Submitted in part fulfilment of the requirements for the degree of Master of Science in Business Analytics

**Credit Default Prediction Using Multimodal Data Integration**

**and Explainable AI**

by

AMRUTHA JEEVA SATHEESAN

URN: 6807428

Faculty of Arts and Social Sciences

University of Surrey

September 2024

Word count: 13760

© AmruthaJeevaSatheesan

# Executive Summary

# This study addresses a key subject in financial risk management: predicting if a customer will default on their credit obligations. This topic has become increasingly important in the present financial landscape, particularly since the 2008 financial crisis, which highlighted serious inadequacies in standard credit risk assessment models. Traditional models frequently relied on limited criteria, such as income and credit history, that failed to reflect borrowers' full financial picture, resulting in incorrect predictions and considerable financial losses. As financial markets and consumer behaviour have become more complicated, there has been an increasing demand for more sophisticated models capable of integrating a greater range of data sources and providing more accurate and transparent predictions. This dissertation seeks to close these gaps by combining multimodal data with explainable AI techniques to increase the accuracy and interpretability of credit default prediction models.

The study utilized the "Default of Credit Card Clients" dataset from the UCI Machine Learning Repository, which comprises data on 30,000 credit card consumers in Taiwan from 2005. The dataset's primary goal is to predict if a client will default on their credit card payment in the coming month. The dataset contains 24 demographic, financial, and behavioural characteristics, such as age, gender, education, marital status, credit limit, and repayment history. These elements provide an extensive overview of the aspects that may influence a client's chance of defaulting, allowing the study to investigate how these various forms of data might be combined to improve the predictability.

The research methodology follows to the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, a structured approach that provides a thorough and systematic analysis of the data. The approach starts with understanding the business and the data, and the study uses exploratory data analysis (EDA) to identify relevant patterns and relationships within the dataset. The data is then pre-processed, which includes dealing with missing values, normalizing the data, and encoding categorical variables. This preparation is critical in ensuring that the data is acceptable for modelling. The study employs four machine learning models: Random Forest, XGBoost, Logistic Regression, and Decision Tree, each chosen for its ability to deal with different aspects of the prediction problem. Explainable AI approaches, such as Permutation Feature Importance, are also utilized to ensure that stakeholders can read and understand the model's decisions.

Each chapter of the dissertation adds to the comprehensive examination of the subject. Chapter 1 discusses the significance of credit default prediction, aim and objectives of this study. The second chapter covers the existing literature on credit risk modelling, with a focus on the role of AI and machine learning in financial sector. Chapter 3 describes the research approach, which includes the CRISP-DM framework and the specific data pre-processing and modelling methodologies used. Chapter 4 summarizes the findings from the exploratory data analysis (EDA), as well as the performance of the various models. Chapter 5 discusses a comparative study. Finally, Chapter 6 concludes the study by summarizing the important findings, limitations and identifying potential areas for future research.

Among the models evaluated, the Random Forest classifier performed the best, with high accuracy and a significant ROC-AUC score, indicating its ability to distinguish between defaulters and non-defaulters. The Random Forest model's ability to produce reliable predictions with fewer false positives makes it particularly beneficial in financial applications where risk minimization is critical. To improve prediction performance, the study also created a hybrid model that averaged the forecasts of the four distinct models. This hybrid approach took advantage of each model's unique strengths, resulting in better predictive accuracy and robustness. The hybrid model performed well on both training and test data, indicating its potential for real-world applications.

To conclude, this study makes a significant contribution to the field of credit risk modelling by proving the efficacy of combining multimodal data and applying explainable AI algorithms to improve credit default prediction. The study demonstrates that models such as Random Forest, when combined in a hybrid method, can provide excellent accuracy and interpretability, making them attractive for financial organizations. These findings have significant implications for enhancing credit risk assessments because they show the possibility of more accurate and transparent decision-making processes. Future study could build on these findings by refining the models and researching new data sources, with the goal of continuously improving credit default models' prediction powers.

# Declaration Of Originality

I hereby declare that this thesis has been composed by myself and has not been presented or accepted in any previous application for a degree. The work, of which this is a record, has been carried out by myself unless otherwise stated and where the work is mine, it reflects personal views and values. All quotations have been distinguished by quotation marks and all sources of information have been acknowledged by means of references including those of the Internet. **I agree that the University has the right to submit my work to the plagiarism detection sources for originality checks.**

Signature: 

Full Name: Amrutha Jeeva Satheesan

Date: 03-09-2024

Table of Contents

[Executive Summary 2](#_Toc176530005)

[Declaration Of Originality 4](#_Toc176530007)

[Acknowledgement 7](#_Toc176530008)

[1.Chapter 1: Introduction 8](#_Toc176530009)

[1.1 Introduction 8](#_Toc176530010)

[1.2 Aim 9](#_Toc176530011)

[1.3 Objectives 10](#_Toc176530012)

[2.Chapter 2: Literature Review 11](#_Toc176530013)

[2.1 Importance of Credit Default 11](#_Toc176530014)

[2.2 AI/ML in Finance Industry 15](#_Toc176530015)

[2.3 Machine Learning Algorithms 20](#_Toc176530016)

[2.3.1 Logistic Regression 20](#_Toc176530017)

[2.3.2. Decision Tree 21](#_Toc176530018)

[2.3.3 Ensemble Learning 22](#_Toc176530019)

[2.3.4 Random Forest Algorithm 22](#_Toc176530020)

[2.3.5 XGBoost Algorithm 23](#_Toc176530021)

[2.3.6 Permutation Feature Importance 24](#_Toc176530022)

[3. Chapter 3: Methodology 25](#_Toc176530023)

[3.1 Overview 25](#_Toc176530024)

[3.2 Sauders Research Onion 25](#_Toc176530025)

[3.3 CRISP-DM Process 28](#_Toc176530026)

[3.3.1. Business Understanding 29](#_Toc176530027)

[3.3.2. Data Understanding 29](#_Toc176530028)

[3.3.3 Data Preparation 34](#_Toc176530029)

[3.3.4 Modelling 35](#_Toc176530030)

[3.3.5 Evaluation 35](#_Toc176530031)

[3.3.6 Deployment 36](#_Toc176530032)

[Chapter 4: Analysis 37](#_Toc176530033)

[4.1 Overview 38](#_Toc176530034)

[4.2 Exploratory Data Analysis (EDA) 38](#_Toc176530035)

[4.3 Modelling 49](#_Toc176530036)

[4.3.1 Model Training 49](#_Toc176530037)

[4.3.2 Trained Model Evaluation 50](#_Toc176530038)

[4.3.3 Testdata 51](#_Toc176530040)

[4.3.4 Testdata Evaluation 51](#_Toc176530041)

[4.3.5 Hybrid Model development 52](#_Toc176530042)

[4.3.6 Permutation Feature Importance 53](#_Toc176530043)

[Chapter 5: Discussions 54](#_Toc176530044)

[5.1 Training Data Evaluation 54](#_Toc176530045)

[5.2 Test Data Evaluation 54](#_Toc176530046)

[5.3 Hybrid Model Evaluation – Train Data 55](#_Toc176530047)

[5.4 Hybrid Model Evaluation – Test Data 56](#_Toc176530048)

[Chapter 6: Conclusion 62](#_Toc176530049)

[6.1 Limitations 63](#_Toc176530050)

[6.2 Future Scope 64](#_Toc176530051)

[References 65](#_Toc176530052)

[Appendix I 72](#_Toc176530053)

[Appendix II 73](#_Toc176530054)

# List Of Tables

Table 3.1- Features and descriptions…………………………………………………………32

Table 3.2- Summary statistics of features…………………………………………………....33

Table 4.1- Train data results of Random Forest Classifier…………………………………...51

Table 4.2- Train data results of XGBoost Classifier…………………………………..……..51

Table 4.3- Train data results of Logistic Regression…………………………………………51

Table 4.4- Train data results of Decision Tree………………………………………………..52

Table 4.5- Test data results of Random Forest Classifier…………………………………….52

Table 4.6- Test data results of XGBoost Classifier…………………………………….…….53

Table 4.7- Test data results of Logistic Regression ………………………………………….53

Table 4.8- Test data results of Decision Tree ………………………………………….…….53

Table 4.9- Hybrid model train data results………………………………….………………..54

Table 4.10- Hybrid model test data results………………………………….………………..54

Table 5.1- Training Data Evaluation Results…………………………………...…………….55

Table 5.2- Test Data Evaluation Results…………………………………………..………….56

Table 5.3- Hybrid Model Results on Training Data………………………….………………56

Table 5.4- Hybrid Model Results on Test Data………………………………………………57

# List Of Figures

Fig 3.1. Saunder’s Research Onion Model……………………………………………….….28

Fig 3.2. CRISP-DM Process…………………………………………………………………30

Fig 3.3. Correlation between features and target variable……………………………………35

Fig 4.1. Histogram of various age groups of customers……………………………………...39

Fig 4.2. Bar graph of various education levels of customers ………………………………..40

Fig 4.3. Histogram of credit limit of customers ……………………………………………..41

Fig 4.4. Bar graph of marital status of customers …………………………………………...42

Fig 4.5. Histograms of Pay status of customers for 6 months………………………………..43

Fig 4.6. Histograms of Bill Amount status of customers for 6 months………………………44

Fig 4.7. Histograms of Pay Amount status of customers for 6 months………………………45

Fig 4.8. Bar graph of Gender of customers ………………………………………………….46

Fig 4.9. Bar graph of limit balance across different education levels ……………………….47

Fig 4.10. Bar graph of limit balance across different marital status ………………………...48

Fig 4.11. Pie chart of target class distribution ………………………………………………49

Fig 5.1 Confusion matrix for test data of hybrid model……………………………………...58

Fig 5.2 Confusion matrix for test data of random forest model……………………………...59

Fig 5.3 Confusion matrix for test data of logistic regression model…………………………60

Fig 5.4. Feature Importance in Random Forest Model………………………………………61

Fig 5.5. Feature Importance in XGBoost Model……………………………………………..62

# Acknowledgement

I would like to express my sincere gratitude to Dr. Vikas Grover, my dissertation supervisor, for his invaluable supervision and support during this process. His insightful inputs, patience, and consistent support have been vital when developing this dissertation, and I am sincerely thankful for his dedication to assisting me in navigating throughout my study.

I would also like to thank our Program Director, Dr. Colin Fu, for his stimulating workshops and continuous encouragement. His leadership and love for learning laid a solid basis for my academic success, keeping me motivated and focused on my objectives.

This dissertation would not have been possible without the collective contributions of both Dr. Grover and Dr. Fu, whose expertise and support have been invaluable in my academic journey.

Lastly, I want to acknowledge the love and support of my family and friends, whose encouragement and belief in me kept me going during this challenging time.

# 1.Chapter 1: Introduction

## 1.1 Introduction

An essential component of financial risk management is credit default prediction, which involves estimating the probability that a borrower won't be able to fulfil their credit obligations. Financial institutions need accurate prediction models in order to reduce potential losses and make well-informed lending decisions. The major financial and regulatory outcomes of defaults highlight the significance of credit default prediction. More reliable predictive tools are required because the 2008 financial crisis exposed the shortcomings in credit risk assessment models, resulting in widespread financial instability (Naili & Lahrichi, 2021).

Good credit default prediction models identify high-risk borrowers and modify lending practices to help financial institutions maintain financial stability. In the past, models depended on restricted characteristics like income, length of credit history, and credit utilization ratios. Although these characteristics have some predictive ability, they frequently fall short of capturing the full financial picture of borrowers, which could result in incorrect classification and monetary loss (Muniroh et al., 2024). The limitations of these traditional models have led to the need for more sophisticated approaches that make use of a wider range of data sources as financial markets and consumer behaviours have grown more complex.

Furthermore, regulatory compliance as well as the financial institutions' profitability depend on the accuracy of credit default forecasts. For equitable lending practices, regulatory agencies demand clear, justified credit decisions. Conventional models, which are frequently viewed as "black boxes," find it difficult to comply with these transparency standards, which presents problems for regulators and financial institutions alike (Lundberg & Lee, 2017). Therefore, improving these models' interpretability is crucial to developing stakeholder trust and adhering to legal requirements.

Credit risk modelling has been transformed by the introduction of machine learning and the availability of vast quantities of data. Large volumes of heterogeneous data can be processed and analysed by advanced algorithms, which can also reveal complex patterns that conventional models might miss. When it comes to capturing non-linear relationships and interactions among variables, machine learning techniques like gradient boosting, random forests, and support vector machines offer significant benefits (Tran et al., 2021). Even with these developments, interpretability of models remains a challenge that requires the incorporation of explainable AI methods to guarantee that intricate models stay transparent and understandable.

## 1.2 Aim

The aim of this study is: Is it possible to predict credit card default prediction by using machine learning algorithms and explainable techniques? By combining the advantages of both traditional and strictly black-box models—the clarity of interpretable models and the predictive power of black-box algorithms—this hybrid approach aims to overcome the imperfections of both approaches. Through the utilization of various data sources, such as billing, payment history, and demographic information, the study seeks to develop a more thorough comprehension of borrower behaviour and risk factors.

To clarify how various features contribute to the predictions, the model will use Explainable AI (XAI) approaches as Permutation Feature Importance. These methods enable stakeholders to better understand and have confidence with the decision -making process by offering insightful information about how particular variables affect model outputs (Wang et al., 2022).

## 1.3 Objectives

The 3 main objectives of this study are:

* Enhancing Credit Default Prediction Accuracy through Multimodal Data Integration:  
  This study aims to improve credit default prediction accuracy by including a greater range of data, such as historical, demographic, billing, and payment information. Using this richer data, the study hopes to discover critical factors impacting credit defaults and improve model performance.
* Improving Interpretability of Multiclass Credit Score Classification Models with explainable techniques: This study focuses on improving the interpretability of black-box models in multiclass credit score classification by utilizing Explainable AI (XAI) techniques such as Permutation Feature Importance. The study compares the accuracy and transparency of these methods to traditional models that are easier to interpret.
* Balancing Accuracy and Interpretability in Hybrid Models for Multiclass Classification: The goal is to investigate the trade-offs between accuracy and interpretability in hybrid models that combine black-box forecasts with understandable explanations. The research uses XAI methodologies to establish a balance that provides high accuracy while maintaining adequate transparency for financial risk management.

Chapter 2 explores the significance of credit default assessment and the expanding role of AI/ML in the finance/banking industry, with a focus on classifier algorithms such as Random Forest, XGBoost, Logistic Regression, and Decision Tree. The methodology in chapter 3 uses the CRISP-DM process and Saunders' research onion to guide the research strategy. Chapter 4 covers exploratory data analysis and modelling, laying the foundation for the upcoming analysis. Chapter 5 presents the results and discussions, which focus on the results and usefulness of models. Finally, Chapter 6 wraps up the study by summarizing and outlining future research options.

# 2.Chapter 2: Literature Review

## 2.1 Importance of Credit Default

Credit default essentially is the failure of a borrower to meet the legal obligations of a loan and is a critical aspect of financial risk management. This concept plays an extremely important role in the financial sector, serving as a foundation for various financial decisions, including loan approvals, pricing, and monitoring. Credit risk assessment estimates the probability of default (PD) over a specified period (usually one year) and is fundamental to this process.

Effective credit risk assessment has a direct impact on the quality of a bank's credit portfolio and financial health. This procedure involves reviewing a borrower's financial statements, industry health, and future estimates to forecast loan repayment volatility. Accurate credit risk assessment enables banks to distinguish between good and poor customers, saving large financial losses from defaults (Wójcicka-Wójtowicz, 2020).

Historically, Altman's (1968) introduction of Multivariate Discriminant Analysis (MDA) laid the foundation for formal credit default modelling. Despite its initial popularity, MDA's restrictive assumptions led to the development of logistic regression models, which offer greater flexibility and fewer biases. As financial markets evolved, so did the approaches to credit risk modelling, with contemporary methods increasingly utilizing machine learning techniques to enhance and improve robustness and accuracy of the predictions (Tran et al., 2021).

Digital financial services are a substantial source of Big Data, with globally payments income reaching $1.9 trillion in 2018 and 14 trillion transactions processed per day. The popularity of these services has prompted increased research into credit risk management in order to reduce financial risks and optimize earnings. According to the Basel Committee on Banking Supervision, banking risks are divided into three categories: credit, market, and operational risks, with credit risk accounting for 60% of bank threats, notably from peer-to-peer (P2P) lending platforms. These platforms enable direct connections between lenders and borrowers, bypassing traditional financial institutions, and have grown significantly, particularly in China. The fundamental problem with P2P lending is determining the creditworthiness of borrowers, who frequently lack credit history, resulting to higher default risks. Credit risk assessment, treated as a binary classification issue, is critical in evaluating loan applicants and controlling the hazards associated with unsecured instalment loans, which are increasingly made possible by P2P platforms (Moscato et al., 2021).

The importance of understanding and managing credit default cannot be overstated. The 2008 financial crisis underscored the consequences of inadequate credit risk management, highlighting the need for more sophisticated and reliable models (Naili and Lahrichi.,2021). Therefore and thereafter, researchers and practitioners have been continuously trying to improve credit risk assessment methodologies. Advances in computational power and availability of data have helped develop more complex models which are capable of capturing the nuances of borrower behaviour and market conditions. These models aim to provide a more accurate estimation of the probability of default, thus enabling financial institutions to manage their portfolios in a better manner and reduce the incidence of default (Tran et al., 2021).

Credit risk and default are key parts of the financial industry, serving as foundational mechanisms for making informed lending decisions, managing credit portfolios, and protecting institutions' financial health. Evaluating the risk of default or credit loss associated with individual borrowers or counterparties is critical to financial institutions' stability and profitability. This assessment often includes examining the borrower's credit history, financial situation, collateral, and macroeconomic conditions (Muñoz-Cancino etc al., 2023).

The credit risk importance is multidimensional. Maintaining a strong loan portfolio guarantees the stability and profitability of financial institutions, while limiting systemic risks from large-scale defaults benefits the larger financial system. Including expert knowledge and qualitative criteria, despite their impreciseness, improves decision-making by offering a thorough risk assessment. Simple Additive Weighting (SAW) and Ordered Fuzzy Numbers (OFNs) are useful methods for handling qualitative data and expert assessments (Wójcicka-Wójtowicz, 2020). ​

Over the years, significant advancements have been made in credit default modelling, learning from and addressing the shortcomings of earlier methods and incorporating sophisticated computational techniques. The accuracy and consistency of credit risk predictions has significantly improved due to the recent shift towards machine learning models, such as Artificial Neural Networks (ANNs) and ensemble methods (Dastile et al., 2020). These models, which require fewer assumptions and can handle more complex data structures, have become the fastest-growing area of research in credit risk modelling. For instance, the application of an entropy-based stacking model has been shown to provide more consistent and less biased performance across different data environments, as demonstrated using datasets from Lending Club (Tran et al., 2021).

Machine learning techniques, such as decision trees, random forests, gradient boosted trees and support vector machines have been widely adopted in credit risk modelling. These methods offer several advantages over traditional statistical models, such as the ability to capture non-linear relationships and interactions between variables. Moreover, they can be trained on large datasets, which is really beneficial since financial transactions generate huge amounts of data. Ensemble methods, which combine the predictions of multiple models, have been particularly effective in improving prediction accuracy and reducing model variance. Ensemble techniques can provide a more robust and reliable assessment of credit risk by aggregating the strengths of individual models (Tran et al., 2021).

There are multiple challenges that still remain even though the sophistication of credit risk models has been increasing. One of the primary issues is the need for a sound conceptual framework to guide the model selection and combination process. The lack of consistency in model benchmarking and the subjective nature of performance criteria selection have led to varying conclusions regarding model efficacy. To address this, researchers have proposed frameworks based on category theory, which provide a unified approach to understanding the relationships between different modelling techniques. Category theory, an abstract mathematical framework, allows for the systematic comparison of models by focusing on their structural components rather than their individual characteristics. This approach helps identify commonalities and differences between models, facilitating the development of more consistent and unbiased credit risk assessments (Tran et al., 2021).

In addition to theoretical advancements, practical considerations are also extremely important in credit risk modelling. The choice of base models, the method of combining model outputs, and the evaluation of model performance all play significant roles in finding the overall effectiveness of a credit risk model. Recent studies have highlighted the importance of using diverse data environments and rigorous performance testing to ensure that models are robust and generalizable. Techniques such as cross-validation, bootstrapping, and significance testing are essential for assessing model performance and reducing the risk of overfitting. Furthermore, the use of metrics like the Matthews Correlation Coefficient (MCC) and accuracy provides a more comprehensive evaluation of model performance, capturing both the consistency and bias of predictions (Tran et al., 2021).

Fintech has transformed credit risk assessment by utilizing big data and machine learning models. Unlike traditional methods, which rely largely on financial history and scorecard models, fintech approaches use different data sources, like transactional data, social media activity, and behavioural patterns, to improve the predictive power of credit risk models. These approaches can capture nonlinear interactions between variables and are more adaptable to shifting economic conditions (Huang et al., 2020).

According to studies, fintech models greatly improve the accuracy of loan default forecasts. For example, MYbank in China outperforms standard scorecard models when it comes to credit risk assessment using big data and machine intelligence. The fintech approach not only gives more accurate default forecasts, but it also improves financial inclusion by effectively assessing credit risk for borrowers with minimal or no traditional credit history​ (Huang et al., 2020).

The incorporation of fintech in credit risk assessment has major implications for policies. Policymakers should encourage fintech lending to increase financial inclusion while maintaining strong regulatory frameworks to control possible risks (Huang et al., 2020).

The importance of credit risk assessment comes from its potential to avoid financial losses due to defaults and to improve financial inclusion by giving accurate credit ratings. Effective credit risk management enables financial institutions to improve their loan portfolios by better distinguishing between low-risk and high-risk borrowers, preserving financial health, and improving economic stability through efficient and responsible credit distribution (Muñoz-Cancino etc al., 2023). The integration of machine learning techniques has significantly taken the field ahead, offering more accurate and reliable predictions of credit risk. However, the ongoing challenges of model consistency, bias, and conceptual clarity highlight the need for continued research and innovation. By utilizing theoretical frameworks like category theory and adopting rigorous evaluation practices, researchers and practitioners can enhance the robustness and reliability of credit risk models, ultimately contributing to more stable and resilient financial management techniques.

2.2 AI/ML in Finance Industry

The banking industry has transformed through AI and ML, enhancing data analysis, risk management, and consumer interaction. These technologies surpass traditional methods in credit scoring and risk assessment by incorporating complex data, such as transactional data and behavioral patterns, to more accurately predict creditworthiness (Pallathadka et al., 2023). By assisting financial institutions in more accurately identifying high-risk borrowers, these cutting-edge algorithms lower default rates and enhance the general stability of loan portfolios.

In the 1980s, AI gained traction in banking, with Expert Systems becoming economically viable. By the 1990s, fraud detection became a priority, exemplified by the FinCEN Artificial Intelligence System (FAIS) in 1993, which evaluated 200,000 transactions weekly, identifying 400 probable money laundering cases totalling $1 billion over two years. Though Expert Systems were eventually phased out, they paved the way for today’s AI in finance. Since 1987, banks have leveraged AI for fraud prevention, organizing activities, investment management, and property management. AI now enhances fraud detection by analysing user behaviour and improves auditing by continuously monitoring large datasets (Kunwar, 2019).

Traditional credit risk assessment methods, inclusing credit scoring and logistic regression, often struggle with the complexities of modern financial markets and the huge amount of data available. They rely heavily on historical data, which may not capture evolving borrower behaviours, and are prone to subjectivity and bias due to human judgment (Muñoz-Cancino etc al., 2023).

Machine learning (ML) techniques are increasingly utilized in banking for credit scoring and risk assessment, using algorithms like extreme gradient boosting (XGB) and neural networks to predict default risk. These models enhance decision-making by leveraging large datasets. Fairness-aware algorithms, including pre-processors, in-processors, and post-processors, are employed to minimize biases, promoting equitable lending. Studies show that reducing algorithmic discrimination is achievable while maintaining profitability, supporting anti-discrimination regulations in credit markets (Kozodoi et al., 2022).​

AI and machine learning have revolutionized banking, enhancing applications like credit risk assessment, fraud detection, and financial forecasting. By processing large datasets, machine learning models such as logistic regression, decision trees, and neural networks identify complex patterns and generate more accurate predictions than traditional statistical methods (Bello, 2023).

AI-driven fraud detection enhances transaction security by identifying subtle patterns and anomalies, surpassing traditional rule-based systems (Ahmed et al., 2022). AI is revolutionizing portfolio management by using automated trading systems and robo-advisors to optimize asset allocation and strategies with minimal human involvement. By analysing historical data and market trends, AI removes emotional bias, boosting efficiency and potentially improving returns (Pallathadka et al., 2023). AI's role in financial analysis is expanding, with robo-advisors set to increase due to their superior capabilities and cost-effectiveness. This shift is driving down commission rates on investments. The rise of "bionic advice," blending precise AI calculations with human intuition, highlights the importance of collaboration between humans and machines. Success in future financial decision-making will hinge on balancing AI's contributions with human intelligence, recognizing AI as an equal partner in navigating the evolving landscape of financial services (Tewari, 2023).

Wealthfront, a robo-advisor, leverages AI to create and manage personalized investment portfolios based on clients' goals and risk tolerance. It automatically rebalances portfolios and monitors outcomes, offering a tailored, efficient investment strategy that attracts many investors (Tewari, 2023). Unsupervised learning clusters similar borrowers and detects high credit risk, while deep learning enhances credit risk models' pattern recognition (Bello, 2023).

AI and ML are revolutionizing customer service in the financial sector. Financial institutions use AI-powered virtual assistants to handle transactions, answer queries, and provide personalized financial advice. By leveraging natural language processing (NLP) and machine learning, these systems enhance operational efficiency, customer experience, and loyalty through tailored financial services and products (Pallathadka et al., 2023). AI models are crucial in finance for predicting market trends, customer behaviour, and risks. In algorithmic trading, AI optimizes investment returns by executing trades based on real-time data analysis. Additionally, AI enhances regulatory compliance by automating the monitoring and reporting of financial transactions to detect suspicious activities (Ahmed et al., 2022). Machine learning is used in trade settlement to identify causes of failed trades, suggest reliable solutions, and predict potential risks. AI-driven financial apps provide personalized advice, track expenses, analyse spending patterns, and highlight savings opportunities, enhancing financial management (Mahalakshmi et al., 2022). Financial institutions are rapidly adopting AI-powered chatbots to enhance customer support and automate various financial activities, boosting efficiency and accessibility. These chatbots offer services such as financial advice, customer service, and account management. An example is Mastercard's "KAI" chatbot, which uses natural language processing and machine learning to provide personalized assistance and financial insights across platforms like SMS, WhatsApp, and Messenger, marking significant progress in delivering tailored financial services (Tewari, 2023).

Machine learning (ML) has gained significant attention in the banking industry, particularly for credit default prediction, due to its superior predictive capabilities compared to traditional statistical methods. Techniques such as support vector machines (SVMs), decision trees, and neural networks excel in processing complex, large-scale data to accurately predict credit risks and defaults (Pallathadka et al., 2023). Credit risk assessments become more accurate as a result of these algorithms' ability to learn from past data and spot patterns and correlations that human analysts would miss.

AI plays an important role in minimizing currency risk. Individuals and small enterprises (SMEs) can now deposit funds in fiat currencies, essentially moving volatility risk to financial intermediaries (FIs) as digital finance has grown (Paul, 2019). Many financial institutions have used bitcoin as a vehicle currency, with the US dollar accounting for 88% of deals (Mhlanga, 2020). By using bitcoin as a vehicle money and employing blockchain systems, both the sender and recipient are protected from the volatility of virtual currencies. This risk mitigation capability allows low-income earners to participate in the financial market, thanks to the strength of AI technology (Paul, 2019).

AI and ML are being used in insurance underwriting to analyse massive volumes of data and assess policy risk (Pattnaik et al., 2024). AI significantly improves insurance underwriting by analysing diverse data, including demographics and health records. Lemonade, an AI-driven Insurtech, uses advanced algorithms to assess claims and underwrite policies, increasing accuracy, reducing fraud, and making insurance more affordable (Tewari, 2023).

According to Hyundai Research Institute, a 'robo-adviser' combines AI, big data, and machine learning to automate asset management in the financial industry. These systems offer automated financial consulting, reducing the need for human advisors by merging financial theories with modern computational methods, streamlining asset management, and optimizing financial decision-making without human intervention (Go et al., 2020).

With great accuracy and resilience, machine learning models like random forests, artificial neural networks (ANNs), and support vector machines (SVMs) have been widely used to predict credit ratings and defaults (Li et al., 2020). In their discussion of the use of these models in credit rating prediction, Li et al. (2020) pointed out that random forests frequently perform better than other models because of their capacity to handle huge datasets and recognize intricate relationships between variables. The study also discovered that SVMs and ANNs work especially well when dealing with scenarios that include default and speculative grade predictions, with ANNs being especially good at identifying non-linear relationships in the data. The researchers underlined the value of combining several models to get the best outcomes, especially in situations involving stress prediction, where the precision of forecasts can have a big influence on decision-making and financial stability.

Using machine learning in credit risk assessment increases credit scoring accuracy, automates decision-making, and optimizes risk management. This leads to more informed lending decisions, less credit losses, and greater financial stability. However, concerns such as data privacy, model interpretability, and regulatory compliance must be addressed in order to ensure that these technologies are used responsibly (Bello, 2023). ZestFinance's Zest Automated Machine Learning (ZAML) system improves loan decision-making processes, lowering default rates by extensively assessing multiple credit risk characteristics and providing precise, dependable credit ratings (Tewari, 2023).

More advanced models for credit default prediction have been developed as a result of recent developments in machine learning techniques. Lahmiri and Bekiros' (2019) bibliometric investigation indicates that big data analytics and nonlinear classifiers have significantly increased the accuracy of bankruptcy forecasts.

The study results of Khandani et al. (2010), who showed that, machine learning significantly outperforms conventional methods in predicting credit card defaults, suggesting that advancements in data processing and algorithms will enhance credit risk management, reducing risks and improving financial stability.

Deep learning has revolutionized credit evaluation in banking by using techniques like neural networks and decision trees to analyse large datasets. By integrating diverse data sources, these models offer a comprehensive view of financial profiles, improving accuracy over traditional methods. Predictive analytics further enhances assessments by continuously learning and predicting future behaviour (Bammidi, 2023).

Deep learning, particularly the combination of Deep Belief Networks (DBNs) and Restricted Boltzmann Machines, has made major advances in financial applications. Following the 2007-2008 financial crisis, the importance of effective credit risk management became clear, prompting an investigation into various machine learning techniques. Traditional statistical methods such as logistic regression and shallow architectures such as support vector machines (SVMs) and multi-layer perceptrons (MLPs) have limitations when dealing with complicated financial data due to their fixed functions and assumptions.

Hinton and Salakhutdinov came up with DBNs in 2006, which solve these restrictions by capturing complicated data patterns using several hidden layers. Empirical studies show that DBNs outperform other models in terms of accuracy and robustness, making them ideal for credit scoring tasks. This breakthrough represents a substantial shift in the finance industry toward more dependable and complex models, which improves the predictive capacity and efficiency of credit risk evaluations (Luo et al., 2017).

AI/ML in finance faces several significant challenges. First, a shortage of specialized talent, including data scientists and machine learning experts, hampers AI implementation. Second, ethical concerns, such as bias, fairness, and data privacy, require financial institutions to ensure transparency and integrity in AI systems. Third, regulatory and compliance hurdles, compounded by strict data protection laws, demand substantial resources and ongoing monitoring. High costs, both in terms of technology and talent, necessitate careful ROI assessment. Additionally, the quality and availability of data are crucial, as AI relies on vast amounts of high-quality data for accurate results. Finally, integrating AI with existing systems, like CRM and ERP, is technically complex and resource-intensive (Moşteanu, 2023).

The banking industry has seen significant gains from AI and ML in portfolio management, fraud detection, credit scoring, and customer service. However, ethical concerns, data privacy, and algorithm biases necessitate evolving regulations to ensure responsible and fair AI use (Bammidi, 2023). Addressing these issues requires a strategy involving explainable AI, ethical adherence, and continuous R&D. As AI advances, its potential in banking is promising, but financial institutions must navigate challenges to fully harness these technologies' benefits.

2.3 Machine Learning Algorithms**:**

### 2.3.1 Logistic Regression

Logistic regression analyses the relationship between independent variables and a binary outcome, mapping predictions to probabilities between 0 and 1. Widely used in healthcare, it estimates disease likelihood. Its simplicity and interpretability make it crucial for transparent decision-making in fields like healthcare and banking (Black et al., 2023).

Logistic regression models the probability of an event using the logit function, which calculates the natural logarithm of the odds. It's particularly effective for small datasets, offering interpretability through feature engineering. In medical research, it's commonly used to predict a patient's risk of developing a condition based on various predictors (Ruchi et al., 2023).

Logistic regression offers competitive performance, ease of implementation, and lower computational cost, making it favoured in areas like credit scoring where model interpretability is crucial. While advanced models like gradient boosting may slightly improve accuracy, logistic regression remains a strong baseline, with ensemble methods enhancing performance while retaining transparency (Song et al., 2023).

Logistic regression is widely used for its simplicity and interpretability, allowing researchers to assess predictor impacts. However, it assumes a linear relationship between predictors and log-odds, which may not always reflect real-world data. Issues like multicollinearity and the need for sufficient samples can also affect its accuracy (Black et al., 2023).

### 2.3.2. Decision Tree

The decision tree algorithm is a widely used, interpretable machine learning technique that splits data into subsets based on attributes, optimizing metrics like Information Gain or Gini Index. Resembling a flowchart, it handles both numerical and categorical data, making it versatile across fields such as banking, healthcare, and education (Lin et al., 2023).

Decision Trees can be pruned to eliminate insignificant branches, which makes the model simpler and more broadly applicable. In order to keep the tree from getting unduly complicated, this pruning procedure involves setting parameters like minimum split size or maximum tree depth (Goyal et al., 2022). Decision trees are used in education to categorize students by performance, enabling personalized learning. Analysing student data helps educators identify key factors affecting achievement and adjust teaching methods accordingly (Lin et al., 2023). Similar to this, similar algorithms are used in finance for credit scoring and risk assessment, analysing an individual's financial history and behaviours to estimate the probability that they will fail on a loan (Goyal et al., 2022). Decision trees are a vital tool in data-driven decision-making processes due to their versatility and simple implementation.

Decision trees have a few benefits, but they also have some drawbacks. Their tendency to overfit, particularly with complicated datasets containing noise, is one of their biggest disadvantages (Goyal et al., 2022). Overfitting occurs when a model captures noise instead of patterns, leading to poor generalization. Pruning helps reduce overfitting but can cause instability, with small data changes altering tree structures.

Large datasets with many features can make decision trees less effective and computationally heavy. Ensemble methods like Random Forests enhance accuracy by combining multiple decision trees (Abdulazeez et al., 2021). Balancing interpretability and model complexity remains crucial. Despite challenges, decision trees are widely used in machine learning due to their simplicity, interpretability, and adaptability, often outweighing their drawbacks.

### 2.3.3 Ensemble Learning

Ensemble learning boosts prediction accuracy by combining multiple models, reducing errors. The three main methods are stacking, boosting, and bagging. Bagging, or Bootstrap Aggregating, trains models on different data subsets to reduce variance and prevent overfitting. Random Forests, a popular bagging approach, enhance accuracy through random feature selection.

Boosting trains models sequentially, focusing on correcting errors from previous iterations. Techniques like AdaBoost adjust instance weights to prioritize difficult cases. Variants like LogitBoost and AdaBoost.M1 handle multi-class problems. Stacking enhances generalization by combining base learners through a meta-learner, with methods like Grading and SCANN boosting efficiency (Rincy and Gupta, 2020).

Ensemble approaches often require significant time and computational resources due to the need for training multiple models, and their results can be complex to interpret. Despite these drawbacks, they are used for face recognition and anomaly detection, as they typically enhance accuracy and robustness compared to single models.

Research findings demonstrate the better performance of ensemble learning techniques such as Random Forests and AdaBoost compared to single-model approaches, highlighting the important influence of ensemble learning on modern data science and artificial intelligence practices (Mienye and Sun, 2022). This broad application emphasizes how crucial it is to carry out more research and development on ensemble approaches in order to increase their efficiency in resolving challenging real-world issues.

### 2.3.4 Random Forest Algorithm

The random forest algorithm uses ensemble learning to improve prediction accuracy by constructing multiple decision trees, each based on random subsets of data and features. For regression, predictions are averaged; for classification, the mode is used. This randomness reduces overfitting and variance, resulting in a more robust and reliable model than individual trees (Otoo et al., 2020; Shah et al., 2020; Sheykhmousa et al., 2020).

The random forest algorithm's adaptability and effectiveness in processing a wide range of datasets make it applicable in a variety of industries. It effectively classifies land cover in remote sensing by utilizing textural and spectral characteristics across industries. (Otoo et al., 2020). By categorizing gene expression patterns and identifying important genes, it supports gene expression analysis in bioinformatics (Shah et al., 2020). Additionally, it is employed in ecological modelling to evaluate environmental impacts and forecast species distribution (Sheykhmousa et al., 2020). It's ideal for large, high-dimensional datasets, handling noise and anomalies well, making it superior for complex real-world applications. (Shah et al., 2020).

Despite its advantages, the random forest algorithm's interpretability diminishes with more trees, complicating the understanding of feature importance and prediction logic, crucial for transparency in healthcare and finance (Otoo et al., 2020; Sheykhmousa et al., 2020). Moreover, random forests can manage big datasets effectively, but as data volume and tree count increase, they may become computationally demanding and need a lot of memory and processing power (Shah et al., 2020). Further, even though the algorithm is typically resistant to overfitting, if it is not adjusted appropriately, it may still overfit on noisy data (Sheykhmousa et al., 2020). To enhance performance, balancing these elements requires thorough parameter tuning and performance optimization (Otoo et al., 2020).

### 2.3.5 XGBoost Algorithm

XGBoost is a powerful machine learning technique that enhances gradient boosting by iteratively improving weak learners. It builds decision trees sequentially, efficiently handles sparse data and missing values, and is faster than traditional gradient boosting methods (Ali et al., 2023).

XGBoost improves performance through tree pruning, parallel boosting, and L1 and L2 regularization to prevent overfitting and control model complexity (Ali et al., 2023). A comparison study shows that while XGBoost outperforms CatBoost in speed and training accuracy, it generalizes less effectively (Bentéjac et al., 2021). XGBoost's versatility and ability to handle large datasets make it popular for regression, classification, advanced system modelling, and predictive analytics in data science applications.

XGBoost's strong default settings, speed, accuracy, and versatility make it a popular choice for both beginners and experts, excelling in regression and classification across industries like marketing, banking, and healthcare (Bentéjac et al., 2021).

XGBoost's growth has been fuelled by a strong open-source community that consistently drives its development. This collaboration has introduced advanced features like GPU acceleration, significantly cutting training times for large datasets. XGBoost's compatibility with popular machine learning libraries like scikit-learn and TensorFlow ensures it integrates smoothly into existing data science workflows, solidifying its role as an essential tool for modern data scientists (Bentéjac et al., 2021).

The use of XGBoost in detecting illicit activities within bitcoin networks shows significant potential in improving anti-money laundering efforts. Researchers introduced Adaptive Stacked eXtreme Gradient Boosting (ASXGB), a modified XGBoost designed to handle concept drift, where data patterns change over time. ASXGB gradually learns from incoming data batches, helping the model adapt to new patterns, which effectively increased recall rates and reduced false negatives in identifying illicit transactions (Vassallo et al., 2021). Additionally, when compared to traditional Random Forest and other gradient boosting methods like LightGBM and CatBoost, XGBoost consistently demonstrated superior accuracy and precision (Benemaran, 2023).

XGBoost has proven to be highly versatile, extending beyond typical classification and regression tasks to feature modification and feature graph construction in geotechnical engineering, particularly for forecasting fracture characteristics. It outperforms other boosting algorithms like Fire Hawk Optimizer and Adaptive Hoeffding Tree in terms of precision and generalization. Sensitivity studies further demonstrate XGBoost's effectiveness in predicting key parameters within complex, multi-dimensional datasets. This adaptability across various data types, including spatial and time-series data, highlights XGBoost's robustness and utility in diverse fields, from engineering to financial crime detection, making it a powerful tool for data-driven challenges (Vassallo et al., 2021).

### 2.3.6 Permutation Feature Importance

Permutation Feature Importance (PFI) is a model-agnostic method for determining feature importance by shuffling the values of each feature and observing the change in model correctness (Wang et al., 2021). A considerable decline in performance suggests a higher feature importance. PFI is computationally efficient and robust since it does not require model refitting or assumptions about feature distributions, making it applicable to a wide range of models. The method's simplicity and broad application have resulted in widespread use, notably in software engineering research. The research emphasizes that PFI consistently achieves significant agreement in feature ranks across diverse datasets, as opposed to classifier-specific approaches that are prone to variability due to feature interactions (Zou et al., 2021).

Permutation Feature Importance (PFI) is an important method for determining feature significance, although it has limits, especially for correlated features and small sample sizes. To address these issues, a cross-validated PFI (CVPFI) method is introduced that improves the stability and accuracy of feature importance judgments by combining cross-validation and taking into account feature correlations. This approach accurately calculates feature significance, even with tiny data samples and highly associated features, resulting in more reliable insights than classical PFI (Kaneko, 2022). The need of addressing biases in PFI, particularly in Random Forest models, to increase interpretability and predictive accuracy is also discussed (Altmann et al., 2010).

# 3. Chapter 3: Methodology

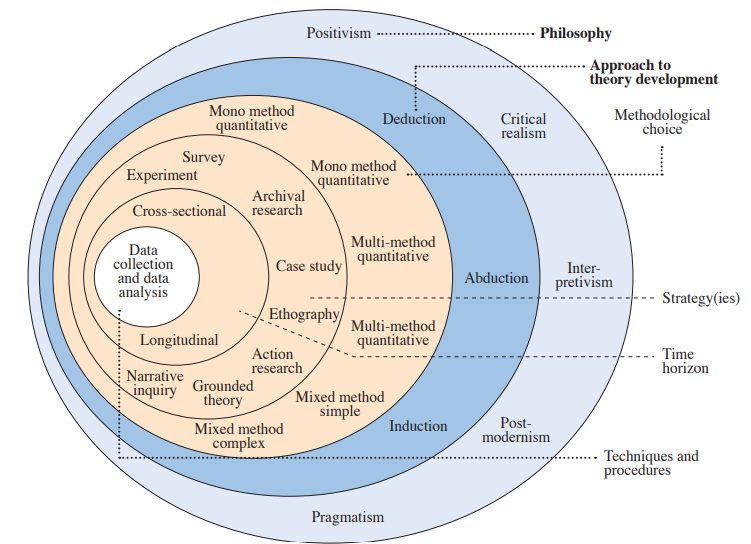
## 3.1 Overview

This chapter describes the methodology used for this study, which follows CRISP-DM framework. CRISP-DM (Cross-Industry Standard Process for Data Mining) is a structured approach to data mining that promises comprehensive and systematic analysis. This study follows the CRISP-DM procedure, which ensures a complete and replicable approach to data analysis while maintaining high levels of rigor and validity throughout the research.

## Sauders Research Onion

The Research Onion, created by Saunders, Lewis, and Thornhill in 2007, is a framework that helps researchers design and implement their research strategies methodically and thoroughly. It establishes an organized progression through the various stages of research methodology, ensuring that all key factors are considered (Melnikovas, 2018). This model offers clarity in selecting suitable methodologies, methods, and techniques, ultimately leading to more robust and coherent research designs (Alturki, 2021).

The Research Onion model has six layers (Fig 3.1.), each representing a different stage of the research design process: research philosophy, approach to theory creation, methodological choice, research strategy, time horizons, and methodologies and procedures (Abdelhakim et al., 2021).



*Fig 3.1. Saunder’s Research Onion Model (Melnikovas, 2018).*

The six main layers of Saunder’s research onion model are;

* Research Philosophy

The research philosophy represents the perspective of the researcher on the topic and incorporates key assumptions that will influence their work. The research philosophy is the basis of the research strategy; It outlines the methodology that will be utilized for answering research questions, the methods for gathering data and procedures, the analysis of the findings, and the presentation of the results of data analysis (Alturki, 2021). This research is pragmatic because it focuses on developing a practical solution to improve credit default prediction by integrating both black-box algorithms and interpretable models, thereby emphasizing the real-world application and effectiveness of the methods.

* Approach to Theory Development

The research approach involves the method of theory development, which can be deductive, inductive, or abductive. Deductive research tests existing theories by forming hypotheses and collecting data for confirmation or rejection, while inductive research forms new theories based on observations and data collection, and abductive research finds the most likely explanation by moving between induction and deduction, especially when starting with a surprising empirical phenomenon (Melnikovas, 2018). A deductive approach is used in this study as it starts with existing theories about credit default prediction and uses quantitative data to test and refine these theories through machine learning models like Random Forest and Logistic Regression.

* Methodological Choice

Research choices include quantitative methods, which focus on numerical data and mathematical operations, and qualitative methods, which involve collecting descriptive data. Mono method research uses either quantitative or qualitative approaches, whereas mixed methods combine both to achieve various objectives and compensate for the limitations of using a single approach. Multi-method research also employs both quantitative and qualitative methods, but one is primary while the other is supplementary, allowing for a more comprehensive exploration of the research topic (Melnikovas, 2018). By using techniques like permutation feature importance, this study focuses on quantifying model performance through metrics such as accuracy, precision, recall, and ROC-AUC to enhance prediction accuracy and interpretability.

* Research Strategy

Research techniques are methods for gathering data that can be applied to both real and hypothetical studies (Hernández et al., 2016). They can be used for investigative, explanatory, and descriptive research, in addition to both deductive and inductive approaches (Abdelhakim et al., 2021). The research challenge and objectives determine the methodologies and strategies used. Secondary data from Kaggle is used to analyse and improve credit card default prediction models, with a focus on understanding the relationship between variables using machine learning algorithms.

* Time Horizon

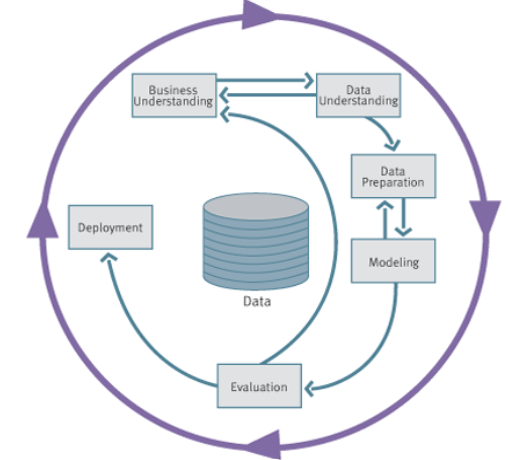
A time horizon is the timeframe in which a study is carried out, including the time range for data collection and analysis. A cross-sectional time horizon collects a single snapshot of data at one moment in time, whereas a longitudinal time horizon takes many snapshots throughout time, allowing for the study of changes and trends (Rindfleisch et al., 2008). Cross-sectional studies are important for establishing immediate correlations between causes and effects, whereas longitudinal studies aid in monitoring changes and determining the consistency of findings across time (Alturki, 2021). The dataset adopts a cross-sectional time horizon, providing a snapshot of credit data over a certain period for examining the immediate relationships between numerous factors and credit card default outcomes.

* Techniques and Procedures

The final layer of the research onion focuses on data collecting and processing, which is influenced by prior study design choices in order to properly address the research topic (Melnikovas, 2018). The techniques and procedures in your dissertation include collecting secondary data from Kaggle, developing models using machine learning and XAI techniques, and conducting quantitative evaluations using performance metrics to assess the effectiveness and interpretability of the proposed models.

## 3.3 CRISP-DM Process

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a popular technique for data mining projects that offers a standardized approach to planning and execution. CRISP-DM has become the industry standard due to its flexibility and application across multiple disciplines. It comprises of six iterative stages as given in Fig 3.2: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. This methodology ensures that data mining initiatives are handled in a systematic manner, from setting business objectives to deploying the finished model. Its iterative nature enables continual improvement, ensuring that data mining solutions efficiently satisfy corporate requirements. CRISP-DM's broad applicability and structured approach have made it the go-to framework for organizations that want to draw actionable insights from their data (Plotnikova et al., 2022).



*Fig 3.2. CRISP-DM Process (Shearer, 2000)*

The 6 iterative phases of CRISP-DM Process are;

### 3.3.1. Business Understanding

The initial phase focuses on identifying the project's business objectives and requirements. The process involves defining the problem, determining data mining goals, and creating a project plan. Key activities include reviewing the present business condition, defining data mining goals, and selecting success criteria. This phase produces a clear business understanding that serves as the foundation for following stages (Schröer et al., 2021). The current business conditions in the banking sector are difficult due to economic volatility and rising credit card default rates, necessitating better credit risk management practices. Traditional credit scoring methods, which focus on limited attributes, frequently fail to capture the whole financial picture of borrowers, resulting in erroneous risk assessments. I chose this study to overcome these constraints by creating a more thorough prediction model that includes demographic, payment history, and billing information. This improved approach intends to increase the accuracy of default predictions, allowing financial institutions to reduce risks more efficiently.

### 3.3.2. Data Understanding

During this phase, data is collected and analysed to provide insights into its characteristics and quality. This process involves collecting initial data, defining it, exploring it, and ensuring its quality. Statistical analysis and data visualization techniques are used to comprehend the structure of data, find anomalies and discover trends. The purpose is to become acquainted with the data and ensure that it is appropriate for the proposed analysis (Jaggia et al., 2020).

#### 3.3.2.1 Dataset

The dataset used for this study is from the UCI Machine Learning Repository, known as the “Default of Credit Card Clients”, which is about the case of customers’ default payments in Taiwan in 2005. The Taiwan Credit Card Clients Default Dataset consists of 30,000 instances, each representing a credit card client. The primary objective of the dataset is to predict whether a client will default on their credit card payment in the next month. A default case, which is the target, is labelled as 1, while a non-default case is labelled as 0. The dataset is highly imbalanced, with 6,626 (22.1%) default cases and 23,374 (77.9%) non-default cases (Islam et al., 2018).

The dataset consists of 24 features, which includes demographic features like ID, Sex, Education, Marriage and Age, financial features like Limit Balance, B ill Amount and Pay Amount and behavioural features like Pay status.

Education, Marriage, Sex and Pay\_0 to Pay\_6 have been encoded for machine learning analysis. Limit Balance, Age, Bill\_Amt1 to Bill\_Amt6 and Pay\_Amt1 to Pay\_Amt6 are continuous variables. Below table summarizes the features:

Table 3.1. Features and descriptions

|  |  |
| --- | --- |
| **Feature Name** | **Description** |
| ID | ID of each client |
| LIMIT\_BAL | Credit Limit of Client (in NT Dollars) |
| SEX | Gender of Client (1-male, 2-female) |
| EDUCATION | Education of Client (1-graduate school, 2-university, 3-high school, 4-others, 5-unknown, 6-unknown) |
| MARRIAGE | Marital Status (1=married, 2=single, 3=others) |
| AGE | Age of Client in years |
| PAY\_0 | Repayment status in September 2005 |
| PAY\_2 | Repayment status in August 2005 |
| PAY\_3 | Repayment status in July 2005 |
| PAY\_4 | Repayment status in June 2005 |
| PAY\_5 | Repayment status in May 2005 |
| PAY\_6 | Repayment status in April 2005 |
| BILL\_AMT1 | Amount of bill statement in September 2005 (in NT Dollars) |
| BILL\_AMT2 | Amount of bill statement in August 2005 (in NT Dollars) |
| BILL\_AMT3 | Amount of bill statement in July 2005 (in NT Dollars) |
| BILL\_AMT4 | Amount of bill statement in June 2005 (in NT Dollars) |
| BILL\_AMT5 | Amount of bill statement in May 2005 (in NT Dollars) |
| BILL\_AMT6 | Amount of bill statement in April 2005 (in NT Dollars) |
| PAY\_AMT1 | Amount paid in September 2005 (in NT Dollars) |
| PAY\_AMT2 | Amount paid in August 2005 (in NT Dollars) |
| PAY\_AMT3 | Amount paid in July 2005 (in NT Dollars) |
| PAY\_AMT4 | Amount paid in June 2005 (in NT Dollars) |
| PAY\_AMT5 | Amount paid in May 2005 (in NT Dollars) |
| PAY\_AMT6 | Amount paid in April 2005 (in NT Dollars) |
| default.payment.next.month | Default payment (1=yes, 0=no) |

In the variables PAY\_0 to PAY\_6, a value of -1 signifies payment made duly, 1 represents a payment delay of one month, 2 indicates a delay of two months, and so on, up to 8 for an eight-months delay and 9 for a delay of nine months or more.

#### 3.3.2.2 Statistical Summary

The following table provides a comprehensive statistical summary of the key features in the dataset, offering insights into the central tendencies and dispersion of the data. This summary includes the mean, median, mode, minimum, maximum, and standard deviation for each feature, allowing for a deeper understanding of the data's distribution and variability.

Table 3.2. Summary statistics of features

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature | Mean | Median | Mode | Min | Max | Std Dev |
| LIMIT\_BAL | 167484.322667 | 140000 | 50000 | 10000 | 1000000 | 129747.661567 |
| SEX | 1.603733 | 2 | 2 | 1 | 2 | 0.489129 |
| EDUCATION | 1.853133 | 2 | 2 | 0 | 6 | 0.790349 |
| MARRIAGE | 1.551867 | 2 | 2 | 0 | 3 | 0.521970 |
| AGE | 35.485500 | 34 | 29 | 21 | 79 | 9.217904 |
| PAY\_0 | -0.016700 | 0 | 0 | -2 | 8 | 1.123802 |
| PAY\_2 | -0.133767 | 0 | 0 | -2 | 8 | 1.197186 |
| PAY\_3 | -0.166200 | 0 | 0 | -2 | 8 | 1.196868 |
| PAY\_4 | -0.220667 | 0 | 0 | -2 | 8 | 1.169139 |
| PAY\_5 | -0.266200 | 0 | 0 | -2 | 8 | 1.133187 |
| PAY\_6 | -0.291100 | 0 | 0 | -2 | 8 | 1.149988 |
| BILL\_AMT1 | 51223.330900 | 22381.5 | 0 | -165580 | 964511 | 73635.860576 |
| BILL\_AMT2 | 49179.075167 | 21200 | 0 | -69777 | 983931 | 71173.768783 |
| BILL\_AMT3 | 47013.154800 | 20088.5 | 0 | -157264 | 1664089 | 69349.387427 |
| BILL\_AMT4 | 43262.948967 | 19052 | 0 | -170000 | 891586 | 64332.856134 |
| BILL\_AMT5 | 40311.400967 | 18104.5 | 0 | -81334 | 927171 | 60797.155770 |
| BILL\_AMT6 | 38871.760400 | 17071 | 0 | -339603 | 961664 | 59554.107537 |
| PAY\_AMT | 5663.580500 | 2100 | 0 | 0 | 873552 | 16563.280354 |
| PAY\_AMT | 5921.163500 | 2009 | 0 | 0 | 1684259 | 23040.870402 |
| PAY\_AMT | 5225.681500 | 1800 | 0 | 0 | 896040 | 17606.961470 |
| PAY\_AMT | 4826.076867 | 1500 | 0 | 0 | 621000 | 15666.159744 |
| PAY\_AMT | 4799.387633 | 1500 | 0 | 0 | 426529 | 15278.305679 |
| PAY\_AMT | 5215.502567 | 1500 | 0 | 0 | 528666 | 17777.465775 |
| default.payment.next.month | 0.221200 | 0 | 0 | 0 | 1 | 0.415062 |

#### 3.3.2.3 Feature to Target Correlation

The Pearson correlation coefficient, referred to as 'r', is a statistical metric that measures the linear relationship between two variables. Its range is -1 to 1. A score of one implies a perfect positive linear relationship, which means that if one variable increases, the other increases accordingly. A value of -1, on the other hand, represents a perfect negative linear relationship in which one variable increases while the other declines proportionally. A value close to 0 suggests little to no linear correlation between the variables, implying that they are likely independent of each other in terms of linear association (Nasir et al., 2020).

The formula for calculating Pearson's correlation coefficient is given by;

*r*(*a,b*) = ,

where and are the mean values of the two sequences, and *n* is the number of samples (Yang et al., 2021).

Also, Pearson Correlation Co-efficient can be calculated using the formula;

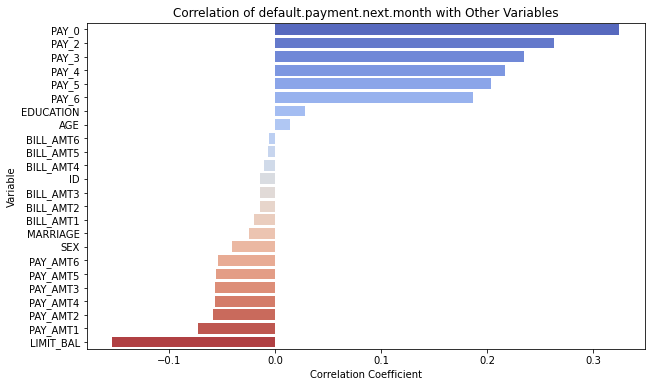
ρX,Y = ,

where *X* and *Y* are the features, *COV(X,Y)* is the covariance and *σ* are the standard deviation of X and Y (Liu et al., 2020).

The correlation graph (Fig 3.) illustrates the linear relationships between all features and the target variable, default.payment.next.month.

The variables PAY\_0 to PAY\_6 show the highest positive correlations with the target variable, which indicates that a higher delay in payments is associated with an increased likelihood of being default. PAY\_0, which is the repayment status for September, shows the strongest correlation, indicating that recent payment behaviour is a significant default predictor. EDUCATION also has a positive correlation, which represents that certain education levels might be related to higher default rates. Age also shows a slight positive correlation, indicating that older individuals may have a higher risk of being default.

LIMIT\_BAL displays the highest negative correlation, suggesting that individuals with higher credit limits are less likely to default, indicating better financial stability. The PAY\_AMT1 to PAY\_AMT6 variables are also negatively correlated with default, meaning that higher payment amounts are associated with a lower risk of default, which aligns with expectations since individuals making larger payments are typically less likely to default.



*Fig 3.3. Correlation between features and target variable*

Features like SEX and MARRIAGE have low correlations, indicating minute linear relationships with the likelihood of default. Similarly, the BILL\_AMT1 to BILL\_AMT6 shows lower correlations, suggesting that current bill amounts have a weaker linear relation with default prediction.

All the other graphs related to exploratory data analysis are discussed in Chapter 4: Analysis.

### 3.3.3 Data Preparation

Data preparation is critical to the effectiveness of the modelling phase. This process involves creating the final dataset from the initial raw data. Tasks include choosing relevant data, cleansing it for missing values or errors, creating new properties, and changing data formats. The output is a prepared dataset that is ready for modelling. This phase often requires iterative methods to modify the data until it fulfils the required quality criteria (Schröer et al., 2021).​

#### 3.3.3.1 Data Pre-processing

Data preprocessing involves applying various techniques to enhance data quality. These techniques include handling missing values, converting feature types, encoding, normalization etc (Sami et al., 2021).

#### 3.3.3.2 Data Cleaning

Data cleaning is an important stage in data preprocessing that improves data quality by removing errors such as missing values, duplicates, noise, outliers, and inconsistencies (Alasadi et al., 2017). It guarantees that datasets are accurate, dependable, and ready for analysis. Missing values must be handled properly to avoid feature loss and retain data integrity, which is frequently accomplished through interpolation or filling with mean, median, or mode values. Removing duplicates and irrelevant information helps to streamline datasets and reduce bias. Correcting structural errors, such as typos and mislabelling, ensures that data is represented uniformly. Also, smoothing noise and recognizing outliers are critical for preserving accurate trends and patterns in data. Consistent data types across characteristics reduces analysis errors (Maharana et al., 2022). The dataset used for this study is already clean, with no missing values, consistent data types, and no duplicate rows. Only data encoding was required, which is done using One-Hot Key encoding.

#### 3.3.3.3 Data Normalization

Data normalization is an important preprocessing procedure that converts attribute values to a common scale or range in order to improve the performance of machine learning algorithms. Min-Max scaling, Z-score normalization, and decimal scaling are three common normalization procedures. In Python, the scikit-learn framework includes a number of useful normalization algorithms, including MinMaxScaler (MMS), MaxAbsScaler (MAS), StandardScaler (SS), RobustScaler (RS), and Normalizer. These techniques help to guarantee that the data is suitably scaled, which reduces biases caused by features with differing units or magnitudes (Halim et al., 2020). The sensitivity of the specified machine learning model to the feature value determines the need for data normalization. Thus, a non-normalized dataset can cause the chosen machine learning model to discover false dependencies in the data and, as a result, impair the efficiency of its work in completing the specified task (Izonin et al., 2022). The MinMaxScaler is a normalization technique that transforms data to fit within a given range, usually between 0 and 1, or -1 to 1 if negative values are present. It achieves this by subtracting the minimum value from each data point and then dividing by the range, which is the difference between the maximum and minimum values. This approach maintains the original data distribution while ensuring that all values are proportionally scaled within the intended range, effectively making the data to fit within the defined constraints (Quang et al., 2024). In this study, I have used the MinMaxScaler normalization technique to scale the dataset.

### 3.3.4 Modelling

The modelling phase involves selecting appropriate modelling techniques and applying them to the prepared dataset. This process includes choosing suitable algorithms, building models, and tuning parameters to optimize performance. Common modelling techniques used in this phase include classification, regression, clustering, and association rules. Each model's performance is evaluated against predefined criteria, ensuring that only the best-performing models are selected for further, more detailed analysis and potential deployment (Schröer et al., 2021). In this study, the whole dataset is split into training data and test data before modelling. Train data contains 22,500 data samples and test data has 7500 data samples. Four machine learning algorithms – Random Forest, XGBoost, Logistic Regression and Decision Tree, are used along with Explainable AI techniques like permutation feature importance.

### 3.3.5 Evaluation

In the evaluation phase, models are rigorously assessed to ensure they meet business objectives and are robust. This involves evaluating performance using various metrics like accuracy, precision, recall, and F1 score. Also, models are reviewed to ensure they align with business perspectives. If the models don't meet the required standards, they may need refinement or redevelopment. The aim is to confirm the models' readiness for deployment and their ability to deliver the anticipated business value, ensuring alignment with organizational goals and strategies (Schröer et al., 2021). Both training data and test data are evaluated using metrics like accuracy, precision, recall, F1 score and ROC-AUC.

### 3.3.6 Deployment

The last phase involves deploying the models into a production environment where they are used to make business decisions. This may include generating reports, integrating the models into existing systems, or developing new applications. The deployment phase also includes planning for monitoring and maintaining the models to ensure they continue to perform well over time. Successful deployment ensures that the insights gained from the data mining project are actionable and provide ongoing value to the business (Schröer et al., 2021).

To conclude, the CRISP-DM framework is still a popular and useful tool for organizing data science initiatives. Its strengths originate from its clear, intuitive phases that build a shared knowledge and vocabulary among team members, making it simple to implement and adapt to iterative processes and the empirical nature of data science. However, the framework has major weaknesses such as a lack of defined team coordination and communication structures, an outdated methodology that may not completely meet the complexities of modern big data projects, and a documentation-heavy process that might obstruct rapid value delivery (Saltz, 2021).

# 

# Chapter 4: Analysis

## 4.1 Overview

The analysis section starts with an in-depth exploratory data analysis (EDA) to identify relevant patterns, trends, and relationships in the dataset, setting the basis for further modelling. After the EDA, the focus is on the construction and evaluation of the four machine learning models, with the goal of accurately and interpretably predicting credit card defaults.

## 4.2 Exploratory Data Analysis (EDA)

Fig 4.1. represents the age distribution in the dataset, providing insights into the demographic profile of the individuals being studied.

A graph of a distribution of age

Description automatically generated

*Fig 4.1. Histogram of various age groups of customers*

The distribution is right-skewed, with the majority of the population categorized between the ages of 20 and 45. The 27-34 age group has the highest frequency, followed by the 34-39 age group, indicating that the majority of the persons in the dataset are relatively young. As age exceeds 40 years, the frequency continuously decreases, indicating a lower proportion of older people in the dataset. The tail end of the distribution, beyond 60 years old, contains very few numbers, indicating a considerable reduction in representation among older age groups.

Fig 4.2. depicts the frequency distribution of data points across various education levels in the dataset.

A graph of a bar chart

Description automatically generated with medium confidence

*Fig 4.2.* *Bar graph of various education levels of customers*

University-level education shows the highest number of customers and defaulters. Graduate school follows, with significant defaulter representation. High school graduates are fewer but still has notable defaults. "Unknown" and "Others" categories have minimal counts and defaults. This distribution indicates a possible association between education level and credit card usage, with higher education levels being more common among credit card holders.

Fig 4.3. illustrates the distribution of the LIMIT\_BAL variable, which is related to the credit limit of the persons in the dataset. The distribution is highly right-skewed, showing that most people have a smaller credit limit, with the majority of individuals falling between 0 and 200,000.

A graph of a graph of a bar graph

Description automatically generated with medium confidence

*Fig 4.3.* *Histogram of credit limit of customers*

The highest frequency of customers, around 5,000, is seen with credit limits between 50,000 and 100,000, where a notable number of defaults also occur. The highest number of defaults, approximately 1500, occurs in the limit balance from 0 to 50,000. As credit limits rise beyond 200,000, both the number of customers and the default rate significantly decrease. Very few customers have credit limits exceeding 500,000, and defaults are rare in this range, which shows that higher credit limits are associated with lower default risk in the dataset.

Fig 4.4. represents the frequency distribution of customers and defaulters in the dataset according to their marital status. The categories are- "Single," "Married," "Others," and "Unknown."

A graph of a number of people

Description automatically generated with medium confidence

*Fig 4.4.* *Bar graph of marital status of customers*

The "Single" category, with nearly 16,000 customers, also has the highest number of defaulters with almost 3500, followed by the "Married" category, which shows a significant defaulter count of around 3500 among its 14,000 customers. The "Others" and "Unknown" categories have minimal representation and few defaulters, emphasizing the higher default risk among single and married individuals.

Fig 4.5. show the distribution of repayment statuses (PAY\_0 to PAY\_6) across six months, from April to September 2005. The repayment status is classified on a scale, with 0 and negative values (-1, -2) indicating timely or early payments and positive values (1–9) indicating payments that were delayed by the number of months.

A graph of a number of bars

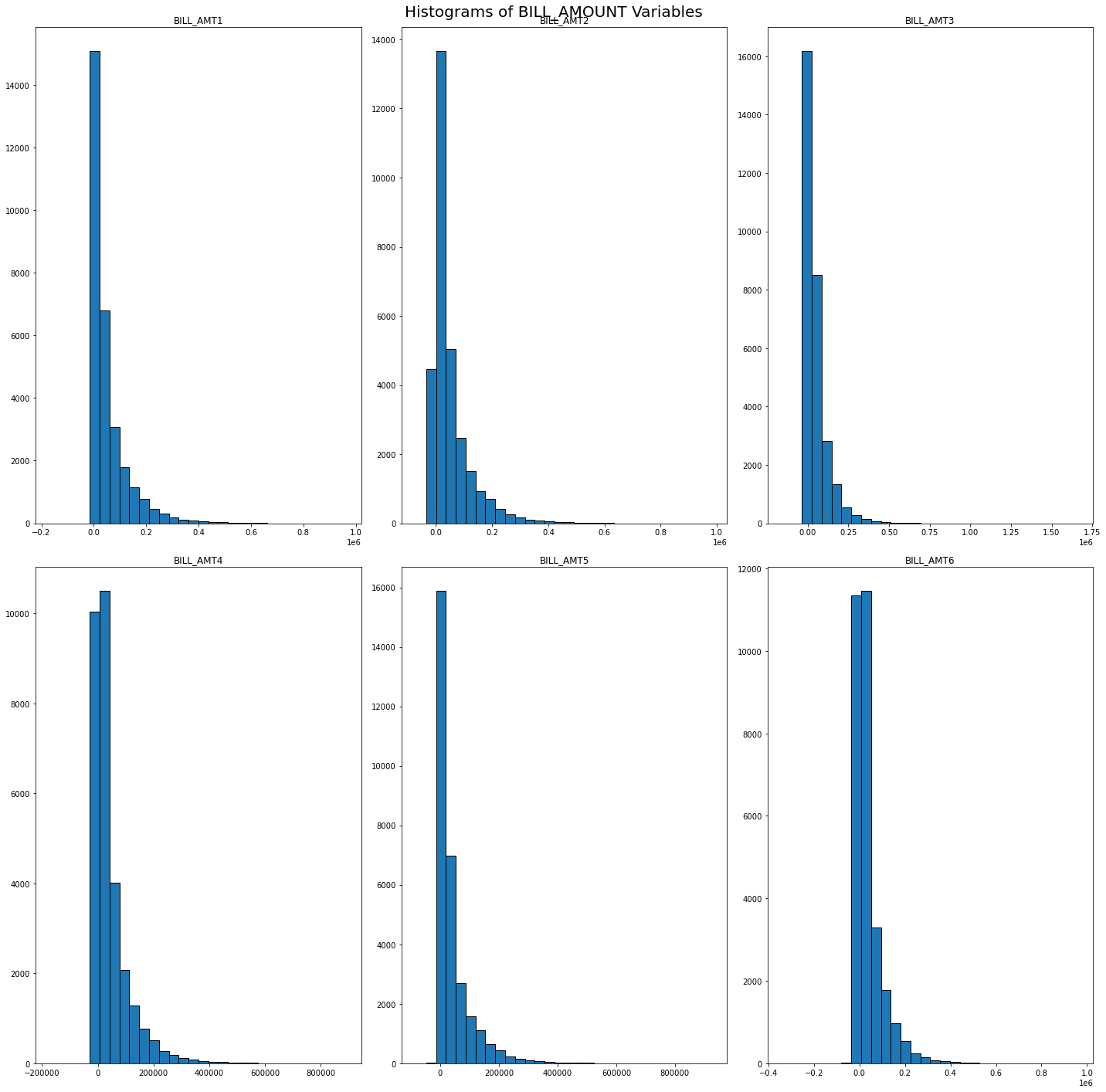
Description automatically generated with medium confidence

*Fig 4.5.* *Histograms of Pay status of customers for 6 months*

Throughout the six months, the majority of persons have a repayment status of 0, showing no delay in payment. This shows that the majority of people in the dataset paid their bills on time during this time period. There are also noticeable rates for payback statuses 1 and 2, which indicate a 1 or 2 months delay in payment, but they are substantially less compared to on-time or early payments.

The small number of customers with repayment statuses greater than 2 (indicating longer delays) shows that significant payment delays were infrequent in this dataset. The regularity of these trends over the last six months suggests that most people's payment behaviour is consistent, with a few having short-term delays. The small number of customers with repayment statuses greater than 2 (indicating lengthier delays) shows that significant payment delays were not common in this dataset. The regularity of these trends over the past six months suggests that most people's payment behaviour is consistent, with a few having short-term delays.

Fig 4.6. is the distribution of bill amounts (in NT dollars) throughout six months, from April to September 2005 (BILL\_AMT1 to BILL\_AMT6). All six distributions are highly right-skewed, indicating that the majority of people have low bill amounts, with only a few having significantly larger amounts. The highest frequency in each histogram is close to zero, indicating that the majority of the bills fall into the lower ranges.



*Fig 4.6.* *Histograms of Bill Amount status of customers for 6 months*

As the bill amount increases, the frequency reduces significantly, indicating that fewer people have greater bill amounts. This pattern remains constant over all the six months, indicating a consistent trend in billing behaviour over time. The distribution indicates that, while the majority of individuals in the sample have low to moderate amounts, a small segment has high payments.

Fig 4.7. represents the distribution of payment amounts (in NT dollars) for six months, from April to September 2005.

A graph of a number of blue lines

Description automatically generated with medium confidence

*Fig 4.7.* *Histograms of Pay Amount status of customers for 6 months*

All six distributions are strongly right skewed, which depicts that majority of people made relatively small payments, with only a small number of customers making much bigger payments. The maximum frequency can be seen at the lower end of the graph, implying that the majority of payments were small. As the payment amount grows, the frequency reduces significantly, indicating that fewer people made higher payments. This pattern repeats itself throughout the course of six months, indicating that payment behaviours have remained stable.

Fig 4.8. shows the frequency distribution of all customers and defaulters by gender within the dataset.

A screenshot of a graph

Description automatically generated

*Fig 4.8.* *Bar graph of Gender of customers*

It shows there are more female customers (over 17,500) than male customers (around 12,500) in the dataset. Females also have a higher number of defaulters compared to males. The proportion of defaulters is similar across both genders, indicating that default risk is not strongly biased by gender, despite the higher female representation.

Fig 4.9. depicts the average limit balance of customers at various educational levels.

A graph of a graph of a number of blue bars

Description automatically generated with medium confidence

*Fig 4.9.* *Bar graph of limit balance across different education levels*

Individuals with a graduate school education and “Others” have a high average limit balance of slightly more than 200,000 NT dollars, showing that those with higher education levels are more likely to acquire larger credit limits. The "Others" category may include people with specialized or international education backgrounds.

Individuals with only a high school education have the lowest average limit balance, which is approximately 120,000 NT dollars. University graduates and individuals in the "Unknown" group have moderate average limit balances, indicating that they fall somewhere between the high school and graduate school levels. This distribution indicates a relationship between education level and creditworthiness, with better education frequently leading to larger credit limits, potentially due to predicted higher income levels and financial responsibility.

Fig 4.10. represents the average credit limit balance for various marital statuses.

A graph of a number of blue rectangular bars

Description automatically generated

*Fig 4.10.* *Bar graph of limit balance across different marital status*

It shows that married people have the highest average limit balance, which is close to 180,000 NT dollars. This shows that financial institutions could consider married people as more financially secure or responsible, possibly due to dual-income households or other marital status-related characteristics.

Single individuals have a somewhat lower average limit amount of roughly 150,000 NT dollars, indicating a good degree of creditworthiness. The "Others" category has the lowest average limit balance, implying that those who do not fall into standard marital classifications may be given lower credit limits, most likely due to perceived increased financial risk or less predictable income.

The "Unknown" group has a relatively low average limit balance, indicating that incomplete or unclear marital status information may result in more conservative credit limit assignments.  
Overall, the graphic indicates that marital status is an important influence in establishing credit limits, with married people being seen as less risky, resulting in higher credit limits.

Fig 4.11. represents the composition of the target class for default payment, giving the percentage of customers that either defaulted (Yes) or did not default (No).

A pie chart with a red circle and numbers

Description automatically generated

*Fig 4.11.* *Pie chart of target class distribution*

The chart shows that 77.9% of customers did not default on their payments, as indicated by the bigger blue area of the pie. In contrast, 22.1% of the customers defaulted, as indicated by the smaller red region. This distribution demonstrates a slight imbalance in the dataset, with a considerably greater proportion of non-defaulters than defaulters. Such an imbalance is common in credit risk databases, as most credit card customers satisfy their payment responsibilities.

## 4.3 Modelling

The modelling procedure starts with a critical step: splitting the whole dataset into traindata and testdata. 7500 datapoints (25%) are allotted to a new dataframe X\_test for testing and the remaining to X\_train for training the models. The next step is separating the features (Features\_train) from the target variable (Labels\_train) of traindata. Proper separation of features and the target is important for training the models appropriately. Following that, the code deals with missing values by replacing all NaN values in Features\_train with -999. This strategy of utilizing a placeholder value ensures that the models can analyse the input without problems, particularly when dealing with algorithms that do not have inbuilt missing value handling capability. Another important preprocessing step is feature normalization with the MinMaxScaler().

### 4.3.1 Model Training

The code then trains four different machine learning models: Random Forest (rf\_model), XGBoost (xgb\_model), Logistic Regression (lr\_model), and Decision Tree (dt\_model). Each of these models had been carefully selected based on its distinct capabilities and capacity to address various aspects of the prediction problem. The Random Forest model uses 100 trees (n\_estimators=100) to reduce variance by averaging. It restricts tree depth to 10 (max\_depth=10) to prevent overfitting. Setting minimal samples for splitting and leaves (min\_samples\_split=5, min\_samples\_leaf=4) reduces complexity while improving generalisation. The XGBoost model is set to a maximum tree depth of 3 (max\_depth=3), which helps to prevent overfitting by limiting complexity. To improve efficiency, it removes label encoding and evaluates using the logloss metric. Random\_state=42 ensures that results are reproducible across various runs. To ensure convergence, the Logistic Regression model is set to a maximum of 500 iterations (max\_iter = 500). The model is implicitly regularised, which helps to balance complexity and prevent overfitting. To avoid overfitting, the Decision Tree model sets a maximum depth of 5 (max\_depth=5). It takes at least four samples to split a node (min\_samples\_split=4) and three samples per leaf (min\_samples\_leaf=3), lowering the chance of the model fitting to noise in the training data.

The training phase comprises of fitting these models to the training data (features\_train and labels\_train). This is the process in which the models learn the deeper trends in the data, with each model adjusting its parameters to reduce prediction errors. The fitting process is critical because it directly affects the model's capacity to generate accurate predictions on unknown data.

### 4.3.2 Trained Model Evaluation

### After training, model performance is evaluated using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. These metrics assess prediction accuracy, positive forecast accuracy, sensitivity, and class distinction ability. Results are stored in a list (train\_results) and converted into a DataFrame (train\_results\_df) for comparison and understanding. The training results are as below in Table 4.1.

Table 4.1. Train data results of Random Forest Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Random Forest Classifier | 0.849 | 0.796 | 0.433 | 0.561 | 0.850 |

Random Forest achieved the highest accuracy (0.849) and ROC-AUC (0.850), showing it is the best at distinguishing between classes. Its precision (0.796) is also the highest, meaning it has fewer false positives. However, its recall (0.433) is lower, suggesting it misses a fair number of positive cases, leading to a moderate F1-Score (0.561).

Table 4.2. Train data results of XGBoost Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| XGBoost Classifier | 0.831 | 0.716 | 0.402 | 0.515 | 0.832 |

Table 4.2 shows results of XGBoost with a balanced performance with good accuracy (0.831), precision (0.716), and ROC-AUC (0.832). However, its recall (0.402) is slightly lower than Random Forest, which results in a lower F1-Score (0.515).

Table 4.3. Train data results of Logistic Regression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Logistic Regression | 0. 810444 | 0.722 | 0.242 | 0.362 | 0. 720619 |

Table 4.3 represents the results of Logistic Regression model, which has the lowest recall (0.242) and F1-Score (0.362), indicating the model’s inability to correctly identify positive cases. Despite this, it maintains reasonable precision (0.722), showing it is more conservative in predicting positives.

Table 4.4. Train data results of Decision Tree

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Decision Tree | 0.825 | 0. 690755 | 0.389 | 0. 497957 | 0.764 |

Decision Tree in Table 4.4. also performed moderately, with an accuracy of 0.825, similar to XGBoost, but with a lower ROC-AUC (0.764). It also has a higher recall (0.389) than Logistic Regression.

### 4.3.3 Testdata

Following the examination of the training data, the algorithm prepares the test data (Features\_test) in the same manner. The features are separated from the target variable, and normalization is performed with the same MinMaxScaler(). This consistency ensures that the models are assessed under the same conditions in which they were trained, allowing for an accurate assessment of the models’ generalization abilities.

### 4.3.4 Testdata Evaluation

The models are then assessed against the testdata using the same set of metrics. This evaluation of testdata is critical because it demonstrates how well the models can generalize their learned patterns to new situations. The results are saved in a separate DataFrame (test\_results\_df) for comparison to the training performance. Comparing these test measures to the training metrics allows one to determine if the models are overfitting or underfitting.

Table 4.5. Test data results of Random Forest Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Random Forest Classifier | 0.811 | 0.661 | 0. 264905 | 0.378 | 0.766 |

Testdata results from Table 4.5 of Random Forest maintains decent accuracy (0.811) and a ROC-AUC (0.766), though both are lower than the training data results. Precision has dropped from 0.796 to 0.661, and recall has almost halved, indicating that the model is struggling to correctly identify positive cases on unseen data. Also, there is a decrease in F1-Score from 0.561 to 0.378.

Table 4.6. Test data results of XGBoost Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| XGBoost Classifier | 0.789 | 0. 553738 | 0.146 | 0.231 | 0. 706102 |

Testdata results from Table 4.6 of XGBoost shows a drop in accuracy (from 0.831 to 0.789) and a sharp decline in recall (from 0.402 to 0.146), which has affected the F1-Score (from 0.515 to 0.231). This significant drop across metrics indicates that XGBoost may have overfitted to the training data, struggling to generalize to new data.

Table 4.7. Test data results of Logistic Regression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Logistic Regression | 0.811 | 0. 646006 | 0.288 | 0. 398640 | 0.720 |

Testdata results from Table 4.7 of Logistic Regression exhibits consistent performance, with slight increase in accuracy (from 0.810 to 0.811) and recall (from 0.242 to 0.288), and a constant ROC-AUC of 0.720. The small changes in F1-Score indicate that Logistic Regression has generalized better compared to more complex models.

Table 4.8. Test data results of Decision Tree

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Decision Tree | 0.819 | 0. 652762 | 0. 355870 | 0. 460621 | 0.733 |

Testdata results from Table 4.8 of Decision Tree has slight decrease in accuracy from 0.825 to 0.819 and ROC-AUC, from 0.764 to 0.733. However, its precision and recall remain somewhat stable, indicating that while it overfits less than XGBoost, it still suffers from some degree of overfitting.

### 4.3.5 Hybrid Model development

To improve prediction performance, a hybrid model is developed by averaging the predicted probability of the four distinct models. This ensemble strategy takes advantage of each model's capabilities, with the goal of producing a more accurate and robust predictions.

The hybrid model is initially evaluated on training data using the same metrics as the separate models, and the results are saved in a DataFrame (hybrid\_results\_train\_df). The results are as depicted in Table 4.9. This initial evaluation determines how effectively the hybrid model has learned from the data.

Table 4.9. Hybrid model train data results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| 0. 830622 | 0. 728626 | 0. 381114 | 0. 500459 | 0. 823387 |

The model is then evaluated using the test data to determine how well it generalizes. The findings of this examination are saved in a separate DataFrame (hybrid\_results\_test\_df). The results are as depicted in Table 4.10.

Table 4.10. Hybrid model test data results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| 0.787867 | 0. 814 | 0. 271051 | 0. 387352 | 0. 761496 |

### 4.3.6 Permutation Feature Importance

The Random Forest and XGBoost models' features are assessed for significance using permutation feature importance. By assessing variations in model performance following feature value shuffles, this approach evaluates the influence of each feature. Plotting the importance allows for the identification of the most important aspects, which helps to clarify how each variable influences the predictions made by these black-box algorithms and offers insights into the decision-making process of the model.

# Chapter 5: Discussions

## 5.1 Training Data Evaluation

The training data evaluation results provided in Table 5.1. gives a comparison of the performance of four machine learning models for the training dataset.

Table 5.1. Training Data Evaluation Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classifier Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Random Forest | 0. 849022 | 0. 795671 | 0. 433021 | 0. 560827 | 0. 850893 |
| XGBoost | 0. 831378 | 0. 715655 | 0. 402476 | 0. 515206 | 0. 832839 |
| Logistic Regression | 0. 810444 | 0. 721957 | 0. 241565 | 0. 362004 | 0. 720619 |
| Decision Tree | 0. 825244 | 0. 690755 | 0. 389299 | 0. 497957 | 0. 764336 |

The training data evaluation reveals that Random Forest performed best because of its high accuracy and precision, but its lower recall suggests that it may have missed some positive examples. XGBoost likewise demonstrated high precision and ROC-AUC, but its lower recall shows that it fails to capture true positives. Logistic Regression performed poorly, with low recall and F1-scores, showing difficulties managing complicated data. The Decision Tree was moderately effective but lacked precision and recall, indicating limits in capturing the data's complexity. Overall, ensemble approaches such as Random Forest and XGBoost performed better, but recall may be improved with tuning.

## 5.2 Test Data Evaluation

Table 5.2. Test Data Evaluation Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classifier Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| Random Forest | 0. 811067 | 0. 661043 | 0. 264905 | 0. 378236 | 0. 765855 |
| XGBoost | 0. 789200 | 0. 553738 | 0. 145667 | 0. 230657 | 0. 706102 |
| Logistic Regression | 0. 811333 | 0. 646006 | 0. 288261 | 0. 398640 | 0. 720298 |
| Decision Tree | 0. 819200 | 0. 652762 | 0. 355870 | 0. 460621 | 0. 732539 |

The test data results indicates that Random Forest retained good accuracy and precision, similar to the training data, but struggled with low recall, indicating that it continues to overlook many positive cases. XGBoost performed the worst overall, with large decreases in recall and F1-score, showing the difficulty of generalizing to previously unknown data. Logistic Regression performed similarly to the training data, with moderate accuracy but low recall, demonstrating that capturing complicated patterns remains a challenge. Decision Tree improved a little in recall and F1-score relative to training data, indicating better generalization but still lagging behind ensemble approaches. This comparison highlights the fact that, while ensemble approaches such as Random Forest perform well, they require further adjustment to balance precision and recall across datasets.

## 5.3 Hybrid Model Evaluation – Train Data

Table 5.3. Hybrid Model Results on Training Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| 0. 830622 | 0. 728626 | 0. 381114 | 0. 500459 | 0. 823387 |

The hybrid model is created by averaging the results of the four individual models shows a balanced performance on the training data. It achieved an accuracy of 0.831, with a precision of 0.729 and a recall of 0.381. The F1-Score is 0.500, and the ROC-AUC is 0.823.

These findings suggest that the hybrid model retains some of the separate models' best features, including good accuracy and a decent trade-off between recall and precision. Still, the recall is marginally lower, indicating that some positive examples may still be missed by the model. The overall performance shows that the hybrid model is a reliable choice, providing a more balanced and generalized approach compared to the individual models alone.

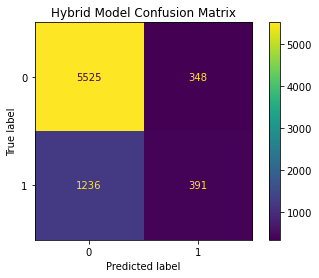
## 5.4 Hybrid Model Evaluation – Test Data

Table 5.4. Hybrid Model Results on Test Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-Score | ROC-AUC |
| 0.787867 | 0. 814 | 0. 271051 | 0. 387352 | 0. 761496 |

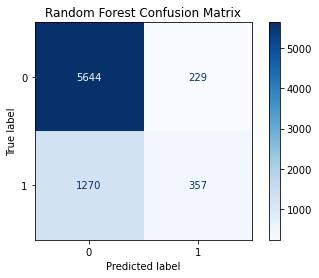
The hybrid model's test data evaluation shows a slight decline in performance compared to the training data evaluation, with accuracy at 0.814 and a ROC-AUC of 0.761. With a precision of 0.678 and a recall of 0.271, the F1-Score is 0.387. This shows an ability for generalisation.

The hybrid approach performs fairly when compared to the individual models, but it is not as accurate or recall-efficient as Random Forest. It does not, however, suffer from the extreme overfitting of models such as XGBoost. Therefore, the hybrid technique is a good option for overall performance and stability, especially when balancing the benefits and drawbacks of different models.



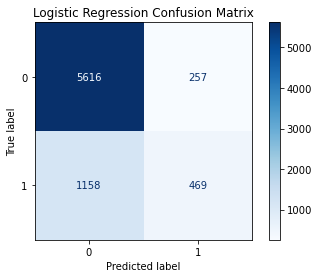
*Fig 5.1 Confusion matrix for test data of hybrid model.*

The confusion matrix in Fig 5.1 demonstrates that the hybrid model performs well in detecting the majority class (label 0), with 5,525 true negatives and only 348 false positives. But the model stumbles with the minority class (label 1), where it correctly recognizes only 391 true positives while missing 1,236 instances, resulting in a significant false negative rate. This shows that, while the model is good at recognizing the majority class, it struggles to recognize the minority class, which could be problematic in cases when both groups are equally important.



*Fig 5.2 Confusion matrix for test data of random forest model.*

The Random Forest model's confusion matrix shows excellent results in correctly determining the majority class (label 0), with 5,644 true negatives and 229 false positives. Still, like with the hybrid model, it struggles with the minority class (label 1), where it properly identifies 357 true positives but 1,270 false negatives. This shows that, while the Random Forest model is effective at categorizing the majority class, it does poorly with the minority class.



*Fig 5.3 Confusion matrix for test data of logistic regression model.*

The confusion matrix for the Logistic Regression model in Fig 5.3 indicates that the model does a good job of identifying the majority class (label 0), with 5,616 true negatives and 257 false positive. Compared to the hybrid and Random Forest models, Logistic Regression performs slightly better with the minority class (label 1), detecting 469 true positives and 1,158 false negatives. This suggests that, while the model is effective in classifying the majority class, it still struggles with the minority class, though less than the other models.

A graph with blue lines

Description automatically generated

*Fig 5.4. Feature Importance in Random Forest Model*

The permutation feature importance graphs for both XGBoost (Fig 5.2) and Random Forest (Fig 5.1.) models show "PAY\_0" as the most influential feature by a significant margin, demonstrating that the repayment status in September (PAY\_0) is the best predictor of the target variable. Other important features are "PAY\_AMT1," "PAY\_5," and a variety of payment amounts (PAY\_AMT2-6). However, the relevance of individual features such as "AGE," "MARRIAGE," and "EDUCATION" varies among models, implying that XGBoost may capture intricate patterns involving these features than Random Forest. The findings highlight the importance of payment history in predicting outcomes.

A graph with blue and white stripes

Description automatically generated

*Fig 5.5. Feature Importance in XGBoost Model*

Overall, the Random Forest model performed the best, particularly in recall and ROC-AUC, which makes it the best choice for cases where accurate classification and fewer false positives are the most important factors. In contrast, the hybrid model reduces overfitting and offers stability across measurements, resulting in a balanced performance. The hybrid model is advised for reliable, generalisable outputs, particularly in situations when there isn't a single better model.

# Chapter 6: Conclusion

The primary aim of this study was to explore whether credit card default predictions is possible. By combining machine learning methods and explainable AI techniques, the study sought to balance predictive accuracy with model interpretability, ultimately providing financial institutions with actionable insights.

The study successfully achieved its aim and objectives. The study created a comprehensive prediction model by combining data from several sources, such as demographic, billing and payment histories and payment behaviour. Classifier machine learning models such as Random Forest, XGBoost, Logistic Regression, and Decision Trees resulted in considerable increases in prediction accuracy. The Random Forest classifier model stood up as the best-performing algorithm, with the highest accuracy and robustness in predicting credit card defaults.

The study also addressed the challenge of interpretability of black-box algorithms like Random Forest and XGBoost by employing Permutation Feature Importance (PFI) as an XAI technique. This approach provided valuable insights on which features contributed most significantly to the model's predictions. The repayment status (PAY\_0 to PAY\_6) was identified as the most significant set of features, indicating that recent payment behaviours are strong predictors of credit default. The credit limit (LIMIT\_BAL) and payment amounts (PAY\_AMT1 to PAY\_AMT6) also played a critical role, with higher credit limits and larger payment amounts being associated with a lower likelihood of default. However, features such as marital status (MARRIAGE) and sex (SEX) showed relatively low significance, suggesting that they have limited influence on the prediction outcomes.

The first objective, enhancing prediction accuracy, was met by combining the rich dataset with machine learning algorithms. The study demonstrated that a hybrid model approach, combining the strengths of different algorithms, provided the most robust and reliable predictions. Among the individual models, Random Forest emerged as the best model, showing excellent accuracy and robustness in identifying default customers. This model outperformed other methods and was particularly effective in reducing the likelihood of misclassification.

The second objective, improving model interpretability, was also achieved. The use of XAI techniques, Permutation Feature Importance (PFI), allowed for a better understanding of the model’s decision-making process, making it easier for stakeholders to trust and act on the predictions.

The third objective, balancing accuracy and interpretability in hybrid models, was successfully explored. The study highlighted the trade-offs between using complex, highly accurate models and simpler, more interpretable ones. The hybrid approach adopted in the study was found to be more robust and reliable, providing a promising direction for financial institutions to maximize both accuracy and transparency. While Random Forest was the best single model, the hybrid model—especially when combined with PFI—stood out as a strong candidate for practical implementation in credit risk assessment.

## 6.1 Limitations

The study used a single dataset from the UCI Machine Learning Repository, which focused on credit card clients in Taiwan from 2005. Although this dataset provides useful insights into credit card default prediction, it also limits the generalizability of the results. The distinct socioeconomic circumstances, regulatory environments, and credit practices found in Taiwan during this time period may not be indicative of other places or timeframes. As a result, the model's performance and predicted accuracy may vary greatly when applied to different situations, such as other nations with different economic conditions, financial systems, or during different economic times. To ensure the model's adaptability and wider applicability, future study should include diverse datasets from different countries and time periods. This allows researchers to evaluate the model's effectiveness in a broader range of scenarios, increasing its overall reliability and usefulness in a variety of applications. This approach would make the model more adaptable and suitable for worldwide application.

Also, the computational resources needed for training and evaluating the hybrid model were essential, especially when dealing with big datasets and applying numerous methods concurrently. This high demand for computational power may provide substantial hurdles for implementation, particularly in situations where computational resources are restricted or expensive to get. As a result, future research efforts could focus on developing more efficient algorithms or optimization strategies to lower these computing needs. Addressing all of this would make the model more accessible, practical, and simple to implement in a wide range of real-world scenarios, even those with limited computational resources. By resolving these obstacles, the model's potential impact and utility across many industries and geographies would be greatly expanded, making it a more feasible tool for broad adoption.

## 6.2 Future Scope

The study's findings suggest several exciting opportunities for future research. One important topic to investigate is the use of other XAI methods besides Permutation Feature Importance (PFI). Techniques like SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) could be utilised to enhance the interpretability of machine learning models. These methods may provide more detailed insights into how diverse aspects influence model decisions, thereby boosting stakeholder confidence and regulatory compliance. Furthermore, building dynamic credit scoring models that respond to changes in borrower behaviour and economic conditions may result in more accurate and timely predictions. To better capture the changing financial scene, this technique may involve using real-time data and more powerful machine learning methodologies, such as deep learning or reinforcement learning.

Another promising area of future research is the integrating of behavioural and psychometric data into credit risk models. Understanding the psychological aspects that drive financial decision-making could help to develop these models and provide better estimates of default risk. Furthermore, because the dataset employed in this study was particular to Taiwan, a comparative analysis across regions would be useful. Assessing the performance of these models in different economic circumstances could help assess their generalisability and robustness. Also, incorporating blockchain technology and fintech advancements into credit scoring systems could be considered for future research.

# References

Abd Halim, K.N., Jaya, A.M. and Fadzil, A.F.A., 2020. Data pre-processing algorithm for neural network binary classification model in bank tele-Marketing. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *9*, pp.272-277.

Abdelhakim, A.S., 2021. Adopted Research Designs by Tourism and Hospitality Postgraduates in The Light‎ of Research Onion. *International Journal of Tourism and Hospitality Management*, *4*(2), pp.98-124.

Ahmed, S., Alshater, M.M., El Ammari, A. and Hammami, H., 2022. Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, *61*, p.101646.

Alasadi, S.A. and Bhaya, W.S., 2017. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, *12*(16), pp.4102-4107.

Ali, Z.A., Abduljabbar, Z.H., Taher, H.A., Sallow, A.B. and Almufti, S.M., 2023. Exploring the power of eXtreme gradient boosting algorithm in machine learning: A review. *Academic Journal of Nawroz University*, *12*(2), pp.320-334.

Altmann, A., Toloşi, L., Sander, O. and Lengauer, T., 2010. Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*(10), pp.1340-1347.

Alturki, R., 2021. Research Onion for Smart IoT‐Enabled Mobile Applications. *Scientific programming*, *2021*(1), p.4270998.

Bammidi, T.R., 2023. Transforming Credit Assessment: The Power of Artificial Intelligence. *International Journal of Interdisciplinary Finance Insights*, *2*(2), pp.1-14.

Bansal, M., Goyal, A. and Choudhary, A., 2022. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal*, *3*, p.100071.

Bello, O.A., 2023. Machine learning algorithms for credit risk assessment: an economic and financial analysis. *International Journal of Management*, *10*(1), pp.109-133.

Benemaran, R.S., 2023. Application of extreme gradient boosting method for evaluating the properties of episodic failure of borehole breakout. *Geoenergy Science and Engineering*, *226*, p.211837.

Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G., 2021. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*, pp.1937-1967.

Black, J.E., Kueper, J.K. and Williamson, T.S., 2023. An introduction to machine learning for classification and prediction. *Family practice*, *40*(1), pp.200-204.

Boateng, E.Y., Otoo, J. and Abaye, D.A., 2020. Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review. *Journal of Data Analysis and Information Processing*, *8*(4), pp.341-357.

Charbuty, B. and Abdulazeez, A., 2021. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), pp.20-28.

Chen, S. and Lin, X., 2023. Application of Decision Tree Algorithm in Educational Data Mining. *Curriculum and Teaching Methodology*, *6*(8), pp.120-127.

Dastile, X., Celik, T. and Potsane, M., 2020. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, *91*, p.106263.

Go, E.J., Moon, J. and Kim, J., 2020. Analysis of the current and future of the artificial intelligence in financial industry with big data techniques. *Global Business & Finance Review (GBFR)*, *25*(1), pp.102-117.

Hernández, J.G.V., Pérez, O.E.A. and Rangel, A.C., 2016. A review of research methods in strategic management. What have been done and what is still missing. *Journal of Knowledge Management, Economics and Information Technology*, *6*(2), pp.1-42.

Huang, Y., Zhang, L., Li, Z., Qiu, H., Sun, T. and Wang, X., 2020. Fintech credit risk assessment for SMEs: Evidence from China.

Islam, S.R., Eberle, W. and Ghafoor, S.K., 2018. Credit default mining using combined machine learning and heuristic approach. *arXiv preprint arXiv:1807.01176*.

Izonin, I., Tkachenko, R., Shakhovska, N., Ilchyshyn, B. and Singh, K.K., 2022. A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain. *Mathematics*, *10*(11), p.1942.

Jaggia, S., Kelly, A., Lertwachara, K. and Chen, L., 2020. Applying the CRISP‐DM framework for teaching business analytics. *Decision Sciences Journal of Innovative Education*, *18*(4), pp.612-634.

Kaneko, H., 2022. Cross‐validated permutation feature importance considering correlation between features. *Analytical Science Advances*, *3*(9-10), pp.278-287.

Khandani, A.E., Kim, A.J. and Lo, A.W., 2010. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, *34*(11), pp.2767-2787.

Kozodoi, N., Jacob, J. and Lessmann, S., 2022. Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, *297*(3), pp.1083-1094.

Kunwar, M., 2019. Artificial intelligence in finance: Understanding how automation and machine learning is transforming the financial industry.

Lahmiri, S. and Bekiros, S., 2019. Can machine learning approaches predict corporate bankruptcy? Evidence from a qualitative experimental design. *Quantitative Finance*, *19*(9), pp.1569-1577.

Li, J.-P., Mirza, N., Rahat, B. and Xiong, D. (2020). Machine learning and credit ratings prediction in the age of fourth industrial revolution. *Technological Forecasting and Social Change*, 161, p.120309.

Liu, Y., Mu, Y., Chen, K., Li, Y. and Guo, J., 2020. Daily activity feature selection in smart homes based on pearson correlation coefficient. *Neural processing letters*, *51*, pp.1771-1787.

Luo, C., Wu, D. and Wu, D., 2017. A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, *65*, pp.465-470.

Mahalakshmi, V., Kulkarni, N., Kumar, K.P., Kumar, K.S., Sree, D.N. and Durga, S., 2022. The role of implementing Artificial Intelligence and Machine Learning technologies in the financial services Industry for creating competitive intelligence. *Materials Today: Proceedings*, *56*, pp.2252-2255.

Maharana, K., Mondal, S. and Nemade, B., 2022. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), pp.91-99.

Melnikovas, A., 2018. Towards an Explicit Research Methodology: Adapting Research Onion Model for Futures Studies. *Journal of futures Studies*, *23*(2).

Mhlanga, D., 2020. Industry 4.0 in finance: the impact of artificial intelligence (ai) on digital financial inclusion. *International Journal of Financial Studies*, *8*(3), p.45.

Mi, X., Zou, B., Zou, F. and Hu, J., 2021. Permutation-based identification of important biomarkers for complex diseases via machine learning models. *Nature communications*, *12*(1), p.3008.

Mienye, I.D. and Sun, Y., 2022. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, *10*, pp.99129-99149.

Moscato, V., Picariello, A. and Sperlí, G., 2021. A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, *165*, p.113986.

Moşteanu, N.R., 2023, June. AI-driven transformation in the financial industry: Navigating change for sustainability. In *Proceedings of the 31st RSEP International Conference on Economics, Finance and Business* (pp. 22-23).

Muñoz-Cancino, R., Bravo, C., Ríos, S.A. and Graña, M., 2023. On the dynamics of credit history and social interaction features, and their impact on creditworthiness assessment performance. *Expert Systems with Applications*, *218*, p.119599.

Naili, M. and Lahrichi, Y., 2022. The determinants of banks' credit risk: Review of the literature and future research agenda. *International Journal of Finance & Economics*, *27*(1), pp.334-360.

Nasir, I.M., Khan, M.A., Yasmin, M., Shah, J.H., Gabryel, M., Scherer, R. and Damaševičius, R., 2020. Pearson correlation-based feature selection for document classification using balanced training. *Sensors*, *20*(23), p.6793.

Pallathadka, H., Mustafa, M., Sanchez, D.T., Sajja, G.S., Gour, S. and Naved, M., 2023. Impact of machine learning on management, healthcare and agriculture. *Materials Today: Proceedings*, *80*, pp.2803-2806.

Pattnaik, D., Ray, S. and Raman, R., 2024. Applications of artificial intelligence and machine learning in the financial services industry: A bibliometric review. *Heliyon*.

Paul, S., 2019. Use of Blockchain and Aritificial Intelligence to Promote Financial Inclusion in India Smita Miglani Indian Council for Research on International Economic Relations.

Plotnikova, V., Dumas, M. and Milani, F.P., 2022. Applying the CRISP-DM data mining process in the financial services industry: Elicitation of adaptation requirements. *Data & knowledge engineering*, *139*, p.102013.

Quang, L.T., Baek, B.H., Yoon, W., Kim, S.K. and Park, I., 2024. Comparison of Normalization Techniques for Radiomics Features From Magnetic Resonance Imaging in Predicting Histologic Grade of Meningiomas. *Investigative Magnetic Resonance Imaging*, *28*(2), pp.61-67.

Rajbahadur, G.K., Wang, S., Oliva, G.A., Kamei, Y. and Hassan, A.E., 2021. The impact of feature importance methods on the interpretation of defect classifiers. *IEEE Transactions on Software Engineering*, *48*(7), pp.2245-2261.

Rincy, T.N. and Gupta, R., 2020, February. Ensemble learning techniques and its efficiency in machine learning: A survey. In *2nd international conference on data, engineering and applications (IDEA)* (pp. 1-6). IEEE.

Rindfleisch, A., Malter, A.J., Ganesan, S. and Moorman, C., 2008. Cross-sectional versus longitudinal survey research: Concepts, findings, and guidelines. *Journal of marketing research*, *45*(3), pp.261-279.

Runchi, Z., Liguo, X. and Qin, W., 2023. An ensemble credit scoring model based on logistic regression with heterogeneous balancing and weighting effects. *Expert Systems with Applications*, *212*, p.118732.

Saltz, J.S., 2021, December. CRISP-DM for data science: strengths, weaknesses and potential next steps. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 2337-2344). IEEE.

Sami, O., Elsheikh, Y. and Almasalha, F. (2021). The Role of Data Pre-processing Techniques in Improving Machine Learning Accuracy for Predicting Coronary Heart Disease. *International Journal of Advanced Computer Science and Applications*, 12(6).

Schröer, C., Kruse, F. and Gómez, J.M., 2021. A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*, pp.526-534.

Shah, K., Patel, H., Sanghvi, D. and Shah, M., 2020. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, *5*(1), p.12.

Shearer, C., 2000. The CRISP-DM model: the new blueprint for data mining. *Journal of data warehousing*, *5*(4), pp.13-22.

Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P. and Homayouni, S., 2020. Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, pp.6308-6325.

Song, Y.X., Yang, X.D., Luo, Y.G., Ouyang, C.L., Yu, Y., Ma, Y.L., Li, H., Lou, J.S., Liu, Y.H., Chen, Y.Q. and Cao, J.B., 2023. Comparison of logistic regression and machine learning methods for predicting postoperative delirium in elderly patients: a retrospective study. *CNS Neuroscience & Therapeutics*, *29*(1), pp.158-167.

Tewari, N., 2023. Artificial Intelligence in Finance and Industry: Opportunities and Challenges. *Decision Strategies and Artificial Intelligence Navigating the Business Landscape. https://doi. org/10.59646/edbookc5/009*.

Tran, C.S., Nicolau, D., Nayak, R. and Verhoeven, P., 2021. Modeling Credit Risk: A Category Theory Perspective. *Journal of Risk and Financial Management*, 14(7), p.298.

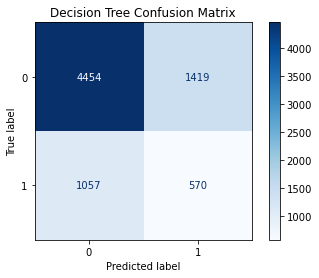
Vassallo, D., Vella, V. and Ellul, J., 2021. Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies. *SN Computer Science*, *2*(3), p.143.

Wójcicka-Wójtowicz, A., 2020. How to include experts’ imprecision in credit risk assessment. In *13th International Scientific Conference Analysis of International Relations* (pp. 234-247).

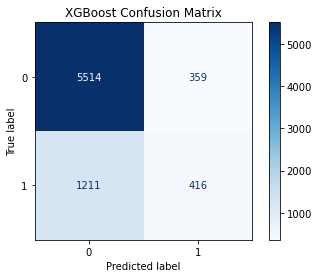
Yang, Q., Kang, Q., Huang, Q., Cui, Z., Bai, Y. and Wei, H., 2021, June. Linear correlation analysis of ammunition storage environment based on Pearson correlation analysis. In *Journal of physics: Conference series* (Vol. 1948, No. 1, p. 012064). IOP Publishing.

# Appendix I

1. Confusion matrix for decision tree model on test data



1. Confusion matrix for XGBoost model on test data



# Appendix II

#%%

# Import the package

import pandas as pd

from sklearn.preprocessing import MinMaxScaler, StandardScaler

from sklearn.model\_selection import train\_test\_split

# Replace 'file\_path' with the path to your CSV file

file\_path = r'C:/Users/aj01421/Downloads/Taiwan.xlsx'# Read the Excel file into a DataFrame

Raw\_DataFrame = pd.read\_excel(file\_path)

#%%

# Display/check the attribute names of the dataframe

print('\n\n\n-----------------------------------------------------------------------')

print('Attribute Names of the Dataframe')

print('-----------------------------------------------------------------------')

print(Raw\_DataFrame.columns)

print('----------------------------------------------------------------------- \n\n\n')

#%%

#Display the types and information about data

print('----------------------------------------------------------------------------')

print('Types and Information about DataFrame')

print('----------------------------------------------------------------------------')

print(Raw\_DataFrame.info())

print('---------------------------------------------------------------------------- \n\n\n')

#%%

# ----------------------------

# Data Dimension

# ----------------------------

# Shape of the Data

DataFrame\_Shape = Raw\_DataFrame.shape

# Number of row

DataFrame\_Row = DataFrame\_Shape[0]

# Number of column

DataFrame\_Col = DataFrame\_Shape[1]

#%%

# ---------------------------

# One-Hot Encoding of Categorical Data

# ---------------------------

# Shape of the Data

DataFrame\_Shape = Raw\_DataFrame.shape

# Number of rows

DataFrame\_Row = DataFrame\_Shape[0]

# Number of columns

DataFrame\_Col = DataFrame\_Shape[1]

# Initialize an empty DataFrame for numeric and one-hot encoded data

Numeric\_DataFrame = pd.DataFrame()

for Attribute in Raw\_DataFrame.columns:

if pd.api.types.is\_numeric\_dtype(Raw\_DataFrame[Attribute]):

print("-----------------------------------------------------------------------------------")

print("Data type of attribute - " + Attribute + " - is numeric, no conversion required")

Numeric\_DataFrame[Attribute] = Raw\_DataFrame[Attribute]

else:

print("-----------------------------------------------------------------------------------")

print("Data type of attribute - " + Attribute + " - is not numeric, conversion required")

# Apply one-hot encoding to this attribute and concatenate with the numeric dataframe

one\_hot\_encoded\_df = pd.get\_dummies(Raw\_DataFrame[Attribute], prefix=Attribute)

Numeric\_DataFrame = pd.concat([Numeric\_DataFrame, one\_hot\_encoded\_df], axis=1)

print("\tData type of attribute - " + Attribute + " - is now one-hot encoded")

#%%

# ----------------------------------------

# Missing value identification

# ----------------------------------------

# Missing Values Calculation

ms = Numeric\_DataFrame.isnull().sum()

# Calculate the percentage of missing values in each column

ms\_percentage = (Numeric\_DataFrame.isnull().sum()/(len(Numeric\_DataFrame)))\*100

# Combine the missing value information into one dataframe

Missing\_Data\_Info = pd.DataFrame({'Total Missings': ms, 'Percentage': ms\_percentage})

# Print them the missing value information on screen

print('----------------------------------------')

print('Missing Data Information')

print('----------------------------------------')

print(Missing\_Data\_Info)

print('----------------------------------------\n\n\n')

#%%

# -------------------

# Data Statistics

# -------------------

# Assuming Raw\_DataFrame is already defined

Data\_Stat = Numeric\_DataFrame.describe().T

# Calculate additional statistics

Data\_Stat['median'] = Numeric\_DataFrame.median()

Data\_Stat['mode'] = Numeric\_DataFrame.mode().iloc[0] # Taking the first mode

Data\_Stat['std'] = Numeric\_DataFrame.std()

# Display the updated statistics summary

print('-------------------------------------------------------------------------------')

print('Data Summary')

print('-------------------------------------------------------------------------------')

print(Data\_Stat)

print('-------------------------------------------------------------------------------\n\n\n')

# To include all data types including objects, categories, etc.

Data\_Stat\_All = Numeric\_DataFrame.describe(include='all').T

# Calculate additional statistics for all data types

# Handle numeric statistics separately

numeric\_cols = Numeric\_DataFrame.select\_dtypes(include='number').columns

# Calculate statistics only for numeric columns

Data\_Stat\_All.loc[numeric\_cols, 'median'] = Numeric\_DataFrame[numeric\_cols].median()

Data\_Stat\_All.loc[numeric\_cols, 'mode'] = Numeric\_DataFrame[numeric\_cols].mode().iloc[0]

Data\_Stat\_All.loc[numeric\_cols, 'std'] = Numeric\_DataFrame[numeric\_cols].std()

# Fill NaN for non-numeric data types with 'N/A'

Data\_Stat\_All.fillna('N/A', inplace=True)

# Display the updated statistics summary for all data types

print('-------------------------------------------------------------------------------')

print('Data Summary for All Data Types')

print('-------------------------------------------------------------------------------')

print(Data\_Stat\_All)

print('-------------------------------------------------------------------------------')

#%%

# Calculate and display the minimum and maximum values for each column

min\_values = Numeric\_DataFrame.min()

max\_values = Numeric\_DataFrame.max()

# Combine the results into a single DataFrame for easy display

min\_max\_df = pd.DataFrame({

'Minimum': min\_values,

'Maximum': max\_values

})

# Calculate the standard deviation for each column

std\_values = Numeric\_DataFrame.std(numeric\_only=True)

# Display the standard deviation values

std\_df = pd.DataFrame({'Standard Deviation': std\_values})

# Display the minimum and maximum values

print('-------------------------------------------------------------------------------')

print('Minimum and Maximum Values for Each Column')

print('-------------------------------------------------------------------------------')

print(min\_max\_df)

print('-------------------------------------------------------------------------------')

print('Standard Deviation for Each Column')

print(std\_df)

#%%

# ========================================

# EDA Univariate Analysis

# ========================================

#Correlation

import seaborn as sns

import matplotlib.pyplot as plt

# Select the target variable

target\_variable = 'default.payment.next.month'

# Calculate the correlation of the target variable with other variables

correlation\_with\_target = Numeric\_DataFrame.corr()[target\_variable].drop(target\_variable)

# Sort the correlations in descending order

correlation\_with\_target\_sorted = correlation\_with\_target.sort\_values(ascending=False)

# Plot the correlation of the target variable with other variables

plt.figure(figsize=(10, 6))

sns.barplot(x=correlation\_with\_target\_sorted.values, y=correlation\_with\_target\_sorted.index, palette='coolwarm')

plt.title('Correlation of {} with Other Variables'.format(target\_variable))

plt.xlabel('Correlation Coefficient')

plt.ylabel('Variable')

plt.show()

#%%

# ----------------------------------------

# Plotting the histogram of attributes

# ----------------------------------------

#histograms of PAY\_ variable

# Import the package

columns = Numeric\_DataFrame.columns

# Set the number of rows and columns for subplots

num\_rows = 2

num\_cols = 3

# Create a new figure and set its size

plt.figure(figsize=(20, 20))

# Exclude specified columns

columns\_to\_exclude = ['ID','LIMIT\_BAL','SEX', 'EDUCATION', 'MARRIAGE','AGE','BILL\_AMT1','BILL\_AMT2','BILL\_AMT3','BILL\_AMT4','BILL\_AMT5','BILL\_AMT6','PAY\_AMT1','PAY\_AMT2','PAY\_AMT3','PAY\_AMT4','PAY\_AMT5','PAY\_AMT6', 'default.payment.next.month']

columns\_to\_plot = [col for col in Numeric\_DataFrame.columns if col not in columns\_to\_exclude]

for i, column in enumerate(columns\_to\_plot):

plt.subplot(num\_rows, num\_cols, i + 1)

plt.hist(Numeric\_DataFrame[column].dropna(), bins=30, edgecolor='black')

plt.title(column)

# Add a title to the whole figure

plt.suptitle('Histograms of PAY Variables', fontsize=20)

# Adjust layout

plt.tight\_layout()

# Show the plot

plt.show()

#%%

#histograms of BILL\_AMOUNT variable

# Set the number of rows and columns for subplots

num\_rows = 2

num\_cols = 3

# Create a new figure and set its size

plt.figure(figsize=(20, 20))

# Exclude specified columns

columns\_to\_exclude = ['ID','LIMIT\_BAL','SEX', 'EDUCATION', 'MARRIAGE','AGE','PAY\_0','PAY\_2','PAY\_3','PAY\_4','PAY\_5','PAY\_6','PAY\_AMT1','PAY\_AMT2','PAY\_AMT3','PAY\_AMT4','PAY\_AMT5','PAY\_AMT6', 'default.payment.next.month']

columns\_to\_plot = [col for col in Numeric\_DataFrame.columns if col not in columns\_to\_exclude]

for i, column in enumerate(columns\_to\_plot):

plt.subplot(num\_rows, num\_cols, i + 1)

plt.hist(Numeric\_DataFrame[column].dropna(), bins=30, edgecolor='black')

plt.title(column)

# Add a title to the whole figure

plt.suptitle('Histograms of BILL\_AMOUNT Variables', fontsize=20)

# Adjust layout

plt.tight\_layout()

# Show the plot

plt.show()

#%%

#histograms of PAY\_AMOUNT variable

# Set the number of rows and columns for subplots

num\_rows = 2

num\_cols = 3

# Create a new figure and set its size

plt.figure(figsize=(20, 20))

# Exclude specified columns

columns\_to\_exclude = ['ID','LIMIT\_BAL','SEX', 'EDUCATION', 'MARRIAGE','AGE','PAY\_0','PAY\_2','PAY\_3','PAY\_4','PAY\_5','PAY\_6','BILL\_AMT1','BILL\_AMT2','BILL\_AMT3','BILL\_AMT4','BILL\_AMT5','BILL\_AMT6', 'default.payment.next.month']

columns\_to\_plot = [col for col in Numeric\_DataFrame.columns if col not in columns\_to\_exclude]

for i, column in enumerate(columns\_to\_plot):

plt.subplot(num\_rows, num\_cols, i + 1)

plt.hist(Numeric\_DataFrame[column].dropna(), bins=30, edgecolor='black')

plt.title(column)

# Add a title to the whole figure

plt.suptitle('Histograms of PAY\_AMOUNT Variables', fontsize=20)

# Adjust layout

plt.tight\_layout()

# Show the plot

plt.show()

#%%

#histograms of LIMIT\_BAL variable

# Separate defaulters and non-defaulters

defaulters = Numeric\_DataFrame[Numeric\_DataFrame['default.payment.next.month'] == 1]

non\_defaulters = Numeric\_DataFrame[Numeric\_DataFrame['default.payment.next.month'] == 0]

plt.figure(figsize=(12, 7))

# Plot the histogram for all customers

plt.hist(Numeric\_DataFrame['LIMIT\_BAL'].dropna(), bins=30, edgecolor='black', alpha=0.6, label='All Customers')

# Plot the histogram for defaulters

plt.hist(defaulters['LIMIT\_BAL'].dropna(), bins=30, edgecolor='red', alpha=0.6, label='Defaulters')

# Adding labels and title

plt.title('Histogram of LIMIT\_BAL Variable with Defaulters')

plt.xlabel('LIMIT\_BAL')

plt.ylabel('Frequency')

plt.xticks(ticks=range(0, int(Numeric\_DataFrame['LIMIT\_BAL'].max()) + 1, 100000), labels=range(0, int(Numeric\_DataFrame['LIMIT\_BAL'].max()) + 1, 100000))

plt.legend()

# Show the plot

plt.show()

#%%

# ----------------------------------------

# Plotting the bargraph

# ----------------------------------------

#Bar graph of SEX variable

# Rename the class for the SEX column

Numeric\_DataFrame['SEX'] = Numeric\_DataFrame['SEX'].map({1: 'Male', 2: 'Female'})

# Calculate the counts of each category

total\_counts = Numeric\_DataFrame['SEX'].value\_counts()

# Calculate the counts of defaulters in each category

defaulter\_counts = Numeric\_DataFrame[Numeric\_DataFrame['default.payment.next.month'] == 1]['SEX'].value\_counts()

# Calculate non-defaulter counts

non\_defaulter\_counts = total\_counts - defaulter\_counts

# Plot the bar graph with same colors for both genders

plt.figure(figsize=(10, 6))

# Plot defaulters (bottom part of the bars)

defaulter\_counts.plot(kind='bar', color='red', edgecolor='black', label='Defaulters')

# Plot non-defaulters (top part of the bars)

non\_defaulter\_counts.plot(kind='bar', color='blue', edgecolor='black', label='Non-Defaulters', bottom=defaulter\_counts)

# Adding labels and title

plt.title('Frequency of Data Points by SEX with Defaulters at the Bottom')

plt.xlabel('SEX')

plt.ylabel('Frequency')

plt.xticks(rotation=0)

# Adding legend

plt.legend()

# Show the plot

plt.show()

#%%

#bar graph of marital status

# Rename the categories

marital\_status = {0: 'Unknown', 1: 'Married', 2: 'Single', 3: 'Others'}

Numeric\_DataFrame['MARRIAGE'] = Numeric\_DataFrame['MARRIAGE'].map(marital\_status)

# Count the frequency of each category

total\_counts = Numeric\_DataFrame['MARRIAGE'].value\_counts()

# Count the frequency of defaulters in each category

defaulter\_counts = Numeric\_DataFrame[Numeric\_DataFrame['default.payment.next.month'] == 1]['MARRIAGE'].value\_counts()

# Calculate non-defaulter counts

non\_defaulter\_counts = total\_counts - defaulter\_counts

# Plot the bar graph with defaulters at the bottom

plt.figure(figsize=(10, 6))

# Plot defaulters (bottom part of the bars)

defaulter\_counts.plot(kind='bar', color='red', edgecolor='black', label='Defaulters')

# Plot non-defaulters (top part of the bars)

non\_defaulter\_counts.plot(kind='bar', color='blue', edgecolor='black', label='Non-Defaulters', bottom=defaulter\_counts)

# Adding labels and title

plt.title('Frequency of Data Points by Marital Status with Defaulters')

plt.xlabel('Marital Status')

plt.ylabel('Frequency')

plt.xticks(rotation=0)

# Adding legend

plt.legend()

# Show the plot

plt.show()

#%%

# Rename the categories and merge 0, 5, 6 as 'Unknown'

education\_status = {1: 'Graduate School', 2: 'University', 3: 'High School', 4: 'Others', 0: 'Unknown', 5: 'Unknown', 6: 'Unknown'}

Numeric\_DataFrame['EDUCATION'] = Numeric\_DataFrame['EDUCATION'].map(education\_status)

# Count the frequency of each category

total\_counts = Numeric\_DataFrame['EDUCATION'].value\_counts()

# Count the frequency of defaulters in each category

defaulter\_counts = Numeric\_DataFrame[Numeric\_DataFrame['default.payment.next.month'] == 1]['EDUCATION'].value\_counts()

# Calculate non-defaulter counts

non\_defaulter\_counts = total\_counts - defaulter\_counts

# Plot the bar graph with defaulters at the bottom

plt.figure(figsize=(10, 6))

# Plot defaulters (bottom part of the bars)

defaulter\_counts.plot(kind='bar', color='red', edgecolor='black', label='Defaulters')

# Plot non-defaulters (top part of the bars)

non\_defaulter\_counts.plot(kind='bar', color='skyblue', edgecolor='black', label='Non-Defaulters', bottom=defaulter\_counts)

# Adding labels and title

plt.title('Frequency of Data Points by Education Level with Defaulters')

plt.xlabel('Education Level')

plt.ylabel('Frequency')

plt.xticks(rotation=0)

# Adding legend

plt.legend()

# Show the plot

plt.show()

#%%

#Histogram for AGE variable

# Plot the histogram

plt.figure(figsize=(10, 6))

plt.hist(Numeric\_DataFrame['AGE'], bins=10, color='skyblue', edgecolor='black')

plt.title('Distribution of Age')

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.show()

#%%

#Bar graph of MARRIAGE-LIMIT\_BAL

# Group data by Marriage and calculate the average Limit Balance

grouped\_data = Numeric\_DataFrame.groupby('MARRIAGE')['LIMIT\_BAL'].mean().reset\_index()

# Plotting the bar chart

plt.figure(figsize=(10, 6))

plt.bar(grouped\_data['MARRIAGE'], grouped\_data['LIMIT\_BAL'], color='skyblue')

plt.xlabel('Marriage')

plt.ylabel('Limit Balance')

plt.title('Average Limit Balance by Marital Status')

plt.xticks(rotation=45)

plt.show()

#%%

#Bar graph of EDUCATION-LIMIT\_BAL

# Group data by Education and calculate the average Limit Balance

grouped\_data = Numeric\_DataFrame.groupby('EDUCATION')['LIMIT\_BAL'].mean().reset\_index()

# Plotting the bar chart

plt.figure(figsize=(10, 6))

plt.bar(grouped\_data['EDUCATION'], grouped\_data['LIMIT\_BAL'], color='skyblue')

plt.xlabel('Education')

plt.ylabel('Limit Balance')

plt.title('Average Limit Balance by Education')

plt.xticks(rotation=45)

plt.show()

#%%

#Piechart of target class distribution

# Calculate the percentage of each category

default\_counts = Numeric\_DataFrame['default.payment.next.month'].value\_counts(normalize=True) \* 100

default\_counts = default\_counts.rename(index={0: 'No', 1: 'Yes'})

# Plotting the pie chart

plt.figure(figsize=(8, 8))

plt.pie(default\_counts, labels=default\_counts.index, autopct='%1.1f%%', startangle=140, colors=['lightblue', 'lightcoral'])

plt.title('Target Class Distribution')

plt.legend(default\_counts.index, title="Default Payment", loc="upper right")

plt.show()

#%%

#drop ID column

Numeric\_DataFrame = Numeric\_DataFrame.drop(columns='ID')

#Train and Test data split

#Randomly select 7500 entries from the dataset for testing

X\_test = Numeric\_DataFrame.sample(n=7500, random\_state=42)

# Remove the randomly selected entries from the original dataset

X\_train = Numeric\_DataFrame.drop(X\_test.index)

#%%

#correct the index value of test data

X\_test.reset\_index(drop=True, inplace=True)

#%%

# ========================================

# Class Distribution of training data

# ========================================

print("-----------------------------------------------")

print("Class Distribution of training data")

print("-----------------------------------------------")

class\_distribution = X\_train['default.payment.next.month'].value\_counts(normalize=True) \* 100

print("Class - 0: {:.3f}%".format(class\_distribution[0]))

print("Class - 1: {:.3f}%".format(class\_distribution[1]))

#%%

# Import the package

import numpy as np

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from xgboost import XGBClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, f1\_score,classification\_report

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_auc\_score

# Ensure all columns are numeric (this is more for safety, as the columns appear to be numeric already)

numeric\_df = Numeric\_DataFrame.apply(pd.to\_numeric, errors='coerce')

# Separate features and target variable from the DataFrame

Features\_train = X\_train.drop(['default.payment.next.month'], axis=1) # Dropping the target variable to keep only features

Labels\_train = X\_train['default.payment.next.month'] # Selecting only the target variable

# Check for any NaN values that might have resulted from coercion

nan\_check = Features\_train.isna().sum()

# Print the results

print("NaN values in each column after coercion:")

print(nan\_check)

# Replace NaN values with a placeholder (-999)

Features\_train.replace(np.nan, -999, inplace=True)

# #-------------------------------

# # Feature normalisation

# #-------------------------------

# # Initialize the MinMaxScaler

scaler = MinMaxScaler()

# # Fit and transform the DataFrame

Features\_train = pd.DataFrame(scaler.fit\_transform(Features\_train), columns=Features\_train.columns)

#%%

# Random Forest

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(Features\_train, Labels\_train)

# XGBoost

xgb\_model = XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss', random\_state=42)

xgb\_model.fit(Features\_train, Labels\_train)

# Logistic Regression

lr\_model = LogisticRegression(random\_state=42, max\_iter=1000)

lr\_model.fit(Features\_train, Labels\_train)

# Decision Tree

dt\_model = DecisionTreeClassifier(random\_state=42)

dt\_model.fit(Features\_train, Labels\_train)

#%%

# Evaluate Models on Training Data

print("\nTraining Data Evaluation:\n")

# Evaluate Models on Training Data

train\_results = []

for name, model in zip(['Random Forest', 'XGBoost', 'Logistic Regression', 'Decision Tree'], [rf\_model, xgb\_model, lr\_model, dt\_model]):

y\_train\_pred = model.predict(Features\_train)

accuracy = accuracy\_score(Labels\_train, y\_train\_pred)

precision = precision\_score(Labels\_train, y\_train\_pred)

recall = recall\_score(Labels\_train, y\_train\_pred)

f1 = f1\_score(Labels\_train, y\_train\_pred)

roc\_auc = roc\_auc\_score(Labels\_train, model.predict\_proba(Features\_train)[:, 1])

train\_results.append([name, accuracy, precision, recall, f1, roc\_auc])

# Create a DataFrame for training results

train\_results\_df = pd.DataFrame(train\_results, columns=['Model', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'ROC-AUC'])

print("\nTraining Data Evaluation:\n")

print(train\_results\_df)

#%%

# Evaluate Models on Test Data

# Separate features and target variable from the DataFrame

Features\_test = X\_test.drop(['default.payment.next.month'], axis=1) # Dropping the target variable to keep only features

Labels\_test = X\_test['default.payment.next.month'] # Selecting only the target variable

# #-------------------------------

# # Feature normalisation

# #-------------------------------

# # Initialize the MinMaxScaler

scaler = MinMaxScaler()

# # Fit and transform the DataFrame

Features\_test = pd.DataFrame(scaler.fit\_transform(Features\_test), columns=Features\_test.columns)

#%%

#imports for confusion matrix

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

test\_results = []

# Loop through each model

for name, model in zip(['Random Forest', 'XGBoost', 'Logistic Regression', 'Decision Tree'],

[rf\_model, xgb\_model, lr\_model, dt\_model]):

# Predict on the test data

y\_test\_pred = model.predict(Features\_test)

# Calculate evaluation metrics

accuracy = accuracy\_score(Labels\_test, y\_test\_pred)

precision = precision\_score(Labels\_test, y\_test\_pred)

recall = recall\_score(Labels\_test, y\_test\_pred)

f1 = f1\_score(Labels\_test, y\_test\_pred)

roc\_auc = roc\_auc\_score(Labels\_test, model.predict\_proba(Features\_test)[:, 1])

# Append the results to the test\_results list

test\_results.append([name, accuracy, precision, recall, f1, roc\_auc])

# Calculate and plot the confusion matrix

cm = confusion\_matrix(Labels\_test, y\_test\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm)

disp.plot(cmap=plt.cm.Blues) # You can customize the colormap if desired

plt.title(f'{name} Confusion Matrix')

plt.show()

# Create a DataFrame for test results

test\_results\_df = pd.DataFrame(test\_results, columns=['Model', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'ROC-AUC'])

print("\nTest Data Evaluation:\n")

print(test\_results\_df)

#%%

#Develop Hybrid Model: Averaging predictions

#evaluation on traindata

y\_pred\_rf\_train = rf\_model.predict\_proba(Features\_train)[:, 1]

y\_pred\_xgb\_train = xgb\_model.predict\_proba(Features\_train)[:, 1]

y\_pred\_lr\_train = lr\_model.predict\_proba(Features\_train)[:, 1]

y\_pred\_dt\_train = dt\_model.predict\_proba(Features\_train)[:, 1]

# Averaging predictions for training data

y\_pred\_hybrid\_train = (y\_pred\_rf\_train + y\_pred\_xgb\_train + y\_pred\_lr\_train + y\_pred\_dt\_train) / 4

y\_pred\_hybrid\_label\_train = (y\_pred\_hybrid\_train >= 0.5).astype(int)

# Hybrid Model Evaluation on Training Data

hybrid\_accuracy\_train = accuracy\_score(Labels\_train, y\_pred\_hybrid\_label\_train)

hybrid\_precision\_train = precision\_score(Labels\_train, y\_pred\_hybrid\_label\_train)

hybrid\_recall\_train = recall\_score(Labels\_train, y\_pred\_hybrid\_label\_train)

hybrid\_f1\_train = f1\_score(Labels\_train, y\_pred\_hybrid\_label\_train)

hybrid\_roc\_auc\_train = roc\_auc\_score(Labels\_train, y\_pred\_hybrid\_train)

# Display Hybrid Model Evaluation on Training Data

hybrid\_results\_train\_df = pd.DataFrame([['Hybrid Model (Train)', hybrid\_accuracy\_train, hybrid\_precision\_train, hybrid\_recall\_train, hybrid\_f1\_train, hybrid\_roc\_auc\_train]],

columns=['Model', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'ROC-AUC'])

print("\nHybrid Model Evaluation on Training Data:\n")

print(hybrid\_results\_train\_df)

#%%

#Develop Hybrid Model: Averaging predictions

#evaluation on testdata

y\_pred\_rf = rf\_model.predict\_proba(Features\_test)[:, 1]

y\_pred\_xgb = xgb\_model.predict\_proba(Features\_test)[:, 1]

y\_pred\_lr = lr\_model.predict\_proba(Features\_test)[:, 1]

y\_pred\_dt = dt\_model.predict\_proba(Features\_test)[:, 1]

# Averaging predictions

y\_pred\_hybrid = (y\_pred\_rf + y\_pred\_xgb + y\_pred\_lr + y\_pred\_dt) / 4

y\_pred\_hybrid\_label = (y\_pred\_hybrid >= 0.5).astype(int)

# Hybrid Model Evaluation

hybrid\_accuracy = accuracy\_score(Labels\_test, y\_pred\_hybrid\_label)

hybrid\_precision = precision\_score(Labels\_test, y\_pred\_hybrid\_label)

hybrid\_recall = recall\_score(Labels\_test, y\_pred\_hybrid\_label)

hybrid\_f1 = f1\_score(Labels\_test, y\_pred\_hybrid\_label)

hybrid\_roc\_auc = roc\_auc\_score(Labels\_test, y\_pred\_hybrid)

# Display Hybrid Model Evaluation

hybrid\_results\_df = pd.DataFrame([['Hybrid Model (Test)', hybrid\_accuracy, hybrid\_precision, hybrid\_recall, hybrid\_f1, hybrid\_roc\_auc]],columns=['Model', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'ROC-AUC'])

print("\nHybrid Model Evaluation on Test Data:\n")

print(hybrid\_results\_df)

# Calculate the confusion matrix for testdata on hybrid model

hybrid\_cm = confusion\_matrix(Labels\_test, y\_pred\_hybrid\_label)

ConfusionMatrixDisplay(confusion\_matrix=hybrid\_cm).plot()

plt.title('Hybrid Model Confusion Matrix')

plt.show()

#%%

from sklearn.inspection import permutation\_importance

# For Random Forest model

rf\_importance = permutation\_importance(rf\_model, Features\_test, Labels\_test, n\_repeats=10, random\_state=42)

rf\_sorted\_idx = rf\_importance.importances\_mean.argsort()

# Print out the importance

print("Random Forest Permutation Feature Importance:")

for i in rf\_sorted\_idx[::-1]: # Reverse order for descending importance

print(f"{X\_test.columns[i]}: {rf\_importance.importances\_mean[i]:.4f}")

#%%

# Sort the features and their importance in descending order

sorted\_importance = rf\_importance.importances\_mean[rf\_sorted\_idx[::-1]]

sorted\_features = X\_test.columns[rf\_sorted\_idx[::-1]]

# Plot the bar chart

plt.figure(figsize=(10, 6))

plt.barh(sorted\_features, sorted\_importance, color='blue')

plt.xlabel('Importance')

plt.ylabel('Feature')

plt.title('Random Forest Permutation Feature Importance')

plt.gca().invert\_yaxis() # To display the highest importance at the top

plt.show()

#%%

# For XGBoost model

xgb\_importance = permutation\_importance(xgb\_model, Features\_test, Labels\_test, n\_repeats=10, random\_state=42)

xgb\_sorted\_idx = xgb\_importance.importances\_mean.argsort()

# Print out the importance

print("XGBoost model Permutation Feature Importance:")

for i in xgb\_sorted\_idx[::-1]: # Reverse order for descending importance

print(f"{X\_test.columns[i]}: {xgb\_importance.importances\_mean[i]:.4f}")

#%%

# Sort the features and their importance in descending order

sorted\_importance = xgb\_importance.importances\_mean[xgb\_sorted\_idx[::-1]]

sorted\_features = X\_test.columns[xgb\_sorted\_idx[::-1]]

# Plot the bar chart

plt.figure(figsize=(10, 6))

plt.barh(sorted\_features, sorted\_importance, color='blue')

plt.xlabel('Importance')

plt.ylabel('Feature')

plt.title('XGBoost Model Permutation Feature Importance')

plt.gca().invert\_yaxis() # To display the highest importance at the top

plt.show()

#%%