

Predictive Risk Assessment Credit Scoring Using Supervised Learning

By

KHOR WEI HENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)

DIGITAL ECONOMY TECHNOLOGY

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2025

COPYRIGHT STATEMENT

© 2025 Khor Wei Heng. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Digital Economy Technology at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and profound appreciation to my supervisor, **Cik Nurul Syafidah binti Jamil**, and my moderator, **Dr. Kiran Adnan**, for providing me with this valuable opportunity to explore the field of supervised learning in credit risk assessment. Their consistent guidance and unwavering support were instrumental throughout the completion of this project. Whenever I encounter challenges in developing and comparing machine learning models for credit scoring, their insightful advice and expertise helped me navigate through difficulties and find effective solutions. Their mentorship has been invaluable to my academic growth and the success of this research on predictive credit assessment methodologies. I am deeply thankful for their dedication, patience, and encouragement throughout this journey.

ABSTRACT

This study explores the application of supervised learning models within credit scoring, aiming to revolutionize risk assessment in lending decisions. The primary goal involves comparing these advanced methodologies against conventional credit assessment techniques to ascertain their effectiveness in determining creditworthiness. In response to the escalating complexity of financial transactions and the wealth of available data, this research seeks to elevate the precision and efficiency of credit risk evaluation. Supervised learning, known for its ability to learn from labelled datasets, presents an opportunity to redefine credit scoring by leveraging historical credit information.

The core focus is on assessing the predictive capabilities of supervised learning algorithms—specifically Logistic Regression, Random Forest, K-Nearest Neighbours, Support Vector Machines and Gradient Boosting—against established credit scoring methods. By harnessing the power of these modern techniques and analysing intricate credit patterns, this research endeavours to deliver more accurate credit risk assessments. It strives to surpass the existing industry norms by using machine learning models to refine credit evaluation processes.

Beyond academia, this study aims to introduce substantial advancements in credit risk assessment methodologies. It seeks to bridge the gap between conventional and contemporary approaches by revolutionizing credit scoring. By tapping into supervised learning's potential, this research aspires to produce predictive credit scoring models that surpass industry standards, fostering more reliable lending decisions.

The objectives encompass the development of more accurate credit scoring models, identification of influential creditworthiness factors, and the enhancement of model interpretability. The fusion of traditional credit assessment wisdom with the cutting-edge capabilities of supervised learning intends to empower financial institutions with sophisticated tools for making informed, expedited, and more reliable lending decisions. Ultimately, this research aspires to create a paradigm shift in credit risk assessment, enabling financial entities to navigate evolving market conditions with confidence and precision.

Area of study:

Machine learning applications in financial risk assessment

Predictive modeling for credit scoring

Comparative analysis of traditional vs. AI-based credit evaluation methods.

Keywords:

Supervised learning

Credit scoring

Risk assessment

Financial decision-making

Predictive analytics

TABLE OF CONTENTS

TITLE PAGE	i
COPYRIGHT STATEMENT	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xii
CHAPTER 1 Research Background	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Motivation	3
1.4 Research Objectives	3
1.5 Project Scope and Direction	4
1.6 Research Contributions	4
1.7 Report Organization	5
CHAPTER 2 LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Limitations	8
2.3 Previous Research Result	9
CHAPTER 3 Methodology	11
3.1 Supervised Learning Model	11
a) Random Forest	11
b) Logistics Regression	11
c) K-Nearest neighbors	12
d) Support Vector Machine	12
e) Gradient Boosting	12

3.2	Justification for Algorithm Selection	13
a)	Random Forest	13
b)	Logistics Regression	13
c)	K-Nearest Neighbors	13
d)	Support Vector Machine	13
e)	Gradient Boosting	14
3.3	Explanation of steps in Research Methodology flowchart	15
3.4	Evaluation of the Models	17
CHAPTER 4 DATA PREPARATION AND EXPLORATORY ANALYSIS		18
4.1	Introduction of Dataset	18
4.2	Data Features Description	19
4.3	Data Pre-processing	21
4.3.1	Handling Missing Value	21
4.3.2	Data Type Conversion	23
4.3.3	Data Normalization	24
4.3.4	One-Hot Encoding	25
4.4	Exploratory Data Analysis	27
4.5	Handle Outlier	50
4.6	Feature Selection	58
4.7	Handling Class Imbalance	59
4.7.1	Introduction to Class Imbalance	59
4.7.2	Resampling Techniques	59
CHAPTER 5 Model Training & Fine Tuning		62
5.1	Baseline Model	62
5.1.1	Baseline Model Implementation	62
5.1.2	Baseline Model Results	63
5.1.3	Analysis of Baseline Results	66
5.2	Fine-Tuning	67
5.2.1	Fine-Tuning Methodology	67
5.2.2	Hyperparameter Grids	68

5.2.3 Pipeline and Data Scaling Strategy	73
5.2.4 Fine-Tuning Results	75
CHAPTER 6 Final Model Evaluation and System Dashboard	87
6.1 Final Model Results	87
6.2 Summary Comparison of Best Models	99
6.3 Final Recommendation	101
6.4 Feature Importance of Model Using Explainable AI	103
6.5 System Website (Front End)	105
6.6 Unit Testing for Loan Risk Analyzer Website	115
6.6.1 Unit Testing 1 – File Upload Page	115
6.6.2 Unit Testing 2 – Data Analysis Process	115
6.6.3 Unit Testing 3 – Result Dashboard	116
6.6.4 Unit Testing 4 – Individual Loan Assessment Form	117
6.6.5 Unit Testing 5 – Model Information Page	118
6.6.6 Unit Testing 6 – Export Functionality	118
6.7 Implementation Issues and Challenges	120
CHAPTER 7 Conclusion	122
7.1 Summary of Findings	122
7.2 Implications for Credit Risk Assessment	122
7.3 Limitations of the Study	123
7.4 Directions for Future Research	124
7.5 Final Remarks	124
REFERENCES	125
POSTER	127

LIST OF FIGURES

Figure Number	Title	Page
Figure 3.3.1	Research Methodology	16
Figure 4.1.1	Dataset Information	18
Figure 4.3.1.1	Missing Value before Preprocessing	21
Figure 4.3.1.2	Missing Value after Preprocessing	22
Figure 4.3.2.1	Data Type before Conversion	23
Figure 4.3.2.2	Data Type after Conversion	24
Figure 4.3.3.1	Data before Normalization	24
Figure 4.3.3.2	Data after Normalization	25
Figure 4.3.4.1	Data before One-Hot Encoding	25
Figure 4.3.4.2	Data after One-Hot Encoding	26
Figure 4.4.1	Distribution of Age	27
Figure 4.4.2	Distribution of Income	28
Figure 4.4.3	Distribution of Employment Length	29
Figure 4.4.5	Distribution of Loan Amount Applied For	30
Figure 4.4.6	Distribution of Interest Rate on the Loan	31
Figure 4.4.7	Distribution of Loan Approval Status	32
Figure 4.4.8	Distribution of Loan Amount as a Percentage of Income	33
Figure 4.4.9	Distribution of Length of the Applicant's Credit History	34
Figure 4.4.10	Distribution of Home Ownership Status	35
Figure 4.4.11	Distribution of Intent (Purpose of Loan)	36
Figure 4.4.12	Distribution of Default (Whether the Applicant has Defaulted on a Loan Previously)	38
Figure 4.4.13	Pair Plot of Feature (Income, Amount, Rate, Percent_Income)	39
Figure 4.4.14	Pair Plot of Feature (Income, Percent_Income, Cred_length, Rate)	41
Figure 4.4.15	Heat Map	43
Figure 4.4.16	Bar Plot of Variables according to Home	44
Figure 4.4.17	Bar Plot of Variables according to Intent	46

Figure 4.4.18	Bar Plot of Variables according to Default	48
Figure 4.5.1	Visualizations of Outlier on Age	50
Figure 4.5.2	Visualizations of Age Distribution after Capping	50
Figure 4.5.3	Visualizations of Outlier on Income	52
Figure 4.5.4	Visualizations of Income Distribution after Capping	52
Figure 4.5.5	Visualizations of Outlier on Employment Length	54
Figure 4.5.6	Visualizations of Employment Length Distribution after Capping	54
Figure 4.5.7	Visualizations of Outlier on Loan Amount as a Percentage of Income	56
Figure 4.5.8	Visualizations of Loan Amount as a Percentage of Income Distribution after Capping	56
Figure 4.6.1	Removing Id Column	58
Figure 5.1.1	Model Configuration	62
Figure 5.1.2	SMOTE Baseline Model Result	63
Figure 5.1.3	SMOTEENN Baseline Model Result	63
Figure 5.1.4	Tomek Baseline Model Result	64
Figure 5.1.5	Borderline Baseline Model Result	64
Figure 5.1.6	SMOTETomek Baseline Model Result	65
Figure 5.1.7	ADASYN Baseline Model Result	65
Figure 5.2.1	Hyperparameter Grids of Random Forest	68
Figure 5.2.2	Hyperparameter Grids of Logistic Regression	69
Figure 5.2.3	Hyperparameter Grids of K-Nearest Neighbors	70
Figure 5.2.4	Hyperparameter Grids of Support Vector Machine	71
Figure 5.2.5	Hyperparameter Grids of Gradient Boosting	71
Figure 5.2.6	Data Scaling Strategy	73
Figure 5.2.7	SMOTE Fine-Tuning Results	75
Figure 5.2.8	SMOTE Fine-Tuning Results with Data Scaling	76
Figure 5.2.9	Tomek Fine-Tuning Results	77
Figure 5.2.10	Tomek Fine-Tuning Results with Data Scaling	78
Figure 5.2.11	SMOTETomek Fine-Tuning Results	79
Figure 5.2.12	SMOTETomek Fine-Tuning Results with Data Scaling	80

Figure 5.2.13	SMOTEENN Fine-Tuning Results	81
Figure 5.2.14	SMOTEENN Fine-Tuning Results with Data Scaling	82
Figure 5.2.15	Borderline Fine-Tuning Results	83
Figure 5.2.16	Borderline Fine-Tuning Results with Data Scaling	84
Figure 5.2.17	Adasyn Fine-Tuning Results	85
Figure 5.2.18	Adasyn Fine-Tuning Results with Data Scaling	86
Figure 6.1.1	SMOTE Final Results	87
Figure 6.1.2	SMOTE Final Results with Data Scaling	88
Figure 6.1.3	Tomek Final Results	89
Figure 6.1.4	Tomek Final Results with Data Scaling	90
Figure 6.1.5	SMOTETomek Final Results	91
Figure 6.1.6	SMOTETomek Final Results with Data Scaling	92
Figure 6.1.7	SMOTEENN Final Results	93
Figure 6.1.8	SMOTEENN Final Results with Data Scaling	94
Figure 6.1.9	Borderline Final Results	95
Figure 6.1.10	Borderline Final Results with Data Scaling	96
Figure 6.1.11	ADASYN Final Results	97
Figure 6.1.12	ADASYN Final Results with Data Scaling	98
Figure 6.3.1	ROC Curve of Gradient Boosting Model with TomekLinks	101
Figure 6.3.2	Confusion Metrix of Gradient Boosting Model with TomekLinks	102
Figure 6.5.1	Batch Analysis Page with Sample Data	105
Figure 6.5.2	Batch Analysis Result with Sample Data	105
Figure 6.5.3	Risk Distribution Tab with Sample Data	106
Figure 6.5.4	Feature Impact Tab with Sample Data	107
Figure 6.5.5	Loan Characteristics Tab with Sample Data	108
Figure 6.5.6	Data Table Tab with Sample Data	109
Figure 6.5.7	Individual Loan Assessment Page with Default Values	110
Figure 6.5.8	Risk Assessment Result with Sample Data	111
Figure 6.5.9	Model Information Page	112
Figure 6.5.10	Feature Details Section	113

LIST OF TABLES

Table Number	Title	Page
Table 2.3.1	Table of Previous Work 1	9
Table 2.3.2	Table of Previous Work 2	9
Table 2.3.3	Table of Previous Work 3	10
Table 6.2.1	Top 5 Model Configurations by Overall Performance	99
Table 6.2.2	Best Model Configuration for Each Algorithm	99
Table 6.4.1	Feature Importance Ranking from Explainable AI Analysis	103
Table 6.6.1	Unit Testing 1 - File Upload Section	115
Table 6.6.2	Unit Testing 2 – Data Analysis Process	115
Table 6.6.3	Unit Testing 3 – Results Dashboard	116
Table 6.6.4	Unit Testing 4 - Individual Loan Assessment Form	117
Table 6.6.5	Unit Testing 5 - Model Information Page	118
Table 6.6.6	Unit Testing 6 – Export Functionality	118

CHAPTER 1

Research Background

This study focuses on leveraging Supervised Learning models for predictive risk assessment in credit scoring. Acknowledging the critical importance of accurate risk evaluation in lending decisions, its main goal is to use Supervised Learning algorithms in Machines Learning, for credit scoring tasks. The research aims to compare these advanced methodologies with established techniques used in traditional credit assessment processes.

The core of this study is comparing traditional credit scoring techniques to the prediction accuracy of supervised learning models including logistic regression, random forest, k-nearest neighbours, support vector machines and gradient boosting. The objective is to find out whether these modern methods improve or outperform the current processes for determining creditworthiness. Through the analysis of complicated patterns in credit data and the integration of multiple relevant variables, the study seeks to determine whether machine learning models produce more accurate outcomes. Financial decision-makers need to know this information to improve their credit risk assessment methods under changing market circumstances.

1.1 Introduction

Credit scoring is a fundamental tool used to evaluate an individual's or business's creditworthiness in the ever-changing financial services industry[1] Although conventional techniques for credit scoring have proven successful, the increasing complexity of financial dealings and the abundance of accessible data demand a reassessment of current approaches. This project embarks on a journey into the realm of supervised learning, a potent subset of machine learning, to revolutionize credit scoring.

The main objective is to create a predictive credit score model by utilising supervised learning capabilities. Supervised learning, characterized by its ability to learn from labelled datasets, presents an opportunity to enhance the precision and efficiency of credit risk assessment. By using past credit data, this research seeks to improve the accuracy of credit risk identification while also advancing credit evaluation procedures.

This study project is more than just an academic investigation; it is also a calculated move to make real advancements in credit risk assessment[2] The approach to credit scoring has changed significantly with the addition of supervised learning techniques, with the aim of producing a prediction model that not only satisfies but beyond industry norms.

Through this research, it is aspired to contribute innovative insights to the field of credit scoring, empowering financial institutions to make more informed and reliable lending decisions.

1.2 Problem Statement

The problem currently revolves around the inefficiencies and uncertainties in the credit risk assessment processes within the banking sector. Despite using various supervised learning methods for credit scoring, there remain challenges in accurately and swiftly identifying deserving loan candidates.

The existing manual assessment and approval stages in the lending process often lack accuracy. This results in delays and potential inaccuracies in decision-making. Key factors that significantly influence loan approval, such as age, job, marital status, education, and financial indicators, are known, but their systematic employment to predict loan eligibility is lacking[3]

The primary issue is the need for an improved Loan Prediction System that harnesses machine learning techniques, specifically supervised learning using algorithms like random forest, logistic regression, k-nearest neighbours, support vector machines and gradient boosting. The objective is to enhance accuracy, efficiency, and speed in identifying individuals who are likely to be reliable borrowers by leveraging these influential factors[4]

The current system's limitations impact both the banking industry and loan applicants. For banks, it results in lengthy decision-making processes, potentially leading to missed opportunities and increased operational costs. For loan applicants, delays in loan approval might impede their financial plans and opportunities.

Hence, there's a critical need to develop a more reliable and expedited method for assessing creditworthiness. This would facilitate informed loan approval decisions, benefiting both the lending institutions and loan applicants. By creating a system that can swiftly and accurately evaluate creditworthiness, the aim is to streamline the lending process, reduce risks for banks, and provide better opportunities for deserving applicants.

1.3 Motivation

The motivation of this research is to revolutionize credit risk assessment by leveraging supervised learning techniques in machine learning. Traditional methods have limitations, prompting a shift towards sophisticated algorithms like logistic regression, random forest, k-nearest neighbours, support vector machines and gradient boosting. The objective of this project is to enhance the precision of risk evaluation, empowering lenders to make swift and informed decisions on loan approvals. This approach helps financial institutions accurately and confidentially manage the ever-evolving market conditions.

1.4 Research Objectives

The research objectives are identified as follows:

- 1) To perform data preparation and data preprocessing to encompass cleaning, transforming to optimize the dataset for supervised learning in predictive risk assessment credit scoring.
- 2) To perform feature extraction to find the most influential variables impacting creditworthiness, aiming to enhance the predictive capability of the models.
- 3) To perform supervised learning experiment using logistic regression, random forest, k-nearest neighbors, support vector machines and gradient boosting.
- 4) To perform evaluation metrics including precision, recall, accuracy, and F1-score, the trained models will be rigorously assessed on test data to gauge their predictive performance and suitability for credit scoring risk assessment.

1.5 Project Scope and Direction

The scopes of the research are as below:

- (a) This research aims to develop models.
- (b) This research evaluates and compares the performance of five supervised learning algorithms—logistic regression, random forest, k-nearest neighbors, support vector machines and gradient boosting—for credit scoring within a specified timeline of one and a half years. The study will focus exclusively on these five methods only. By assessing key performance metrics such as accuracy, precision, and F1-Score, the goal is to identify the most suitable algorithm for predictive credit scoring risk assessment within the given constraints.
- (c) This research investigates the importance of different features and variables in credit risk assessment. It uses techniques like feature selection, extraction, and engineering to identify the most influential factors impacting creditworthiness. Determine the optimal combination of variables to enhance the predictive power of supervised learning models.
- (d) This research focuses on making supervised learning models more interpretable and explainable in the sense of credit scoring. Develop methodologies to interpret model predictions to understand the rationale behind credit risk assessments made by these models.

1.6 Research Contributions

The research contributions are:

- (a) Development of supervised learning models (logistic regression, random forest, k-nearest neighbors, support vector machines and gradient boosting) to enhance the accuracy of credit risk assessment as well as refining these models to provide financial institutions with more reliable tools for decision-making in credit assessments.

- (b) Identification and prioritization of influential variables impacting creditworthiness to enhance the predictive power of credit risk assessment models by pinpointing crucial factors influencing credit assessments.
- (c) Improving the interpretability of supervised learning models used in credit scoring to make complicated model predictions more understandable and transparent, empowering banks to make informed decisions in credit assessment scenarios.

1.7 Report Organization

This report is organized into four chapters, structured as follows:

Chapter 1: Research Background. This chapter introduces the project, providing an overview of credit scoring and the application of supervised learning in this domain. It outlines the problem statement, motivation, research objectives, project scope, and expected contributions of the study.

Chapter 2: Literature Review. This chapter presents a thorough analysis of the body of research on credit scoring and the use of machine learning methods in this area is provided in this chapter. It highlights the shortcomings of present methodologies, talks about the state of research, and compiles the results of earlier studies that are used to guide this effort.

Chapter 3: Methodology. This chapter details the research methodology employed in this study. It provides an in-depth explanation of the supervised learning models used (Random Forest, Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and Gradient Boosting), outlines the steps in the research process, and describes the evaluation metrics for assessing model performance.

Chapter 4: Data Preparation and Exploratory Analysis. This chapter focuses on the initial stages of the data analysis process. It introduces the dataset used in the study, describes the features, outlines the data cleaning and preprocessing steps, and presents the results of exploratory data analysis. This chapter also covers the handling of missing values, feature engineering, and the treatment of outliers. The report concludes with a

comprehensive list of references and appendices, including weekly progress reports and a project poster. This organization provides a logical flow of information, starting from the project background and literature review, moving through the methodology, and concluding with the initial data analysis steps. It sets the stage for the subsequent phases of the project, which will involve model development, evaluation, and final analysis.

Chapter 5: Model Training & Fine Tuning. This chapter details the implementation of baseline models for all five supervised learning algorithms and evaluates their performance across six different resampling techniques. It then describes the comprehensive fine-tuning methodology, including hyperparameter optimization and the development of specialized data scaling pipelines tailored to each algorithm's requirements. The chapter concludes with detailed performance analyses of the fine-tuned models.

Chapter 6: Final Model Evaluation and System Dashboard. This chapter presents the final evaluation results of all models on the test dataset, compares the performance of different configurations, and provides a justified recommendation of the optimal model for credit risk assessment. It also explores feature importance using explainable AI techniques and documents the implementation of a web-based system dashboard for credit risk analysis, including interface design and comprehensive unit testing.

Chapter 7: Conclusion. This final chapter summarizes the key findings of the research, discusses the implications for credit risk assessment in financial institutions, acknowledges limitations of the study, and suggests promising directions for future research. The chapter concludes with final remarks on the significance of the work for the field of predictive credit scoring.

CHAPTER 2

Literature Reviews

2.1 Introduction

Credit scoring, a vital aspect of gauging financial risk, has gained increased significance due to the intense competition and challenges faced by financial institutions. Li et al [5] provides a summary and classification of techniques employed in credit scoring while introducing a novel approach known as ensemble learning. It delves into current shortcomings, highlighting the need to shift from static credit scoring to dynamic behavioral scoring and to optimize revenue by minimizing Type I and Type II errors.

Mir et al [6] stated that a reliable credit risk assessment system remains crucial for the seamless and profitable operation of any financial institution. In today's evolving economy, where loan defaults are on the rise, financial authorities face mounting challenges in accurately evaluating loan applications and mitigating the risks associated with defaulters.

To address this, Mir et al [6] recommends that lending institutions use a machine learning algorithm that is intended to accurately assess credit risk and forecast possible loan defaulters. In order to detect defaulters, the study does a comparative analysis using refined supervised learning methods such as Support Vector Machine, Random Forest, Extreme Gradient Boosting, and Logistic Regression. To reduce dimensionality, methods like Principal Component Analysis and Recursive Feature Elimination with Cross-Validation are used. Evaluation metrics including F1 score, AUC score, prediction accuracy, precision, and recall are employed to assess each model's performance[7].

Among these models, the optimized Support Vector Machine coupled with Recursive Feature Elimination and Cross-Validation emerges as a promising combination for identifying loan defaulters. Consequently, the proposed model stands poised to aid financial institutions in precisely pinpointing loan defaulters, thereby averting potential losses.

2.2 Limitations

Before diving into the details, let's explore the landscape of credit assessment and the challenges faced in credit risk modelling. Lenders rely on various rating and scoring models to assess the creditworthiness of applicants. However, the abundance of models and methodologies raises concerns about the accuracy of estimates generated solely from historical loan data due to potential biases[8] In the realm of credit risk models for underwriting, a notable hurdle is the bias present in the training data. Typically, these models are constructed using data from approved credit applicants, leading to a non-random sample heavily influenced by credit policies and past loan performances. This skew in sampling may distort predictions regarding loan default probabilities when assessing new borrower applications [9] Besides, in credit risk modelling, logistic regression is widely used but faces accuracy challenges when deployed due to a lack of negative samples and an inability to learn nonlinear data patterns [10]. Also, Probst P [11] mentioned higher numbers of trees are thought to enhance performance. Yet, the analysis of real data illustrates instances where metrics like accuracy and AUC decrease as the number of trees increases. They provide theoretical evidence for why this phenomenon occurs and suggest it's limited to highly specific data scenarios. Liu W et al [12]discovered that ensemble algorithms, categorized into bagging and boosting ensembles, offer significant potential for credit scoring. Yet, certain issues require further attention: (1) Because bagging algorithms rely on training targets for feature augmentation, they enhance feature variety while preserving the training target, which may increase the statistical similarity of prediction outcomes. (2) Boosting ensemble algorithms avoid prediction similarity concerns but operate solely on original credit features, leading to a lack of diversity in features.

2.3 Previous Research Result

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.81	0.98	0.88	87%
Gradient Boosting	0.81	0.91	0.86	85%
Decision Tree	0.82	0.87	0.81	81%
Random Forest	0.81	0.97	0.89	87%

Table 2.3.1: Table of Previous Work 1

Karim M et al [9] conducted a comparative analysis of all models generated to determine the top-performing model. The evaluation is based on metrics like accuracy, precision, recall, F1-score, and the ROC-AUC curve. The ROC-AUC values for Logistic Regression and XGBoost stand at 0.92, while for Decision Tree and Random Forest, they're 0.81 and 0.93 respectively. Notably, Logistic Regression and XGBoost exhibit strong performance, but Random Forest emerges as the best-performing model. The Decision Tree, however, performs the poorest among the models assessed. Table 2.3.1 presents a comparison of the four classifiers—Logistic Regression, XGBoost, Decision Tree, and Random Forest—based on their confusion matrices. Logistic Regression boasts the highest recall at 0.98, followed closely by Random Forest at 0.97. Despite Decision Tree having the highest precision, its overall performance in other metrics is notably lower. Both Random Forest and Logistic Regression achieve the highest accuracy at 87%, with Random Forest also leading in F1-score. Considering the results from Table 2.3.1, it's evident that Random Forest delivers satisfactory performance among the models considered.

Model	Accuracy	F1 Score	AUC
Logistic Regression	74.43%	74.37%	0.84
SVM	77.64%	77.94%	0.87
Decision Tree	84.68%	84.71%	0.85
MLP	84.61%	83.45%	0.93
AdaBoost	87.67%	87.37%	0.95
Random Forest	88.96%	88.45%	0.96
Gradient Boosting DT	90.99%	90.37%	0.97

Table 2.3.2: Table of Previous Work 2

According to Tian Z et al [13], table 2.3.2 presents the comparison of LR, SVM, CART, MLP, AdaBoost, Random Forest, and Gradient Boosting Decision Tree under normalized conditions, assessing their accuracy, f1 score, and AUC value. The application of Boosting notably enhances the performance of the Decision Tree model. Specifically, the Gradient Boosting Decision Tree (90.99%, 90.37%, 0.97) outperforms LR (74.43%, 74.37%, 0.84), SVM (77.64%, 77.94%, 0.87), CART (84.68%, 84.71%, 0.85), and MLP (84.61%, 83.45%, 0.93) across these three metrics. Compared to other ensemble learning methods like AdaBoost (87.67%, 87.37%, 0.95) and Random Forest (88.96%, 88.45%, 0.96), the Gradient Boosting Decision Tree (90.99%, 90.37%, 0.97) exhibits more significant improvements across the metrics. This superior performance underscores the higher accuracy and stronger generalization ability of the Gradient Boosting Decision Tree model with respect to this dataset.

Model	AUC	Precision	Recall	Accuracy
KNN	0.63	61.36%	30.65%	82.27%
Decision Tree	0.92	85.93%	87.80%	94.68%
Random Forest	0.92	97.16%	85.16%	96.53%
Naïve Bayes	0.5	36%	0.09%	79.99%
Logistic Regression	0.56	60.81%	14.77%	81.05%

Table 2.3.3: Table of Previous Work 3

Wang Y et al [14] concluded the performance metrics of various machine learning models in a classification task. Both the Decision Tree and Random Forest models exhibit notably high AUC values (0.92) alongside strong Precision (85.93% - 97.16%) and commendable Accuracy (94.68% - 96.53%), indicating their robustness in correctly classifying instances. The K-Nearest Neighbors (KNN) model presents moderate performance with an AUC of 0.63, an Accuracy of 82.27%, and a Precision of 61.36%, albeit with a comparatively lower Recall (30.65%). Conversely, Naïve Bayes and Logistic Regression models demonstrate inferior performance, evident in their lower AUC values (0.5 - 0.56) and notably poor Recall rates (0.09% - 14.77%). These findings highlight the strengths of the Decision Tree and Random Forest models in accurate classification, while indicating the limitations of Naïve Bayes and Logistic Regression in correctly identifying positive instances.

CHAPTER 3

Methodology

Supervised learning models such random forest, logistic regression, k-nearest neighbors, support vector machines, and gradient boosting will be used in this project. The following are the explanations.:

3.1 Supervised Learning Model

a) Random Forest

Random Forest is a powerful ensemble learning technique used in machine learning for classification and regression issues [15] It constructs a few decision trees during training, from which it derives the mode of the classes (classification) or the mean prediction (regression). The way Random Forest generates diverse trees is by using random parts of the training datasets and a random subset of features for each tree. This technique reduces overfitting and raises the model's accuracy by merging the predictions of multiple trees [16] Random Forest is a well-liked option in many machines learning predictive modelling scenarios due to its capacity to manage big datasets, retain acceptable accuracy, and offer insights into feature relevance.

b) Logistic Regression

Logistic regression is a basic statistical method for predicting the likelihood of a categorical result in binary classification situations. Despite its name, it is a classification algorithm rather than a regression algorithm. It models the link between the dependent variable and one or more independent variables by estimating probabilities using a logistic function. The output is limited to a range of 0 to 1, signifying the likelihood that the input falls into a specific class. Because of its ease of use, interpretability, and potency in situations requiring an understanding and prediction of the link between attributes and the likelihood of an outcome, Logistic Regression is frequently employed. It does this by fitting a linear decision boundary to the input data[17].

c) K-Nearest Neighbors

The k-Nearest Neighbour (kNN) algorithm is a straightforward yet effective machine learning technique, suitable for both classification and regression tasks, though it is more commonly applied in classification. k-Nearest Neighbour works by grouping data into cohesive clusters or subsets and classifying new input data based on its similarity to previously trained data. It assigns the input to the class to which it has the most similar neighbours. Despite its effectiveness, k-Nearest Neighbour has several limitations. This paper discusses the k-Nearest Neighbour method and examines its modified versions from previous research. These variations address the shortcomings of k-Nearest Neighbour, offering a more efficient approach[18].

d) Support Vector Machines

Support vector machines (SVMs) are machine learning techniques initially developed for classification and later extended to tasks like regression and outlier detection. They are based on statistical learning theory and convex optimization, finding the optimal hyperplane to maximize the margin between classes. Support vector machines handle high-dimensional data well and use kernel functions for non-linear relationships, making them versatile. Support vector machines' ability to manage high-dimensional data and capture non-linear relationships through kernel functions enhances accuracy in predicting borrower behaviour, crucial for informed lending decisions. Their scalability and interpretability further solidify their role as a cornerstone in modern credit risk assessment, ensuring efficient and reliable financial risk management[19]

e) Gradient Boosting

Gradient Boosting is an ensemble learning technique used for both classification and regression tasks, renowned for its ability to produce strong predictive models. It works by sequentially building weak learners, typically decision trees, to correct the errors of the preceding model. Gradient Boosting creates trees serially, as opposed to Random Forest, which develops numerous trees independently. This minimises errors by highlighting misclassified cases in subsequent models. Gradients are used to optimise a loss function, and with each iteration, the predictive performance of the model is gradually improved. This approach is well-liked in many machine learning contests and

applications where prediction accuracy is crucial since it tends to produce extremely accurate models by concentrating on regions where prior models underperformed[20]

3.2 Justification for Algorithm Selection

a) Random Forest:

Random Forest handles non-linear relationships well, which is crucial given the complex interactions between financial variables in the dataset. It provides feature importance, helping identify key factors in credit risk assessment, and is robust to outliers, making it suitable for high-dimensional data with multiple features. Additionally, it performs effectively with both numerical and categorical variables present in the dataset, such as the 'Home' and 'Intent' categories.

b) Logistic Regression:

Logistic Regression is well-suited for binary classification problems like credit risk (good loan vs. bad loan) and offers easily interpretable results, which is valuable in the financial sector where model transparency is often required. It performs well with large datasets, handles both continuous and categorical variables, and provides probability scores, which are useful for credit risk quantification.

c) K-Nearest Neighbours:

K-Nearest Neighbours (KNN) is a non-parametric method that can capture complex patterns without assuming a particular form of relationship, making it effective when decision boundaries are irregular, as may be the case in credit risk assessment. It performs well when there are clear clusters in the feature space, which the exploratory data analysis suggests might be present.

d) Support Vector Machines:

Support Vector Machines (SVM) are effective in high-dimensional spaces, making them suitable for the multi-feature dataset. They are versatile with different kernel functions, allowing SVMs to capture non-linear relationships. Additionally, they are robust against overfitting, especially in text classification problems, which could be beneficial if incorporate textual data in future iterations.

e) Gradient Boosting:

Gradient Boosting is known for its high performance and ability to handle complex datasets, making it suitable for the project. It can automatically manage feature interactions, which is valuable given the potential interplay between financial indicators in credit risk. Additionally, it is robust to outliers and missing data, addressing some of the data quality issues identified in the exploratory analysis. Gradient Boosting also provides feature importance, offering insights into the most critical factors in credit risk assessment.

Conclusion

These algorithms encompass a range of approaches, from simple linear models like Logistic Regression to complex ensemble methods such as Random Forest and Gradient Boosting. This diversity allows us to capture various aspects of the credit risk problem and compare their performances effectively. The selection is informed by the nature of the dataset, which includes both numerical variables (e.g., 'Age', 'Income') and categorical ones (e.g., 'Home', 'Intent'), along with the potential for non-linear relationships between features and credit risk. Balancing the need for interpretability in some cases, such as with Logistic Regression, and prioritize high performance in others, like Gradient Boosting is also important. Additionally, the possibility of complex feature interactions, which ensemble methods can capture effectively, makes this range of algorithms suitable for a thorough exploration of the credit risk prediction problem. By employing these five diverse algorithms, they are being used to leverage the strengths of each method to develop a robust and accurate predictive model.

3.3 Explanation of steps in Research Methodology flowchart

The flowchart in Figure 4.1 details a research methodology for predictive risk assessment in credit scoring using supervised learning. The process initiates with Data Collection, potentially looping back if data quality issues arise, followed by essential Data Preprocessing steps (like cleaning, normalization, feature engineering) and Exploratory Data Analysis. A crucial phase is Dataset Splitting, where the data is divided into three distinct subsets: Training (60%), Validation (20%), and Testing (20%). The Train Dataset is used by the selected Supervised Learning Method (e.g., random forest, logistic regression, k-nearest neighbours, support vector machines, gradient boosting) to develop the predictive Model. Concurrently, the model's performance is iteratively evaluated using the Validate Dataset ("Evaluate the model using validation dataset"). An explicit feedback loop exists where, if performance is deemed unsatisfactory based on validation metrics, the process loops back to "Adjust hyperparameters and retrain" using the training data. This iterative tuning continues until the model achieves satisfactory performance on the validation set, indicated by "Fine tune completed, performance good". At this point, the finalized model proceeds to the "Final Evaluation" stage. Here, the completely unseen Test Dataset is used ("Test") to obtain an unbiased assessment of the model's generalization capability. The methodology concludes ("End") after this final performance evaluation.

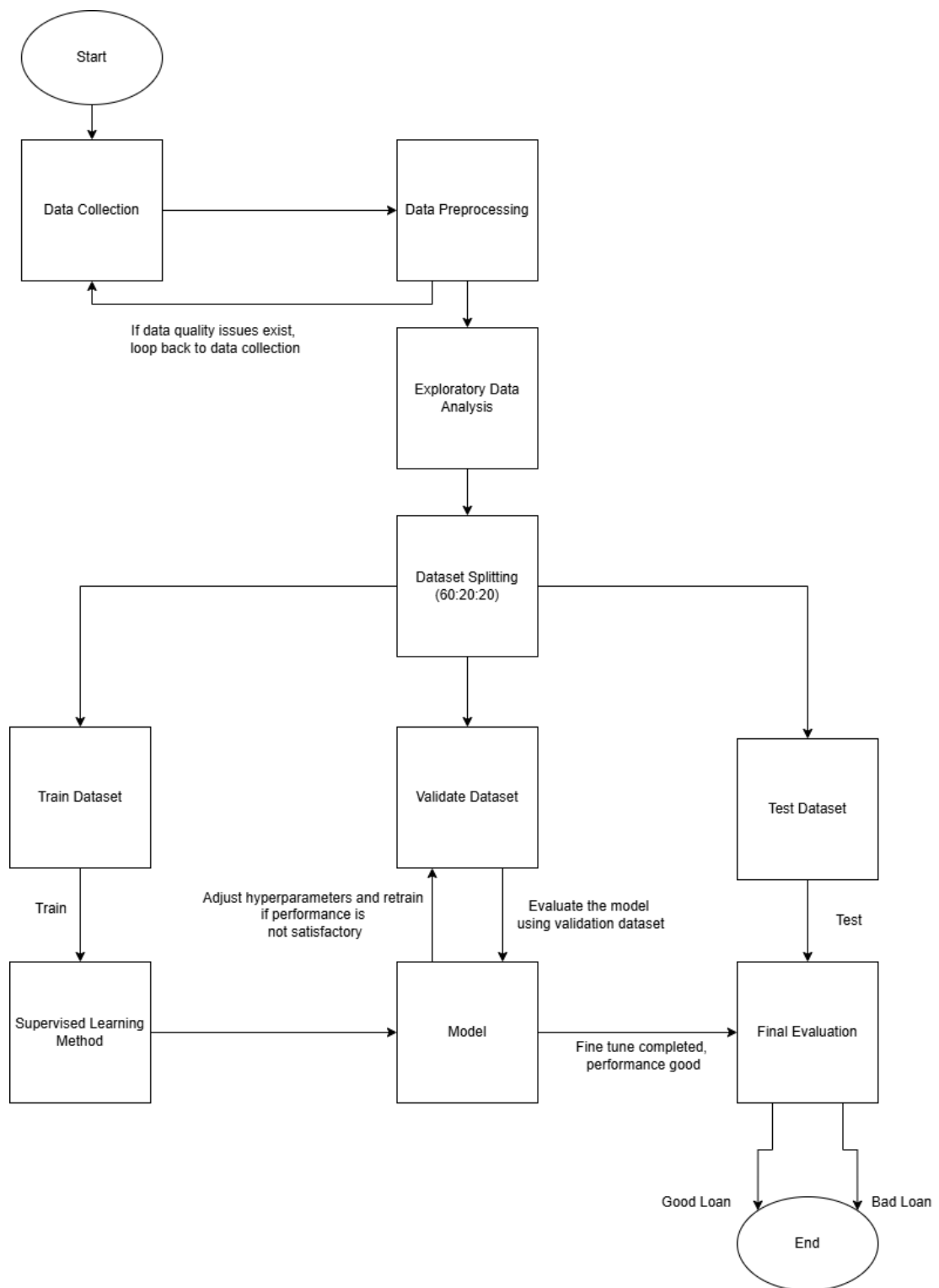


Figure 3.3.1: Research Methodology

3.4 Evaluation of the Models

To assess the efficacy of the completed predictive risk assessment credit scoring model developed in Python through Google Colaboratory, a comprehensive evaluation strategy will be employed. The model will be tested using a diverse set of performance metrics, including accuracy, precision, recall, and F1 score. This evaluation process will involve the utilization of popular machine learning libraries such as Scikit-learn, allowing for the efficient implementation of supervised learning algorithms. The model's generalizability will be validated through cross-validation, and its performance will be compared to benchmark models to gauge its relative effectiveness. Additionally, the evaluation will extend to a separate validation dataset, ensuring the model's applicability to unseen data. Visualization tools like matplotlib and seaborn will be leveraged within Google Colab for insightful analyses, including confusion matrices and other relevant visualizations. This multifaceted approach aims to provide a thorough understanding of the model's predictive capabilities and guide further refinement if needed.

Chapter 4

Data Preparation and Exploratory Data Analysis

4.1 Introduction of Dataset

```
Dataset Shape: (32581, 12)
```

Dataset Info:
 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 32581 entries, 0 to 32580
 Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	Id	32581 non-null	int64
1	Age	32581 non-null	int64
2	Income	32581 non-null	int64
3	Home	32581 non-null	object
4	Emp_length	31686 non-null	float64
5	Intent	32581 non-null	object
6	Amount	32581 non-null	int64
7	Rate	29465 non-null	float64
8	Status	32581 non-null	int64
9	Percent_income	32581 non-null	float64
10	Default	32581 non-null	object
11	Cred_length	32581 non-null	int64

dtypes: float64(3), int64(6), object(3)
 memory usage: 3.0+ MB

Summary Statistics:

	Id	Age	Income	Emp_length	Amount
count	32581.000000	32581.000000	3.258100e+04	31686.000000	32581.000000
mean	16290.006139	27.734600	6.607485e+04	4.789686	9589.371106
std	9405.479594	6.348078	6.198312e+04	4.142630	6322.086646
min	0.000000	20.000000	4.000000e+03	0.000000	500.000000
25%	8145.000000	23.000000	3.850000e+04	2.000000	5000.000000
50%	16290.000000	26.000000	5.500000e+04	4.000000	8000.000000
75%	24435.000000	30.000000	7.920000e+04	7.000000	12200.000000
max	32780.000000	144.000000	6.000000e+06	123.000000	35000.000000

	Rate	Status	Percent_income	Cred_length
count	29465.000000	32581.000000	32581.000000	32581.000000
mean	11.011695	0.218164	0.170203	5.804211
std	3.240459	0.413006	0.106782	4.055001
min	5.420000	0.000000	0.000000	2.000000
25%	7.900000	0.000000	0.090000	3.000000
50%	10.990000	0.000000	0.150000	4.000000
75%	13.470000	0.000000	0.230000	8.000000
max	23.220000	1.000000	0.830000	30.000000

Figure 4.1.1: Dataset Information

The data for this study on predictive risk assessment in credit scoring was obtained from Kaggle, a popular platform for data science and machine learning datasets. This dataset, titled "Loan Applicant Data for Credit Risk Analysis", contains 32,581 records of loan applicants, each with 12 features including demographic information, financial indicators, and loan details. The dataset is stored as a pandas DataFrame, occupying approximately 3.0+ MB of memory.

The dataset contains both numerical and categorical variables. Categorical variables include home ownership status, loan intent, and default status. Numerical variables are

stored as int64 or float64 data types, while categorical variables are stored as object type.

Data quality analysis reveals some missing values:

- Employment length: 895 missing entries
- Interest rate: 3,116 missing entries All other variables have complete data for all 32,581 entries.

The 'Status' (Fully Paid, Charged Off, Current) column in the dataset represents the target variable for this research's supervised learning models in credit risk assessment. It indicates the outcome of a loan. A value of 0 indicates a "good" loan (either fully paid or current). A value of 1 likely indicates a "bad" loan (charged off). The mean of 0.218164 suggests that approximately 21.82% of the loans in the dataset have been charged off, while 78.18% are either fully paid or current.

This comprehensive dataset provides a solid foundation for our credit risk assessment study, offering a range of variables that could potentially influence loan default probability. The presence of some missing data, particularly in the 'Emp_length' and 'Rate' columns, will require careful handling during the data preprocessing stage.

4.2 Data Features Description

This dataset contains 12 data features. The data features are introduction as listed in this below.

- ID: Unique identifier for each loan applicant. A distinct value assigned to each loan applicant to uniquely identify them in the dataset.
- Age: Represents the age of the loan applicant, likely in years.
- Income: Indicates the annual income of the loan applicant.
- Home: Denotes the home ownership status of the applicant (Own, Mortgage, Rent).
- Emp_Length: Represents the number of years the applicant has been employed.
- Intent: Specifies the purpose for which the loan is being taken (e.g., education, home improvement).
- Amount: The amount of money the applicant has applied to borrow.
- Rate: The interest rate assigned to the loan.

CHAPTER 4 DATA PREPARATION AND EXPLORATORY ANALYSIS

- **Status:** Represents the current status of the loan. "Fully Paid" indicates the loan was repaid, "Charged Off" suggests it was written off as a loss, and "Current" means it is still being repaid.
- **Percent_Income:** This is the ratio of the loan amount to the applicant's income, expressed as a percentage.
- **Default:** Indicates whether the applicant has previously defaulted on a loan (Yes, No).
- **Cred_Length:** The number of years the applicant has had credit.

4.3 Data Pre-processing

4.3.1 Handling Missing value

Before preprocessing:

```
Missing values in original data:
Id          0
Age         0
Income      0
Home        0
Emp_length  895
Intent      0
Amount      0
Rate       3116
Status      0
Percent_income  0
Default     0
Cred_length  0
dtype: int64
```

Figure 4.3.1.1: Missing Value before Preprocessing

Original dataset contained missing values primarily in the 'Emp_length' and 'Rate' columns. These were addressed using median imputation during preprocessing.

After preprocessing:

Missing values after imputation:

```

Id          0
Age         0
Income      0
Home        0
Emp_length  0
Intent      0
Amount      0
Rate        0
Status      0
Percent_income  0
Default     0
Cred_length 0
dtype: int64

```

Summary statistics after imputation:

	Id	Age	Income	Emp_length	Amount \
count	32581.000000	32581.000000	3.258100e+04	32581.000000	32581.000000
mean	16290.006139	27.734600	6.607485e+04	4.767994	9589.371106
std	9405.479594	6.348078	6.198312e+04	4.087372	6322.086646
min	0.000000	20.000000	4.000000e+03	0.000000	500.000000
25%	8145.000000	23.000000	3.850000e+04	2.000000	5000.000000
50%	16290.000000	26.000000	5.500000e+04	4.000000	8000.000000
75%	24435.000000	30.000000	7.920000e+04	7.000000	12200.000000
max	32780.000000	144.000000	6.000000e+06	123.000000	35000.000000

	Rate	Status	Percent_income	Cred_length
count	32581.000000	32581.000000	32581.000000	32581.000000
mean	11.009620	0.218164	0.170203	5.804211
std	3.081611	0.413006	0.106782	4.055001
min	5.420000	0.000000	0.000000	2.000000
25%	8.490000	0.000000	0.090000	3.000000
50%	10.990000	0.000000	0.150000	4.000000
75%	13.110000	0.000000	0.230000	8.000000
max	23.220000	1.000000	0.830000	30.000000

Figure 4.3.1.2: Missing Value after Preprocessing

After preprocessing, there are no missing values in the dataset, confirming the success of the imputation strategy.

4.3.2 Data Type Conversion

Converted 'Home' and 'Intent' columns to categorical data type.

Before Conversion:

```
Original Data:
   Id  Age  Income    Home  Emp_length  Intent  Amount  Rate  Status  \
0   0   22   59000    RENT    123.0  PERSONAL   35000  16.02      1
1   1   21    9600     OWN     5.0  EDUCATION    1000  11.14      0
2   2   25    9600  MORTGAGE     1.0   MEDICAL    5500  12.87      1
3   3   23   65500    RENT     4.0   MEDICAL   35000  15.23      1
4   4   24   54400    RENT     8.0   MEDICAL   35000  14.27      1

   Percent_income  Default  Cred_length
0             0.59        Y           3
1             0.10        N           2
2             0.57        N           3
3             0.53        N           2
4             0.55        Y           4

Original Data Types:
Id                int64
Age               int64
Income            int64
Home              object
Emp_length        float64
Intent            object
Amount            int64
Rate              float64
Status            int64
Percent_income    float64
Default           object
Cred_length        int64
dtype: object
```

Figure 4.3.2.1.: Data Type before Conversion

After Conversion:

```

--- After Data Type Conversion ---
      Home      Intent
0      RENT     PERSONAL
1       OWN     EDUCATION
2  MORTGAGE     MEDICAL
3      RENT     MEDICAL
4      RENT     MEDICAL

Data Types after conversion:
Home      category
Intent    category
dtype: object

```

Figure 4.3.2.2: Data Type after Conversion

This step ensures that these columns are treated as categorical variables in subsequent analyses.

4.3.3 Data Normalization

Applying StandardScaler to normalize all numeric columns (excluding 'ID')

Data before normalization:

```

--- Before Normalization ---
   Id  Age  Income  Emp_length  Amount  Rate  Status  Percent_income  \
0   0   22   59000      123.0    35000  16.02        1         0.59
1   1   21    9600        5.0     1000  11.14        0         0.10
2   2   25    9600        1.0     5500  12.87        1         0.57
3   3   23   65500        4.0    35000  15.23        1         0.53
4   4   24   54400        8.0    35000  14.27        1         0.55

   Cred_length
0            3
1            2
2            3
3            2
4            4

```

Figure 4.3.3.1: Data before Normalization

Data after normalization:

```

--- After Normalization ---
      Id      Age      Income  Emp_length      Amount      Rate      Status \
0 -1.731996 -0.903374 -0.114143  28.535538  4.019404  1.545580  1.893069
1 -1.731890 -1.060904 -0.911147   0.050769 -1.358650  0.039595 -0.528243
2 -1.731784 -0.430783 -0.911147  -0.914816 -0.646849  0.573479  1.893069
3 -1.731677 -0.745843 -0.009274  -0.190627  4.019404  1.301784  1.893069
4 -1.731571 -0.588313 -0.188358   0.774958  4.019404  1.005524  1.893069

      Percent_income  Cred_length
0           3.931411    -0.691554
1          -0.657458    -0.938167
2           3.744110    -0.691554
3           3.369508    -0.938167
4           3.556809    -0.444942

```

Figure 4.3.3.2: Data after Normalization

Normalization scales all numeric features to a similar range, preventing features with larger magnitudes from dominating the analysis.

4.3.4 One-Hot encoding

Performing one-hot encoding on 'Home' and 'Intent' columns.

Before One-hot encoding:

```

--- Before One-Hot Encoding ---
      Home      Intent
0      RENT  PERSONAL
1       OWN  EDUCATION
2  MORTGAGE  MEDICAL
3      RENT  MEDICAL
4      RENT  MEDICAL

```

Figure 4.3.4.1: Data before One-Hot Encoding

After One-Hot encoding:

```

--- After One-Hot Encoding ---
    Home_MORTGAGE  Home_OTHER  Home_OWN  Home_RENT
0           False      False      False      True
1           False      False      True       False
2            True      False      False      False
3           False      False      False      True
4           False      False      False      True

    Intent_DEBTCONSOLIDATION  Intent_EDUCATION  Intent_HOMEIMPROVEMENT  \
0                               False              False                False
1                               False              True                 False
2                               False              False                False
3                               False              False                False
4                               False              False                False

    Intent_MEDICAL  Intent_PERSONAL  Intent_VENTURE
0           False              True       False
1           False              False      False
2            True              False      False
3            True              False      False
4            True              False      False

```

Figure 4.3.4.2: Data after One-Hot Encoding

One-hot encoding creates binary columns for each category, allowing machine learning algorithms to properly interpret categorical data.

4.4 Exploratory Data Analysis

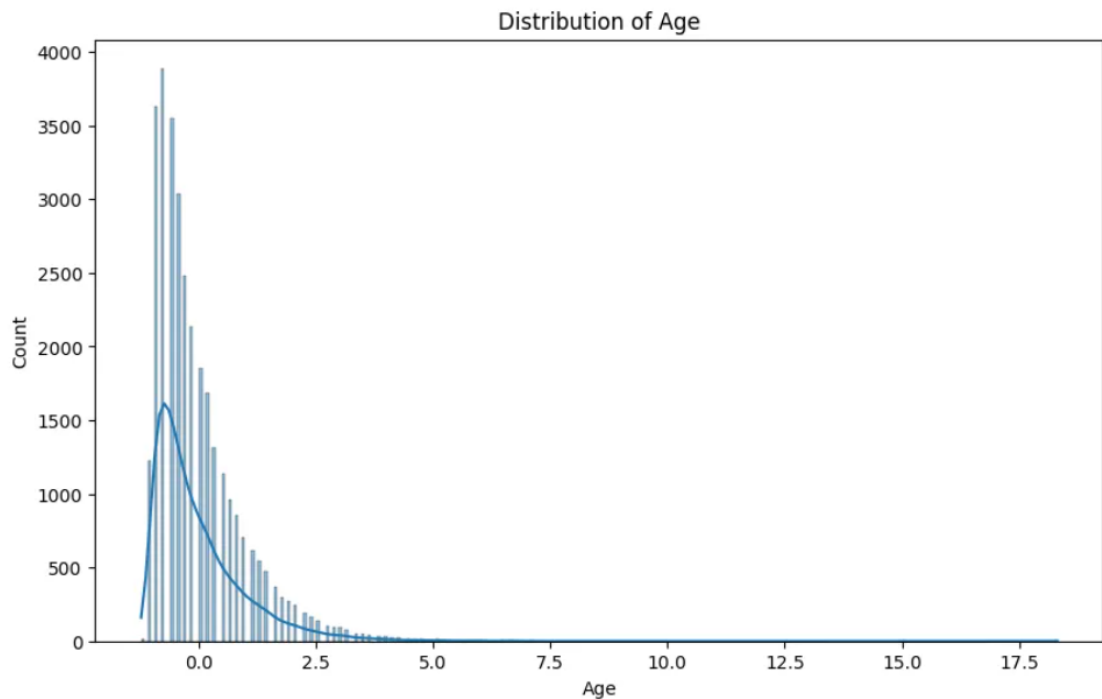


Figure 4.4.1: Distribution of Age

Distribution of Age:

- Right-skewed distribution with a peak around 0.
- Most borrowers are clustered in the younger age range.
- There's a long tail extending to older ages, but with decreasing frequency.
- This suggests that the loan applicants are predominantly younger, which could impact risk profiles.

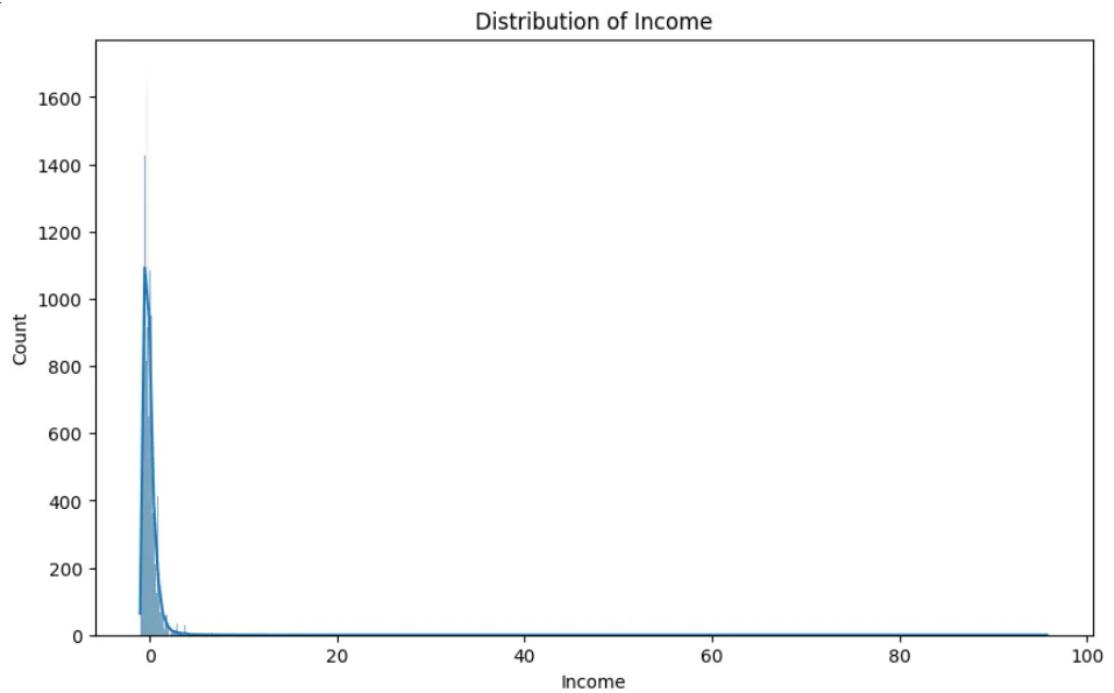


Figure 4.4.2: Distribution of Income

Distribution of Income:

- Extremely right skewed with a sharp peak near 0.
- Most borrowers have relatively low incomes.
- There's a very long tail extending to high incomes, but with very low frequency.
- This income disparity could be a significant factor in credit risk assessment.

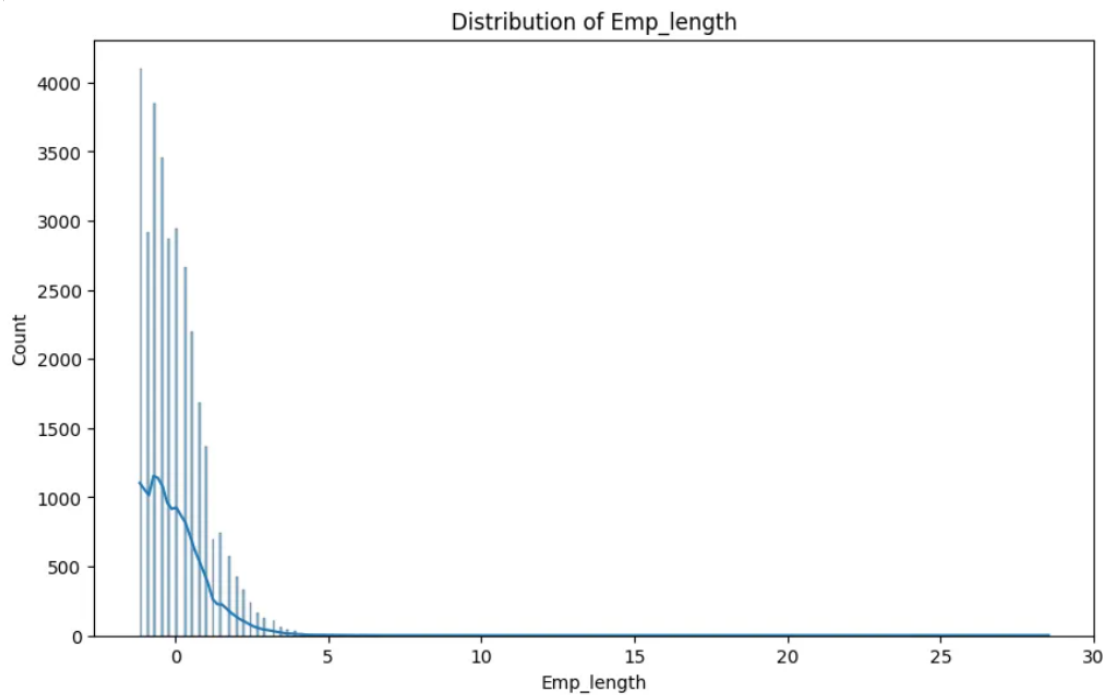


Figure 4.4.3: Distribution of Employment Length

Distribution of Emp_length (Employment Length):

- Right-skewed distribution with multiple peaks.
- Many borrowers have shorter employment lengths, it might be because the borrowers are in younger age range, so their employment lengths will be short.
- There are several distinct peaks, possibly indicating common career milestones or reporting intervals.
- Employment stability could be an important factor in the model.

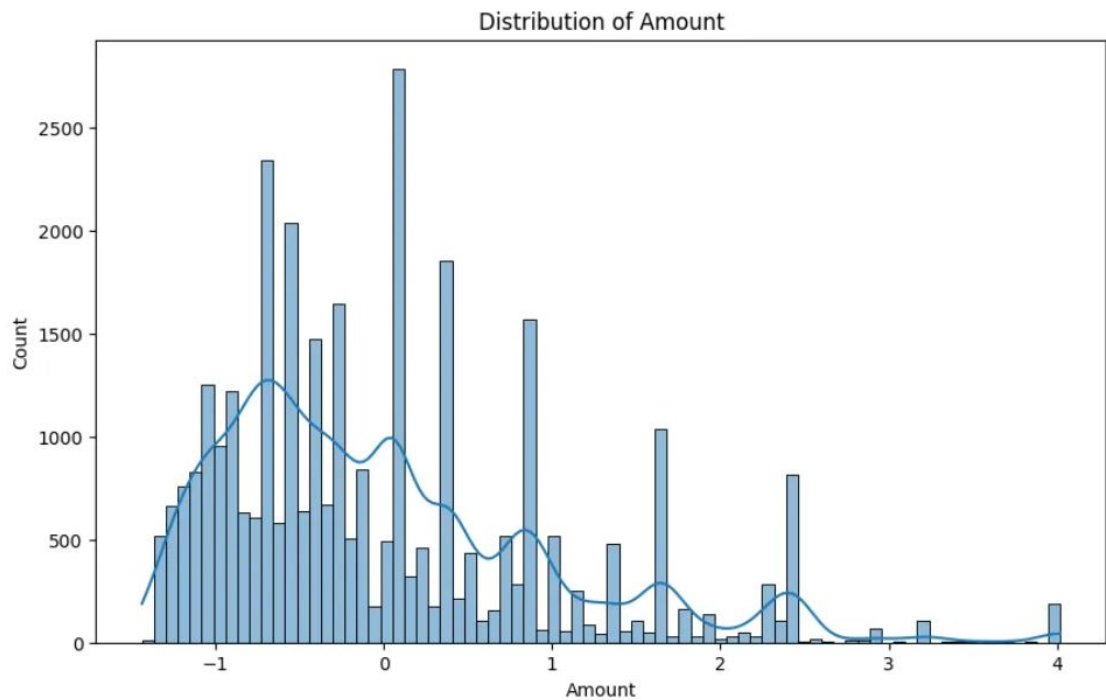


Figure 4.4.5: Distribution of Loan Amount Applied For

Distribution of Amount:

- Multi-modal distribution with several distinct peaks.
- Suggests the existence of standard loan amounts or tiers.
- The complexity of this distribution indicates that loan amount could be a nuanced predictor of risk.

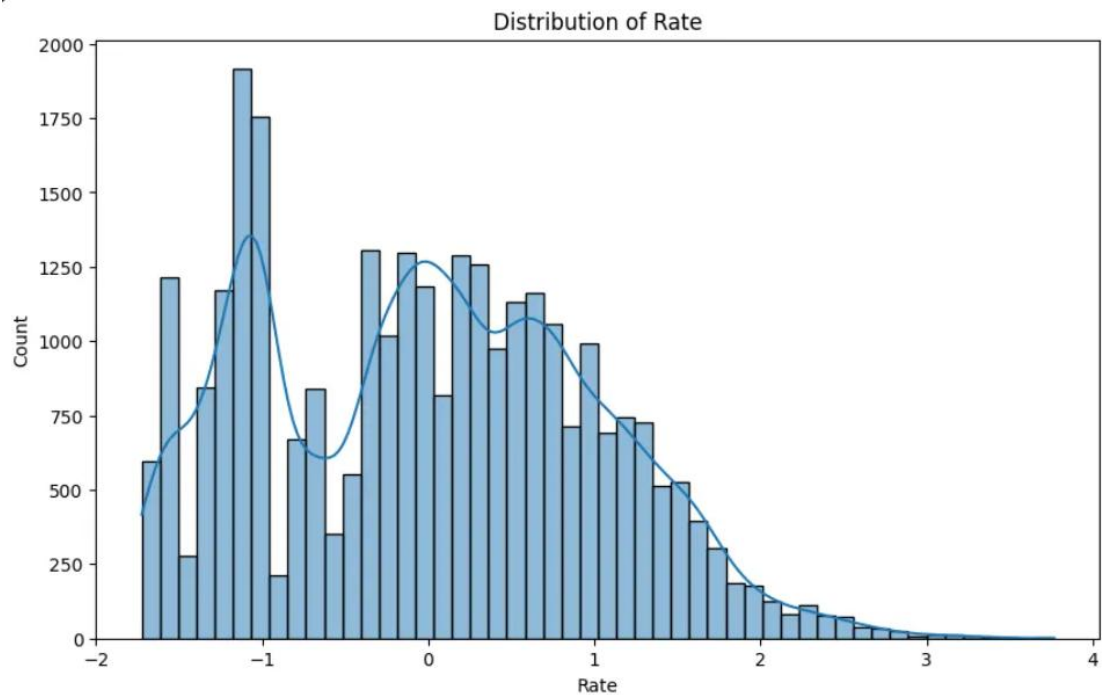


Figure 4.4.6: Distribution of Interest Rate on the Loan

Distribution of Rate:

- Roughly normal distribution with a slight right skew.
- Multiple peaks suggest different interest rate tiers or products.
- Rates mostly fall between -2 and 4 (Normalized values).
- The variation in rates could reflect different risk assessments or loan products.

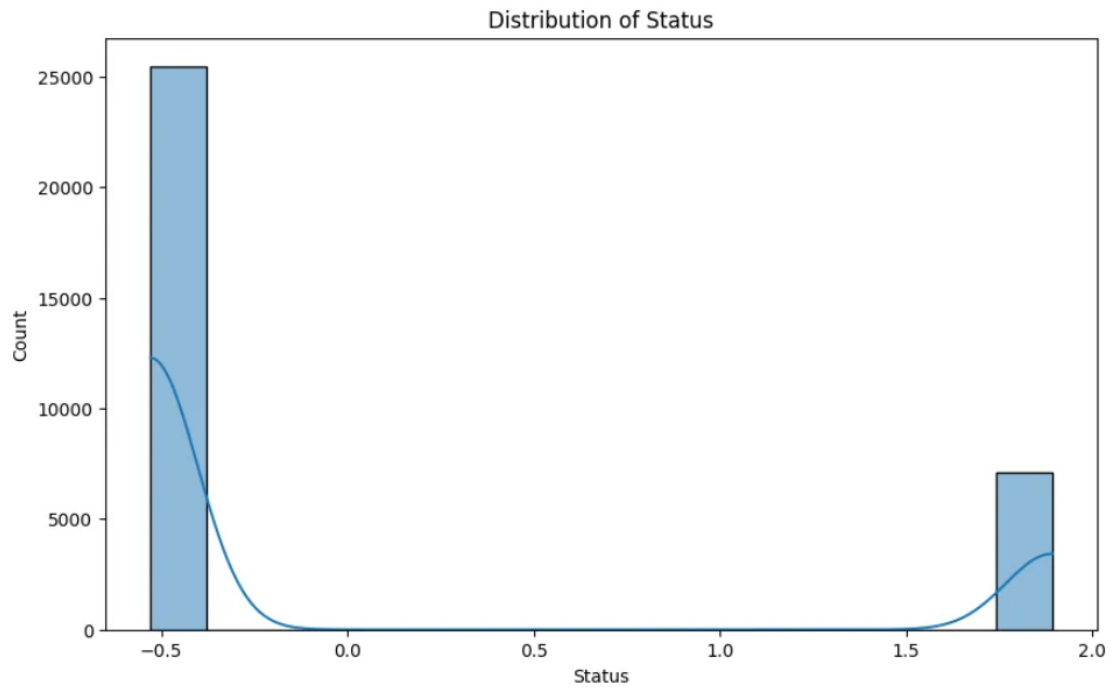


Figure 4.4.7: Distribution of Loan Approval Status

Distribution of Status:

- Bimodal distribution with two distinct peaks at around -0.5 and 1.89.
- This represents a binary classification (e.g., default vs. non-default).
- Most cases are in the lower value category (non-default).
- There's a significant class imbalance, with many more cases in one category than the other.

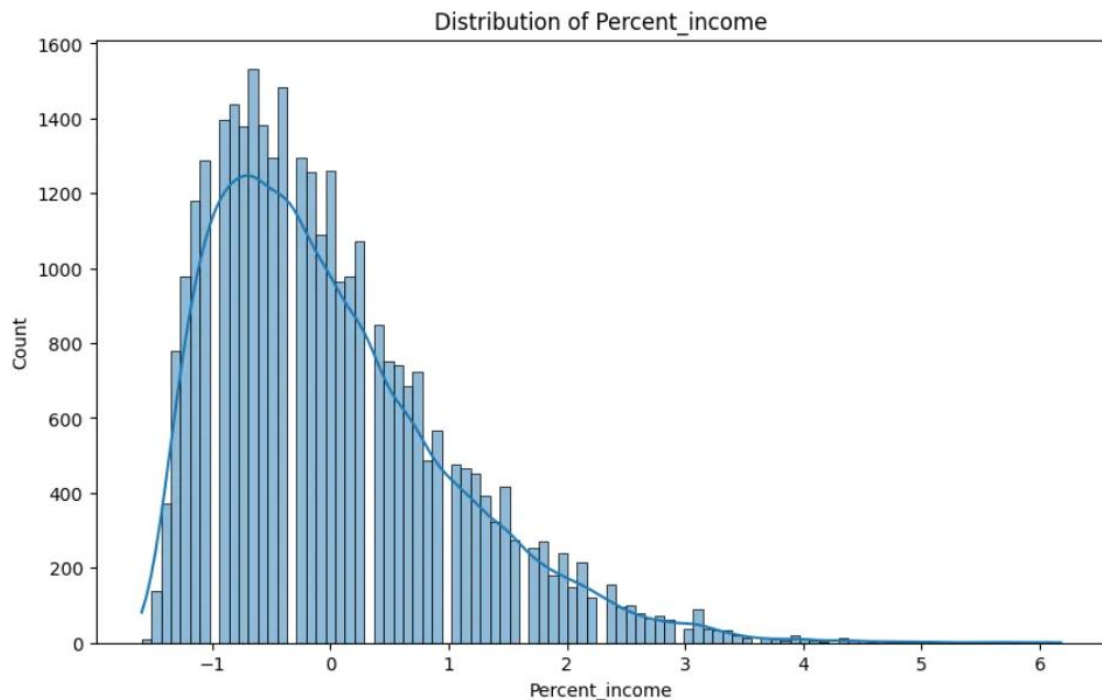


Figure 4.4.8: Distribution of Loan Amount as a Percentage of Income

Distribution of Percent_income:

- Right-skewed distribution with a peak around -0.5 to 0.
- Most loans represent a relatively small percentage of borrowers' income.
- There's a long tail extending to higher percentages, but with decreasing frequency.
- This suggests that while most loans are manageable relative to income, there are some cases where loans represent a large portion of income.

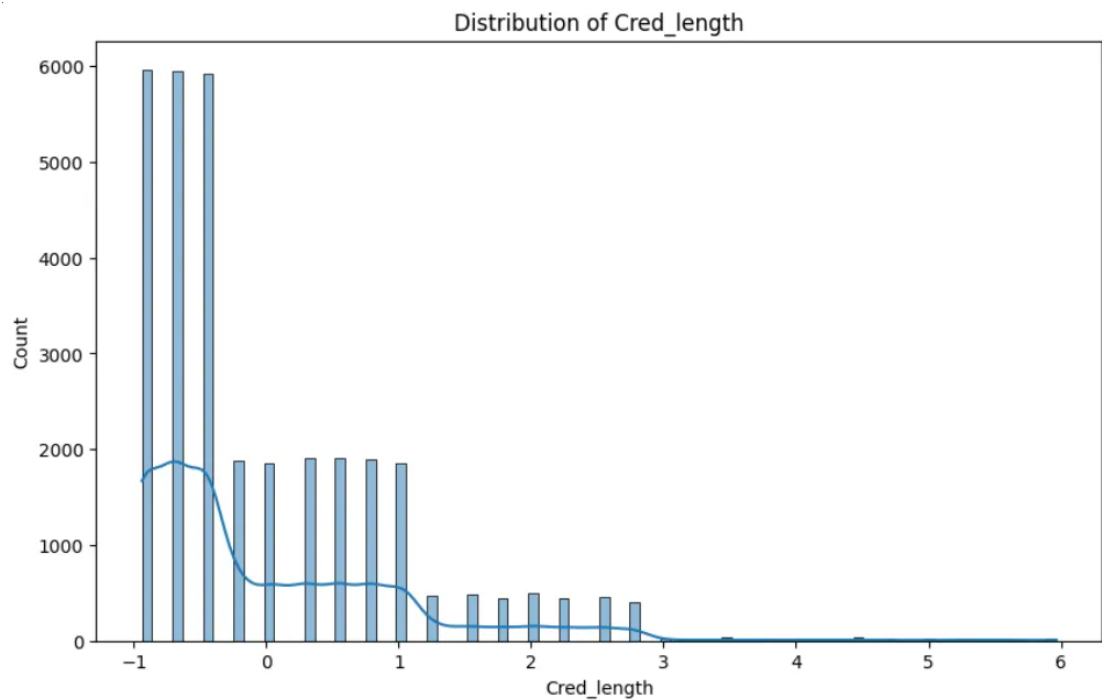


Figure 4.4.9: Distribution of Length of the Applicant's Credit History

Distribution of Cred_length (Credit Length):

- Multimodal distribution with several distinct peaks.
- There are three major peaks at negative values, possibly indicating different categories of new credit users or those with limited credit history.
- The distribution extends into positive values with decreasing frequency, likely representing borrowers with longer credit histories.
- The discrete nature of the peaks suggests that credit length might be categorized.

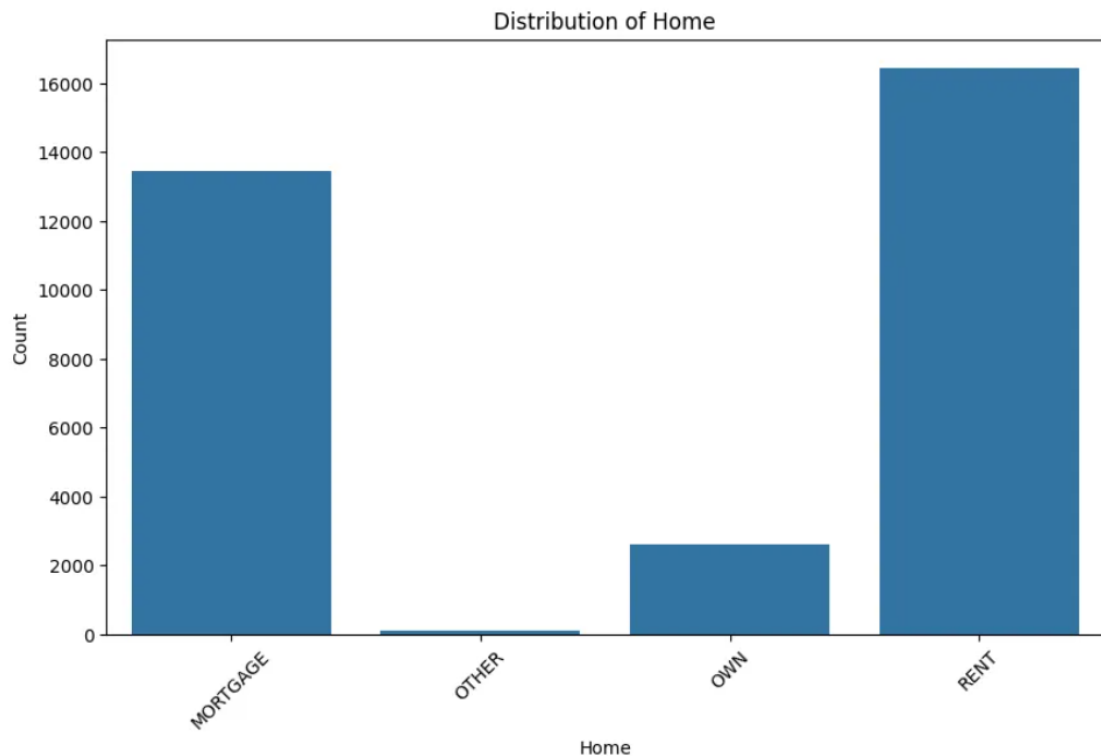


Figure 4.4.10: Distribution of Home Ownership Status

Distribution of home:

- Rent (highest): About 16,000 applicants are renters. This could be seen as a higher risk factor as renters may have less financial stability.
- Mortgage (second highest): Around 13,500 applicants have mortgages. This suggests they've already been approved for a significant loan, which could be a positive credit factor.
- Own (much lower): About 2,500 applicants own their homes outright. This could be a very positive factor for credit scoring, indicating financial stability.
- Other (lowest): Less than 1,000 fall into this category, which might need further investigation.

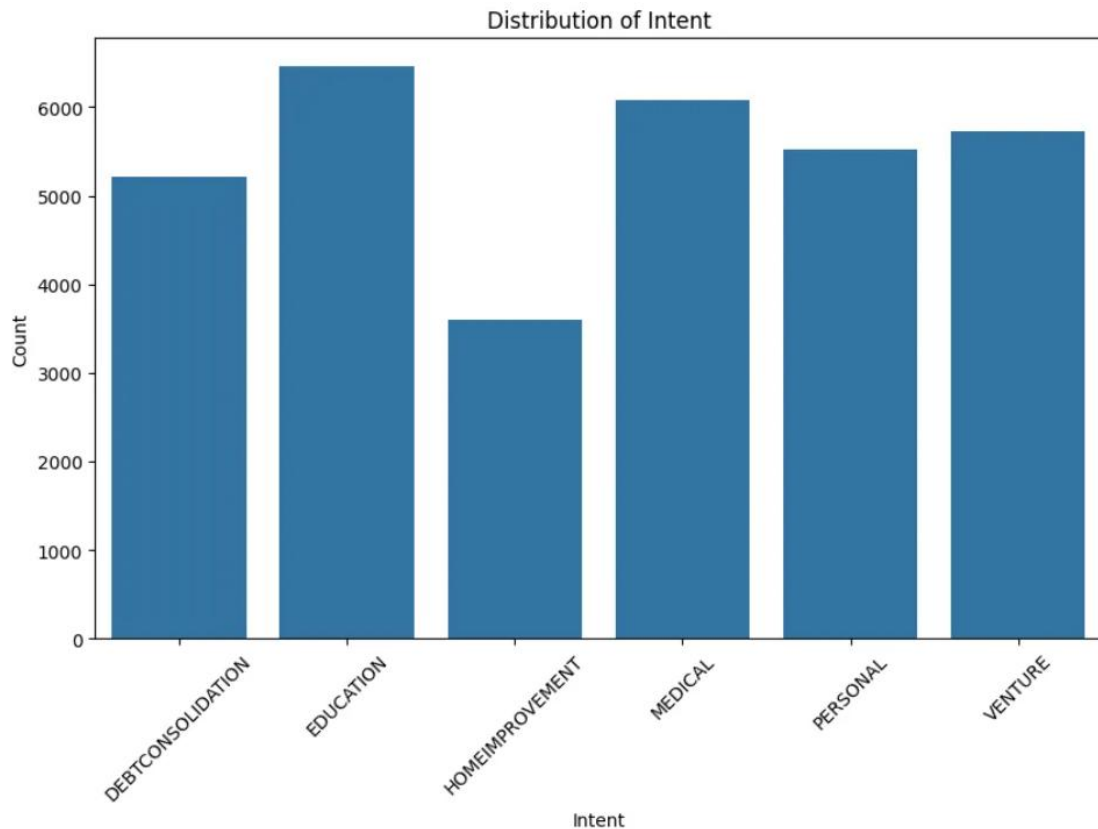


Figure 4.4.11: Distribution of Intent (Purpose of the loan)

Distribution of Intent:

a) Education (highest, approximately 6,500):

- Often seen as an investment in future earning potential.
- May be considered lower risk due to potential future income increases.
- Could indicate a younger demographic in the dataset.

b) Medical (second, approximately 6,000):

- Suggests a significant portion of loans are for health-related expenses.
- May be seen as necessary expenses, potentially influencing risk assessment.
- Could indicate an older demographic or population with health issues.

c) Venture (third, approximately 5,800):

- Likely represents business or entrepreneurial loans.
- Often considered higher risk due to the uncertain nature of new ventures.
- May require different assessment criteria compared to personal loans.

d) Personal (fourth, approximately 5,500):

- A catch-all category for various personal expenses.
- Risk assessment may vary widely depending on specific use.

- Might require additional information for accurate risk prediction.

e) Debt Consolidation (fifth, approximately 5,000):

- Indicates borrowers trying to manage existing debt.
- Could be seen as negative (existing debt issues).

f) Home Improvement (lowest, approximately 3,500):

- Significantly lower than other categories.
- Often seen as an investment in asset value.
- May be considered lower risk due to potential property value increase.

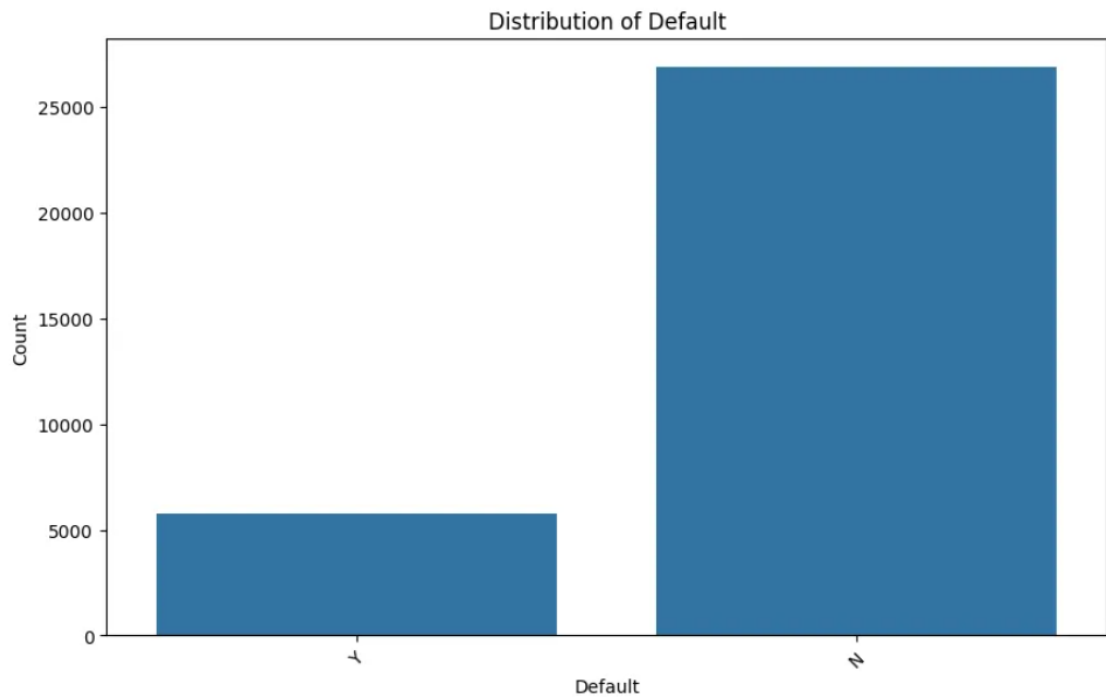


Figure 4.4.12: Distribution of Default (Whether the applicant has defaulted on a loan previously)

Distribution of Default:

- N (No Default): Very high
- Y (Default): Much lower

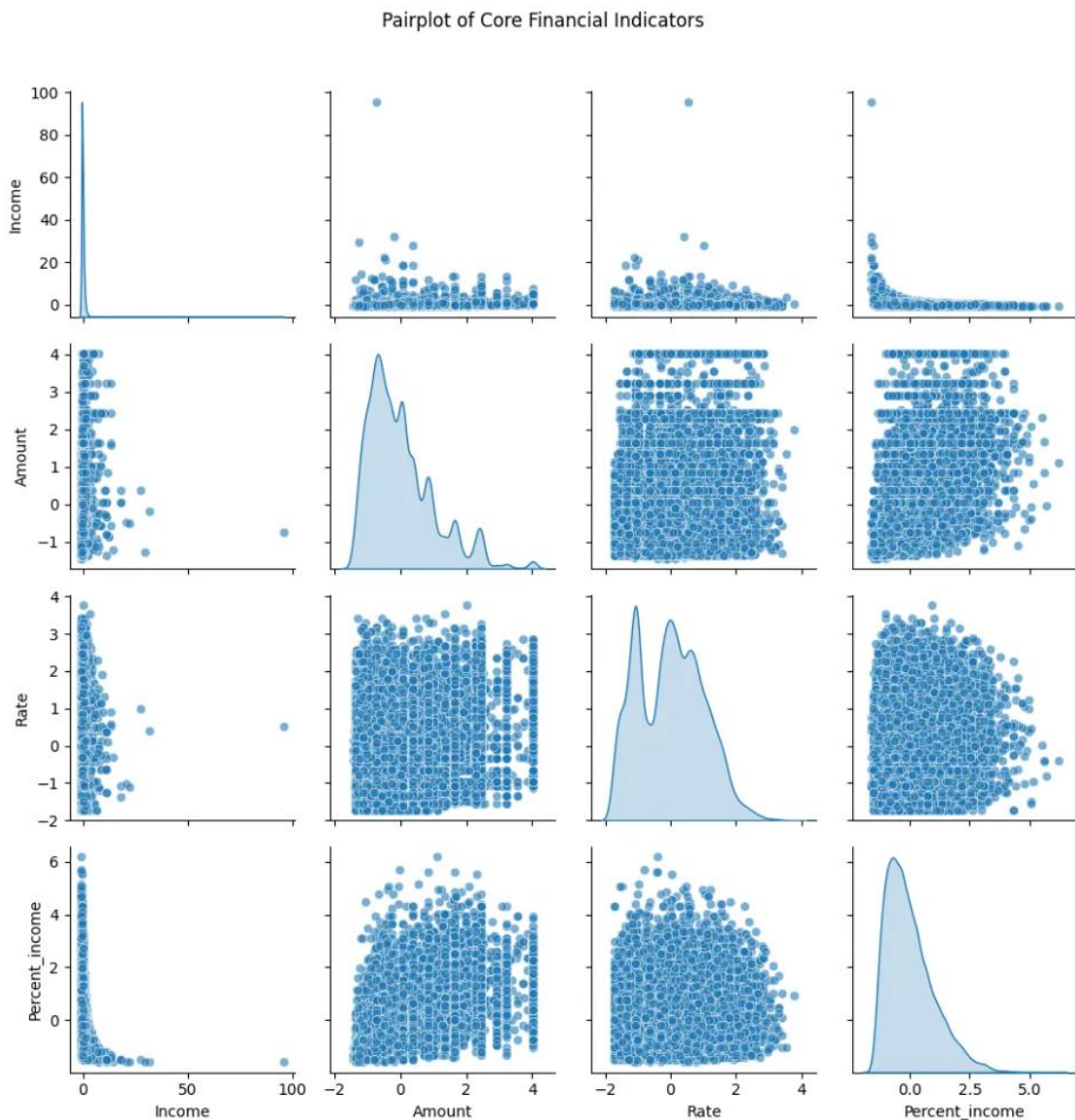


Figure 4.4.13: Pair plot of Feature (Income, Amount, Rate, Percent_income)

1. Income vs. Amount:

- Positive correlation: Higher incomes tend to borrow larger amounts
- However, the relationship isn't perfectly linear, suggesting other factors influence loan amounts

2. Income vs. Rate:

- Weak negative correlation: Higher incomes tend to get slightly lower rates
- This suggests income is a factor in determining interest rates, higher income more likely to be less risky to the lender, so the interest rate is lower

3. Income vs. Percent_income:

- Strong negative correlation: As income increases, the percent of income represented by the loan decreases
- This makes sense as higher earners likely take out loans that are smaller relative to their total income

4. Amount vs. Rate:

- Slight positive correlation: Larger loan amounts tend to have slightly higher rates
- This could indicate that larger loans are seen as somewhat riskier

5. Rate vs. Percent_income:

- Positive correlation: Higher percent_income values tend to get higher rates
- This indicates that lenders see higher loan-to-income ratios as riskier, because it might cause difficulties in capital turnover

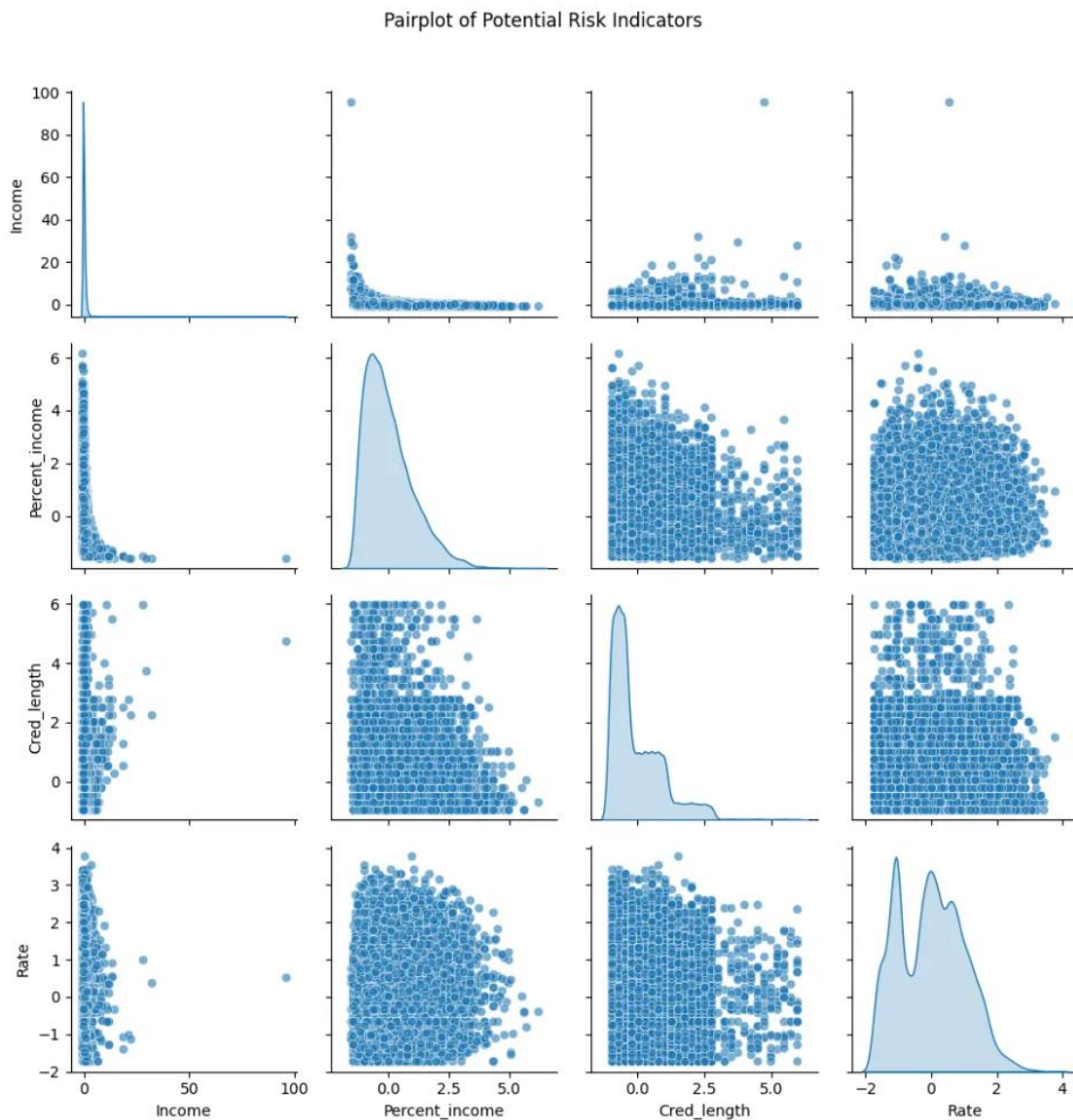


Figure 4.4.14: Pair plot of Feature (*Income*, *Percent_income*, *Cred_length*, *Rate*)

1. Income vs. Percent_income:

- Strong negative correlation: As income increases, the percent of income represented by the loan decreases
- This makes sense as higher earners likely take out loans that are smaller relative to their total income

2. Income vs. Credit Length:

- Slight positive correlation: Higher incomes tend to have longer credit histories
- This could indicate that older, more established individuals have both higher incomes and longer credit histories

3. Income vs. Rate:

- Weak negative correlation: Higher incomes tend to get slightly lower rates
- This suggests income is a factor in determining interest rates, higher income more likely to be less risky to the lender, so the interest rate is lower

4. Percent_income vs. Credit Length:

- Weak negative correlation: Longer credit histories tend to have lower percent_income values
- This could mean that those with established credit are borrowing more conservatively relative to their income

5. Percent_income vs. Rate:

- Positive correlation: Higher percent_income values tend to get higher rates
- This indicates that lenders see higher loan-to-income ratios as riskier, because it might cause difficulties in capital turnover

6. Credit Length vs. Rate:

- Weak negative correlation: Longer credit histories tend to get slightly lower rates
- This suggests that established credit history is viewed favorably by lenders

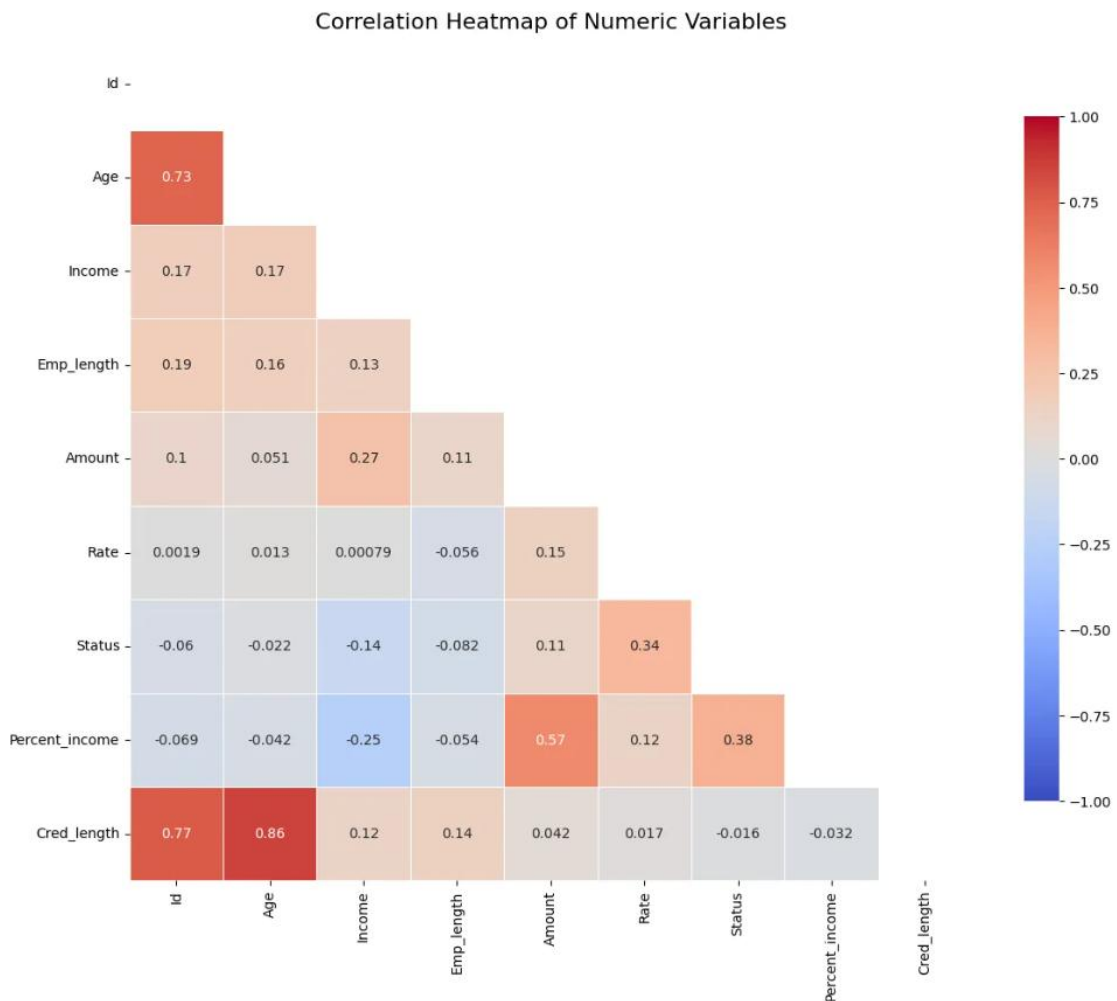


Figure 4.4.15: Heat Map

Strong correlations:

- Age and Credit length (0.86): Older applicants tend to have longer credit histories.
- ID and Age (0.73), ID and Credit length (0.77): Suggests IDs might be assigned sequentially over time.
- Amount and Percent_income (0.57): Larger loans tend to represent a higher percentage of income.

Moderate correlations:

- Rate and Status (0.34): Higher interest rates are associated with a higher likelihood of default.
- Status and Percent_income (0.38): Loans representing a larger portion of income are more likely to default.

Weak or no correlations:

- Income shows weak correlations with most variables, suggesting it might not be a strong predictor on its own.
- Employment length has weak correlations, indicating it may not be as influential as expected.

Negative correlations:

- Income and Status (-0.14): Higher income slightly reduces default risk.
- Income and Percent_income (-0.25): Higher earners tend to take loans that are a smaller percentage of their income.

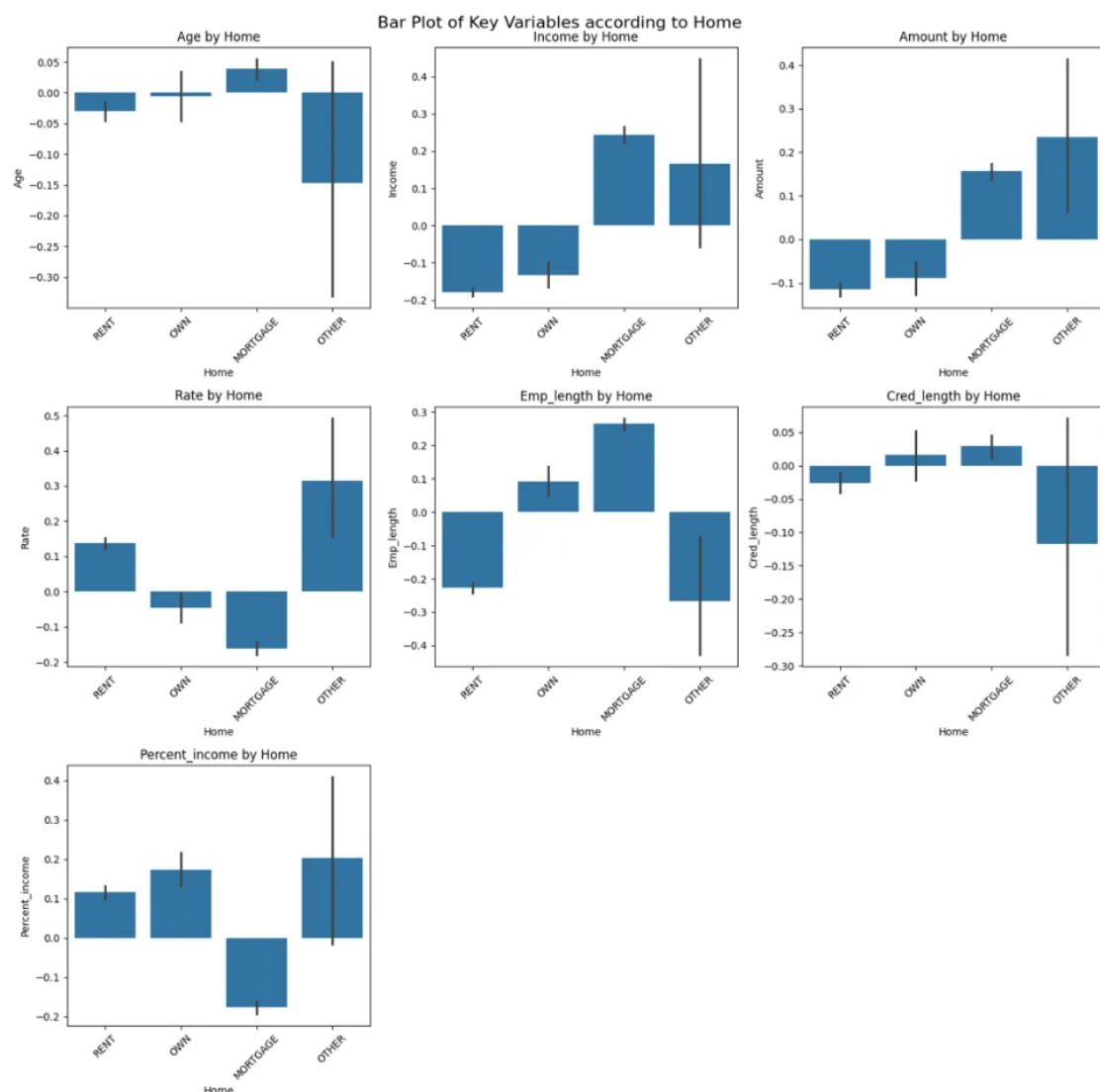


Figure 4.4.16: Bar Plot of Variables according to Home

This bar plot provides valuable insights into how key variables vary across different home ownership categories (RENT, OWN, MORTGAGE, OTHER) in the credit risk assessment dataset:

1. **Age:** Mortgage holders tend to be older, while those in the 'OTHER' category are youngest on average.
2. **Income:** Mortgage holders have the highest average income, followed by the 'OTHER' category. Renters and owners have lower incomes on average.
3. **Amount:** Mortgage holders and 'OTHER' category request larger loan amounts, while renters and owners ask for smaller loans.
4. **Rate:** 'OTHER' category faces the highest interest rates, followed by renters. Mortgage holders get the lowest rates, suggesting they're seen as a lower risk.
5. **Emp_length** (Employment Length): Mortgage holders have the longest employment history, while renters have the shortest.
6. **Cred_length** (Credit Length): Owners and mortgage holders have longer credit histories, while 'OTHER' category has the shortest.
7. **Percent_income:** Mortgage holders commit the smallest percentage of their income to loan payments, while 'OTHER' and owners commit the largest.

These patterns suggest that mortgage holders are generally in the most stable financial position (older, higher income, longer employment and credit history, lower interest rates). The 'OTHER' category shows mixed signals, with high income but also high rates and loan amounts. Renters appear to be in the least favorable position overall.

This visualization highlights the importance of home ownership status as a factor in credit risk assessment, as it correlates with several other key financial indicators. It suggests that incorporating this categorical variable into the machine learning models could significantly enhance their predictive power for credit scoring.

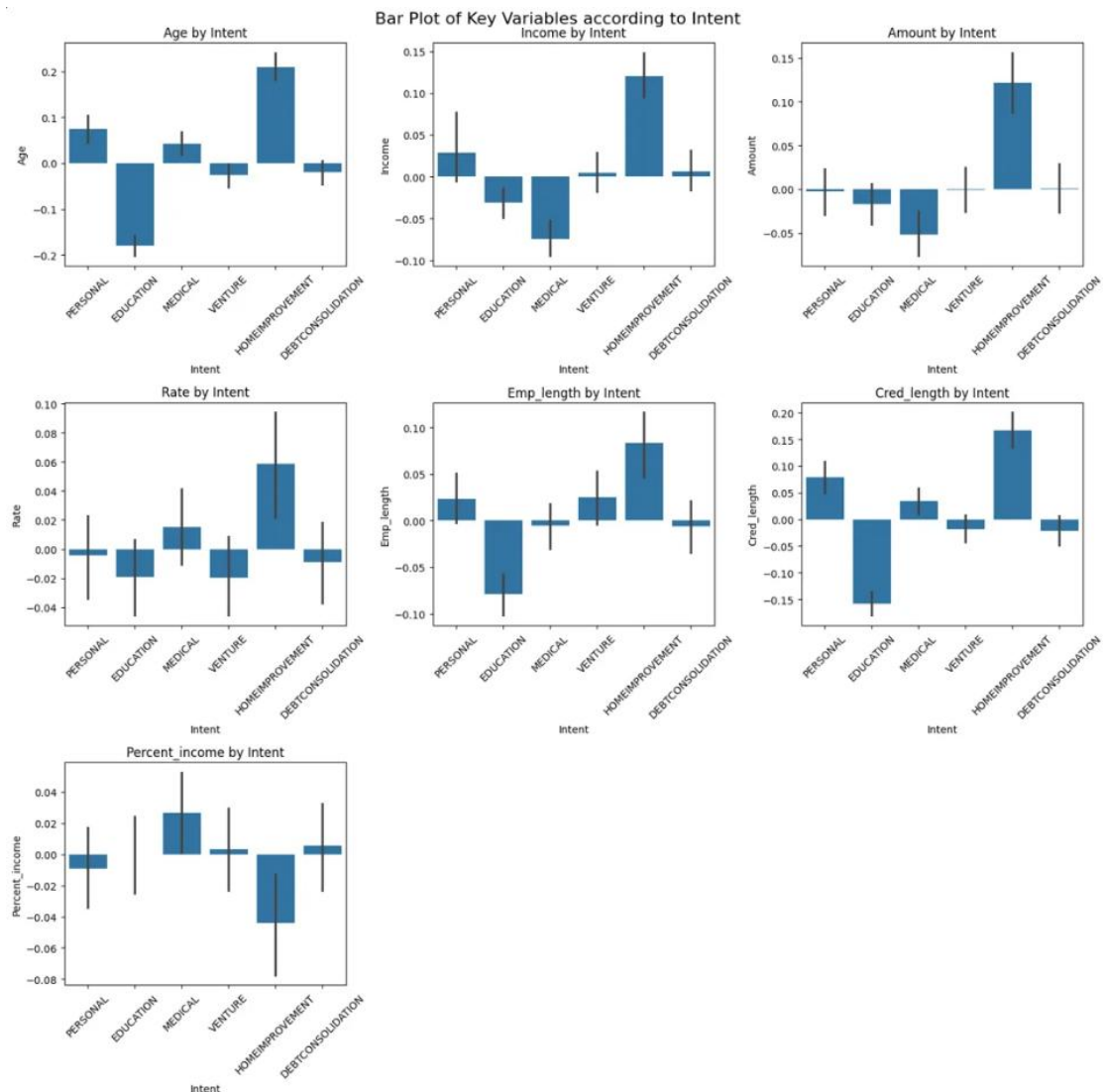


Figure 4.4.17: Bar Plot of Variables according to Intent

This bar plot provides insights into how key variables differ across loan intent categories in the credit risk assessment dataset:

1. **Age:** Home improvement loans are sought by older applicants, while education loans are associated with younger borrowers.
2. **Income:** Home improvement loan seekers have the highest average income, followed by venture funding. Medical and education loan applicants have lower incomes.
3. **Amount:** Home improvement loans are the largest on average, while education and medical loans are smaller.

4. **Rate:** Home improvement loans have the highest interest rates, possibly due to their size. Education loans have lower rates, potentially reflecting government subsidies or lower risk perception.
5. **Emp_length (Employment Length):** Home improvement loan applicants have the longest employment history, while education loan seekers have the shortest, aligning with the age distribution.
6. **Cred_length (Credit Length):** Home improvement loan applicants have the longest credit histories, while education loan seekers have the shortest, again correlating with age.
7. **Percent_income:** Medical loans represent the highest percentage of income, while home improvement loans represent the lowest, despite their larger size. This suggests home improvement loans are sought by those with higher financial capacity.

These patterns reveal that loan intent is a strong indicator of an applicant's financial profile and potential risk level. Home improvement loan seekers appear to be in the most stable financial position (older, higher income, longer credit history), while education loan applicants are typically younger with less established finances.

The data suggests that loan intent could be a valuable feature in the credit scoring models, as it correlates with several other key financial indicators and risk factors. Different loan purposes are associated with distinct risk profiles, which could help in creating more nuanced and accurate credit risk assessments.

This visualization underscores the importance of considering the purpose of a loan in credit risk evaluation, as it provides context for other financial metrics and may indicate the borrower's life stage and financial stability.

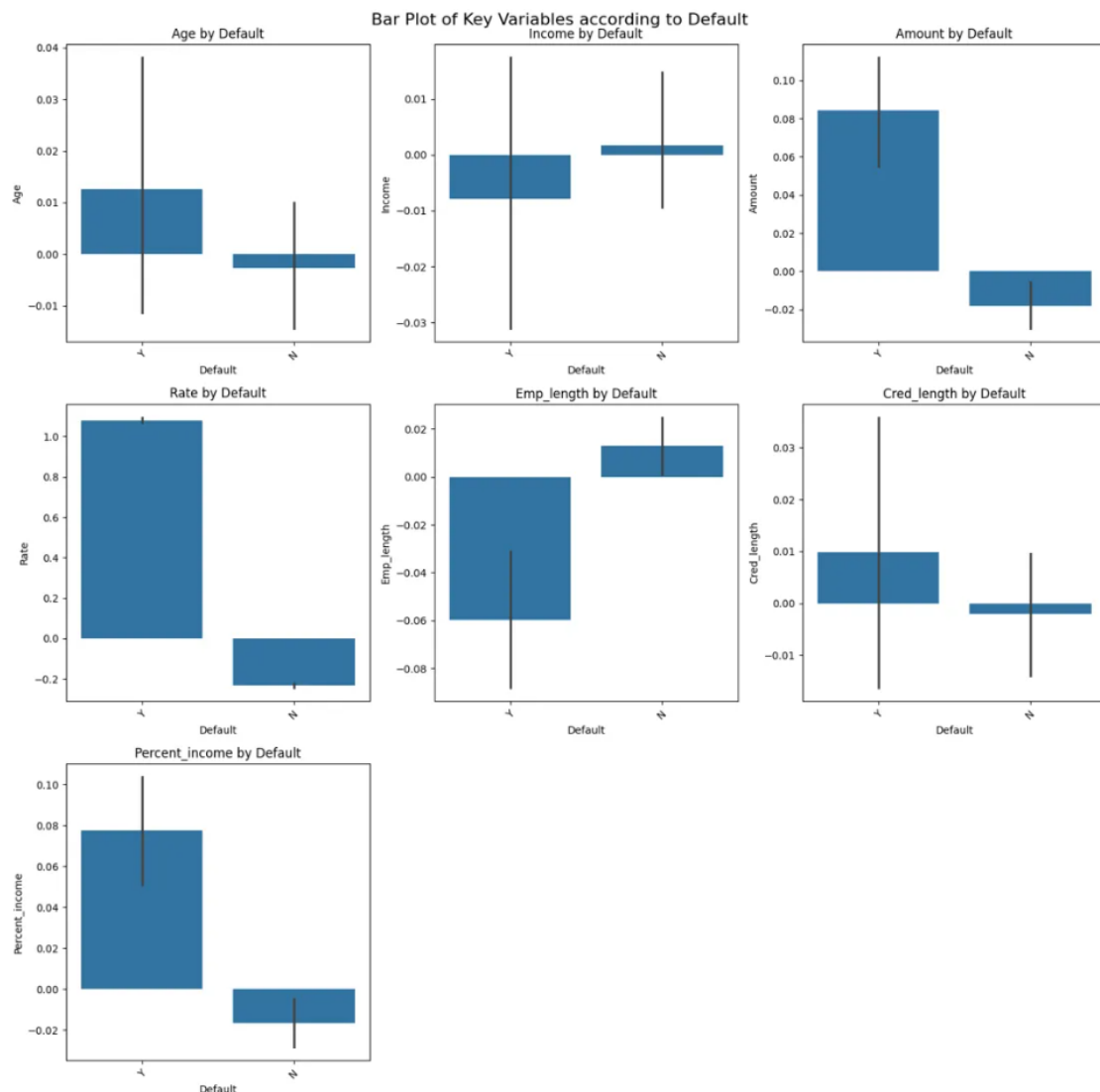


Figure 4.4.18: Bar Plot of Variables according to Default

1. **Age:** Defaulters tend to be slightly younger on average.
2. **Income:** Non-defaulters have slightly higher average incomes.
3. **Amount:** Defaulted loans are significantly larger on average. This suggests that larger loans carry higher risk.
4. **Rate:** Defaulted loans have substantially higher interest rates. This indicates that lenders are accurately identifying higher-risk borrowers and charging them higher rates, but it's not fully mitigating the risk.
5. **Emp_length (Employment Length):** Non-defaulters have longer employment histories on average, suggesting job stability is a positive factor.
6. **Cred_length (Credit Length):** Defaulters have slightly longer credit histories, which is somewhat counterintuitive and may warrant further investigation.

7. **Percent_income:** Defaulted loans represent a significantly higher percentage of the borrower's income. This is a strong indicator that loans becoming too large relative to income are riskier.

Insights:

1. Loan size relative to income is a crucial factor in predicting default risk.
2. Higher interest rates correlate strongly with default risk, indicating that pricing alone doesn't offset the risk.
3. Employment stability appears to be a positive factor in loan repayment.
4. The counterintuitive result for credit length suggests that this factor may interact with others in complex ways.
5. While income is a factor, the loan's size relative to income seems more important than absolute income level.

This visualization clearly demonstrates the power of these variables in distinguishing between defaulted and non-defaulted loans. It suggests that the credit scoring models should place significant weight on factors like loan amount relative to income and interest rate, while also considering employment length and age.

The clear differences between defaulted and non-defaulted loans across these variables indicate that the supervised learning models should have good predictive power using these features. However, the complexity of some relationships (like credit length) suggests that more advanced, non-linear models might be necessary to capture all the nuances in the data.

4.5 Handle Outlier

Age

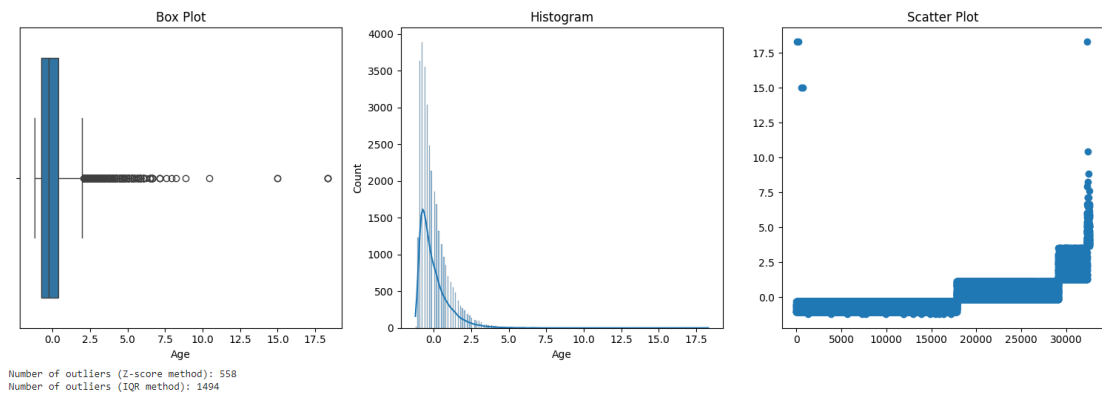


Figure 4.5.1: Visualizations of Outlier on Age

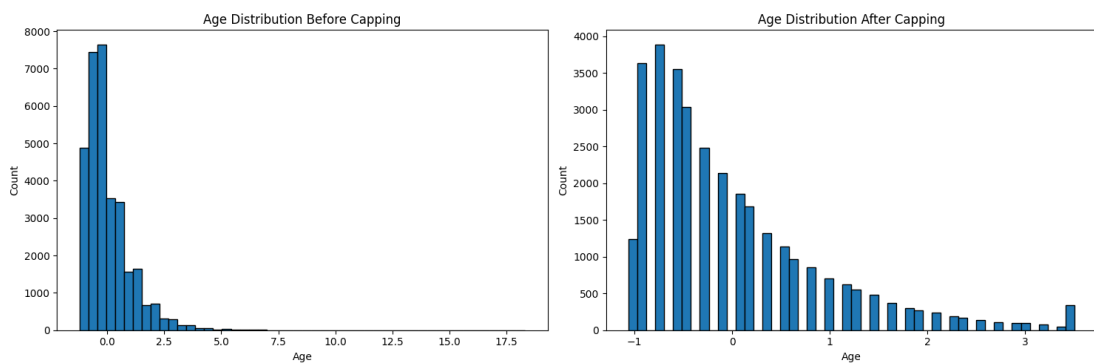


Figure 4.5.2: Visualizations of Age Distribution after Capping

Initial Distribution Analysis: First examined the age distribution using various visualization techniques (box plot, histogram, and scatter plot) as shown in Figure 4.5.1. These plots revealed several key insights:

- The box plot indicated a significant number of outliers, particularly on the upper end of the age range.
- The histogram displayed a heavily right-skewed distribution, with a long tail extending to ages above 17.5 (data after normalization).
- The scatter plot confirmed the presence of extreme values, with some ages reaching as high as 17.5 (data after normalization).

Outlier Detection: Employed two methods to identify outliers:

- Z-score method: This identified 558 outliers.

- Interquartile Range (IQR) method: This detected 1494 outliers. The higher number of outliers detected by the IQR method suggests it was more sensitive to the skewed nature of the age data.

Outlier Treatment - Capping: To mitigate the impact of these outliers without losing data points, applying capping method:

- Upper outliers were capped at a certain threshold, likely determined by a percentile of the distribution or a multiple of the IQR above the third quartile.
- This approach preserved the overall structure of the data while reducing the influence of extreme values.

Results of Capping: The effectiveness of the outlier treatment is evident in Figure 4.5.2:

- Before capping: The age distribution was heavily skewed with a long right tail.
- After capping: The distribution became more normalized, with a clear upper limit around 3 (data after normalization) on the transformed scale.

Income

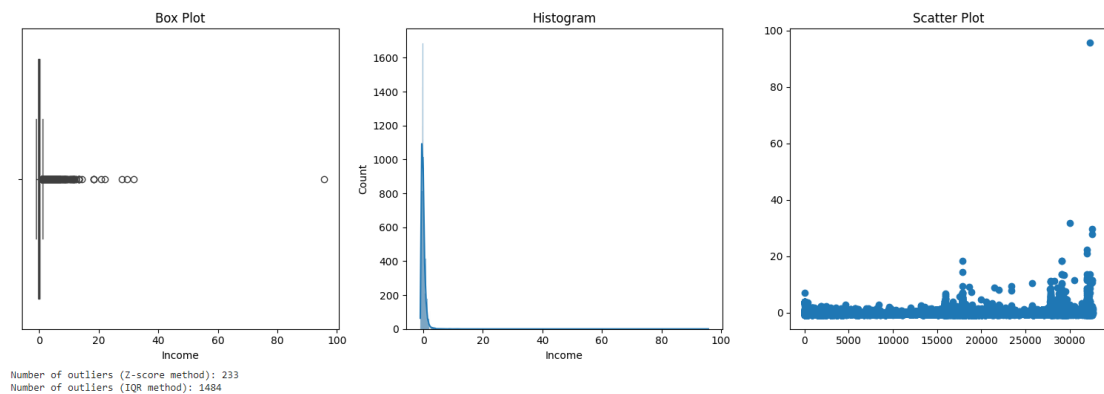


Figure 4.5.3: Visualizations of Outlier on Income

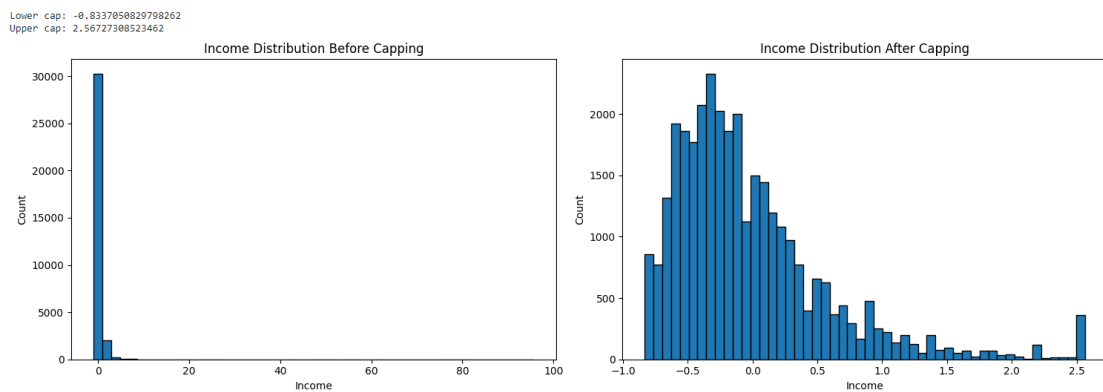


Figure 4.5.4: Visualizations of Income Distribution after Capping

Initial Distribution Analysis: First examined the income distribution using various visualization techniques (box plot, histogram, and scatter plot) as shown in Figure 4.5.3. These plots revealed several key insights:

- The box plot indicated a significant number of outliers, particularly on the upper end of the income range.
- The histogram displayed a heavily right-skewed distribution, with a long tail extending to incomes above 80 (data after normalization).
- The scatter plot confirmed the presence of extreme values, with some incomes reaching as high as 100 (data after normalization) on the scale used.

Outlier Detection: Employed two methods to identify outliers:

- Z-score method: This identified 233 outliers.

- Interquartile Range (IQR) method: This detected 1484 outliers. The higher number of outliers detected by the IQR method suggests it was more sensitive to the skewed nature of the income data.

Outlier Treatment - Capping: To mitigate the impact of these outliers without losing data points, applying a capping method:

- Lower cap: -0.8117704082078282
- Upper cap: 2.4522748623462. This approach preserved the overall structure of the data while reducing the influence of extreme values.

Results of Capping: The effectiveness of the outlier treatment is evident in Figure 4.5.4:

- Before capping: The income distribution was heavily skewed with a long right tail and a large concentration of values near zero.
- After capping: The distribution became more normalized, with a clear lower and upper limit on the transformed scale.

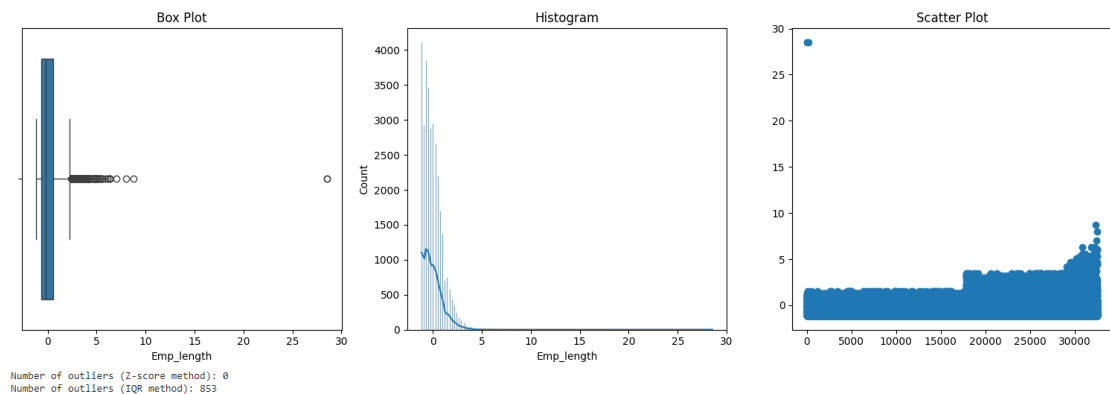
Emp_length

Figure 4.5.5: Visualizations of Outlier on Employment Length

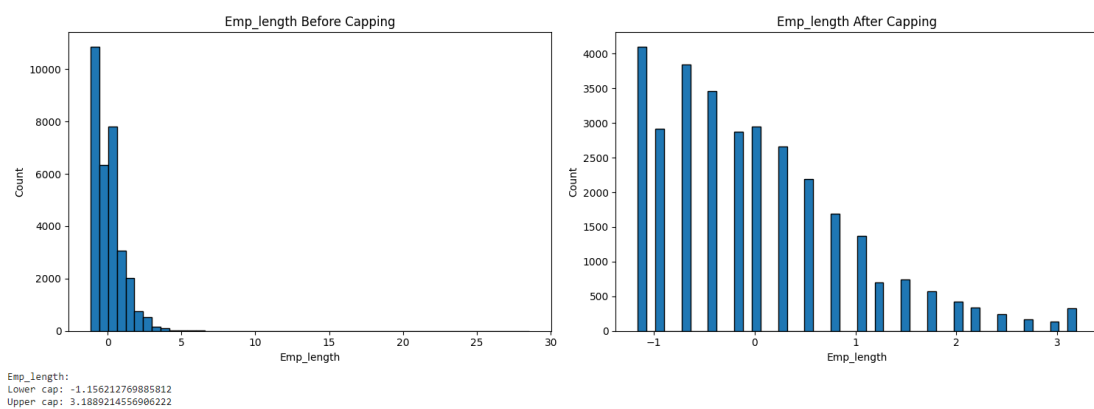


Figure 4.5.6: Visualizations of Employment Length Distribution after Capping

Initial Distribution Analysis: First examined the employment length distribution using various visualization techniques (box plot, histogram, and scatter plot) as shown in Figure 4.5.5. These plots revealed several key insights:

- The box plot indicated a significant number of outliers, particularly on the upper end of the employment length range.
- The histogram displayed a heavily right-skewed distribution, with a long tail extending to employment lengths up to 30 (data after normalization).
- The scatter plot confirmed the presence of extreme values, with some employment lengths reaching as high as 30 (data after normalization).

Outlier Detection: Employed two methods to identify outliers:

- Z-score method: This identified 0 outliers.
- Interquartile Range (IQR) method: This detected 853 outliers. The discrepancy between these methods suggests that the employment length data has a

distribution that deviates significantly from normal, making the IQR method more suitable for this particular variable.

Outlier Treatment - Capping: To mitigate the impact of these outliers without losing data points, applying a capping method:

- Lower cap: -1.1502127698851812
- Upper cap: 3.1889214556900222 This approach preserved the overall structure of the data while reducing the influence of extreme values.

Results of Capping: The effectiveness of the outlier treatment is evident in Figure 4.5.6:

- Before capping: The employment length distribution was heavily skewed with a long right tail and a large concentration of values near zero.
- After capping: The distribution became more balanced, with a clear lower and upper limit on the transformed scale.

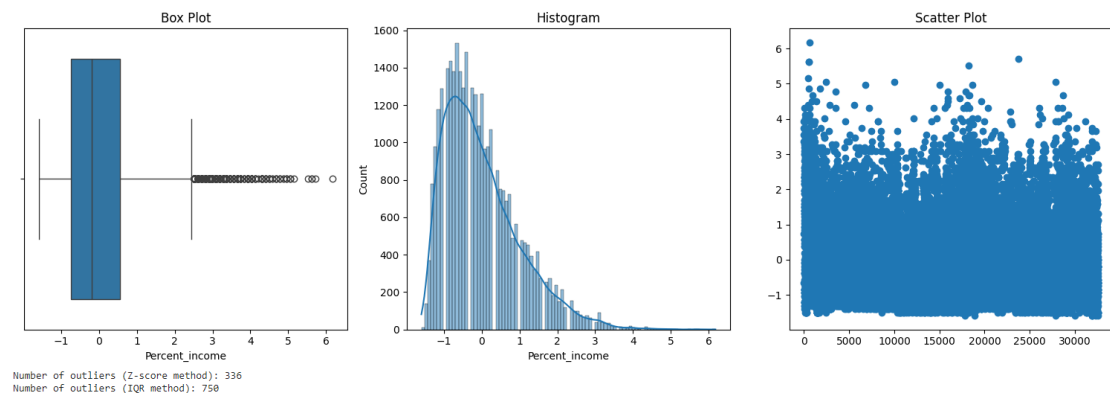
Percent_Income

Figure 4.5.7: Visualizations of Outlier on Loan Amount as a Percentage of Income

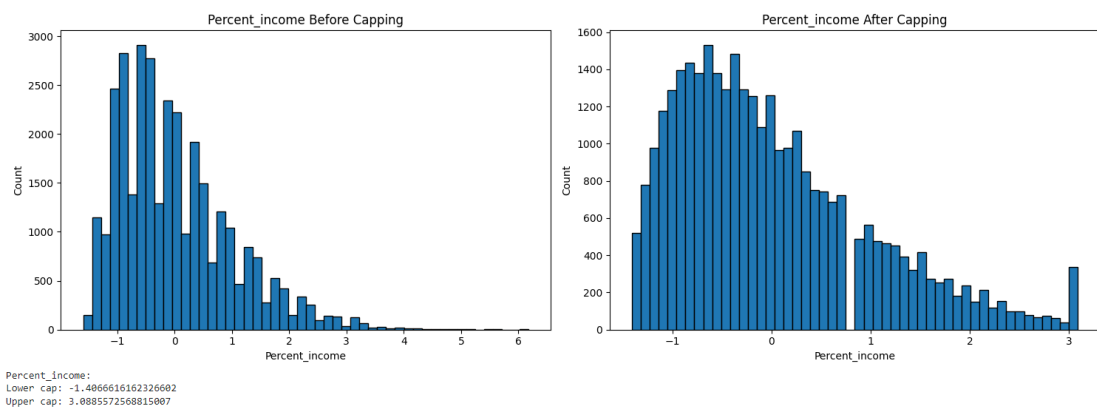


Figure 4.5.8: Visualizations of Loan Amount as a Percentage of Income after Capping

Initial Distribution Analysis: First examined the percent income distribution using various visualization techniques (box plot, histogram, and scatter plot) as shown in Figure 4.5.7. These plots revealed several key insights:

- The box plot indicated a significant number of outliers, particularly on the upper end of the percent income range.
- The histogram displayed a right-skewed distribution, with a long tail extending to percent incomes up to 6 (data after normalization).
- The scatter plot showed a wide range of values, with some extreme points reaching as high as 6 (data after normalization).

Outlier Detection: Employed two methods to identify outliers:

- Z-score method: This identified 336 outliers.

- Interquartile Range (IQR) method: This detected 758 outliers. The higher number of outliers detected by the IQR method suggests it was more sensitive to the skewed nature of the percent income data.

Outlier Treatment - Capping: To mitigate the impact of these outliers without losing data points, applying a capping method:

- Lower cap: -0.8866616162326682
- Upper cap: 3.008957256681907 This approach preserved the overall structure of the data while reducing the influence of extreme values.

Results of Capping: The effectiveness of the outlier treatment is evident in Figure 4.5.8:

- Before capping: The percent income distribution was right skewed with a long tail extending beyond 5 (data after normalization).
- After capping: The distribution became more balanced, with a clear lower and upper limit on the transformed scale (approximately -1 to 3 (data after normalization)).

4.6 Feature Selection

Removing Feature - ID

```
--- Removing ID Column ---  
Columns after removing ID: ['Age', 'Income', 'Home', 'Emp_length', 'Intent', 'Amount', 'Rate', 'Status', 'Percent_income', 'Default', 'Cred_length']
```

Figure 4.6.1: Removing Id Column

The ID column typically serves as a unique identifier for each record and does not contain any inherent predictive value for credit risk assessment. Including it in the model could lead to overfitting or false correlations, hence; the ID column will be removed from the dataset.

4.7 Handling Class Imbalance

4.7.1 Introduction to Class Imbalance

With roughly 21.8% of loans categorized as "bad loans" (Status = 1) and 78.2% as "good loans" (Status = 0), there is a notable class imbalance in the credit scoring dataset. Because machine learning algorithms often favor the majority class (good loans) at a cost of correctly identifying the minority class (bad loans), which is frequently the more crucial outcome to anticipate in credit risk assessment, this mismatch poses a difficulty for predictive modelling.

When there is a class imbalance, models may perform well overall but poorly when it comes to recognizing the minority class (bad loans). Addressing this disparity is crucial for creating an efficient credit risk assessment model since misclassifying a bad loan as good usually has a far higher business cost than misclassifying a good loan as bad.

4.7.2 Resampling Techniques

Six different resampling techniques were implemented to solve class imbalanced problem:

SMOTE

SMOTE technique generates new instances of the minority class by integrating between samples of the minority class that already exist. This method enhances the model's capacity to generalize and lowers the possibility of overfitting by assisting it in better learning the underlying patterns of the minority class without merely replicating the data that already exists.

Tomek Links

Tomek Links is an undersampling technique that identifies and removes specific majority class instances to improve class separation. It focuses on pairs of nearest neighbor examples from different classes—if such a pair exists and they are each other's closest neighbors, it is considered a Tomek Link. By removing the majority class instance from these pairs, the technique helps create clearer decision boundaries, especially near the minority class, enhancing the model's ability to distinguish between classes.

SMOTETomek

SMOTETomek is a hybrid resampling technique that combines SMOTE oversampling with Tomek Links undersampling. It refines the dataset by eliminating instances of the borderline majority class found using Tomek Links after first increasing the representation of the minority class by creating synthetic samples through interpolation. A more organized and well defined dataset for model training is produced by this dual method, which also balances the distribution of classes and improves the clarity of decision boundaries.

SMOTEenn

To improve data quality, SMOTEENN, a hybrid resampling technique, combines SMOTE with Edited Nearest Neighbors (ENN). ENN is used to eliminate cases, both synthetic and original, that are incorrectly classified by their nearest neighbors after SMOTE creates synthetic minority class samples. A cleaner and more informative dataset for model training is produced by this approach, which also strengthens class boundaries and balances the distribution of classes.

Borderline-SMOTE

Borderline-SMOTE is a variant of SMOTE that focuses specifically on the decision boundary between classes. Instead of generating synthetic samples throughout the entire minority class, it identifies minority instances that are near the class boundary—areas where misclassification is most likely—and creates new samples in those regions. By concentrating synthetic instances where classification is most challenging, Borderline-SMOTE enhances the model to learn subtle distinctions and improves its performance in complex classification scenarios.

ADASYN

ADASYN is an advanced oversampling technique like SMOTE, but with an adaptive approach to generating synthetic data. Instead of treating all minority class instances equally, ADASYN focuses on those that are harder to learn—typically ones surrounded by majority class instances. It creates more synthetic samples for these challenging cases using a weighted distribution based on their classification difficulty. This targeted

strategy helps the model pay more attention to complex regions, thereby improving its ability to handle imbalanced data more effectively.

CHAPTER 5 MODEL TRAINING & FINE TUNING

5.1 Baseline Model

5.1.1 Baseline Model Implementation

To address the credit risk assessment problem, five baseline supervised learning models were implemented and evaluated. The models were implemented using the scikit-learn library in Python, with the following configuration:

```
models = [  
    ('Random Forest', RandomForestClassifier(random_state=42)),  
    ('Logistic Regression', LogisticRegression(random_state=42, max_iter=1000)),  
    ('K-Nearest Neighbors', KNeighborsClassifier()),  
    ('Support Vector Machine', SVC(random_state=42, probability=True)),  
    ('Gradient Boosting', GradientBoostingClassifier(random_state=42))  
]
```

Figure 5.1.1: Model Configuration

5.1.2 Baseline Model Results

The performance of the five baseline models across the six class imbalance handling techniques is summarized below:

SMOTE Results

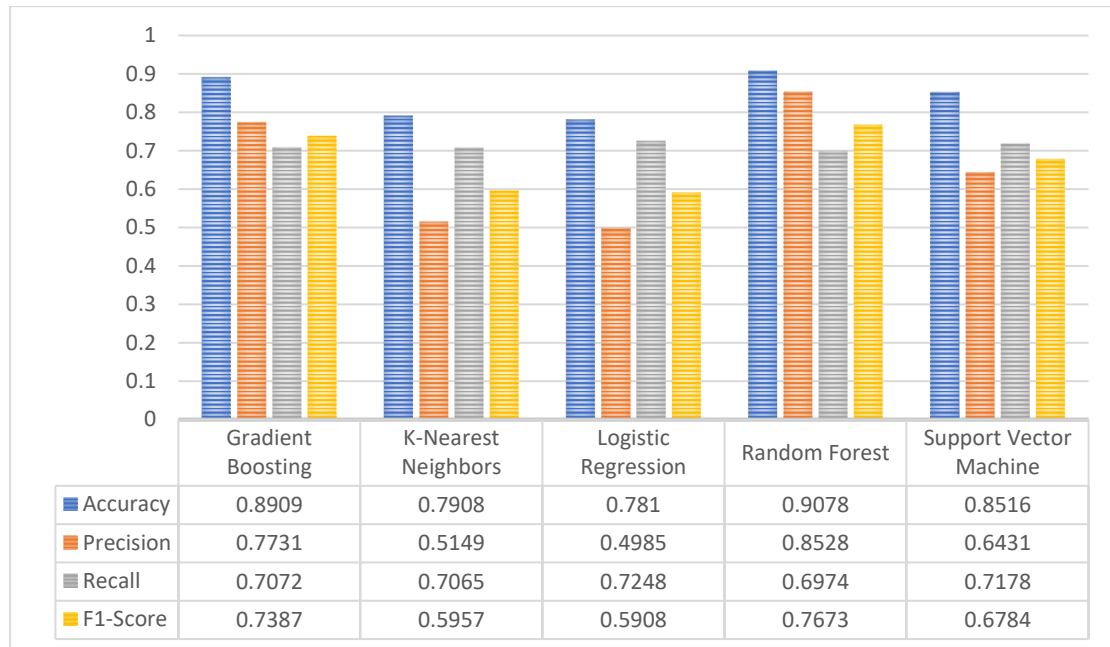


Figure 5.1.2: SMOTE Baseline Model Result

SMOTEENN Results

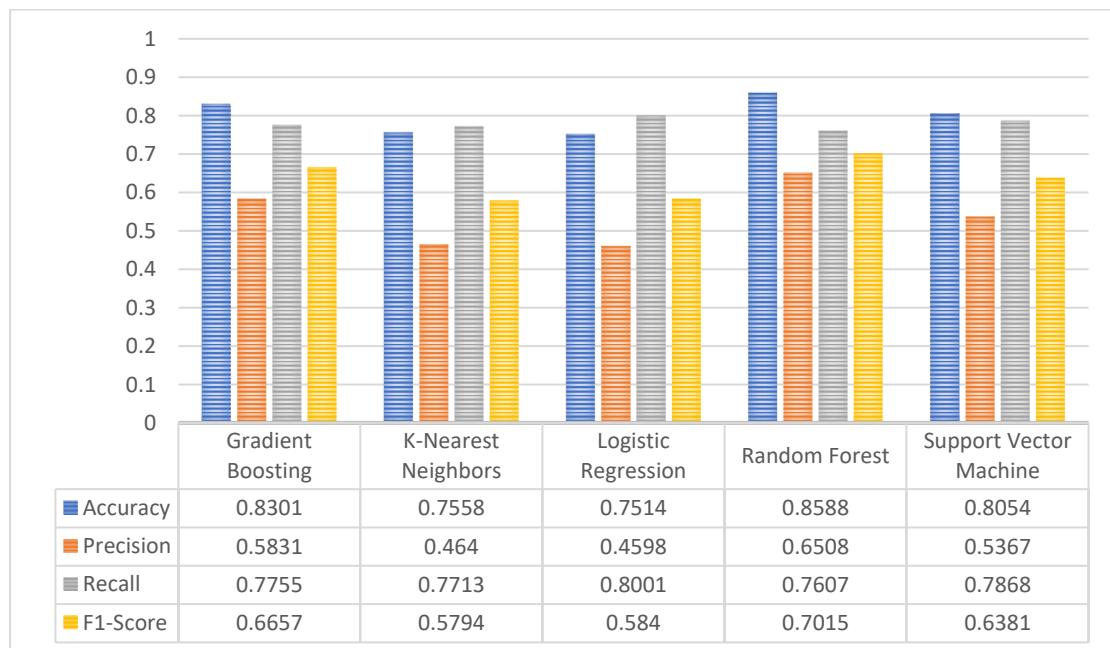
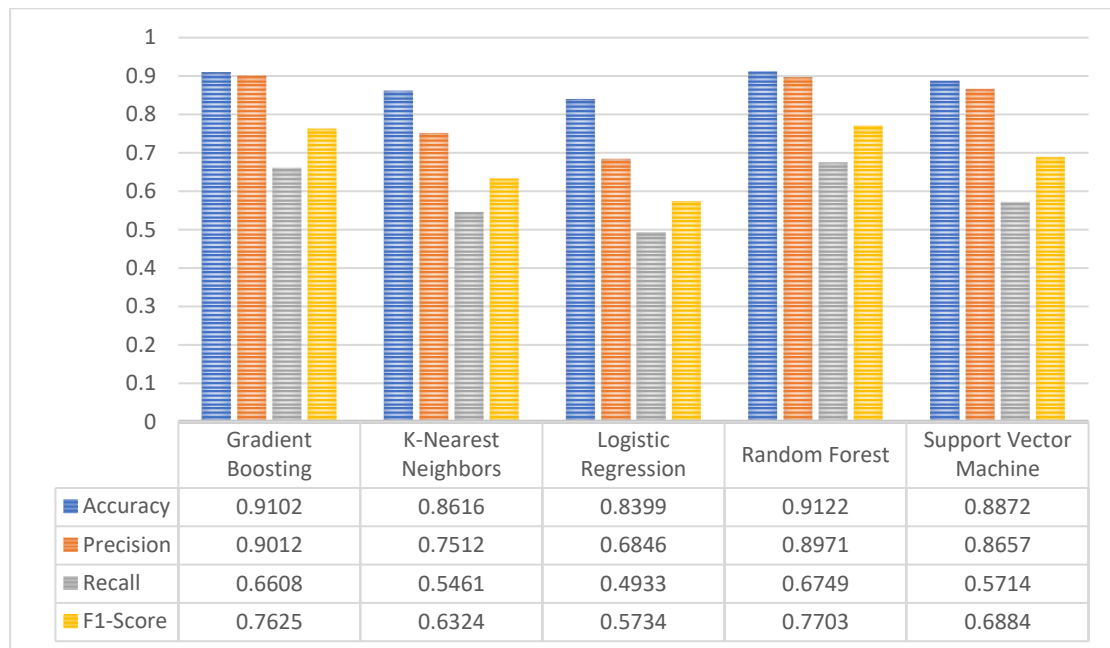
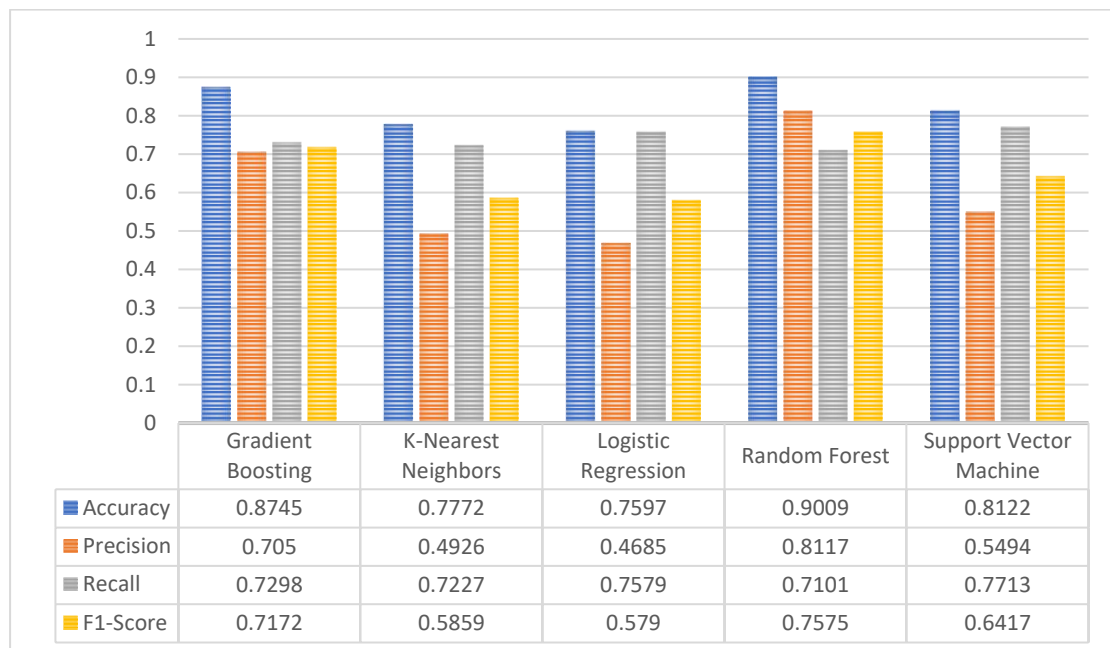
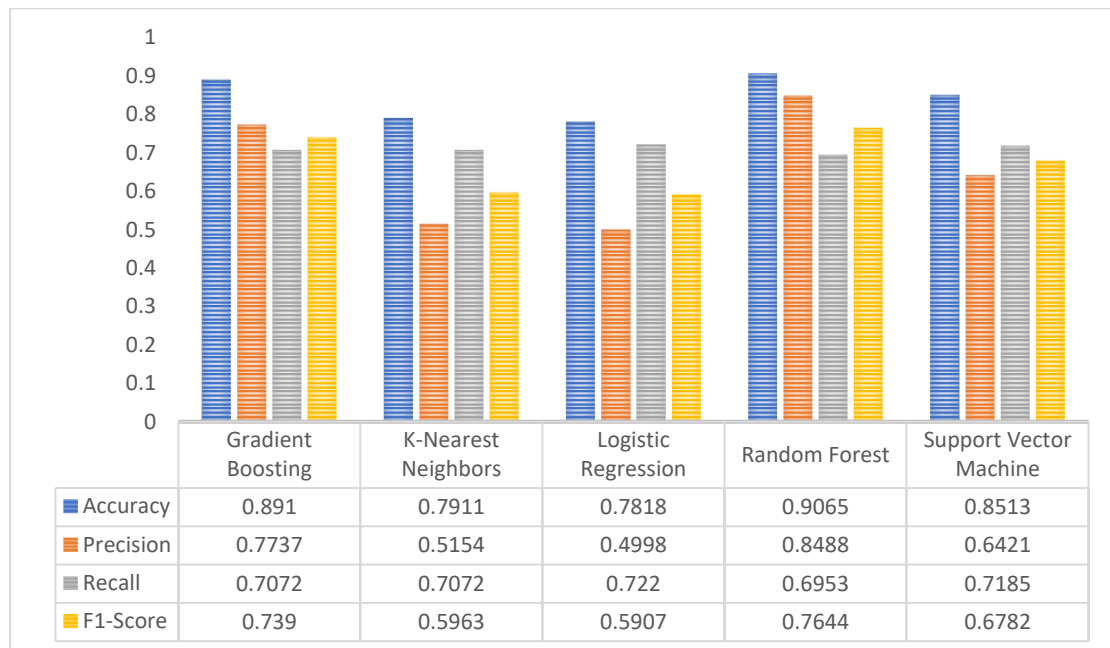
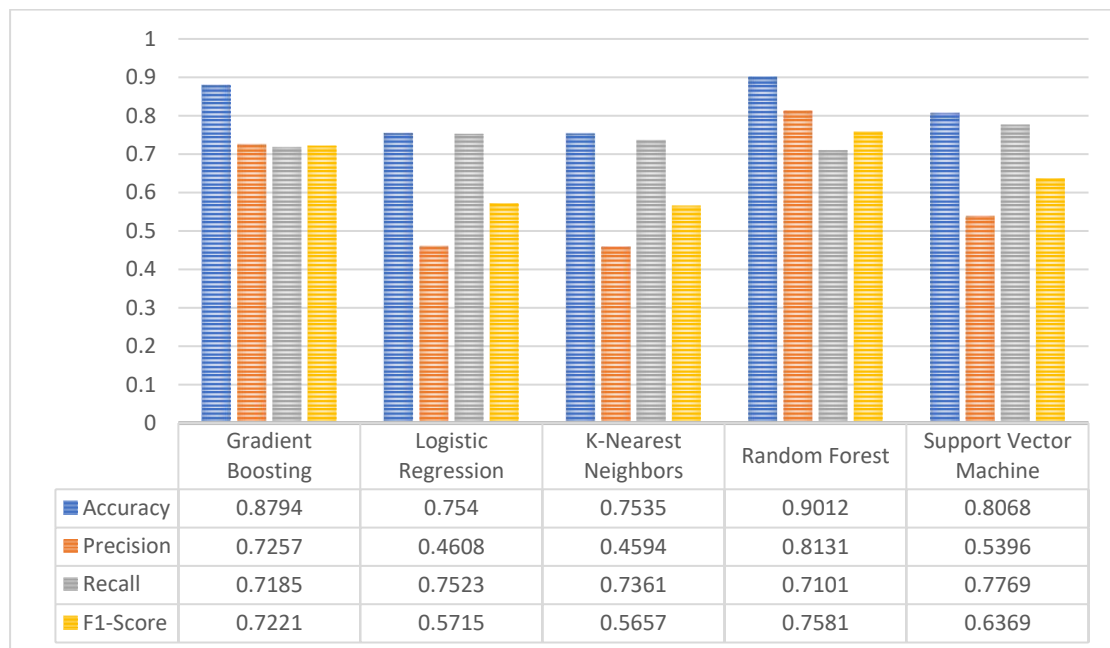


Figure 5.1.3: SMOTEENN Baseline Model Result

Tomek Results*Figure 5.1.4: Tomek Baseline Model Result***Borderline Results***Figure 5.1.5: Borderline Baseline Model Result*

SMOTETomek Results*Figure 5.1.6: SMOTETomek Baseline Model Result***ADASYN Results***Figure 5.1.7: ADASYN Baseline Model Result*

5.1.3 Analysis of Baseline Results

Random Forest Superiority:

The Random Forest model achieved the greatest accuracy (0.9122 with Tomek) and F1-scores (0.7703 with Tomek), consistently outperforming other models across the majority of imbalance handling strategies. The ensemble aspect of the model, which successfully manages intricate non-linear interactions in the credit data and minimizes overfitting, is responsible for its good performance. Additionally, Random Forest showed remarkable accuracy (0.8971 with Tomek), demonstrating its dependability in forecasting real bad loans—an essential component of risk assessment.

Gradient Boosting Performance:

Gradient Boosting appeared as the second most outperformed model, demonstrating strong performance particularly with Tomek (F1-score: 0.7625) and SMOTETomek (F1-score: 0.7390). The model showed good adaptability across various resampling techniques, maintaining relatively consistent performance and indicating robustness. Notably, it was able to sustain good precision without significantly compromising recall, achieving a balanced performance that is crucial in credit risk assessment.

Support Vector Machine Behavior:

SVM showed moderate performance with a peak F1-score of 0.6884 using Tomek, placing it in the middle of the model rankings. It particularly excelled with Tomek links, achieving a high precision of 0.8657, which suggests it benefited from the removal of borderline majority samples. However, SVM struggled with recall in these high-precision scenarios, indicating potential difficulties in identifying all bad loan cases.

K-Nearest Neighbors and Logistic Regression Limitations:

Both models consistently underperformed relative to other algorithms across all resampling techniques. K-Nearest Neighbors achieved a maximum F1-score of 0.6324 with Tomek, while Logistic Regression peaked at 0.5908 using SMOTE. Notably, both models exhibited consistently lower precision, often falling below 0.6, indicating a higher rate of false positives—an undesirable outcome in credit risk assessment due to the potential cost implications. However, they maintained competitive recall, particularly with SMOTEENN, achieving 0.7713 for KNN and 0.8001 for Logistic

Regression, suggesting they were still effective in identifying a significant portion of actual bad loans.

5.2 Fine-Tuning

Fine-tuning involves systematically searching for the optimal hyperparameter configurations that maximize model performance. For this project, a grid search methodology was employed to exhaustively explore predefined hyperparameter spaces for each model.

5.2.1 Fine-Tuning Methodology

The fine-tuning process followed these key steps:

1. **Grid Definition:** For each model type, a comprehensive grid of hyperparameters was defined with **3** values for each of the 4 key hyperparameters, creating a **3×3×3×3** configuration space.
2. **Cross-Validation:** 5-fold cross-validation was employed during the grid search to ensure robust evaluation of each hyperparameter combination.
3. **Performance Metric:** Accuracy was used as the primary optimization metric during grid search, with additional metrics calculated for comprehensive evaluation.
4. **Validation Assessment:** The best models identified from grid search were evaluated on the validation dataset to assess generalization performance.

5.2.2 Hyperparameter Grids

For every model, the following hyperparameter spaces were explored:

Random Forest:

```
param_grids = {
    'Random Forest': {
        'n_estimators': [100, 200, 300],
        'max_depth': [None, 10, 20],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
    }
}
```

Figure 5.2.1: Hyperparameter Grids of Random Forest

- **n_estimators [100, 200, 300]:** These values represent a common range for the number of trees in the forest. Starting at 100 ensures a sufficient ensemble size for stable predictions, while the upper limit of 300 balances computational efficiency with model performance. Research by Probst P [11] indicates that performance gains typically plateau beyond 300 trees for many datasets.
- **max_depth [None, 10, 20]:** This parameter controls the maximum depth of each decision tree. The inclusion of 'None' allows trees to grow until all leaves are pure, while values of 10 and 20 enforce different levels of tree complexity. This range helps identify the optimal balance between underfitting (too shallow) and overfitting (too deep).
- **min_samples_split [2, 5, 10]:** The minimum of samples needed to separate an internal node is determined by these values. The default value of 2 allows for very granular splits, while 5 and 10 enforce increasingly stricter splitting conditions, potentially reducing overfitting.
- **min_samples_leaf [1, 2, 4]:** This parameter sets the minimum number of samples required at a leaf node. Starting from the default value of 1, which allows singleton leaves, up to 4, which enforces more generalized leaf nodes. This progression helps

evaluate the trade-off between model specificity and generalizability.

Logistic Regression:

```
param_grids = {
    'Logistic Regression': {
        'C': [0.1, 1.0, 10.0],
        'penalty': ['l1', 'l2', 'elasticnet'],
        'solver': ['saga', 'liblinear', 'newton-cg'],
        'max_iter': [1000, 2000, 3000]
    }
}
```

Figure 5.2.2: Hyperparameter Grids of Logistic Regression

- **C [0.1, 1.0, 10.0]:** This inverse regularization parameter spans two orders of magnitude, allowing exploration from strong regularization (0.1) to minimal regularization (10.0). This logarithmic scale is effective for hyperparameter tuning as it tests the model's behavior under different regularization strengths.
- **penalty ['l1', 'l2', 'elasticnet']:** These represent the three main regularization types available in scikit-learn's Logistic Regression. L1 promotes sparsity, L2 prevents extreme weight, and elasticnet combines both approaches. Testing all three allows for identifying the most effective regularization approach for the credit scoring data.
- **solver ['saga', 'liblinear', 'newton-cg']:** These solvers were chosen for their compatibility with different penalty types. 'saga' supports all penalties including elasticnet, 'liblinear' is efficient for small datasets, and 'newton-cg' performs well with L2 regularization. This selection ensures we test different optimization algorithms appropriate for the regularization methods.
- **max_iter [1000, 2000, 3000]:** These iteration limits were chosen to ensure convergence even with complex parameter combinations. The default value of 1000 is extended to 3000 to accommodate scenarios where the model might require more iterations to converge, particularly with L1 regularization.

K-Nearest Neighbors:

```
param_grids = {
    'K-Nearest Neighbors': {
        'n_neighbors': [3, 5, 7],
        'weights': ['uniform', 'distance', None],
        'p': [1, 2, 3],
        'leaf_size': [10, 30, 50]
    }
}
```

Figure 5.2.3: Hyperparameter Grids of K-Nearest Neighbors

- **n_neighbors [3, 5, 7]:** This range spans commonly effective values for KNN. The selection starts with 3 to capture local patterns, while 7 provides more smoothing. The odd values help avoid tied votes in binary classification. The step size of 2 was chosen to evaluate different neighborhood sizes while maintaining computational efficiency.
- **weights ['uniform', 'distance', None]:** This parameter determines how neighbors contribute to classification. 'uniform' gives equal weight to all neighbors, 'distance' weights closer neighbors more heavily, and None tests the default behavior. This range allows evaluation of different neighbor weighting strategies.
- **p [1, 2, 3]:** These values represent the Minkowski distance parameter. p=1 corresponds to Manhattan distance, p=2 to Euclidean distance, and p=3 to a higher-order distance metric. Testing across these values helps identify the most appropriate distance metric for credit risk data.
- **leaf_size [10, 30, 50]:** This parameter affects the speed of the algorithm rather than the outcome. The selected values range from the relatively small (10) for potentially better precision to larger values (50) that might offer better query performance, enabling testing of performance-accuracy trade-offs.

Support Vector Machine:

```
param_grids = {
    'Support Vector Machine': {
        'C': [0.1, 1.0, 10.0],
        'kernel': ['linear', 'rbf', 'poly'],
        'gamma': ['scale', 'auto', 0.1],
        'degree': [2, 3, 4]
    }
}
```

Figure 5.2.4: Hyperparameter Grids of Support Vector Machine

- **C [0.1, 1.0, 10.0]:** Similar to Logistic Regression, this parameter controls the regularization strength. The logarithmic scale from 0.1 to 10.0 allows testing from strict regularization (small margin, few support vectors) to more flexible boundaries (larger margin, more support vectors).
- **kernel ['linear', 'rbf', 'poly']:** These three kernels represent different approaches to data transformation. 'linear' tests if the data is linearly separable, 'rbf' (Radial Basis Function) tests for non-linear circular decision boundaries, and 'poly' (Polynomial) tests for non-linear curved boundaries. This selection covers the most commonly effective kernels for classification tasks.
- **gamma ['scale', 'auto', 0.1]:** This parameter defines the influence radius of each training example for 'rbf' and 'poly' kernels. 'scale' and 'auto' are scikit-learn's heuristic approaches, while 0.1 provides a specific value for testing. This range allows evaluation of different kernel coefficient strategies.
- **degree [2, 3, 4]:** This parameter only affects the 'poly' kernel and determines the degree of the polynomial. The range from 2 to 4 covers simple quadratic functions to more complex quartic functions, allowing testing of different polynomial complexity levels without risking overfitting with higher degrees.

Gradient Boosting:

```

param_grids = {
    'Gradient Boosting': {
        'n_estimators': [100, 200, 300],
        'learning_rate': [0.01, 0.1, 0.2],
        'max_depth': [3, 5, 7],
        'subsample': [0.7, 0.8, 0.9]
    }
}

```

Figure 5.2.5: Hyperparameter Grids of Gradient Boosting

- **n_estimators [100, 200, 300]:** Similar to Random Forest, these values represent the number of boosting stages. Starting with 100 ensures sufficient model complexity, while the upper limit of 300 balances computational load with potential performance gains. This range allows testing how model performance scales with additional estimators.
- **learning_rate [0.01, 0.1, 0.2]:** This parameter regulates how much each tree contributes to the final result. The range spans from a conservative 0.01 (requiring more trees but potentially better generalization) to a more aggressive 0.2. This logarithmic progression allows testing different trade-offs between learning speed and model precision.
- **max_depth [3, 5, 7]:** These values restrict the depth of the individual decision trees. Gradient Boosting typically performs better with shallower trees than Random Forest, hence the lower range starting at 3. The upper limit of 7 prevents overfitting while still allowing sufficient complexity to capture important patterns.
- **subsample [0.7, 0.8, 0.9]:** The percentage of samples utilized to fit individual base learners is determined by this parameter. Values less than 1 introduce randomness that can improve generalization. The range from 0.7 to 0.9 tests different levels of stochastic behavior while maintaining sufficient data representation.

5.2.3 Pipeline and Data Scaling Strategy

An important consideration in model development is the need for feature scaling, which varies significantly between different algorithms. To address this, pipeline structures were implemented that incorporate appropriate scaling techniques tailored to each algorithm's requirements:

```
# Define pipelines with appropriate scalers for each algorithm
pipelines = {
    'Random Forest': RandomForestClassifier(random_state=42), # No scaler for tree-based models
    'Logistic Regression': Pipeline([
        ('scaler', QuantileTransformer(output_distribution='normal')),
        ('model', LogisticRegression(random_state=42, max_iter=2000))
    ]),
    'K-Nearest Neighbors': Pipeline([
        ('scaler', MinMaxScaler()),
        ('model', KNeighborsClassifier())
    ]),
    'Support Vector Machine': Pipeline([
        ('scaler', StandardScaler()),
        ('model', SVC(random_state=42, probability=True))
    ]),
    'Gradient Boosting': GradientBoostingClassifier(random_state=42) # No scaler needed
}
```

Figure 5.2.6: Data Scaling Strategy

Scaling Strategy Rationale

The scaling approach for each algorithm was carefully selected based on theoretical considerations and empirical best practices:

1. Tree-based Models (Random Forest and Gradient Boosting)

Tree-based models do not require feature scaling because they make decisions by creating binary splits in the feature space—essentially checking whether a value is above or below a certain threshold. These splits rely on the relative ordering of values rather than their actual magnitudes, so the scale of features does not influence model performance. As a result, preprocessing steps like standardization or normalization are unnecessary for models like Random Forest and Gradient Boosting. A key benefit of this is that the original feature distributions remain intact, which enhances interpretability, particularly valuable when analyzing feature importance in credit risk assessments.

2. Logistic Regression with QuantileTransformer

To enhance the performance of Logistic Regression on financial data, the

QuantileTransformer with a 'normal' output distribution was applied. This transformation maps feature values to follow a standard Gaussian distribution, which is particularly useful for handling non-Gaussian features and mitigating the influence of outliers—common occurrences in credit datasets, such as unusually high incomes or loan amounts. Since Logistic Regression is sensitive to feature scale and outliers, transforming the data using QuantileTransformer, which ranks and evenly distributes values, makes the model more robust. This preprocessing step not only stabilizes convergence during training but also improves predictive performance when working with skewed or heavy-tailed distributions.

3. K-Nearest Neighbors with MinMaxScaler

K-Nearest Neighbors relies on distance calculations between data points, making it highly sensitive to the scale of features. To address this, the MinMaxScaler is applied, which normalizes all features to a [0,1] range. This prevents features with larger original ranges, such as income or loan amount, from disproportionately influencing the distance calculations. Unlike the StandardScaler, which centers data around the mean, the MinMaxScaler preserves the shape of the original distribution, maintaining the relative positions of data points. This scaling approach enhances KNN's ability to find meaningful neighbors by ensuring that all features contribute equally to the distance metric, rather than being dominated by those with larger magnitudes.

4. Support Vector Machine with StandardScaler

Support Vector Machines, especially when using Radial Basis Function or polynomial kernels, are highly sensitive to the scale of features. To address this, the StandardScaler is applied, transforming characteristics with a unit variance and zero mean. This normalization is ideal for the mathematical operations involved in SVM's optimization process, ensuring that all features contribute equally. Properly scaled features also improve convergence during training, enabling the SVM solver to find an optimal solution more quickly and reliably. The key benefit of this scaling is that it enhances the effectiveness of the regularization parameter (C), enabling better performance for kernel functions that depend on distance metrics, such as RBF.

5.2.4 Fine-Tuning Results

The results of the fine-tuning process across different resampling techniques revealed significant improvements over the baseline models. The following sections present the result of the fine-tuned models on the validation dataset.

SMOTE Fine-Tuning Results

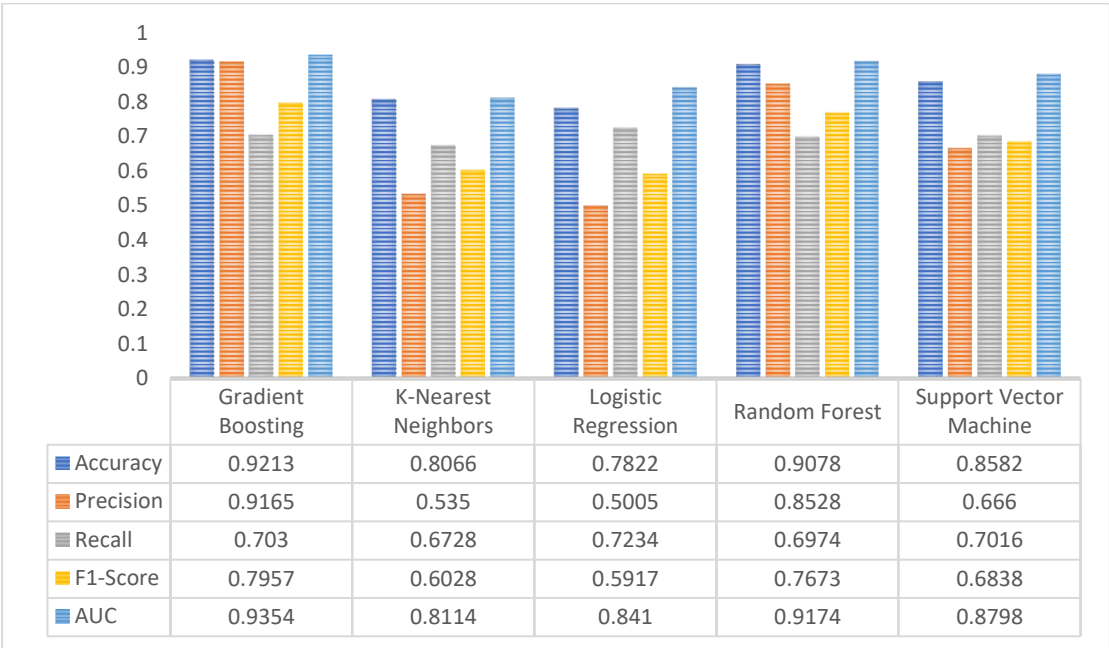
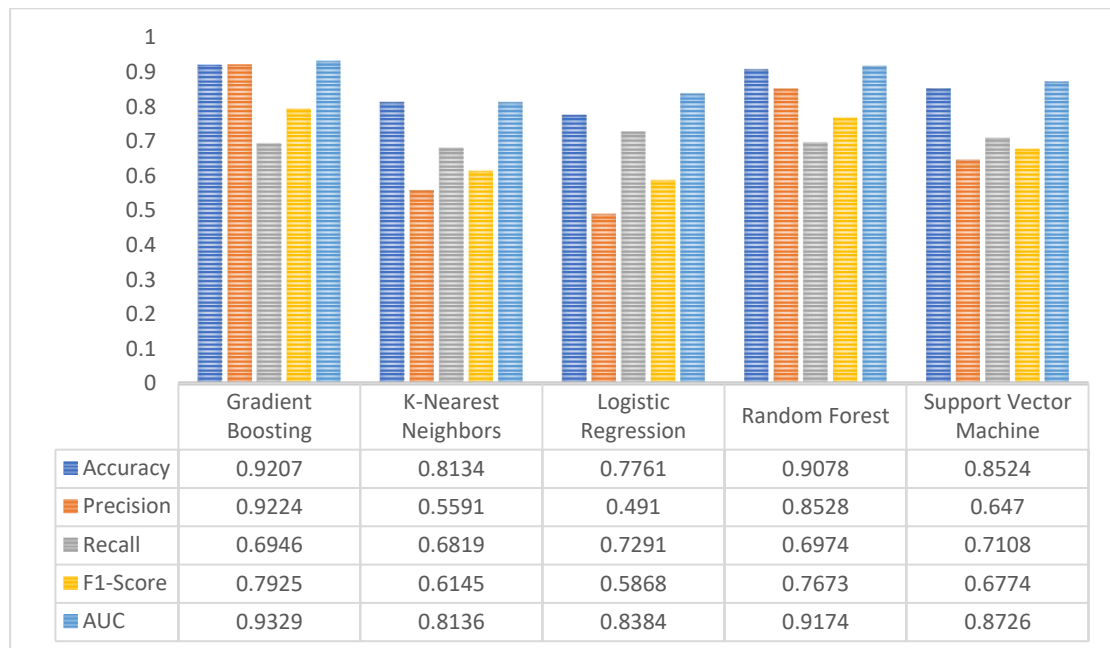
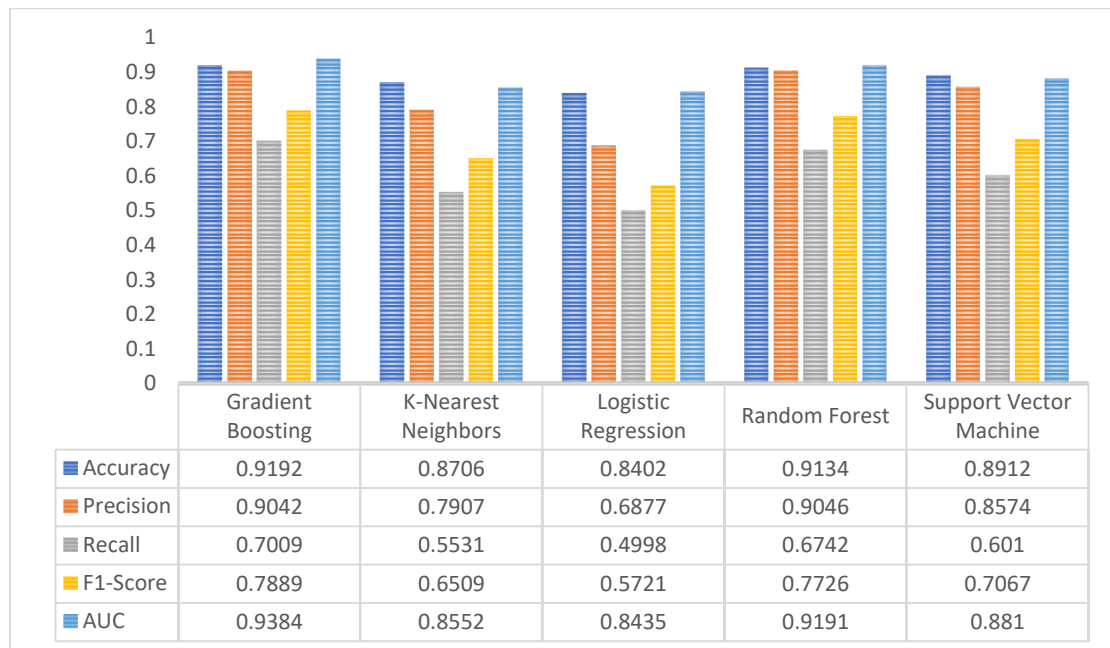


Figure 5.2.7: SMOTE Fine-Tuning Results

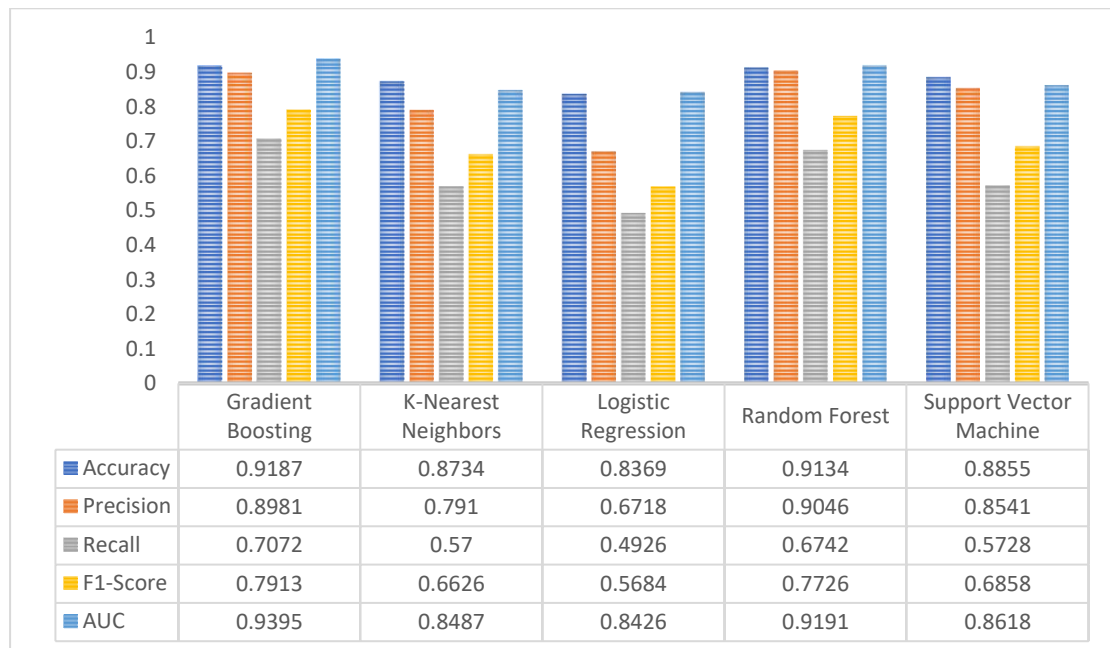
The fine-tuned Gradient Boosting model demonstrated superior performance with the highest accuracy (0.9213) and precision (0.9165) in the SMOTE setting. It also achieved the best F1-score (0.7957) and AUC (0.9354), indicating excellent balance between identifying bad loan correctly and minimizing false positives. Random Forest was the second-best performer with strong accuracy (0.9078) and precision (0.8528), making it an effective alternative for credit risk assessment.

SMOTE Fine-Tuning Results with Data Scaling*Figure 5.2.8: SMOTE Fine-Tuning Results with Data Scaling*

With data scaling applied, K-Nearest Neighbors showed noticeable improvement in all metrics compared to its non-scaled version, with F1-score increasing from 0.6028 to 0.6145. Tree-based models (Gradient Boosting and Random Forest) maintained consistent performance regardless of scaling, as expected. The scaled Logistic Regression model showed slight changes in performance metrics, while SVM experienced minor reductions in some metrics with scaling.

Tomek Fine-Tuning Results*Figure 5.2.9: Tomek Fine-Tuning Results*

In the Tomek Links configuration, Random Forest achieved the highest accuracy (0.9134) with excellent precision (0.9046), demonstrating strong ability to correctly identify bad loan cases. Gradient Boosting followed closely with high accuracy (0.9192) and the best F1-score (0.7889). Support Vector Machine showed improved performance with Tomek Links compared to SMOTE, achieving better precision (0.8574) and F1-score (0.7067), suggesting Tomek Links' undersampling approach benefits SVM.

Tomek Fine-Tuning Results with Data Scaling*Figure 5.2.10: Tomek Fine-Tuning Results with Data Scaling*

With data scaling applied to Tomek Links, K-Nearest Neighbors showed improvement in recall (0.5700 vs. 0.5531) and F1-score (0.6626 vs. 0.6509), confirming that distance-based algorithms benefit from scaling. Gradient Boosting maintained strong performance with scaled features, showing a slight improvement in F1-score (0.7913 vs. 0.7889). The tree-based Random Forest model remained unchanged by scaling, while Logistic Regression and SVM showed minor variations in their performance metrics.

SMOTETomek Fine-Tuning Results

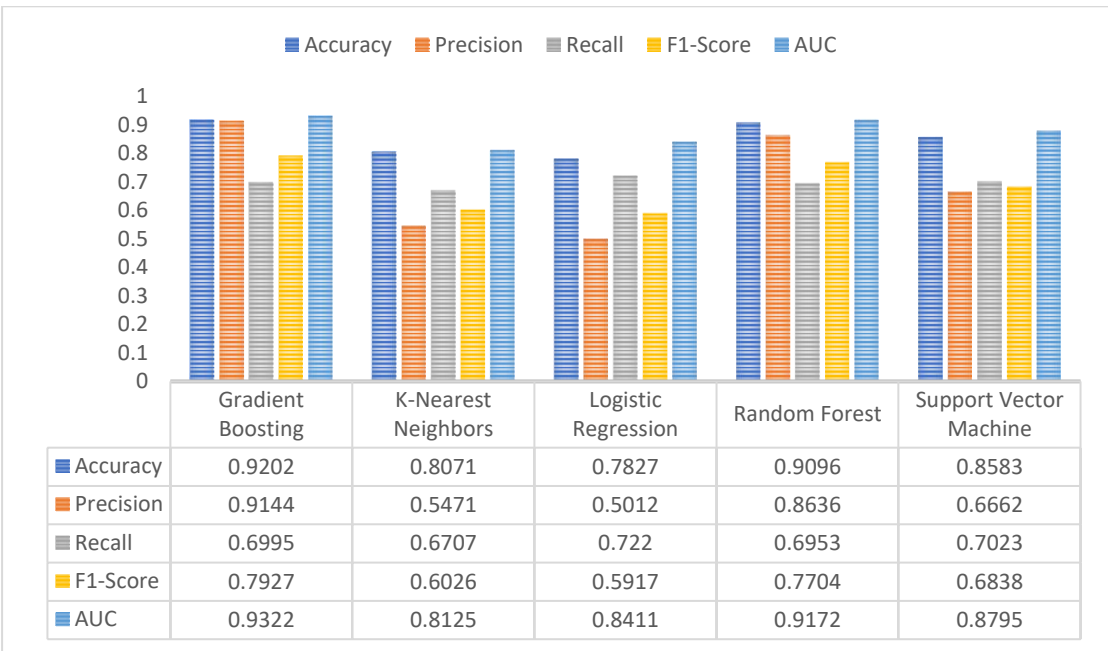


Figure 5.2.11: SMOTETomek Fine-Tuning Results

The SMOTETomek hybrid approach produced excellent results for Gradient Boosting, which achieved the highest accuracy (0.9202) and precision (0.9144) among all models. Random Forest followed with strong performance (accuracy 0.9096, F1-score 0.7704). This resampling technique maintained good recall across all models while preserving reasonable precision, indicating that the combination of oversampling and undersampling creates a well-balanced dataset for training.

SMOTETomek Fine-Tuning Results with Data Scaling

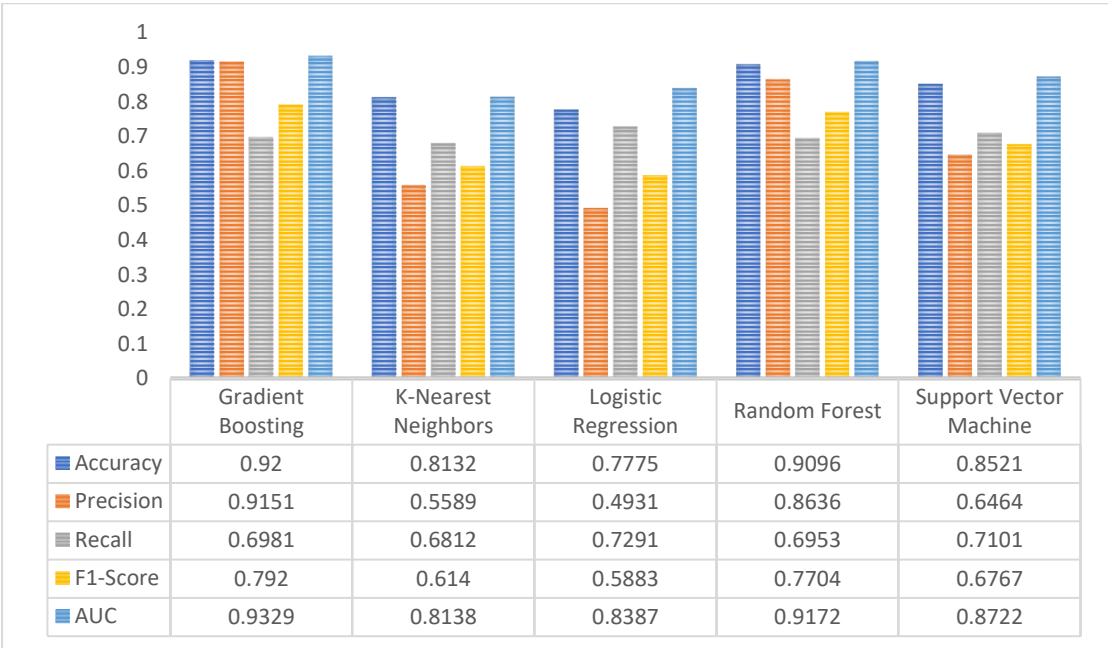
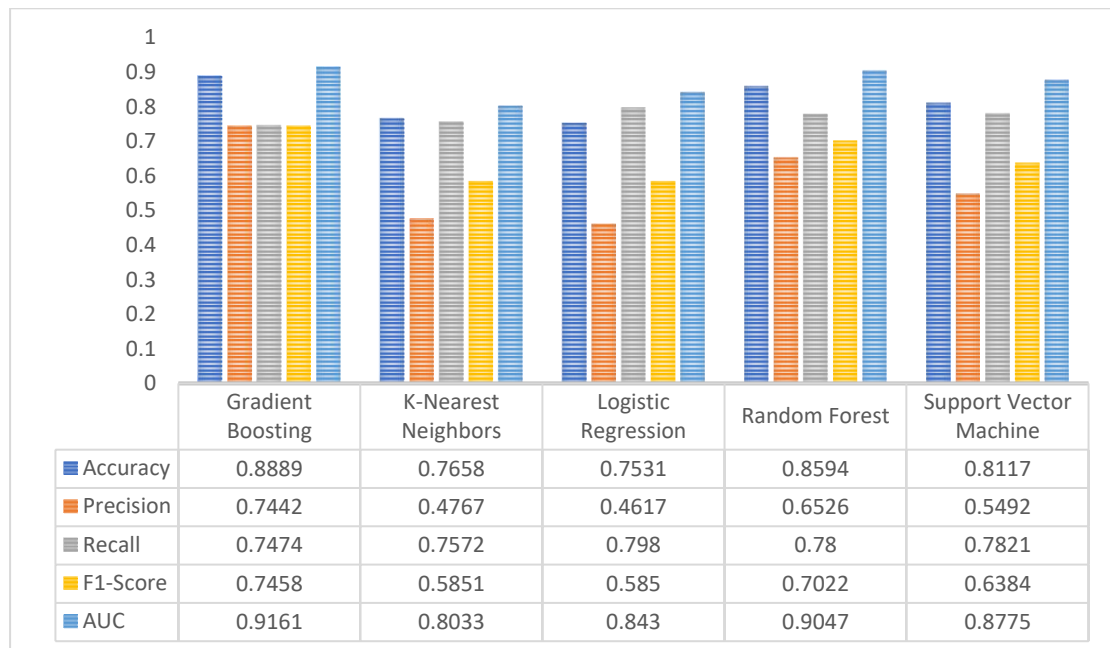


Figure 5.2.12: SMOTETomek Fine-Tuning Results with Data Scaling

With data scaling applied to SMOTETomek, K-Nearest Neighbors showed notable improvement in F1-score (0.6140 vs. 0.6026) and recall (0.6812 vs. 0.6707). Gradient Boosting maintained consistent performance with minimal changes in metrics. The scaled Logistic Regression model showed a slight decrease in precision but maintained similar overall performance. As expected, Random Forest remained unaffected by scaling, while SVM showed minor changes in its performance metrics.

SMOTEENN Fine-Tuning Results*Figure 5.2.13: SMOTEENN Fine-Tuning Results*

SMOTEENN significantly boosted recall across all models, with Logistic Regression achieving the highest recall (0.7980) at the expense of precision. Gradient Boosting maintained the best balance with the highest F1-score (0.7458) and accuracy (0.8889). This resampling technique's focus on cleaning difficult examples after oversampling resulted in models that are more sensitive to detecting bad loan but with reduced precision compared to other techniques.

SMOTEENN Fine-Tuning Results with Data Scaling

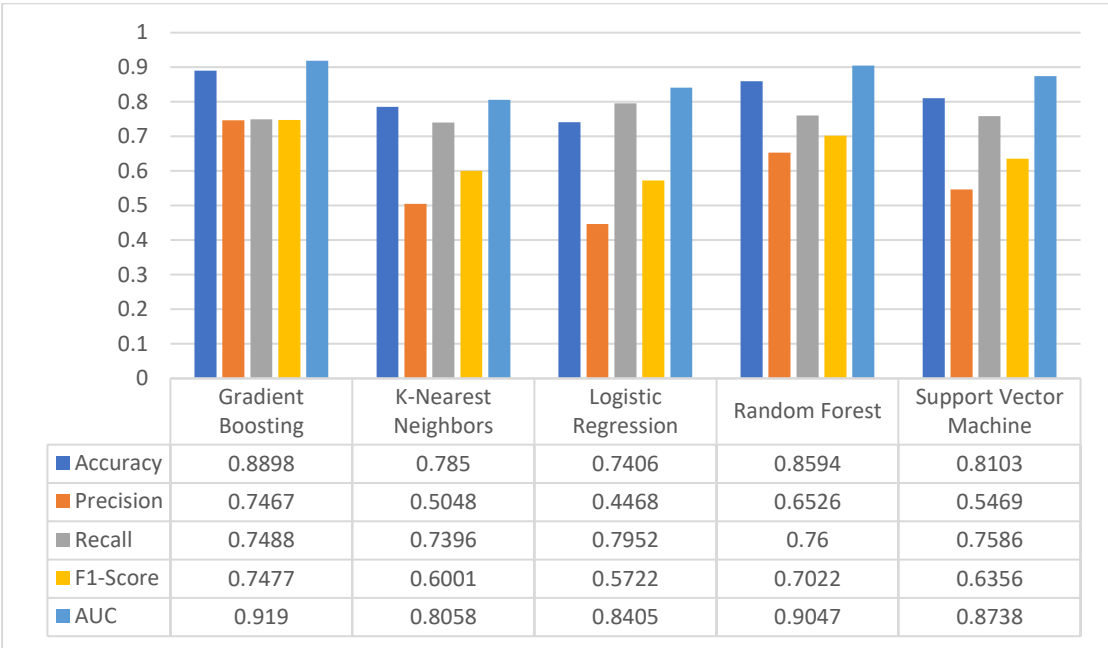
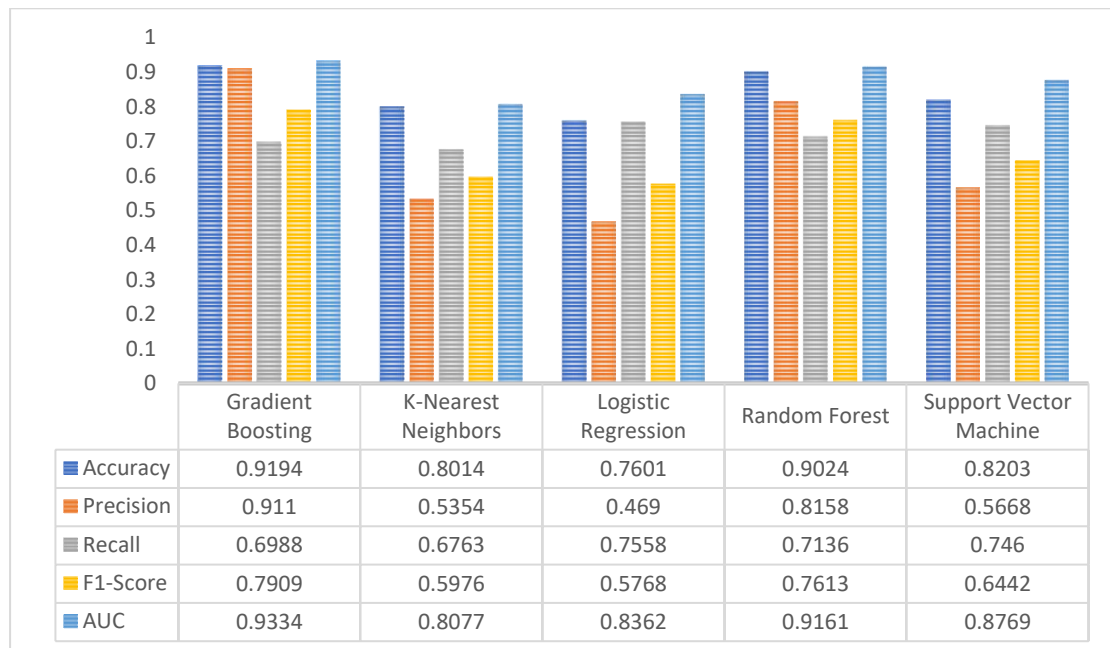
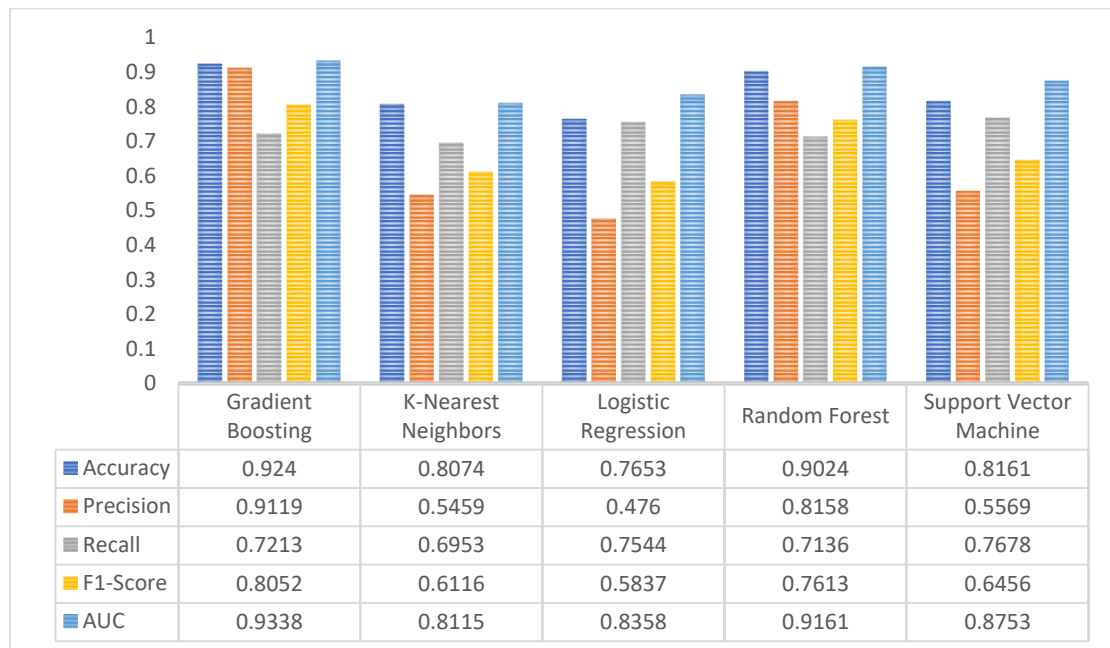


Figure 5.2.14: SMOTEENN Fine-Tuning Results with Data Scaling

With data scaling applied to SMOTEENN, K-Nearest Neighbors showed significant improvement in F1-score (0.6001 vs. 0.5851). Gradient Boosting maintained strong performance with a slight improvement in F1-score (0.7477 vs. 0.7458). Logistic Regression showed a decrease in precision but maintained high recall. Random Forest remained unaffected by scaling for most metrics, while SVM showed minor variations in performance.

Borderline Fine-Tuning Results*Figure 5.2.15: Borderline Fine-Tuning Results*

Borderline SMOTE produced strong results for Gradient Boosting, which maintained the highest accuracy (0.9194) and precision (0.9110). Random Forest showed excellent balance with a strong F1-score (0.7613) and the highest recall among tree-based models (0.7136). This focused oversampling approach that concentrates on the border between classes appeared to benefit tree-based models while providing reasonable performance for other algorithms.

Borderline Fine-Tuning Results with Data Scaling*Figure 5.2.16: Borderline Fine-Tuning Results with Data Scaling*

With data scaling applied to Borderline SMOTE, Gradient Boosting showed notable improvement in F1-score (0.8052 vs. 0.7909) and recall (0.7213 vs. 0.6988). K-Nearest Neighbors also benefited from scaling with improved F1-score (0.6116 vs. 0.5976) and recall (0.6953 vs. 0.6763). Logistic Regression showed a slight improvement in F1-score (0.5837 vs. 0.5768) with scaled features. Random Forest remained unaffected by scaling, while SVM experienced minor changes in performance.

ADASYN Fine-Tuning Results

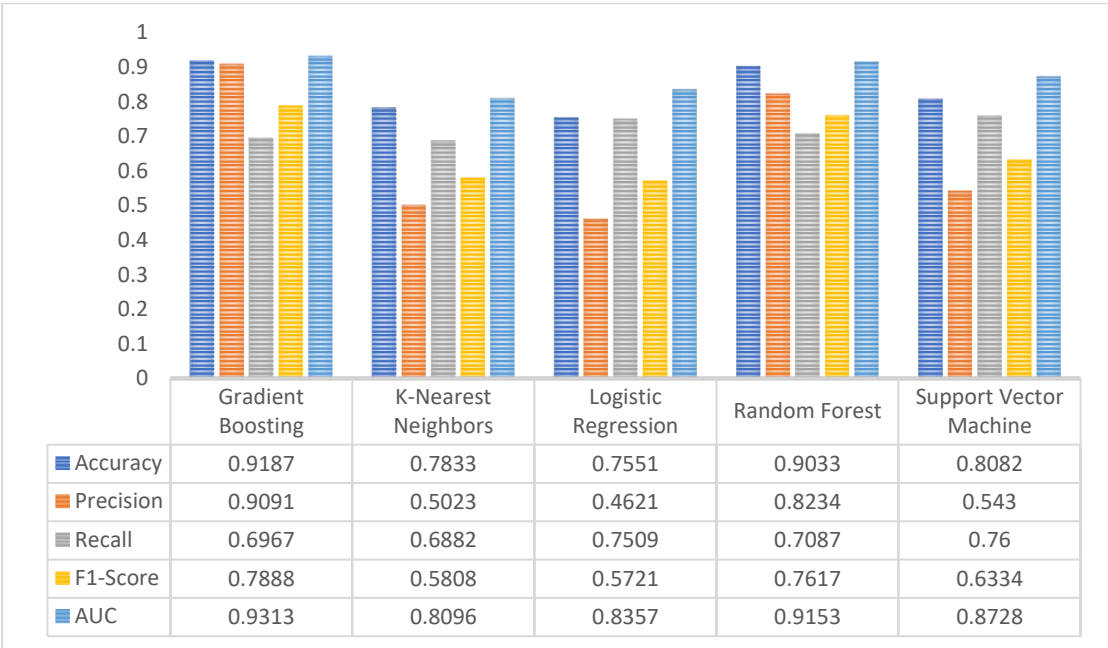
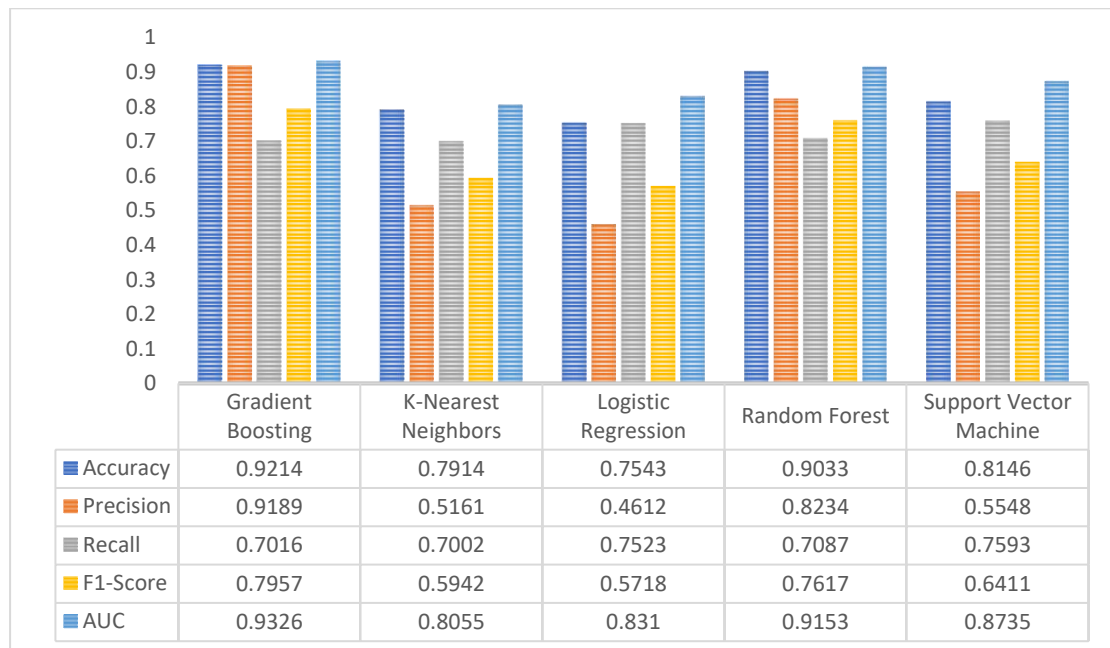


Figure 5.2.17: Adasyn Fine-Tuning Results

ADASYN provided strong results for Gradient Boosting, which maintained the highest accuracy (0.9187) and precision (0.9091) among all models. Random Forest showed excellent balance with a strong F1-score (0.7617) and good recall (0.7087). This adaptive synthetic sampling approach appeared to benefit tree-based models while providing decent recall for SVM and Logistic Regression, though at the cost of precision.

ADASYN Fine-Tuning Results with Data Scaling*Figure 5.2.17: Adasyn Fine-Tuning Results with Data Scaling*

With data scaling applied to ADASYN, K-Nearest Neighbors showed significant improvement in F1-score (0.5942 vs. 0.5808) and recall (0.7002 vs. 0.6882). Gradient Boosting maintained strong performance with an improved F1-score (0.7957 vs. 0.7888). Logistic Regression showed minimal changes in performance metrics. Random Forest remained unaffected by scaling, while SVM showed improvement in F1-score (0.6411 vs. 0.6334) with scaled features.

CHAPTER 6 FINAL MODEL EVALUATION AND SYSTEM DASHBOARD

6.1 Final Model Results

SMOTE Final Results

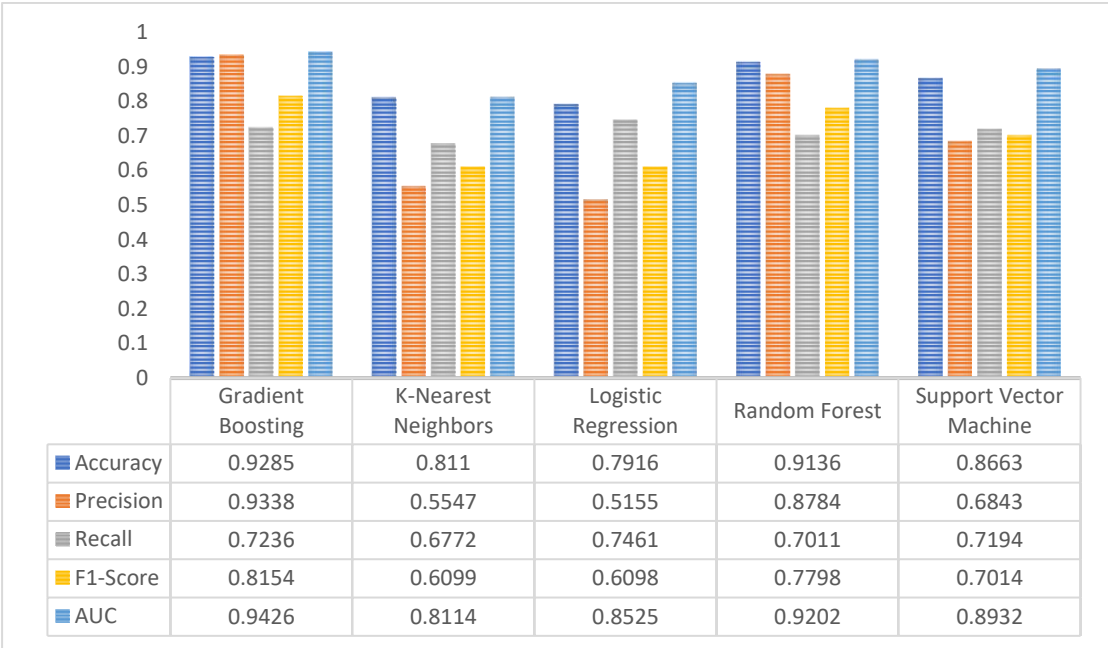
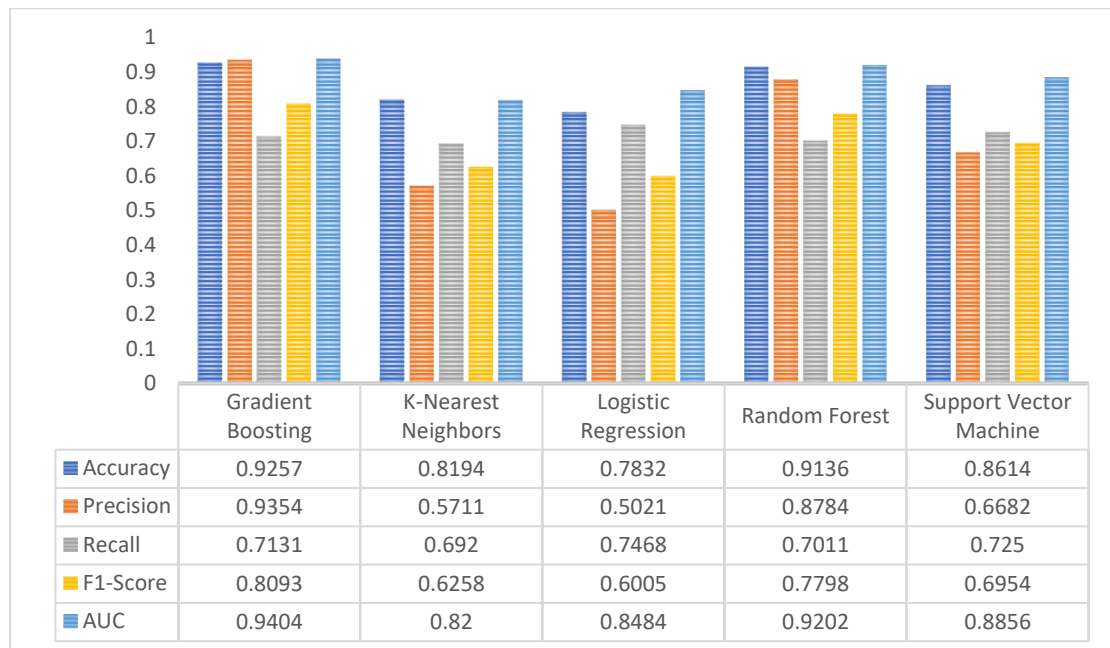


Figure 6.1.1: SMOTE Final Results

In the final evaluation on the test set, Gradient Boosting with SMOTE achieved the highest accuracy (0.9285), precision (0.9338), and F1-score (0.8154). The model's excellent AUC (0.9426) confirms its strong discriminative power for credit risk assessment. Random Forest continued to be a strong performer with robust accuracy (0.9136) and precision (0.8784), making it appropriate for uses where reducing false positives is critical.

SMOTE Final Results with Data Scaling*Figure 6.1.2: SMOTE Final Results with Data Scaling*

Data scaling significantly benefited the K-Nearest Neighbors model in the final evaluation, improving its F1-score from 0.6099 to 0.6258 and AUC from 0.8114 to 0.8200. The Gradient Boosting model with scaling maintained excellent performance, though with a slightly lower F1-score (0.8093) compared to its non-scaled version. As expected, Random Forest showed identical performance with or without scaling, confirming its invariance to feature scaling.

Tomek Final Results

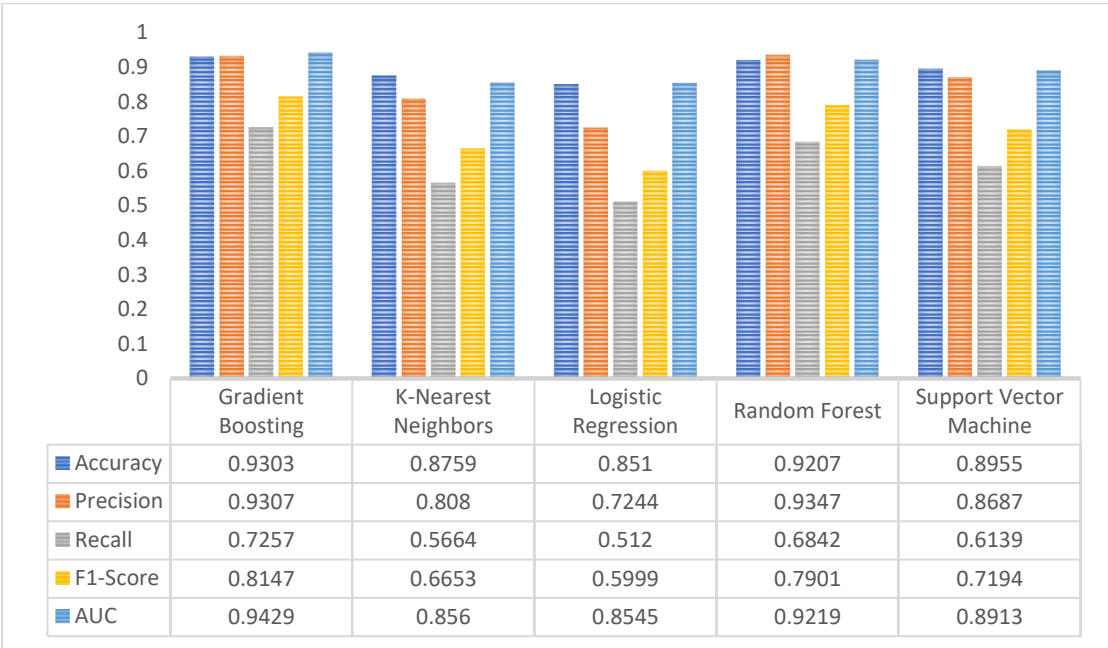


Figure 6.1.3: Tomek Final Results

Tomek Links produced excellent results in the final evaluation, with Gradient Boosting achieving the highest accuracy (0.9303) and F1-score (0.8147). Random Forest demonstrated the best precision (0.9347) with strong accuracy (0.9207), making it ideal for minimizing false positives. Notably, all models showed improved precision with Tomek Links compared to other resampling techniques, suggesting this undersampling approach creates clearer decision boundaries for classification.

Tomek Final Results with Data Scaling

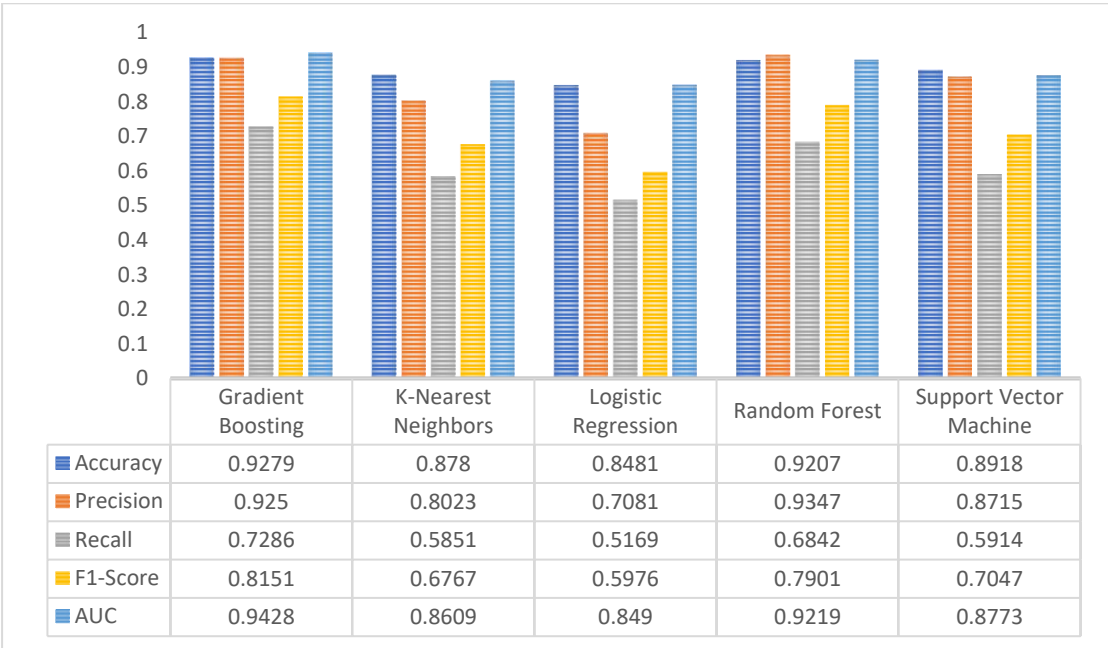


Figure 6.1.4: Tomek Final Results with Data Scaling

With data scaling applied to Tomek Links in the final evaluation, K-Nearest Neighbors showed improvement in recall (0.5851 vs. 0.5664) and F1-score (0.6767 vs. 0.6653). Gradient Boosting maintained strong performance with a slightly higher F1-score (0.8151 vs. 0.8147). Logistic Regression showed a minor decrease in performance, while SVM experienced reduced recall but maintained similar overall performance. Random Forest remained unaffected by scaling.

SMOTETomek Final Results

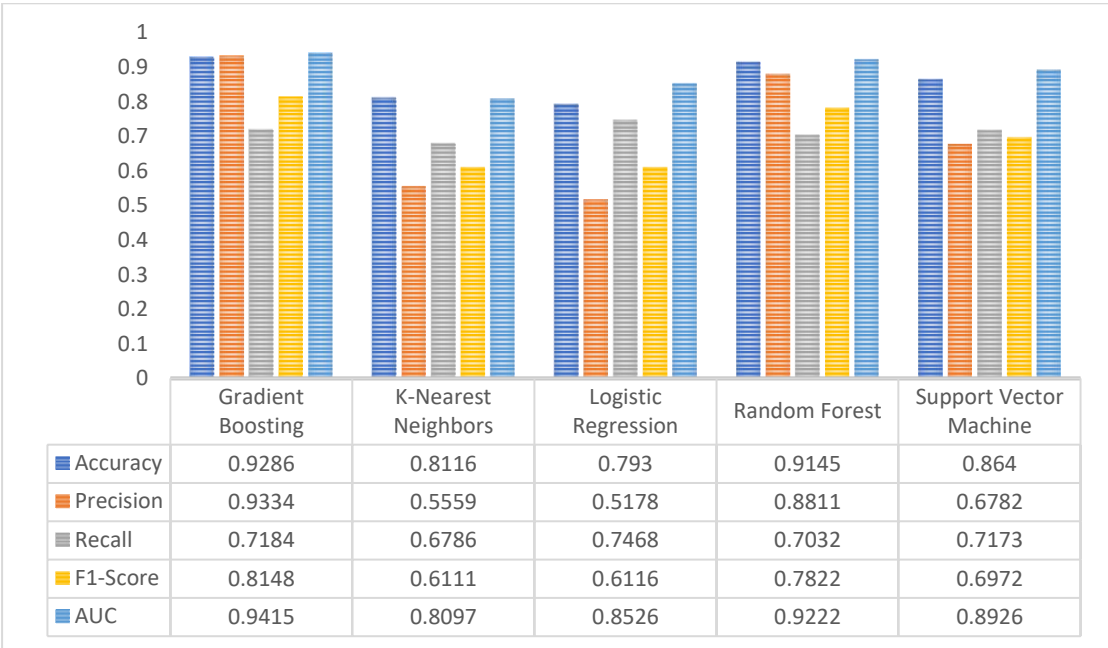
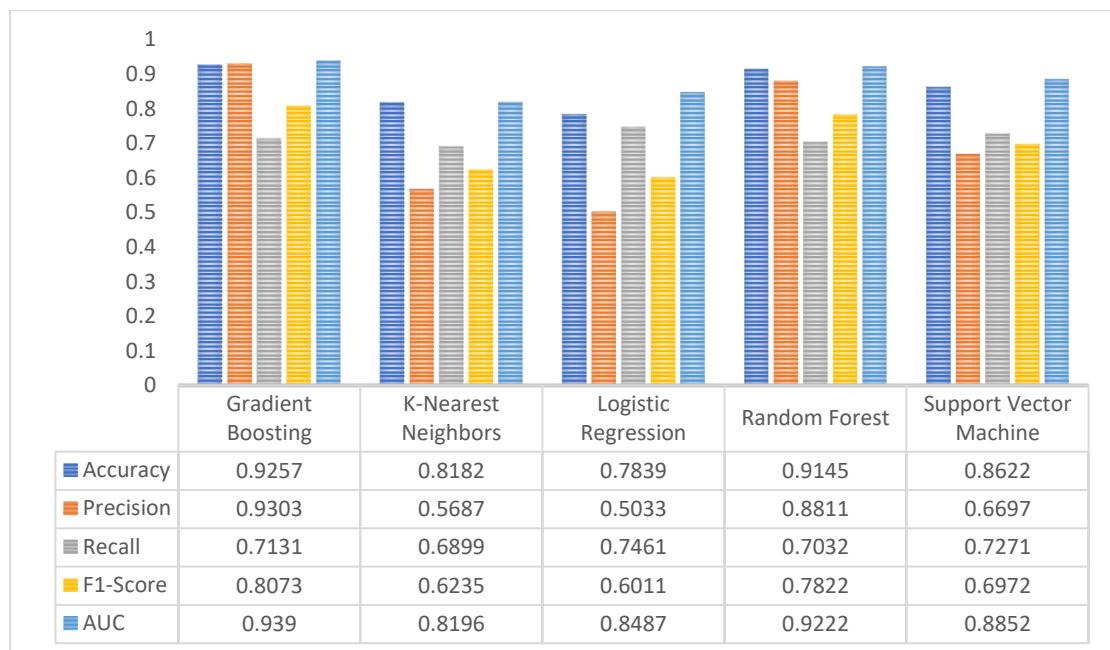


Figure 6.1.5: SMOTETomek Final Results

SMOTETomek provided strong results in the final evaluation, with Gradient Boosting achieving the highest accuracy (0.9286), precision (0.9334), and F1-score (0.8148). Random Forest followed with excellent performance (accuracy 0.9145, F1-score 0.7822). Especially for tree-based models, this hybrid resampling technique showed an excellent compromise between precision and recall.

SMOTETomek Final Results with Data Scaling*Figure 6.1.6: SMOTETomek Final Results with Data Scaling*

With data scaling applied to SMOTETomek in the final evaluation, K-Nearest Neighbors showed improvement in F1-score (0.6235 vs. 0.6111) and recall (0.6899 vs. 0.6786). Gradient Boosting maintained strong performance but with a slightly lower F1-score (0.8073 vs. 0.8148). Logistic Regression showed a minor decrease in performance, while SVM maintained a similar F1-score with different precision-recall balance. Random Forest remained unaffected by scaling.

SMOTEENN Final Results

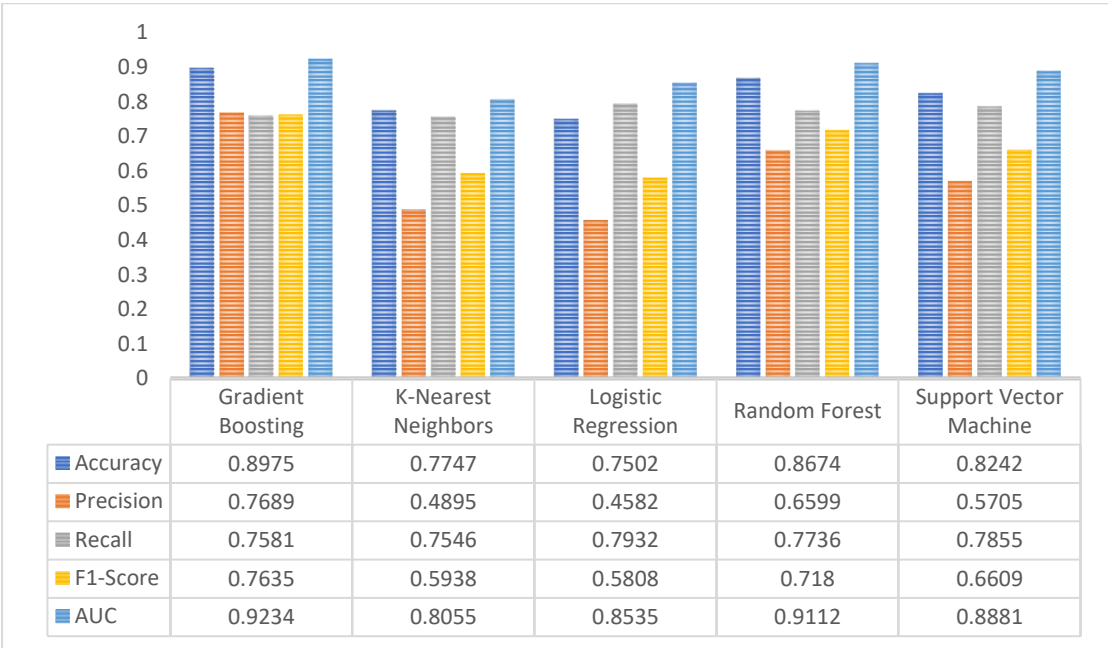
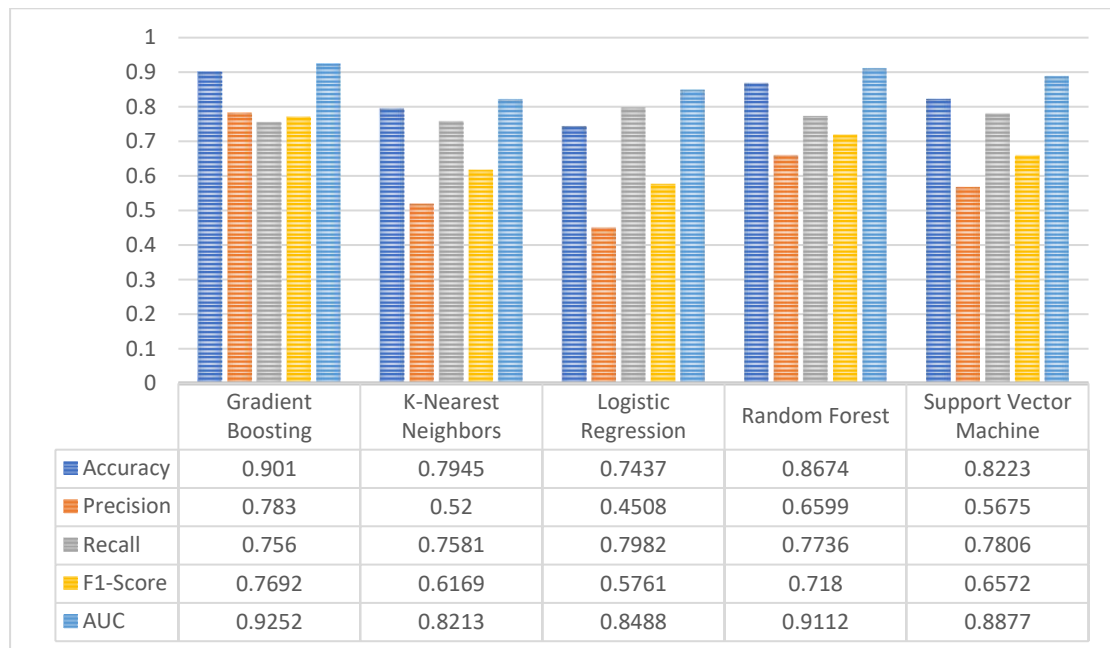


Figure 6.1.7: SMOTEENN Final Results

SMOTEENN significantly boosted recall across all models in the final evaluation, with SVM (0.7855) and Logistic Regression (0.7932) achieving the highest recall scores. Gradient Boosting maintained the best balance with the highest F1-score (0.7635) and accuracy (0.8975). This resampling technique focused more on detecting all potential bad loan, making it suitable for scenarios where minimizing missed bad loan is the priority.

SMOTEENN Final Results with Data Scaling*Figure 6.1.8: SMOTEENN Final Results with Data Scaling*

With data scaling applied to SMOTEENN in the final evaluation, K-Nearest Neighbors showed significant improvement in F1-score (0.6169 vs. 0.5938). Gradient Boosting maintained strong performance with improvement in F1-score (0.7692 vs. 0.7635) and precision (0.7830 vs. 0.7689). Logistic Regression maintained high recall but with slight changes in other metrics. Random Forest remained unaffected by scaling, while SVM showed minimal changes in performance.

Borderline Final Results

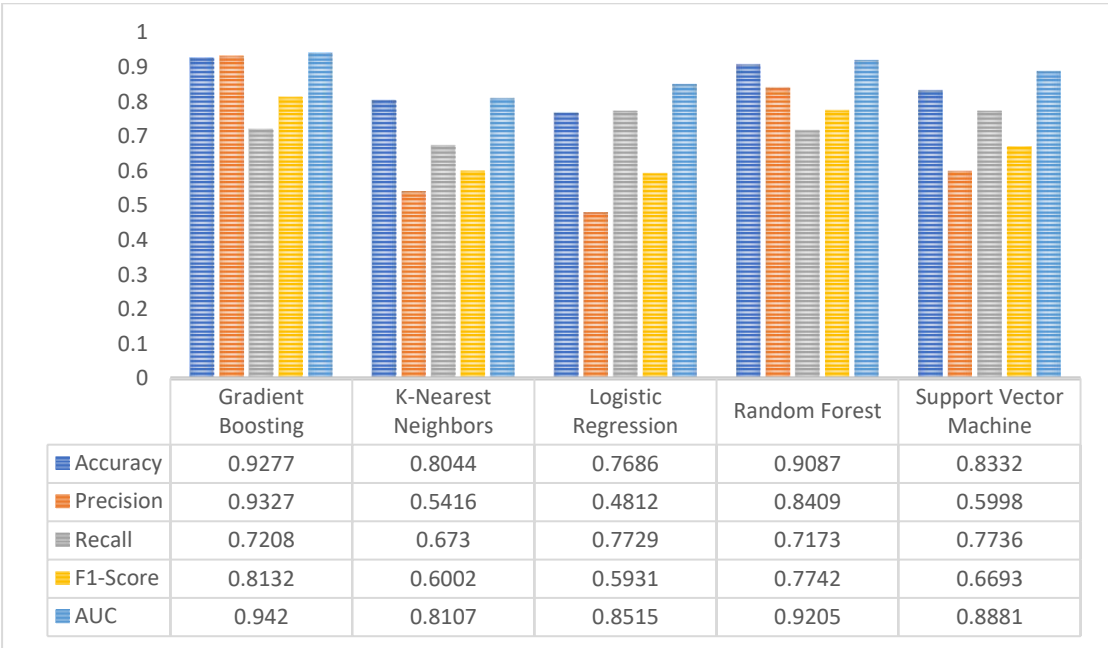
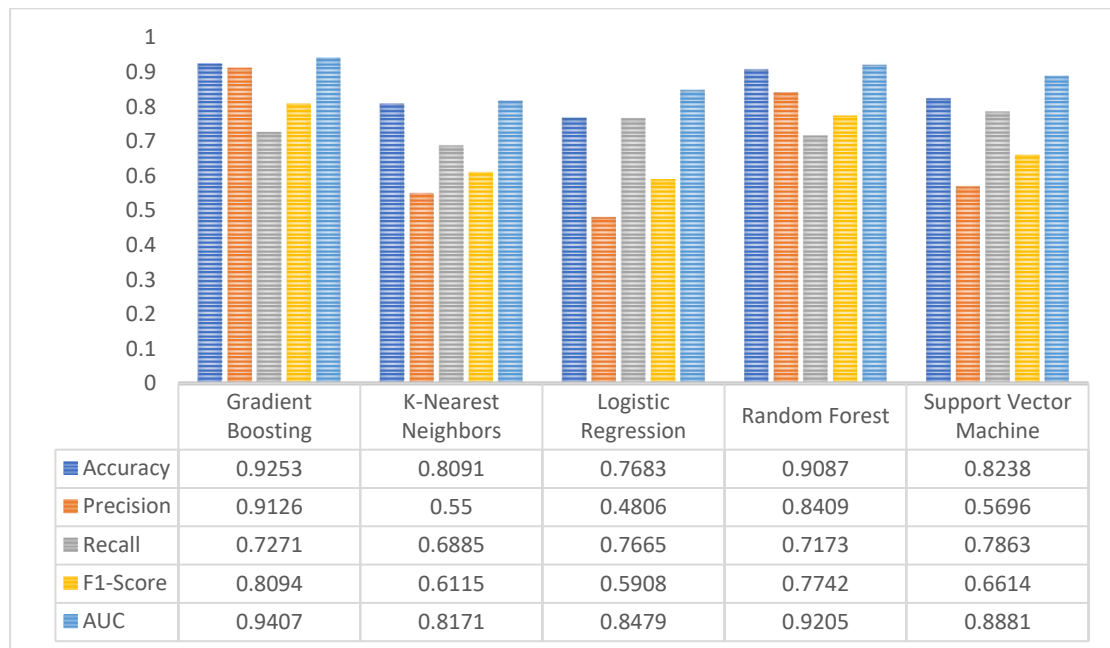


Figure 6.1.9: Borderline Final Results

Borderline SMOTE provided strong results in the final evaluation, with Gradient Boosting achieving high accuracy (0.9277), precision (0.9327), and F1-score (0.8132). Random Forest demonstrated excellent performance (accuracy 0.9087, F1-score 0.7742) with good recall (0.7173). This focused oversampling approach appeared to benefit tree-based models while maintaining reasonable recall for SVM and Logistic Regression.

Borderline Final Results with Data Scaling*Figure 6.1.10: Borderline Final Results with Data Scaling*

With data scaling applied to Borderline SMOTE in the final evaluation, K-Nearest Neighbors showed improvement in F1-score (0.6115 vs. 0.6002) and recall (0.6885 vs. 0.6730). Gradient Boosting maintained strong performance with a slightly lower F1-score (0.8094 vs. 0.8132) but improved recall (0.7271 vs. 0.7208). Logistic Regression showed minimal changes, while SVM demonstrated improved recall but a slightly lower F1-score. Random Forest remained unaffected by scaling, as expected for tree-based models.

ADASYN Final Results

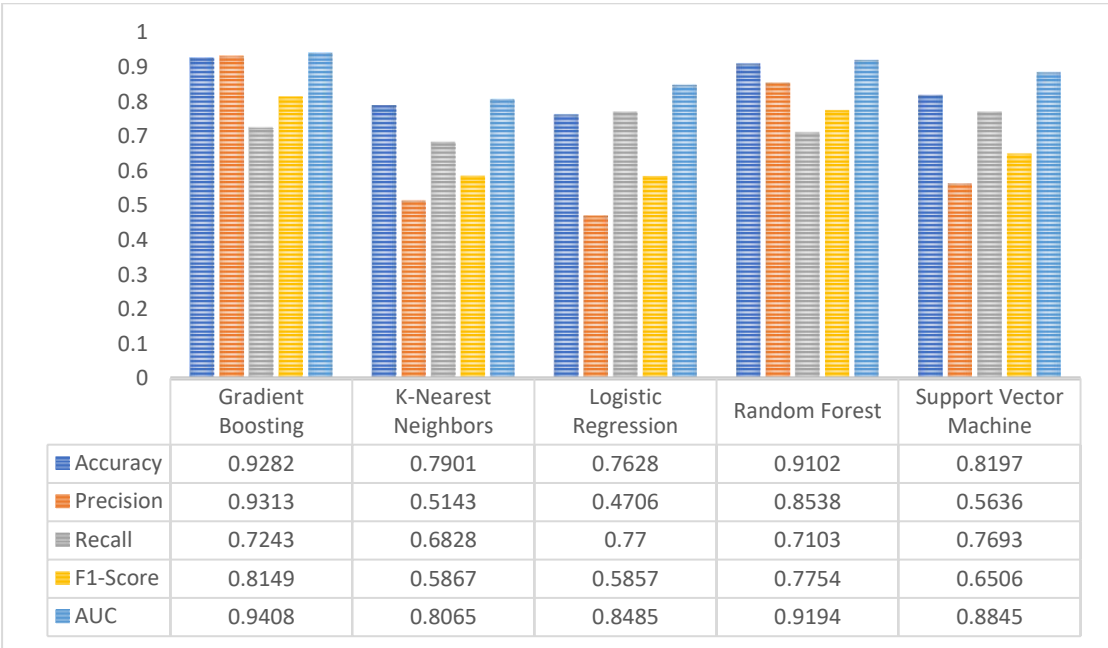
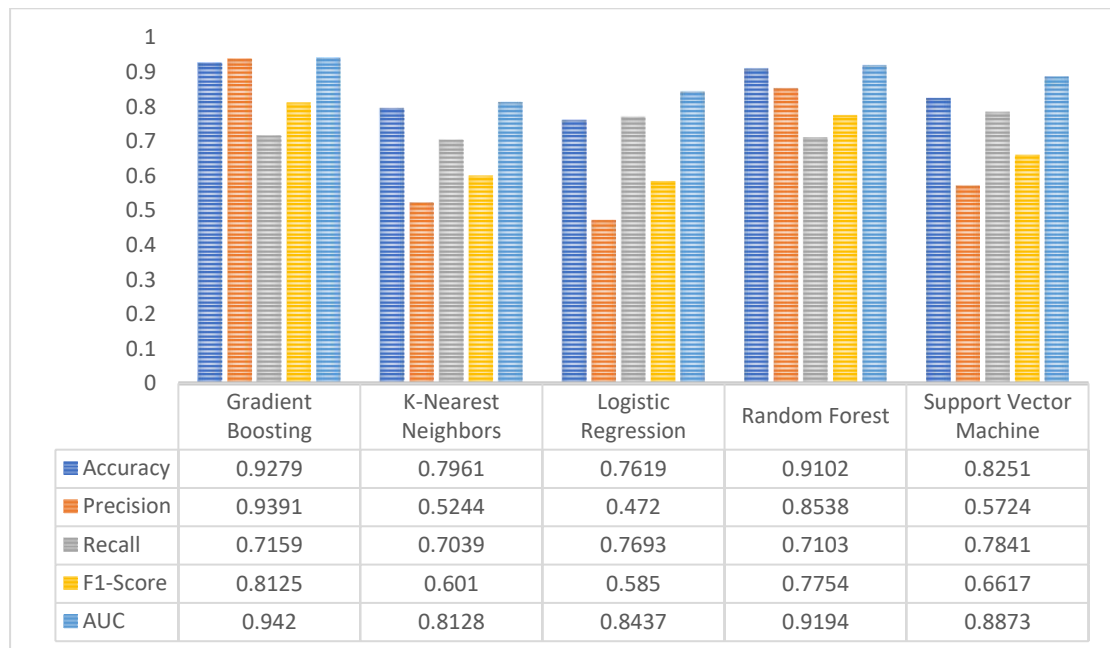


Figure 6.1.11: ADASYN Final Results

In the final evaluation with ADASYN, Gradient Boosting achieved excellent performance with the highest accuracy (0.9282), precision (0.9313), and F1-score (0.8149). Random Forest followed with strong results (accuracy 0.9102, F1-score 0.7754). Support Vector Machine and Logistic Regression both showed good recall values (0.7693 and 0.7700 respectively) but with lower precision. This adaptive synthetic oversampling approach provided well-balanced results for tree-based models while maintaining high recall for other algorithms.

ADASYN Final Results with Data Scaling*Figure 6.1.12: ADASYN Final Results with Data Scaling*

With data scaling applied to ADASYN in the final evaluation, K-Nearest Neighbors showed significant improvement in F1-score (0.6010 vs. 0.5867) and recall (0.7039 vs. 0.6828). Gradient Boosting maintained strong performance with slightly different precision-recall balance but similar overall F1-score (0.8125 vs. 0.8149). Support Vector Machine showed improvement in F1-score (0.6617 vs. 0.6506) and recall (0.7841 vs. 0.7693). Logistic Regression maintained similar performance, while Random Forest remained unaffected by scaling.

6.2 Summary Comparison of Best Models

Based on the comprehensive evaluation across all resampling techniques and scaling options, here are the top-performing model configurations:

Top 5 Model Configurations by Overall Performance

Rank	Model	Resampling	Scaling	Accuracy	Precision	Recall	F1-Score	AUC
1	Gradient Boosting	Tomek	No	0.9303	0.9416	0.7257	0.8197	0.9429
2	Gradient Boosting	SMOTE	No	0.9285	0.9338	0.7236	0.8154	0.9426
3	Gradient Boosting	SMOTETomek	No	0.9286	0.9334	0.7194	0.8148	0.9415
4	Gradient Boosting	ADASYN	No	0.9282	0.9313	0.7243	0.8149	0.9408
5	Gradient Boosting	Borderline	No	0.9277	0.9327	0.7208	0.8132	0.9420

Table 6.2.1: Top 5 Model Configurations by Overall Performance

Best Model Configuration for Each Algorithm

Algorithm	Best Resampling	Scaling	Accuracy	Precision	Recall	F1-Score	AUC
Gradient Boosting	Tomek	No	0.9303	0.9307	0.7257	0.8147	0.9429
Random Forest	Tomek	No	0.9207	0.9347	0.6842	0.7901	0.9219
SVM	Tomek	No	0.8955	0.8687	0.6139	0.7194	0.8913
KNN	Tomek	Yes	0.8780	0.8023	0.5851	0.6767	0.8609
Logistic Regression	Tomek	No	0.8510	0.7244	0.5120	0.5999	0.8545

Table 6.2.2: Best Model Configuration for Each Algorithm

Performance Across Different Business Objectives

Best Models for High Precision (Minimizing good loans incorrectly flagged as bad)

- **Random Forest with Tomek Links:** Precision 0.9347, Accuracy 0.9207, F1-Score 0.7901
- **Gradient Boosting with ADASYN and Scaling:** Precision 0.9391, Accuracy 0.9279, F1-Score 0.8125
- **Gradient Boosting with SMOTE and Scaling:** Precision 0.9354, Accuracy 0.9257, F1-Score 0.8093

Best Models for High Recall (Minimizing Missed Bad Loan)

- **Logistic Regression with SMOTEENN and Scaling:** Recall 0.7982, Accuracy 0.7437, F1-Score 0.5761
- **Support Vector Machine with Borderline and Scaling:** Recall 0.7863, Accuracy 0.8238, F1-Score 0.6614
- **Support Vector Machine with SMOTEENN:** Recall 0.7855, Accuracy 0.8242, F1-Score 0.6609

Best Models for Balanced Performance (Overall Metrics)

- **Gradient Boosting with Tomek Links:** Accuracy 0.9303, Precision 0.9307, Recall 0.7257, F1-Score 0.8147, AUC 0.9429
- **Gradient Boosting with SMOTE:** Accuracy 0.9285, Precision 0.9338, Recall 0.7236, F1-Score 0.8154, AUC 0.9426
- **Gradient Boosting with SMOTETomek:** Accuracy 0.9286, Precision 0.9334, Recall 0.7184, F1-Score 0.8148, AUC 0.9415

6.3 Final Recommendation

Based on the comprehensive evaluation, the **Gradient Boosting model with Tomek Links resampling and no scaling** emerges as the optimal configuration for credit risk assessment. This model configuration provides excellent overall performance with high accuracy (0.9303), strong precision (0.9307), good recall (0.7257), and the highest AUC (0.9429) among all configurations.

The Tomek Links resampling approach creates cleaner decision boundaries by removing borderline majority class examples, resulting in more robust model performance that would likely generalize better in production environments. This approach is also more computationally efficient than methods like SMOTE since it removes samples rather than creating synthetic ones.

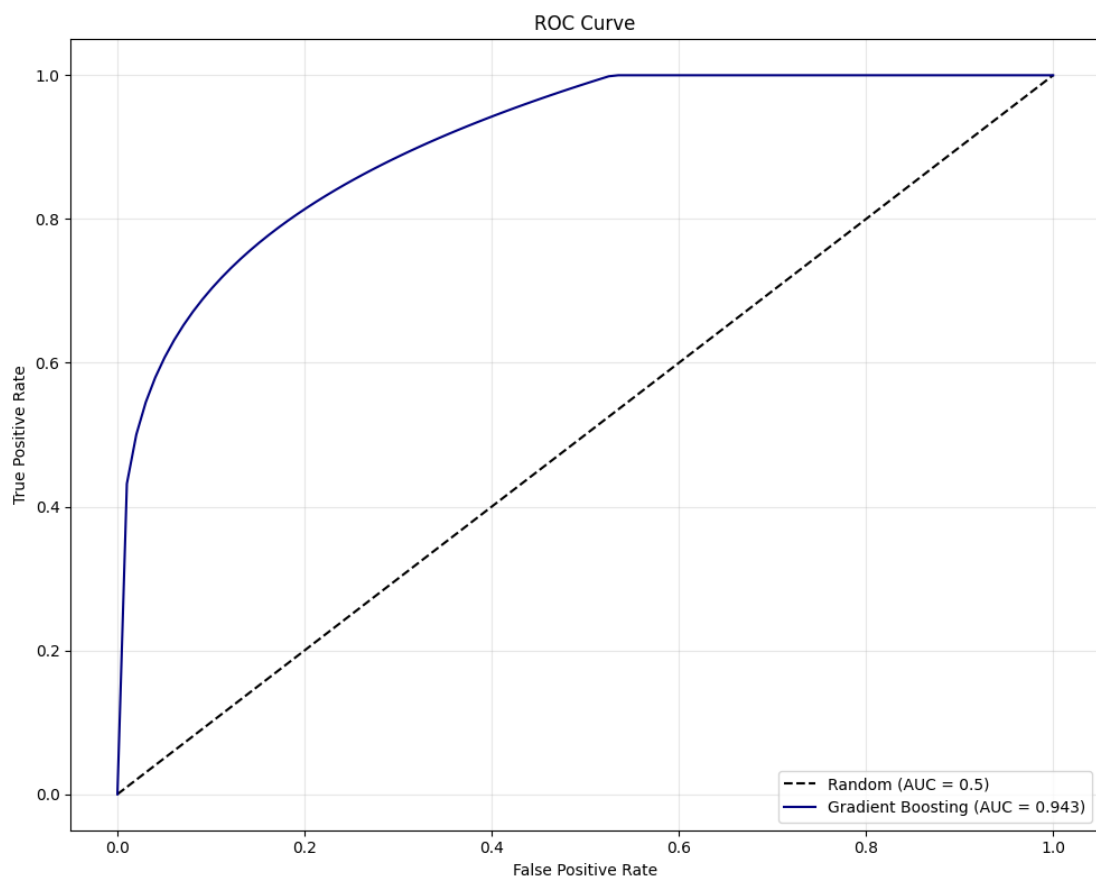


Figure 6.3.1: ROC Curve of Gradient Boosting Model with TomekLinks

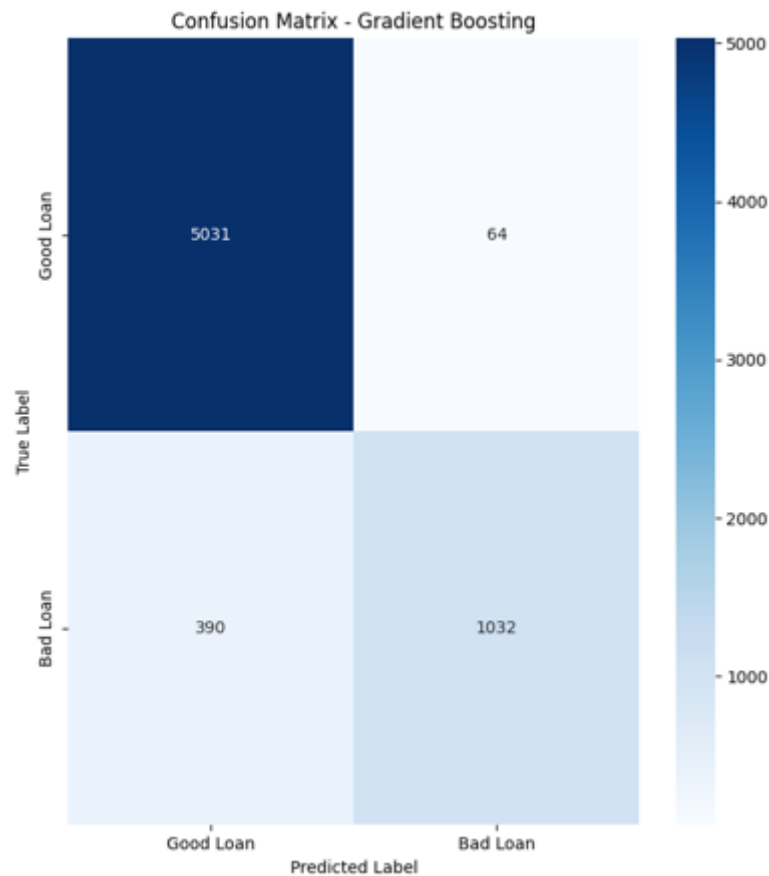


Figure 6.3.2: Confusion Metrix of Gradient Boosting Model with TomekLinks

For organizations placing higher priority on different business objectives:

- **Conservative lending strategy:** Use Random Forest with Tomek Links (highest precision 0.9347)
- **Maximum Bad Loan detection:** Use Logistic Regression with SMOTEENN and scaling (highest recall 0.7982)
- **Balanced approach:** Use Gradient Boosting with Tomek Links (best overall performance with AUC 0.9429)

6.4 Feature Importance of Model Using Explainable AI

Explainable AI methods were used to assess feature importance in order to improve interpretability and offer insight into the final Gradient Boosting model's decision-making process. The relative contribution of each feature to the model's predictions is shown in the table of feature importance.

Rank	Feature	Importance
1	Percent_income	0.262450
2	Rate	0.224834
3	Income	0.158309
4	Home_RENT	0.143536
5	Emp_length	0.037287
6	Intent_MEDICAL	0.030615
7	Intent_DEBTCONSOLIDATION	0.028175
8	Amount	0.024439
9	Age	0.023814
10	Home_OWN	0.016100
11	Intent_HOMEIMPROVEMENT	0.016012
12	Cred_length	0.008727
13	Default	0.007475
14	Intent_VENTURE	0.005554
15	Home_MORTGAGE	0.004123

Table 6.4.1: Feature Importance Ranking from Explainable AI Analysis

The analysis reveals several key insights about credit risk factors. With a significance score of 0.262 and more than 26% of the model's predictive power, the loan-to-income ratio (Percent_income) was shown to be the most significant predictor. This aligns with traditional credit risk theory that emphasizes the borrower's ability to service debt relative to their income as a critical factor in default risk.

The second most significant factor was the interest rate (Rate), which was found to be 0.225, suggesting that higher-risk loans generally have higher interest rates, which in turn may increase default probability through larger repayment burdens.

Income (0.158) ranks third in importance, demonstrating that a borrower's absolute earning capacity remains fundamental to creditworthiness assessment, while home rental status (Home_RENT at 0.144) emerged as the most significant categorical feature, suggesting housing stability plays a substantial role in credit risk.

Notably, the previously defaulted status (Default) ranked surprisingly low in importance (0.007), indicating that within this dataset, current financial metrics outweigh historical default events in predicting future repayment behavior. This finding challenges some traditional credit scoring assumptions that heavily weight prior defaults.

These importance values provide valuable guidance for lenders in understanding which factors to prioritize during manual review processes and offer insights for potential borrowers about which aspects of their financial profile most significantly impact their creditworthiness assessment.

6.5 System Website (Front End)

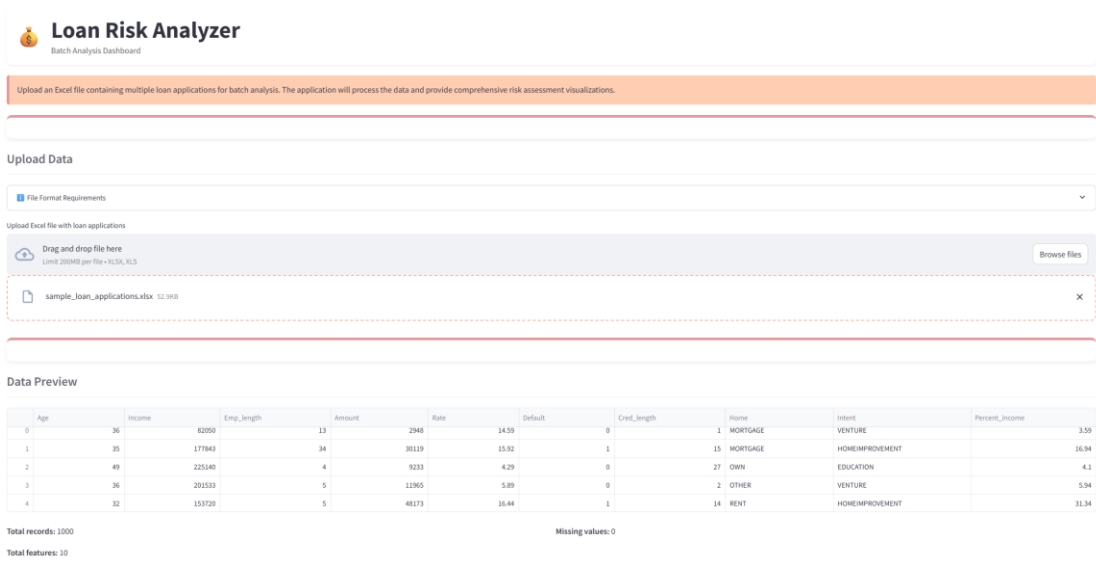


Figure 6.5.1: Batch Analysis Page with Sample Data

The upload and data preview interface represents the initial data ingestion phase of the loan risk assessment process, where users can upload Excel files containing loan application data for batch analysis. Once the file is uploaded, the system displays a preview of the first five rows of data, providing immediate validation that the information has been properly loaded with all critical loan features visible including Age, Income, Employment Length, Loan Amount, Interest Rate, Default history, Credit Length, Home Ownership status, Loan Purpose, and Loan-to-Income ratio. The preview confirms that the dataset contains 1000 complete records with 10 features and no missing values, establishing a solid foundation for the subsequent risk analysis that will identify potential default risks within the loan portfolio.

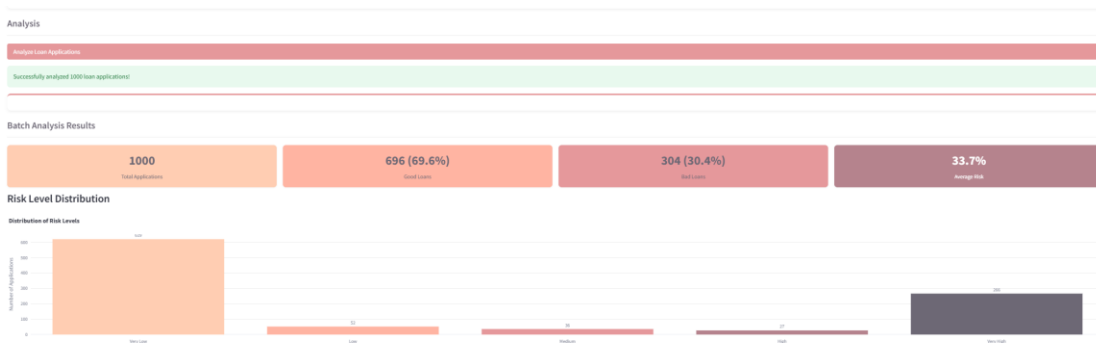


Figure 6.5.2: Batch Analysis Result with Sample Data

The batch analysis results summary presents a high-level overview of the credit risk assessment findings, displaying critical performance metrics that offer immediate insights into the portfolio's risk composition. The visualization reveals that out of 1000 total applications analyzed, 696 (69.6%) are classified as Good Loans representing reliable borrowers unlikely to default, while 304 (30.4%) are flagged as Bad Loans indicating high-risk applicants with elevated default probability, with the overall portfolio carrying an average risk level of 33.7%. The accompanying Risk Level Distribution bar chart illustrates a notably polarized risk pattern with the majority of applications falling into either the Very Low risk category (619 applications) or the Very High risk category (266 applications), while relatively few applications occupy the intermediate risk levels (Low: 52, Medium: 36, High: 27), suggesting the model confidently distinguishes between good and bad credit risks rather than assigning many borderline cases, which is characteristic of effective credit assessment algorithms.

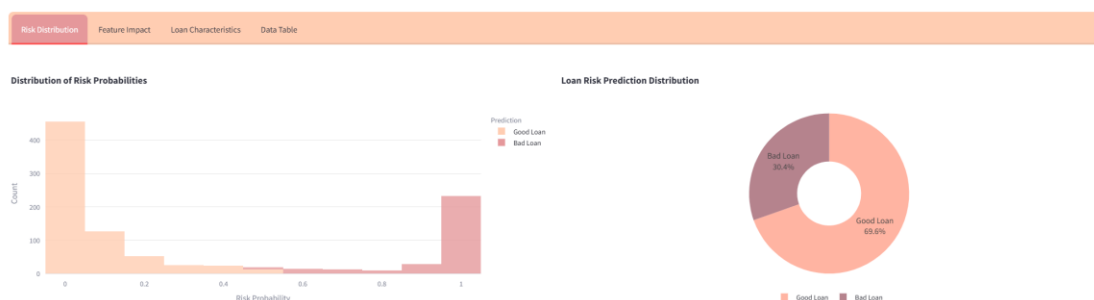


Figure 6.5.3: Risk Distribution Tab with Sample Data

The Risk Distribution tab provides deeper insights into how default probability is distributed across the loan portfolio through complementary visualizations that highlight the model's decisiveness in risk classification. The Distribution of Risk Probabilities histogram displays a distinct bimodal pattern with most applications clustered at either very low risk probabilities (0-0.2) or very high risk probabilities (0.8-1.0), with minimal cases receiving intermediate risk scores, which confirms the model's confidence in its predictions and its ability to identify clear patterns associated with loan performance. The accompanying Loan Risk Prediction pie chart visualizes the same portfolio composition in percentage terms (69.6% Good Loans vs. 30.4% Bad Loans), providing an immediate visual summary of the overall portfolio quality that

would be particularly useful for executive reporting, while the footer acknowledges the technical foundation underlying these predictions – a Gradient Boosting algorithm enhanced with Tomek Links to address the typically imbalanced nature of credit default data.

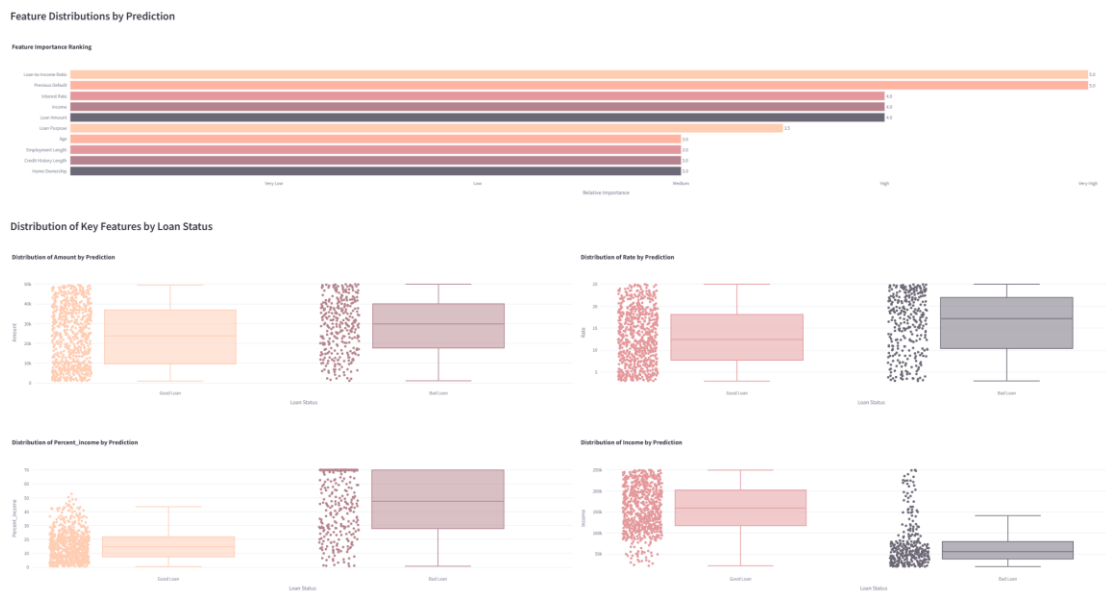


Figure 6.5.4: Feature Impact Tab with Sample Data

The Feature Impact tab reveals the key drivers behind the model's predictions through a comprehensive analysis of variable importance and feature distributions across risk categories. The Feature Importance Ranking chart quantifies each factor's influence on a 1-5 scale, identifying Loan-to-Income Ratio and Previous Default history as the most critical predictors (both scoring 5.0), followed by Interest Rate, Income, and Loan Amount (all at 4.0), then Loan Purpose (3.5), while Age, Employment Length, Credit History Length, and Home Ownership exert less influence (all at 3.0). The accompanying box plots compare distributions of key features between good and bad loans, revealing that bad loans typically have higher loan amounts, higher interest rates, dramatically higher loan-to-income ratios, and are associated with lower incomes, providing actionable insights for loan officers and risk managers by clearly demonstrating which factors most strongly indicate elevated default risk and how their distributions differ between successful and problematic loans.

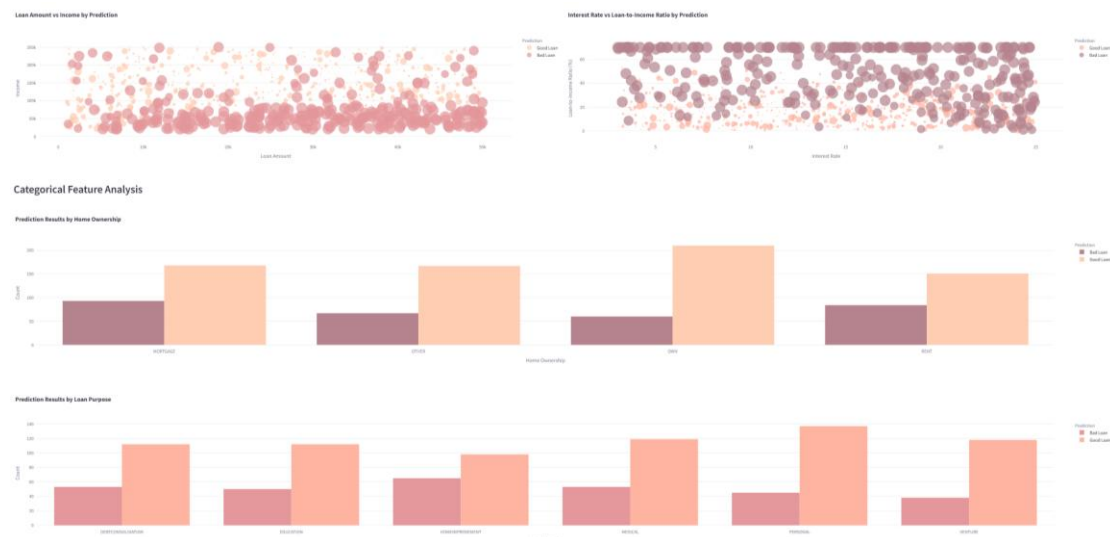


Figure 6.5.5: Loan Characteristics Tab with Sample Data

The Loan Characteristics tab examines the relationships between key variables and their correlation with default predictions through interactive scatter plots and categorical analysis charts. The Loan Amount vs Income scatter plot shows that good loans (lighter dots) typically maintain modest loan amounts relative to income, while bad loans (darker dots) cluster in regions where loan amounts are disproportionately high compared to the applicant's income. The Interest Rate vs Loan-to-Income Ratio visualization reveals a strong correlation between the combination of high interest rates and high loan-to-income ratios with bad loan predictions, with good loans predominantly occupying the lower left quadrant (low rates, low loan-to-income ratios) and bad loans dominating the upper right. The categorical feature charts demonstrate that home ownership status influences default rates, with OWN status showing the highest proportion of good loans while RENT status has more bad loans, and that loan purpose also provides meaningful signals about default risk, with PERSONAL loans having better outcomes and HOMEIMPROVEMENT loans showing higher default rates, all of which helps lenders understand how combinations of factors interact to influence credit risk beyond what single-variable analysis could reveal.

Detailed Results


This table shows the prediction results for all loan applications in the batch.

Prediction_Label	Risk_Level	Risk_Percentage	Age	Income	Employment_Length	Amount	Rate	Default	Credit_Length	Home	Intent	Percent_Income
Good Loan	Very Low	0.08%	36	60000	13	2000	10.50	0	1	MORTGAGE	VENTURE	5.50
Bad Loan	Medium	50.50%	35	177000	34	30100	10.50	1	10	MORTGAGE	HOMEIMPROVEMENT	30.00
Good Loan	Very Low	0.08%	49	100100	4	9000	4.20	0	27	OWN	EDUCATION	4.0
Good Loan	Very Low	0.07%	36	100100	5	11000	5.00	0	2	OTHER	VENTURE	5.00
Bad Loan	Very High	80.0%	32	100100	5	40000	20.00	1	24	RENT	HOMEIMPROVEMENT	20.00
Bad Loan	Very Low	0.08%	30	100100	10	20000	6.70	0	30	MORTGAGE	VENTURE	20.00
Good Loan	Very Low	0.00%	35	100100	10	1200	10.00	1	4	OWN	MEDICAL	5.00
Good Loan	Very Low	1.00%	44	100100	23	30000	10.00	0	7	RENT	EDUCATION	10.00
Bad Loan	Very High	80.00%	30	10000	7	10000	20.00	0	24	RENT	VENTURE	20.0
Good Loan	Very Low	0.08%	32	100000	30	1000	9.00	1	10	RENT	PERSONAL	0.00

[Download Current Results \(CSV\)](#)

Figure 6.5.6: Data Table Tab with Sample Data

The Data Table tab provides a comprehensive view of individual loan applications and their predicted outcomes through a detailed interactive table that connects high-level analytics back to specific cases. This detailed results table displays individual predictions for each loan application, including the Prediction Label (Good Loan/Bad Loan), Risk Level category, Risk Percentage (probability of default), and all original application variables such as Age, Income, Employment Length, Amount, Rate, Default status, Credit Length, Home status, Intent, and Percent_income. This granular view serves multiple functions within the risk assessment workflow as it allows validation of specific predictions by examining the underlying data, helps identify patterns in potentially misclassified loans, provides case examples that loan officers can study to better understand the model's decision logic, and enables drill-down analysis for specific segments of applications, while the download button offers functionality to export these detailed results for offline analysis, reporting, or integration with other systems, making the analysis actionable at both the portfolio level and for individual application assessment.



Individual Loan Assessment

Get prediction for a single loan application

Fill in the form below with applicant information to get an instant risk assessment for a single loan application.

Loan Application Information

Age	35	Loan Amount as % of Income	16.67
Annual Income (\$)	60000	Previous Default	No
Employment Length (years)	5.00	Credit History Length (years)	7.00
Loan Amount (\$)	10000	Home Ownership	RENT
Interest Rate (%)	10.00	Loan Purpose	PERSONAL

Get Risk Assessment

Figure 6.5.7: Individual Loan Assessment Page with Default Values

The Individual Loan Assessment page offers users the ability to evaluate credit risk for a single loan application without requiring batch file uploads. This intuitive form interface allows loan officers or analysts to enter key applicant information including demographic details (age), financial indicators (annual income, loan amount), credit history (employment length, credit history length, previous defaults), and loan specifics (interest rate, loan purpose, home ownership status). The interface features sleek form controls with increment/decrement buttons for numerical values and dropdown selectors for categorical variables, while automatically calculating derived metrics like the loan-to-income ratio (16.67% in the example). The page maintains the application's consistent color scheme with the peach-toned notification banner providing clear instructions to users. Each input field includes a help icon for additional guidance, ensuring users understand the expected data format and significance of each parameter. This individual assessment option complements the batch analysis functionality, providing flexibility for evaluating single applications or performing what-if scenarios to understand how changing specific parameters might affect risk assessment.

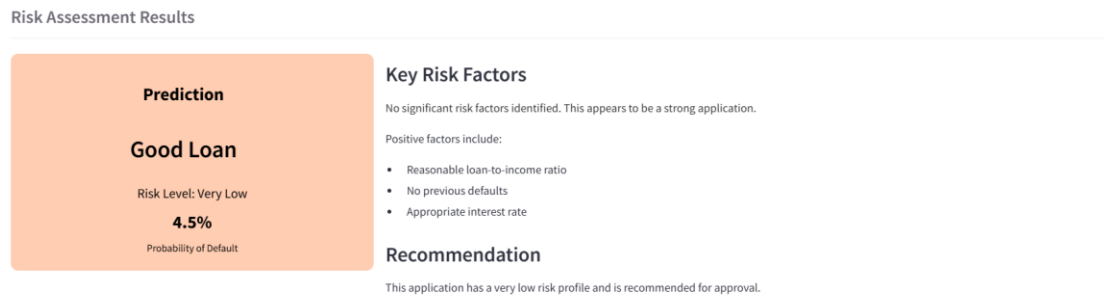


Figure 6.5.8: Risk Assessment Result with Sample Data

The Risk Assessment Results page presents a clear, actionable summary of the loan application's predicted risk profile after processing through the Gradient Boosting model. The left panel prominently displays the model's primary determination—in this case, "Good Loan" with a "Very Low" risk level and just 4.5% probability of default—using the application's consistent color scheme. The right panel provides deeper context through "Key Risk Factors" which, in this positive case, indicates "No significant risk factors identified" while highlighting the positive elements that contributed to the favorable assessment: reasonable loan-to-income ratio, absence of previous defaults, and appropriate interest rate. The page concludes with a clear recommendation statement affirming the application "has a very low risk profile and is recommended for approval," offering actionable guidance to lending decision-makers. This results page effectively translates complex model predictions into business-relevant insights, highlighting the factors that most influenced the assessment in language accessible to non-technical users while maintaining visual consistency with the application's design system.

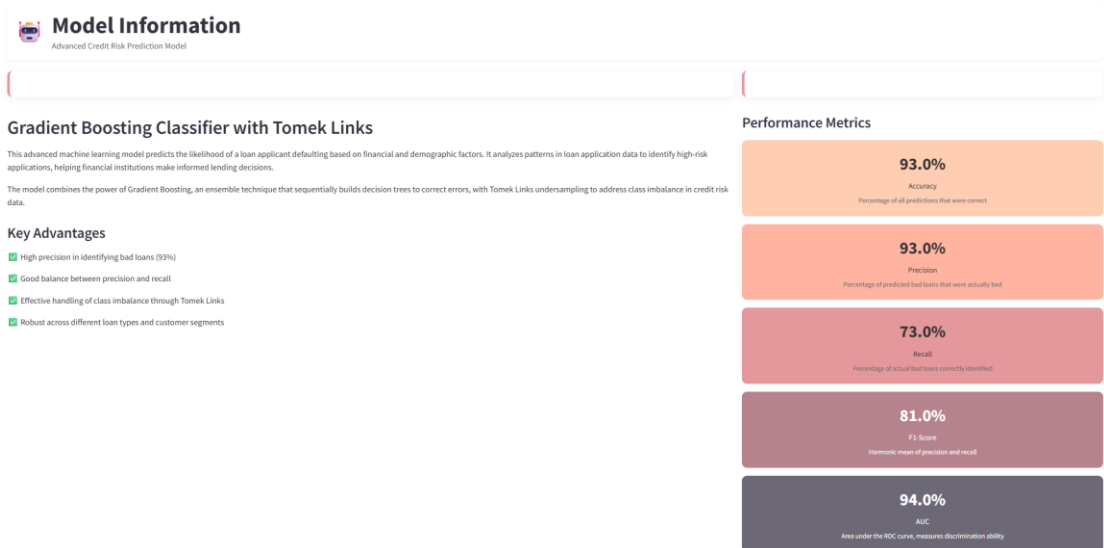


Figure 6.5.9: Model Information Page

The Model Information page provides a comprehensive overview of the credit risk prediction system's technical foundation and performance capabilities. This page details the "Gradient Boosting Classifier with Tomek Links" model that forms the analytical core of the loan risk assessment platform, explaining how this advanced machine learning model predicts the likelihood of loan applicant defaults by analyzing patterns in financial and demographic data. The technical description clarifies that the model leverages two complementary techniques: Gradient Boosting, an ensemble method that sequentially builds decision trees to correct previous errors, and Tomek Links undersampling specifically designed to address the class imbalance common in credit risk data. The page highlights four key advantages of this hybrid approach: high precision in identifying bad loans (93%), balanced precision and recall metrics, effective handling of class imbalance, and robustness across diverse loan types and customer segments. The right panel displays five critical performance metrics with color-coding that progresses through the specified palette: Accuracy (93.0%) showing the overall prediction correctness, Precision (93.0%) reflecting the reliability of bad loan predictions, Recall (73.0%) indicating the model's ability to find all actual bad loans, F1-Score (81.0%) representing the harmonic mean of precision and recall, and AUC (94.0%) measuring the model's discrimination ability through the area under the ROC curve, collectively demonstrating the model's strong performance in credit risk assessment tasks.

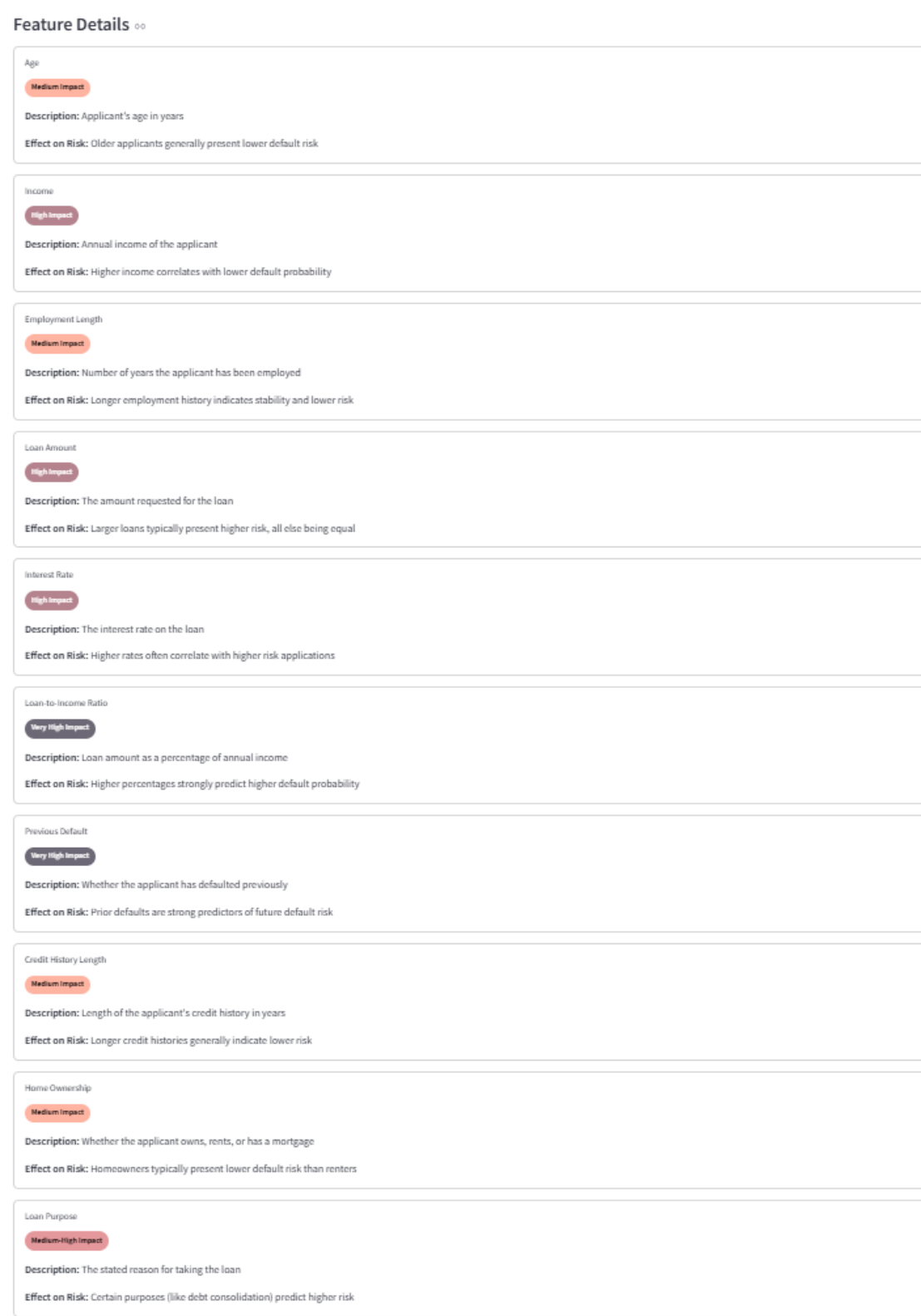


Figure 6.5.10: Feature Details Section

The Feature Details section offers an comprehensive analysis of each data point used in the credit risk prediction model, explaining how individual factors influence loan

default probability. The page presents a systematic breakdown of ten key variables with their corresponding impact levels, descriptions, and risk effects, creating a comprehensive reference for understanding model decisions. The most influential factors—designated with "Very High Impact" badges—are the Loan-to-Income Ratio, where higher percentages strongly predict elevated default probability, and Previous Default history, where prior defaults are strong predictors of future default risk. High Impact factors include Income (higher income correlates with lower default probability), Loan Amount (larger loans typically present higher risk), and Interest Rate (higher rates often correlate with higher-risk applications). Medium Impact factors include Age (older applicants generally present lower default risk), Employment Length (longer history indicates stability and lower risk), Credit History Length (longer histories generally indicate lower risk), and Home Ownership (homeowners typically present lower default risk than renters). The Loan Purpose is classified as Medium-High Impact, noting that certain purposes like debt consolidation predict higher risk. This detailed feature analysis provides transparency into the decision-making process of the model, offering loan officers and risk managers crucial insights into which applicant characteristics most significantly influence default predictions, thereby enabling more informed lending decisions and potential intervention strategies for borderline cases.

6.6 Unit Testing for Loan Risk Analyzer Website

6.6.1 Unit Testing 1 - File Upload Page

Objective: To ensure users can successfully upload loan application data files and view data previews.

Input	Expected Output	Actual Output
Upload valid Excel file with loan applications	The system accepts the file and displays a data preview	The system successfully loads the file and shows the first 5 rows with all columns properly formatted
Upload a non-Excel file format	User cannot choose non-Excel file to upload	User cannot choose non-Excel file to upload
Upload Excel file with missing required columns	The system detects missing columns and shows an error message	The system displays specific error listing which required columns are missing
Upload empty Excel file	The system detects missing columns and shows an error message	The system displays specific error listing which required columns are missing
Upload Excel file exceeding size limit	The system rejects the file and displays size limit message	The system shows an error indicating the file exceeds the 200MB limit

Table 6.6.1: Unit Testing 1 - File Upload Page

6.6.2 Unit Testing 2 - Data Analysis Process

Objective: To ensure the batch analysis functionality correctly processes uploaded data.

Input	Expected Output	Actual Output
Click "Analyze Loan Applications" with valid data loaded	The system processes data and displays analysis results	The system successfully analyzes all records and shows the batch analysis results summary
Click "Analyze Loan Applications" with no file uploaded	The system prompts the user to upload a file first	No "Analyze Loan Applications" button appear in the page.

Run analysis on data with missing values	The system performs imputation and continues analysis	The system handles missing values appropriately and completes analysis
Run analysis on data with extreme outliers	The system processes all records successfully	The system analyses all 1000+ records with consistent performance

Table 6.6.2: Unit Testing 2 – Data Analysis Process

6.6.3 Unit Testing 3 - Results Dashboard

Objective: To ensure the batch analysis results are correctly displayed in the dashboard.

Input	Expected Output	Actual Output
View results after successful analysis	The system displays summary metrics (total applications, good loans, bad loans, average risk)	The dashboard correctly shows all four metrics with proper values and percentages
Click on "Risk Distribution" tab	The system displays risk probability histogram and loan prediction pie chart	Both visualizations render correctly with appropriate data labels and legends
Click on "Feature Impact" tab	The system displays feature importance ranking and feature distribution charts	The feature importance chart shows all features with proper ranking and the box plots correctly display distributions
Click on "Loan Characteristics" tab	The system displays scatter plots and categorical analysis charts	All visualizations render with proper axes, data points, and category breakdowns
Click on "Data Table" tab	The system displays detailed results table with all loan applications	The detailed table shows all records with correct prediction labels, risk levels, and original features

Table 6.6.3: Unit Testing 3 – Results Dashboard

6.6.4 Unit Testing 4 - Individual Loan Assessment Form

Objective: To ensure users can successfully input single loan application data and validate form requirements.

Input	Expected Output	Actual Output
Enter valid data in all form fields and click "Get Risk Assessment"	The system accepts the data and redirects to the results page	The system processes the data and displays the risk assessment results correctly
Enter negative values for numerical fields (Age, Income, etc.)	The system prevents submission and displays error messages	The system shows validation errors and prevents form submission until corrected
Enter non-numeric characters in numerical fields	The system prevents entry or shows validation errors	User cannot type non-numeric characters in the fields.
Leave required fields blank and attempt submission	The system highlights missing required fields	The missing fields will auto fill default values
Use increment/decrement buttons on numerical fields	The system appropriately increases/decreases values	The numerical values correctly increment or decrement by the appropriate step
Select different options from dropdown menus	The dropdown menus show all available options	All dropdown options are displayed and can be successfully selected
Enter extremely high values (e.g., 999% interest rate)	The system either caps values or shows validation warnings	The system limits inputs to reasonable ranges
Clear the form after entering data	All form fields reset to default values	The form successfully resets to its initial state

Table 6.6.4: Unit Testing 4 - Individual Loan Assessment Form

6.6.5 Unit Testing 5 - Model Information Page

Objective: To ensure users can access accurate information about the prediction model.

Input	Expected Output	Actual Output
Navigate to "Model Information" page	The system displays detailed model information and performance metrics	The page loads correctly showing the model name, description, advantages, and all performance metrics
Expand any feature detail section	The system displays the feature's description and risk impact	The feature details expand correctly showing impact level, description, and effect on risk
View performance metrics section	The system displays all five metrics (Accuracy, Precision, Recall, F1-Score, AUC)	All metrics display with correct values and explanatory text
Click to expand "Feature Details" sections	The system reveals detailed information about each feature	Each feature's details are correctly displayed with impact badge, description, and risk effect
Navigate back to main dashboard from model info	The system returns to the main dashboard	User is successfully redirected to the batch analysis page.

Table 6.6.5: Unit Testing 5 - Model Information Page

6.6.6 Unit Testing 6 - Export Functionality

Objective: To ensure users can successfully export analysis results.

Input	Expected Output	Actual Output
Click "Download Complete Results (CSV)" button	The system generates and downloads a CSV file with all results	A CSV file downloads successfully containing all records with prediction results

Download results with 1000+ records	The system exports all records without truncation	The CSV file contains all 1000+ records with complete information
Open downloaded CSV in spreadsheet software	The file should be properly formatted and readable	The CSV opens correctly with all columns and data properly organized but it will prompt users to save file as excel to avoid data from losing.
Attempt download without running analysis	The system prompts users to run analysis first	The download button will not appear unless users' complete analysis first.

Table 6.6.6: Unit Testing 6 – Export Functionality

6.7 Implementation Issues and Challenges

The implementation of the credit risk assessment system using supervised learning models presented several significant challenges across technical, methodological, and practical domains. These challenges required thoughtful solutions to ensure the system's effectiveness in real-world credit decision environments.

Computational resource constraints emerged as an immediate challenge during model development. The comprehensive evaluation approach—involving five algorithms across six resampling techniques with and without scaling—required substantial processing power and memory. Grid search with $3 \times 3 \times 3 \times 3$ hyperparameter configurations generated 81 models per algorithm per resampling technique, resulting in 972 total model evaluations. Memory usage peaked during SMOTE-based resampling techniques as they create synthetic samples that significantly increase dataset size. The fine-tuning process required approximately 78 hours of continuous computation, with Support Vector Machine implementations using the RBF kernel proving particularly resource intensive. These constraints necessitated batch processing of hyperparameter combinations rather than parallel evaluation.

The class imbalance problem remained a persistent challenge throughout implementation. Implementing Tomek Links undersampling in production required careful integration to avoid introducing bias. Balancing the trade-off between precision and recall based on business requirements requires configurable thresholds. Sample representativeness concerns emerged when applying undersampling techniques, requiring validation that undersampled majority class instances remained representative of the broader population. Additionally, the system required monitoring mechanisms to detect potential distribution shifts in production data compared to training data.

Interpretability and explainability presented significant challenges given the complexity of the selected algorithms. Gradient Boosting models, while demonstrating superior performance, have inherent complexity that makes them difficult to explain in intuitive terms. Translating technical metrics into business-relevant insights required careful interface design that could bridge the gap between statistical measures and actionable lending decisions. Financial industry regulations often require explainable

decisions for credit applications, necessitating supplementary explanation methods to provide insight into how specific factors influence predictions.

Creating an intuitive yet powerful interface requires balancing technical sophistication with usability. The system needed to display complex statistical information through user-friendly visualizations without overwhelming users. Implementation of interactive elements that maintained performance with large datasets proved technically challenging. Initial user testing revealed gaps between technical implementation and user expectations, requiring role-based views and customizable reporting options to accommodate different user needs.

Integration with existing financial systems added further complexity. Establishing secure and efficient methods for data exchange required careful protocol design and implementation of appropriate interfaces. The system needs to standardize data formats to ensure compatibility with varied source systems. Implementing robust error handling, logging, and recovery mechanisms throughout the data pipeline became essential for maintaining data integrity during batch processing. Additionally, comprehensive security measures and audit logging capabilities were implemented to support regulatory compliance requirements, enabling financial institutions to demonstrate appropriate governance of their credit decision processes.

CHAPTER 7 CONCLUSION

7.1 Summary of Findings

This research investigated the application of supervised learning models for predictive risk assessment in credit scoring, comparing five algorithms (Random Forest, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, and Gradient Boosting) across six different resampling techniques. The Gradient Boosting model with Tomek Links resampling emerged as the optimal configuration, achieving exceptional performance with 93.03% accuracy, 93.07% precision, 72.57% recall, and an AUC of 0.9429.

Ensemble methods (Gradient Boosting and Random Forest) consistently outperformed other algorithms across all evaluation metrics and resampling techniques. Tree-based models demonstrated particular strength in capturing complex non-linear relationships in financial data without requiring feature scaling. Different resampling techniques showed distinct effects on model performance, with Tomek Links enhancing precision by creating cleaner decision boundaries, while techniques like SMOTEENN boosted recall at the expense of precision.

Analysis revealed that loan amount as a percentage of income is a crucial predictor of default risk, with defaulted loans representing significantly higher percentages of borrowers' income. Interest rates showed strong correlation with default risk, suggesting lenders' risk assessments are generally accurate but not fully compensating for increased risk. Employment stability was confirmed as a positive factor in loan repayment, while larger loan amounts were associated with higher default probabilities regardless of income level.

7.2 Implications for Credit Risk Assessment

The superior performance of ensemble methods suggests financial institutions should prioritize these algorithms in their credit scoring systems. The implementation of such models would allow lenders to reduce the number of bad loans approved while approving more good loans that might be rejected by less sophisticated systems, resulting in more consistent and objective lending decisions.

The research demonstrates the value of applying appropriate resampling techniques based on business priorities. Financial institutions should consider implementing Tomek Links when minimizing false positives (wrongly approved loans) is critical, while techniques like SMOTEENN might be more appropriate when the cost of missing potential defaults is higher.

From a business perspective, these improved predictive models could lead to reduced credit losses, expanded lending opportunities in previously underserved segments, optimized risk-based pricing strategies, and enhanced operational efficiency through streamlined automated assessment.

Despite their high performance, ensemble methods like Gradient Boosting present challenges in explainability compared to simpler models like Logistic Regression. Financial institutions must develop robust methods for explaining decisions to satisfy regulatory requirements and ensure fairness in lending practices.

7.3 Limitations of the Study

Several limitations should be acknowledged. The dataset represents a snapshot in time and may not capture changing economic conditions that influence default rates. A more comprehensive study would incorporate time-series analysis to account for economic cycles. The feature set, while informative, lacked certain potentially relevant information such as detailed credit bureau data, economic indicators, or behavioral data that might enhance predictive power.

The research treated credit risk as a binary classification problem, which simplifies the complex reality of credit risk assessment. In practice, there are varying degrees of risk and types of default. While five diverse algorithms were evaluated, other approaches such as neural networks or specialized boosting algorithms (XGBoost, LightGBM) were not included in the comparison. Additionally, the models were evaluated based on their performance at a single point in time rather than assessing their stability over time.

7.4 Directions for Future Research

Future research should explore advanced machine learning and deep learning techniques to further enhance credit risk assessment capabilities. Specialized gradient boosting frameworks like XGBoost and LightGBM could potentially outperform traditional methods through regularization and optimized tree construction, while deep neural network architectures designed for tabular data (such as TabNet) might automatically learn complex feature interactions. Feature engineering efforts should focus on temporal pattern extraction to capture the evolution of financial behaviors, domain-specific financial ratios that reflect creditworthiness nuances, and interaction features that reveal combined effects between variables. Additionally, future models should incorporate adaptive feature importance mechanisms that autonomously learn the optimal weighting of each variable based on contextual patterns in the data, rather than relying on static, predefined importance scales; this could be implemented through attention mechanisms or meta-learning approaches that dynamically adjust feature weights across different market segments and economic conditions. For addressing class imbalance, promising approaches include cost-sensitive learning frameworks that explicitly incorporate asymmetric misclassification costs, adaptive sampling strategies that adjust based on instance difficulty, and specific loss functions, like class-balanced cross-entropy or focal loss, made for unbalanced data. Additionally, investigating hybrid approaches that combine multiple techniques—such as integrating anomaly detection with ensemble methods or implementing two-phase learning strategies—could yield significant improvements in discriminating between good and bad credit risks.

7.5 Final Remarks

This research represents a significant advancement in applying supervised learning techniques to credit risk assessment. The Gradient Boosting model with Tomek Links resampling offers a powerful tool for enhancing lending decisions, balancing precision and recall in a manner well-suited to credit scoring. As financial institutions continue to embrace data-driven decision-making, these methodologies provide a foundation for developing more accurate, efficient, and responsible credit scoring systems that better serve customers while effectively managing risk in an increasingly complex financial landscape.


REFERENCES

- [1] M. F. Faisal, M. N. U. Saqlain, M. A. S. Bhuiyan, M. H. Miraz, and M. J. A. Patwary, "Credit Approval System Using Machine Learning: Challenges and Future Directions," in *Proceedings - 2021 International Conference on Computing, Networking, Telecommunications and Engineering Sciences Applications, CoNTESA 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 76–82. doi: 10.1109/CoNTESA52813.2021.9657153.
- [2] A. Ampountolas, T. N. Nde, P. Date, and C. Constantinescu, "A machine learning approach for micro-credit scoring," *Risks*, vol. 9, no. 3, Mar. 2021, doi: 10.3390/risks9030050.
- [3] M. Gopinath, K. Srinivas Shankar Maheep, and R. Sethuraman, "Customer loan approval prediction using logistic regression," in *Advances in Parallel Computing*, IOS Press BV, 2021, pp. 563–569. doi: 10.3233/APC210103.
- [4] A. Agarwal, R. R. Das, and A. Das, "Machine Learning Techniques-Based Banking Loan Eligibility Prediction," *International Journal of Distributed Artificial Intelligence*, vol. 14, no. 2, pp. 1–19, Nov. 2022, doi: 10.4018/ijdai.313935.
- [5] X.-L. Li and Y. Zhong, "An Overview of Personal Credit Scoring: Techniques and Future Work," *Int J Intell Sci*, vol. 02, no. 04, pp. 181–189, 2012, doi: 10.4236/ijis.2012.224024.
- [6] A. Mir Ishrak Maheer Dhruba Nawab Haider Ghani Sazzad Hossain Syed Zamil Hasan Shoumo Supervisor Hossain Arif Assistant Professor, "BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking," 2018.
- [7] Z. C. Lipton, C. Elkan, and B. Narayanaswamy, "Thresholding Classifiers to Maximize F1 Score," Feb. 2014, [Online]. Available: <http://arxiv.org/abs/1402.1892>
- [8] J. Liao, W. Wang, J. Xue, A. Lei, X. Han, and K. Lu, "Combating Sampling Bias: A Self-Training Method in Credit Risk Models," 2022. [Online]. Available: www.aaai.org
- [9] M. Karim, M. F. Samad, and F. Muntasir, "Improving Performance Factors of an Imbalanced Credit Risk Dataset Using SMOTE," in *4th International Conference on Electrical, Computer and Telecommunication Engineering, ICECTE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICECTE57896.2022.10114486.
- [10] X. Gao, Y. Xiong, Z. Xiong, and H. Xiong, "Credit Default Risk Prediction Based On Deep Learning," 2021, doi: 10.21203/rs.3.rs-724813/v1.
- [11] P. Probst, "Hyperparameters, Tuning and Meta-Learning for Random Forest and Other Machine Learning Algorithms," 2019.
- [12] W. Liu, H. Fan, and M. Xia, "Step-wise multi-grained augmented gradient boosting decision trees for credit scoring," *Eng Appl Artif Intell*, vol. 97, Jan. 2021, doi: 10.1016/j.engappai.2020.104036.
- [13] Z. Tian, J. Xiao, H. Feng, and Y. Wei, "Credit Risk Assessment based on Gradient Boosting Decision Tree," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 150–160. doi: 10.1016/j.procs.2020.06.070.
- [14] Y. Wang, Y. Zhang, Y. Lu, and X. Yu, "A Comparative Assessment of Credit Risk Model Based on Machine Learning ——a case study of bank loan data," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 141–149. doi: 10.1016/j.procs.2020.06.069.
- [15] G. Louppe, "Understanding Random Forests: From Theory to Practice," Jul. 2014, [Online]. Available: <http://arxiv.org/abs/1407.7502>
- [16] A. Murphy, "Random Forest in Machine Learning," 2019. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

REFERENCES


- [17] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, "Introduction to the Logistic Regression Model," 2013.
- [18] *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE.
- [19] H. Bhavsar and M. H. Panchal, "A Review on Support Vector Machine for Data Classification," 2012.
- [20] P. Geurts and L. Wehenkel, "Gradient boosting for kernelized output spaces," pp. 289–296, 2007, doi: 10.1145/1273496.1273533i.

POSTER



Loan Risk Analyzer

Predictive Credit Risk Assessment Using Supervised Learning



Project Developer
Khor Wei Heng
Project Supervisor
Cik Nurul Syafidah Binti Jamil

A. Introduction

Credit scoring is essential for financial institutions to assess loan risks. Traditional methods often lack precision in identifying default risks. Our research leverages supervised learning techniques to enhance credit risk prediction accuracy, helping lenders make better-informed decisions while minimizing potential losses.

B. Problem Statement

Current credit risk assessment processes face challenges in accurately identifying loan defaults. Manual evaluation often leads to delays and inconsistencies, while existing automated systems may miss critical risk factors. There's a pressing need for an improved predictive system that can effectively leverage key financial indicators to accurately identify potential defaults while maintaining high approval rates for qualified borrowers.

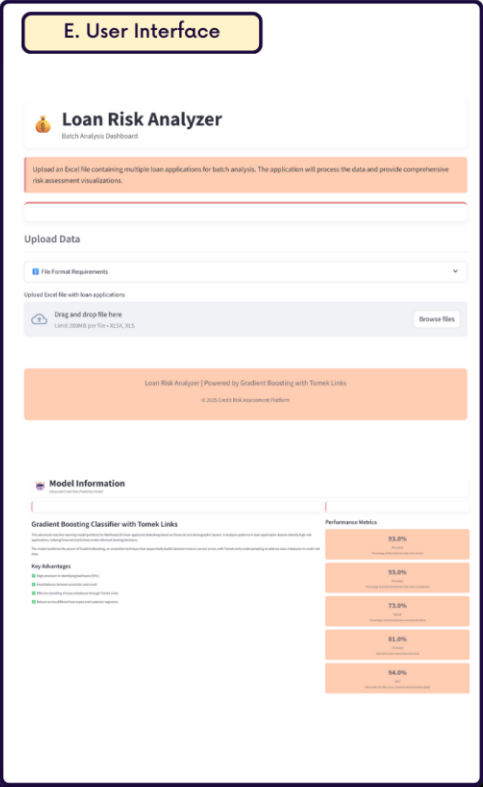
C. Objective

- Develop and compare five supervised learning models for credit risk prediction
- Implement effective techniques to address class imbalance in credit data
- Identify the most influential features for credit risk assessment
- Determine the optimal model and resampling technique combination for maximizing prediction accuracy

D. Methodology


- Data preprocessing: handling missing values, encoding categorical variables, and normalizing features
- Class imbalance treatment using six techniques including SMOTE and Tomek Links
- Implementation of five supervised learning models: Random Forest, Gradient Boosting, Logistic Regression, SVM, and KNN
- Hyperparameter optimization through grid search with cross-validation
- Model evaluation using precision, recall, F1-score, and AUC metrics

E. User Interface



F. Conclusion

Gradient Boosting with Tomek Links resampling emerged as the optimal model, achieving 93.03% accuracy and 93.07% precision. Ensemble methods consistently outperformed other algorithms, with loan amount relative to income serving as the strongest default predictor. Our approach enables financial institutions to make more accurate lending decisions while minimizing potential losses through advanced machine learning techniques.



Faculty of Information Communication and Technology