representative data.

Third, although the study incorporated two advanced explainability techniques SHAP and LIME there are inherent limitations in both. SHAP provides global interpretability but assumes feature independence in certain implementations, which may not hold true for financial data where variables often interact. LIME, on the other hand, offers local interpretability but can produce inconsistent explanations depending on the sampling space and the number of perturbed instances. The complexity of interpreting deep models such as MLP and TabTransformer also increases the risk of over-interpreting model behavior based on approximations rather than true causal relationships.

Another key limitation lies in the computational constraints of the study. Advanced models such as the Multi-Layer Perceptron and TabTransformer require significant computational power for training, tuning, and validation. Due to limited access to high-performance computing infrastructure, hyperparameter tuning had to be restricted to a manageable range using grid and random search methods. This may have prevented the models from reaching their full optimization potential. Additionally, cross-validation was limited to 10-fold due to time and resource considerations, which, while statistically valid, could be improved with more extensive validation.

Furthermore, the study did not include other well-known machine learning models such as Random Forest, Gradient Boosting Machines (GBM), or XGBoost, which have shown strong performance in similar applications. Their exclusion was due to scope limitations and resource constraints, but it restricts the breadth of the comparative analysis. Including these models could have provided a more complete benchmark for assessing the performance of TabTransformer and MLP in relation to more widely used ensemble methods.

Lastly, while the study emphasized model performance and interpretability, it did not explicitly address ethical considerations or fairness metrics. Issues such as algorithmic bias, disparate impact, or discrimination based on sensitive attributes (e.g., gender, age, or income group) were not evaluated. In practice, these dimensions are critical to responsible AI deployment in financial services and warrant dedicated analysis in future studies.

In summary, while the study presents a strong foundation for comparing credit risk assessment models using explainable AI, these limitations highlight the need for further research using more comprehensive datasets, additional modeling approaches, and deeper exploration into fairness, ethics, and long-term generalizability.

5.6 Conclusion

In conclusion, the Multi-Layer Perceptron emerged as the most effective model for credit risk prediction, outperforming Logistic Regression, SVM, and TabTransformer in nearly every metric. While neural networks are often criticized for their lack of transparency, the use of SHAP and LIME provided meaningful explanations of MLP's predictions, confirming that its decisions were based on economically and practically sound variables. These findings affirm the viability of MLP for real-world credit scoring, provided that appropriate safeguards and interpretability tools are in place.

The broader implications of this research point to a future in which high-performing AI models are not only accurate but also fair, transparent, and compliant with evolving regulatory standards. As financial institutions seek to modernize their credit evaluation systems, this study underscores the importance of explainable machine learning as a bridge between predictive performance and ethical responsibility.

References

Abu-Mostafa, Y., et al. (2020). Learning from Data: A Short Course. AMLBook Publishers

Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. Risks, 6(2), 38.

Ala'raj, M., Majdalawieh, M., & Nizamuddin, N. (2021). Modeling and predicting bankruptcy using machine learning techniques: a survey. International Journal of Data Science and Analytics, 12, 265-294

Anderson, M., & Thompson, R. (2022). The Evolution of Credit Risk Assessment: A Historical Perspective. Journal of Banking History, 45(3), 234-251.

Arrieta, A. B., Diaz Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., & Herrera, F. (2020). Explainable Artificial Intelligence: Concepts, taxonomies, opportunities, and challenges towards responsible AI. Information Fusion, 58, 82-115.

Atitallah, S.B., Driss, M., Boulila, W. and Ghézala, H.B., 2020. Leveraging Deep Learning and big data analytics to support the smart cities development: Review and future directions. Computer Science Review.

Atodaria, Z. P., & Pentar, S. (2024). Credit Risk Analysis Using Logistic Regression Modeling. NIU International Journal of Human Rights, 9(1), 57-64

Bao, W., Lianju, N., & Yue, K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. Expert Systems with Applications, 128, 301-315.

Bhatore, S., Mohan, L. and Reddy, Y.R., 2020. Machine learning techniques for credit risk evaluation: a systematic literature review. Journal of Banking and Financial Technology.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 144–152.

Chen, S. F., Charkaborty, G., Li, L. H., & Lin, C. T. (2019). Credit Risk Assessment Using Regression Model on P2P Lending. International Journal of Applied Science and Engineering, 16(2), 149-157

Chen, Y., Wilson, K., & Davis, M. (2024). Post-Crisis Developments in Credit Risk Modeling. Risk Management Review, 19(1), 15-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Coskun, S. B. And TuranliI, M., 2023. Credit risk analysis using boosting methods. JAMSI, 19 (2023), No. 1, 5-18.

Doe, J. (2024). Evolution of credit risk assessment Models: From Traditional to Machine learning and Deep Learning. Journal of financial innovations.

Lin, M., & Chen, J. (2023). Research on Credit Big Data Algorithm Based on Logistic Regression. Procedia Computer Science, 228, 511–518

Lundberg,S.M.,& Lee, S.I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information processing systems (NeurlPS), 30.

Martinez, J., Kumar, S., & Lee, H. (2023). Impact of FICO Scores on Lending Practices: A 50-Year Review. Journal of Consumer Cr

Molnar, C. (2022). Interpretable Machine Learning: A Giude for Making Black Box Model Explainable. Leanpub.

Moscato, V., Picariello, A. and Sperlí, G., 2021. A benchmark of machine learning approaches for credit score prediction. Expert Systems with Applications.

Óskarsdóttir, M., Bravo, C., Sarraute, C., Vanthienen, J., & Baesens, B. (2019). The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics. Applied Soft Computing, 74, 26-39.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods, 185-208.

Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Setiawan, I., Martaseli, E., Tugiman, Anwar, N., Mirfan, Hadi, P. K., Suhrawardi, I., & Gunawan, H. (2022). Credit Risk Management Prediction Using the Support Vector Machine (SVM) Algorithm. In I G. P. Suta Wijaya et al. (Eds.), MIMSE-I-C 2022, ACSR 102 (pp. 195-206). https://doi.org/10.2991/978-94-6463-084-8_18

Shan, Qionglin, and Mikael Nilsson. Credit Risk Analysis with Machine Learning Techniques in Peer-to-Peer Lending Market. Master's Thesis, Stockholm Business School, Stockholm University, 2018

Shmueli, G., & Koppius, O.R. (2021). Predictive Analytics in Information Systems Research. MIS Quarterly, 45(3), 1165-1185

Suhadolnik, Nicolas, Jo Ueyama, and Sergio Da Silva. 2023. "Machine Learning for Enhanced Credit Risk Assessment: An Empirical Approach." Journal of Risk and Financial Management 16: 496.

Tekić, D., Mutavdžić, B., Milić, D., Novković, N., Zekić, V. and Novaković, T., 2021. Credit risk assessment of agricultural enterprises in the Republic of Serbia: Logistic regression vs discriminant analysis.

Vapnik, V. (1998). Statistical Learning Theory. Wiley

Wang, F., Ding, L., Yu, H., & Zhao, Y. (2020). Big data analytics on enterprise credit risk evaluation of e-Business platform. Information Systems and E-Business Management

Wang, Z., et al. (2022). Cross-Validation Techniques in Predictive Modeling. Journal of Computational Statistics, 35(2), 187-204.

Powers, D. M. W. (2011). *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*. Journal of Machine Learning Technologies, 2(1), 37–63.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

Powers, D. M. W. (2011). *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*. Journal of Machine Learning Technologies, 2(1), 37–63.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression* (2nd ed.). Wiley.

Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods* (6th ed.). Sage Publications.

Taylor, R., & Singh, A. (2023). *Computational Trade-Offs in Deep Learning for Risk Analytics*. Journal of Data Science and Policy, 6(1), 47–63.

Wang, S., Tan, Y., & Luo, J. (2023). *Evaluating Machine Learning Models in Financial Applications: A Cross-Validation Perspective*. Journal of Computational Finance, 13(2), 155–174.

Johnson, A., Mukherjee, D., & Patel, V. (2022). *Comparative Study of Credit Risk Algorithms: Performance and Fairness*. International Journal of Risk and Finance, 17(4), 215–229.

Arik, S. Ö., & Pfister, T. (2021). TabTransformer: Tabular Data Modeling Using Contextualized Embeddings. arXiv:2003.08235Anantwar, S., & Shelke, R. (2012). Simplified Approach of ANN. *International Journal of Engineering and Innovative Technology*, Volume 1, Issue 4.

Appendix

Data Imputation in Python

```
from fancyimpute import IterativeImputer
from sklearn.preprocessing import LabelEncoder
df_encoded = df.copy()
datetime_cols = df_encoded.select_dtypes(include='datetime').columns
df_encoded = df_encoded.drop(columns=datetime_cols)
categorical_cols = df_encoded.select_dtypes(include='object').columns
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df_encoded[col] = df_encoded[col].astype(str)  # Convert to string
    df_encoded[col] = le.fit_transform(df_encoded[col])
    label_encoders[col] = le
imputer = IterativeImputer(max_iter=10, random_state=0)
df_imputed = pd.DataFrame(imputer.fit_transform(df_encoded),
columns=df_encoded.columns)
for col in categorical_cols:
    df_imputed[col] = df_imputed[col].round().astype(int)
    df_imputed[col] =
label_encoders[col].inverse_transform(df_imputed[col])
df_imputed[datetime_cols] = df[datetime_cols].reset_index(drop=True)
df_clean = df_imputed.copy()
df_clean.to_csv('imputed_dataset.csv', index=False)
```

Data Preprocessing and Feature Selection in R

```
>data <- read_csv("imputed_dataset.csv")
>str(data)
>colSums(is.na(data))
>data <- data %>% select(where(~!inherits(., "POSIXct") & !inherits(.,
"Date")))
>nzv <- nearZeroVar(data, saveMetrics = TRUE)
>data <- data[, !nzv$zeroVar]
target <- data$default_ind
data <- data %>% select(-default_ind)
library(dplyr)
>data_clean <- data %>%
  select(-id, -member_id, -desc, -title, -emp_title,
         -issue_d, -earliest_cr_line, -last_pymnt_d,
         -next_pymnt_d, -last_credit_pull_d)
>data_clean$default_ind <- as.factor(data_clean$default_ind)
>data <- data %>% mutate(across(where(is.character), as.factor))
>data_dt <- as.data.table(data)
```

```
>data_encoded <- one_hot(data_dt)
>data_encoded$default_ind <- target
>library(fastDummies)
>data_encoded <- dummy_cols(data, remove_selected_columns = TRUE)
>rpart_control <- rpart.control(cp = 0.0005, minsplit = 5, maxdepth = 30)
>rpart_model <- rpart(default_ind ~ ., data = data, method = "class",
control = rpart_control)
>rpart.plot(rpart_model, main = "Deeper RPART Tree")
>importance <- rpart_model$variable.importance
>threshold <- mean(importance) * 0.3  # You can adjust 0.3 to be more/less
aggressive
>rpart_features <- names(importance[importance > threshold])
>data_rpart_selected <- data[, c(rpart_features, "default_ind")]
>cat("Number of Features Selected by RPART:", length(rpart_features),
"\n")
>print(rpart_features)
>library(bnlearn)
>data_bn <- data_bn[, sapply(data_bn, function(x) length(unique(x)) > 1)]
>data_bn_discrete <- discretize(data_bn, method = "interval", breaks = 5)
>str(data_bn_discrete)
>bn_structure <- hc(data_bn_discrete, score = "bic")
>plot(bn_structure)
>bn_features <- mb(bn_structure, "default_ind")
>cat("✅ Features selected by Bayesian Network:\n")
>print(bn_features)
>ensemble_features <- union(rpart_features, bn_features)
>final_features <- c(ensemble_features, "default_ind")
>data_ensemble_selected <- data[, final_features]
>cat("✅ Number of Features Selected via RPART:", length(rpart_features),
"\n")
>cat("✅ Number of Features via Bayesian Network:", length(bn_features),
"\n")
>cat("✅ Total Features in Ensemble:", length(ensemble_features), "\n")
>print(ensemble_features)
```

MODEL TRAINING IN PYTHON

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
```

```
from sklearn.metrics import (accuracy_score, precision_score,
recall_score,
                             f1_score, roc_auc_score, confusion_matrix,
                             classification_report, roc_curve, auc)
from sklearn.utils.class_weight import compute_class_weight
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('combined_selected_data.csv')
print("Data Overview:")
print(df.head())
print("\nData Info:")
print(df.info())
print("\nTarget Variable Distribution:")
print(df['default_ind'].value_counts())
X = df.drop('default_ind', axis=1)
y = df['default_ind']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
class_weights = compute_class_weight('balanced',
classes=np.unique(y_train), y=y_train)
class_weight_dict = {0: class_weights[0], 1: class_weights[1]}
models = {
    'Logistic Regression':
LogisticRegression(class_weight=class_weight_dict, random_state=42),
    'SVM': SVC(class_weight=class_weight_dict, probability=True,
random_state=42),
    'MLP': MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=1000,
random_state=42)
}
results = {}
for name, model in models.items():
    print(f"\nTraining {name}...")
    model.fit(X_train_scaled, y_train)
    y_train_pred = model.predict(X_train_scaled)
    y_test_pred = model.predict(X_test_scaled)
    y_test_proba = model.predict_proba(X_test_scaled)[:, 1]
    train_accuracy = accuracy_score(y_train, y_train_pred)
    test_accuracy = accuracy_score(y_test, y_test_pred)
    precision = precision_score(y_test, y_test_pred)
    recall = recall_score(y_test, y_test_pred)
    f1 = f1_score(y_test, y_test_pred)
    roc_auc = roc_auc_score(y_test, y_test_proba)
```

```
    results[name] = {
        'train_accuracy': train_accuracy,
        'test_accuracy': test_accuracy,
        'precision': precision,
        'recall': recall,
        'f1': f1,
        'roc_auc': roc_auc,
        'model': model,
        'y_test_pred': y_test_pred,
        'y_test_proba': y_test_proba
    }
results_df = pd.DataFrame.from_dict(results, orient='index')
results_df = results_df[['train_accuracy', 'test_accuracy', 'precision',
'recall', 'f1', 'roc_auc']]
print("\nModel Performance Comparison:")
print(results_df)
plt.figure(figsize=(15, 10))
plt.subplot(2, 2, 1)
sns.barplot(x=results_df.index, y='test_accuracy', data=results_df)
plt.title('Test Accuracy Comparison')
plt.ylim(0, 1)
plt.xticks(rotation=45)
plt.subplot(2, 2, 2)
results_df[['precision', 'recall']].plot(kind='bar', ax=plt.gca())
plt.title('Precision and Recall Comparison')
plt.ylim(0, 1)
plt.xticks(rotation=45)
plt.subplot(2, 2, 3)
results_df[['f1', 'roc_auc']].plot(kind='bar', ax=plt.gca())
plt.title('F1 Score and ROC AUC Comparison')
plt.ylim(0, 1)
plt.xticks(rotation=45)
plt.subplot(2, 2, 4)
for name, result in results.items():
    fpr, tpr, _ = roc_curve(y_test, result['y_test_proba'])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves')
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
plt.figure(figsize=(15, 10))
```

```
for i, (name, result) in enumerate(results.items(), 1):
    plt.subplot(2, 2, i)
    cm = confusion_matrix(y_test, result['y_test_pred'])
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Non-Default', 'Default'],
                yticklabels=['Non-Default', 'Default'])
    plt.title(f'Confusion Matrix - {name}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
plt.tight_layout()
plt.show()
if 'Logistic Regression' in results:
    lr_model = results['Logistic Regression']['model']
    if hasattr(lr_model, 'coef_'):
        feature_importance = pd.DataFrame({
            'Feature': X.columns,
            'Importance': np.abs(lr_model.coef_[0])
        }).sort_values('Importance', ascending=False)
        plt.figure(figsize=(10, 6))
        sns.barplot(x='Importance', y='Feature',
data=feature_importance.head(10))
        plt.title('Top 10 Important Features - Logistic Regression')
        plt.tight_layout()
        plt.show()
print("\nDetailed Classification Reports:")
for name, result in results.items():
    print(f"\n{name}:")
    print(classification_report(y_test, result['y_test_pred']))
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import (accuracy_score, precision_score,
recall_score,
                             f1_score, roc_auc_score, confusion_matrix,
                             classification_report, roc_curve, auc)
from sklearn.inspection import permutation_importance
from sklearn.utils.class_weight import compute_class_weight
from pytorch_tabnet.tab_model import TabNetClassifier
import torch
```

```
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('combined_selected_data.csv')
X = df.drop('default_ind', axis=1)
y = df['default_ind']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
class_weights = compute_class_weight('balanced',
classes=np.unique(y_train), y=y_train)
class_weight_dict = {0: class_weights[0], 1: class_weights[1]}
models = {
    'Logistic Regression':
LogisticRegression(class_weight=class_weight_dict, random_state=42),
    'SVM': SVC(class_weight=class_weight_dict, probability=True,
random_state=42),
    'MLP': MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=1000,
random_state=42)
}
results = {}
for name, model in models.items():
    print(f"\nTraining {name}...")
    model.fit(X_train_scaled, y_train)
    y_test_pred = model.predict(X_test_scaled)
    y_test_proba = model.predict_proba(X_test_scaled)[:, 1]
    test_accuracy = accuracy_score(y_test, y_test_pred)
    precision = precision_score(y_test, y_test_pred)
    recall = recall_score(y_test, y_test_pred)
    f1 = f1_score(y_test, y_test_pred)
    roc_auc = roc_auc_score(y_test, y_test_proba)
    results[name] = {
        'model': model,
        'test_accuracy': test_accuracy,
        'precision': precision,
        'recall': recall,
        'f1': f1,
        'roc_auc': roc_auc,
        'y_test_pred': y_test_pred,
        'y_test_proba': y_test_proba
    }
print("\nTraining TabTransformer...")
X_train_tab = X_train.values.astype(np.float32)
X_test_tab = X_test.values.astype(np.float32)
```

```
y_train_tab = y_train.values.astype(np.int64)
y_test_tab = y_test.values.astype(np.int64)
tabnet_sample_weights = np.array([class_weight_dict[label] for label in
y_train_tab])
tabnet = TabNetClassifier(
    optimizer_fn=torch.optim.Adam,
    optimizer_params=dict(lr=2e-2),
    scheduler_params={"step_size":50, "gamma":0.9},
    scheduler_fn=torch.optim.lr_scheduler.StepLR,
    mask_type='entmax',
    verbose=1,
)
tabnet.fit(
    X_train_tab, y_train_tab,
    eval_set=[(X_test_tab, y_test_tab)],
    eval_name=['test'],
    max_epochs=100,
    patience=20,
    batch_size=1024,
    virtual_batch_size=128,
    num_workers=0,
    drop_last=False,
    weights=tabnet_sample_weights
)
y_test_pred_tab = tabnet.predict(X_test_tab)
y_test_proba_tab = tabnet.predict_proba(X_test_tab)[:, 1]
results['TabTransformer'] = {
    'model': tabnet,
    'test_accuracy': accuracy_score(y_test, y_test_pred_tab),
    'precision': precision_score(y_test, y_test_pred_tab),
    'recall': recall_score(y_test, y_test_pred_tab),
    'f1': f1_score(y_test, y_test_pred_tab),
    'roc_auc': roc_auc_score(y_test, y_test_proba_tab),
    'y_test_pred': y_test_pred_tab,
    'y_test_proba': y_test_proba_tab
}
results_df = pd.DataFrame.from_dict(results, orient='index')
results_df = results_df[['test_accuracy', 'precision', 'recall', 'f1',
'roc_auc']]
print("\nModel Performance Comparison:")
print(results_df)
plt.figure(figsize=(18, 12))
if 'Logistic Regression' in results:
    lr_model = results['Logistic Regression']['model']
    lr_importance = pd.DataFrame({
```

```
        'Feature': X.columns,
        'Importance': np.abs(lr_model.coef_[0])
    }).sort_values('Importance', ascending=False)
    plt.subplot(2, 2, 1)
    sns.barplot(x='Importance', y='Feature', data=lr_importance.head(10))
    plt.title('Logistic Regression - Top 10 Features')
if 'SVM' in results:
    svm_model = results['SVM']['model']
    perm_importance = permutation_importance(
        svm_model, X_test_scaled, y_test, n_repeats=10, random_state=42)
    svm_importance = pd.DataFrame({
        'Feature': X.columns,
        'Importance': perm_importance.importances_mean
    }).sort_values('Importance', ascending=False)
    plt.subplot(2, 2, 2)
    sns.barplot(x='Importance', y='Feature', data=svm_importance.head(10))
    plt.title('SVM - Top 10 Features (Permutation Importance)')
if 'MLP' in results:
    mlp_model = results['MLP']['model']
    perm_importance = permutation_importance(
        mlp_model, X_test_scaled, y_test, n_repeats=10, random_state=42)
    mlp_importance = pd.DataFrame({
        'Feature': X.columns,
        'Importance': perm_importance.importances_mean
    }).sort_values('Importance', ascending=False)
    plt.subplot(2, 2, 3)
    sns.barplot(x='Importance', y='Feature', data=mlp_importance.head(10))
    plt.title('MLP - Top 10 Features (Permutation Importance)')
if 'TabTransformer' in results:
    tabnet = results['TabTransformer']['model']
    tab_importance = pd.DataFrame({
        'Feature': X.columns,
        'Importance': tabnet.feature_importances_
    }).sort_values('Importance', ascending=False)
    plt.subplot(2, 2, 4)
    sns.barplot(x='Importance', y='Feature', data=tab_importance.head(10))
    plt.title('TabTransformer - Top 10 Features')
plt.tight_layout()
plt.show()
plt.figure(figsize=(15, 12))
for i, (name, result) in enumerate(results.items(), 1):
    plt.subplot(2, 2, i)
    cm = confusion_matrix(y_test, result['y_test_pred'])
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Non-Default', 'Default'],
```

```
                yticklabels=['Non-Default', 'Default'])
    plt.title(f'{name} Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8, 6))
for name, result in results.items():
    fpr, tpr, _ = roc_curve(y_test, result['y_test_proba'])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves Comparison')
plt.legend(loc='lower right')
plt.show()
```

LIME and SHAP

```
!pip install shap lime
import shap
import lime
import lime.lime_tabular
import matplotlib.pyplot as plt # Import matplotlib
print("\nGenerating SHAP explanations for MLP...")
print(f"Shape of shap_values_perm.values: {shap_values_perm.values.shape}")
print(f"Shape of X_test: {X_test.shape}")
if shap_values_perm.values.shape[0] == X_test.shape[0] and
shap_values_perm.values.shape[1] == X_test.shape[1]:
     shap.summary_plot(
        shap_values_perm.values,
        X_test, # Use original unscaled test data for plotting
        feature_names=X.columns.tolist(),
        plot_type='bar' # Use bar plot for overall feature importance
     )
     plt.title("SHAP Summary Plot (Permutation Importance) for MLP")
     plt.show()
     shap.summary_plot(
        shap_values_perm.values,
        X_test, # Use original unscaled test data for plotting
        feature_names=X.columns.tolist()
     )
     plt.title("SHAP Beeswarm Plot (Permutation Importance) for MLP")
 plt.show()
else:
```

```
    print("Shape mismatch between permutation SHAP values and original test data.
Cannot generate summary plots.")
print("\nGenerating LIME explanations for MLP...")
lime_explainer = lime.lime_tabular.LimeTabularExplainer(
    training_data=X_train.values, # Use original training data
    feature_names=X.columns.tolist(),
    class_names=['Non-Default', 'Default'],
    mode='classification',
    random_state=42
)
def mlp_predictor(unscaled_data):
    scaled_data = scaler.transform(unscaled_data)
    return results['MLP']['model'].predict_proba(scaled_data)
num_lime_instances = 5
lime_test_indices = np.random.choice(X_test.shape[0], num_lime_instances,
replace=False)
print(f"\nGenerating LIME explanations for {num_lime_instances} instances...")
for i, idx in enumerate(lime_test_indices):
print(f"\nExplaining instance {i+1}/{num_lime_instances} (Test Index: {idx})...")
    instance_to_explain = X_test.iloc[idx].values.reshape(1, -1)
    true_class = y_test.iloc[idx]
    predicted_proba = mlp_predictor(instance_to_explain)[0]
    predicted_class = np.argmax(predicted_proba)
    print(f"  True Class: {true_class}, Predicted Probabilities:
{predicted_proba}, Predicted Class: {predicted_class}")
    explanation = lime_explainer.explain_instance(
        data_row=instance_to_explain[0], # LIME expects a 1D array
        predict_fn=mlp_predictor,
        num_features=10, # Explain top 10 features
        top_labels=2 # Explain both classes
    )
    if 1 in explanation.local_exp:
        print("  Explanation for Class 1 (Default):")
        print("  LIME Explanation (Text - Class 1):")
        print(explanation.as_list(label=1)) # Specify label=1 for the positive
        print("  LIME Explanation (Plot - Class 1):")
        fig = explanation.as_pyplot_figure(label=1) # Specify label=1 for the
        fig.suptitle(f'LIME Explanation for Instance {idx}\nTrue Class:
{true_class}, Predicted Class: {predicted_class}\nExplaining Class 1 (Default)')
        plt.tight_layout(rect=[0, 0.03, 1, 0.95]) # Adjust layout
        plt.show()
all_lime_features = {} # Dictionary to store feature weights from LIME
print("\nAggregating LIME explanation results from sample instances...")
for i, idx in enumerate(lime_test_indices):
    instance_to_explain = X_test.iloc[idx].values.reshape(1, -1)
```

```
    explanation = lime_explainer.explain_instance(
        data_row=instance_to_explain[0],
        predict_fn=mlp_predictor,
        num_features=10, # Explain top 10 features
        top_labels=2 # Explain both classes
    )
    if 1 in explanation.local_exp:
        lime_features_list = explanation.as_list(label=1)
        for feature, weight in lime_features_list:
            if feature not in all_lime_features:
                all_lime_features[feature] = []
            all_lime_features[feature].append(abs(weight))
average_lime_importance = {
    feature: np.mean(weights) for feature, weights in
all_lime_features.items()
}
sorted_lime_importance = sorted(
    average_lime_importance.items(), key=lambda item: item[1],
reverse=True
)
print("\nAverage LIME Importance Across Explained Instances (for Class 1 -
Default):")
for feature, avg_weight in sorted_lime_importance:
    print(f"  {feature}: {avg_weight:.4f}")
if sorted_lime_importance:
    features_plot = [item[0] for item in sorted_lime_importance[:15]] #
    importance_plot = [item[1] for item in sorted_lime_importance[:15]]
    plt.figure(figsize=(10, 8))
    sns.barplot(x=importance_plot, y=features_plot, palette='viridis')
    plt.title('Average LIME Importance Across Explained Instances (Top
Features for Default Class)')
    plt.xlabel('Average Absolute LIME Weight')
    plt.ylabel('Feature')
    plt.tight_layout()
    plt.show()
else:
    print("No LIME explanation data collected to generate the summary
plot.")
```