LOAN DEFAULT PREDICTION USING MACHINE LEARNING APPROACHES

By

Motunrayo Aduragbemi ADEREMI

Supervisor: Mr Joseph Annan

Contents

ABSTRACT

Loan default prediction using machine learning approaches.

by

Motunrayo Aderemi A.

Due to the evolving financial sector in recent years and the growing trend of loan applications, a substantial portion of the population seeks bank loans. However, a significant concern confronted by the banking sector in this constantly evolving economy is the rising prevalence of loan defaults. Banking authorities are encountering heightened challenges in accurately appraising loan applications and effectively addressing the hazards associated with individuals failing to meet loan obligations. These loan applications undergo scrutiny and approval based on a diverse range of criteria, including Lifetime Value, income levels, employment statuses, and risk evaluation mechanisms.

The aim of this research was to explore different machine learning algorithms with the purpose of creating a consumer loan approval system capable of accurately forecasting the likelihood of an applicant's default before the approval process. The primary aim of this research is two-fold: (i) To present a comprehensive visualization of loan performance using Power BI dashboards, and (ii) To conduct an extensive and comparative analysis of various machine learning algorithms for predicting loan defaults. The study employs authentic borrower loan data from Lending Club to train and assess multiple machines learning models, including Decision Tree, Logistic Regression, Support Vector Machines, k-Nearest Neighbors, and Artificial Neural Networks, while employing a diverse set of evaluation metrics.

Artificial Neural Network (ANN) yielded an accuracy rate of 69.18% and AUC of 65.80%. Consequently, the ANN model emerges as a more favourable choice compared to the other algorithms for predicting potential loan defaulters among consumers. The findings suggest that financial institutions should prioritize applicants with higher Lifetime Value and Income during the pre-approval stage.

Keywords: personal Loan, Fully Paid, Default, Machine Learning, Visualization, Algorithm

CHAPTER ONE

1.1 Introduction

With the constant evolution of the global economy and the growing competition in the financial sector, individuals worldwide increasingly rely on banks to obtain loans for diverse purposes. Financial loans are an important source of capital for both individuals and businesses, not only in upcoming economies but also in well-established capital markets. The growth of lending activities undertaken by financial institutions and banks is considered a pivotal factor that impacts a nation's inflation rate and interest rates. This, in turn, plays a significant role in driving economic growth and serves as a reflection of the broader economic status of the country. (Wyman, 2015). These loans serve as a means to overcome financial limitations and accomplish personal goals. Given the dynamic nature of the economy and the prevalent competition in the financial industry, seeking loans has become an inevitable practice (Aslam *et al., 2019*). Furthermore, both small and large banking institutions rely on lending activities to generate profits and effectively manage their operations, particularly during periods of financial constraints. Loans represent a significant source of income for the banking sector, but they also pose substantial financial risks. A substantial portion of a bank's assets stems from the interest earned on loans disbursed (Aslam *et al., 2019*).

In previous times, the primary approach to forecast loan default was through human screening. This method invariably consumed substantial amounts of time and required a large workforce, often leading to challenges due to the sheer volume of data involved. However, the utilization of machine learning for predicting the probability of loan default is a time-efficient and labour-saving alternative. This approach significantly enhances effectiveness and accuracy in loan default prediction (Jozef, 2007). The rise in popularity of electronic transactions, online and mobile banking, and the advancements in third-party mobile and online payment platforms have led to an increased collection of customer data by banks through various internal and external channels. This extensive data serves as the underlying basis for utilizing big data analytics to predict loan default.

The research study will involve the analysis of an extensive dataset that includes historical loan records, encompassing borrower information, loan characteristics, and loan outcomes (default or non-default). This dataset will serve as the fundamental groundwork for training and evaluating various machine learning models. This study aims to achieve several objectives in diverse settings, including Developing machine learning models capable of accurately predicting the default rate of loan applicants, incorporating models such as K-Nearest Neighbors, Logistic Regression, Artificial Neural Networks, Decision Trees, and Support Vector Machines, conducting a comparative analysis of these models using a range of evaluation metrics to determine the most suitable model, employing

exploratory data analysis techniques to gain insights into the dataset and explore internal correlations and addressing class imbalance issues during historical data training and testing to enhance model performance.

1.1.1 Research Area (a brief literature review)

The literature review focuses on exploring various machine learning methods used for predicting loan defaults. It discusses the strengths, weaknesses, and assessment metrics of these approaches. The aim is to provide insights into the effectiveness of different algorithms and guide future research in this field. Additionally, the study highlights the crucial steps taken to refine loan datasets for analysis, including data cleansing, feature selection, factor analysis, and handling imbalanced data. It emphasizes the role of data preprocessing in shaping the performance of machine learning models in a classification context.

Researchers at Vignan's Nirula Institute of Technology & Science for Women in India developed a loan approval prediction model using R-Studio. Tejaswini et al. (2020) found that their study demonstrated the Decision Tree model's superior performance in comparison to other supervised learning methods like Logistic Regression and Random Forest. This comprehensive analysis highlighted the effectiveness of the Decision Tree model for loan approval prediction at the mentioned institute.

Personal Loan: A loan entails a financial arrangement where one party lends funds to another, expecting repayment as mutually agreed (Amin et al., 2015). These loans can fall under categories such as Commercial loans or Personal loans. Personal loans, in particular, can be classified as either secured, involving collateral (Hamid & Ahmed, 2016), or Unsecured, not requiring collateral.

When a borrower fails to meet the contractual obligation of repaying a loan as per the agreed terms, it results in a personal loan default. To comprehend the potential influences on such defaults, various predictor variables come into play. These variables encompass the Debt-Income Ratio, Credit Score, Debt-to-Income Ratio, Lifetime Value, and Creditworthiness.

- Debt-Income Ratio: This ratio examines the proportion of a borrower's total debt obligations in relation to their income. A higher ratio suggests a heavier debt burden relative to their earnings, potentially increasing the risk of loan default.
- Credit Score: Representing an individual's creditworthiness based on credit history, a credit score serves as a critical factor in assessing the likelihood of timely loan repayments. Lower credit scores indicate higher credit risk and could lead to an elevated chance of default.

- Debt-to-Income Ratio: Similar to the Debt-Income Ratio, this metric analyses the percentage of a borrower's debt compared to their income. A higher ratio indicates a more substantial debt load relative to their earnings, which may heighten the risk of loan default.
- Lifetime Value: Evaluating the potential long-term profitability of the borrower for the lender, lifetime value provides insights into the borrower's commitment to fulfilling financial obligations over time.
- Creditworthiness: This broader assessment considers various factors, including credit history, income stability, and overall financial reliability, to gauge a borrower's ability to repay debts. A higher creditworthiness indicates a lower risk of loan default.

By comprehensively analysing these predictor variables, lenders and financial institutions can make well-informed decisions to assess credit risk and mitigate the likelihood of loan defaults.

Machine Learning

Machine learning is defined as the field of study that empowers computers to learn and improve from data without being explicitly programmed. This is divided into Supervised, Unsupervised and Reinforcement machine learning.

Supervised machine learning utilizes labelled data to train algorithms, enabling accurate predictions on new data by recognizing patterns and relationships from examples.

Unsupervised machine learning trains algorithms on unlabelled data to unveil patterns and relationships without explicit guidance, useful for clustering, anomaly detection, and dimensionality reduction.

Reinforcement learning empowers agents to make decisions by interacting with an environment, receiving rewards or penalties based on actions. Its aim is to maximize cumulative rewards over time, applied in areas like gaming, robotics, and recommendations.

1.3 Aims

The primary goal of this research is to employ machine learning techniques to make predictions regarding the probability of customers defaulting on their loans.

1.3.2 Objectives

1. Literature Review

A literature review is crucial before practical work, involving comprehensive analysis of research papers to gain insights into relevant theories, methods, and the field's progress.

2. Methodology

Methodology is the structured process used to plan, execute, and control a project, outlining steps for achieving objectives and ensuring efficient management.

3. Implementation

This is the implementation phase where planned strategies are put into action through tasks and solutions to achieve project objectives.

4. Results interpretation and discussion

This step entails interpreting and contextualizing findings, understanding their relevance to project goals, discussing significance, and addressing limitations.

5. Conclusion

This encompasses a thorough evaluation of project performance, outcomes, findings, strengths, weaknesses, and limitations to arrive at a final informed decision or judgment.

Expected output of practical/investigative element

The practical and investigative aspect of this study will involve the comprehensive analysis of a dataset containing historical loan records. This dataset includes pertinent information about borrowers, loan attributes, and loan outcomes (whether they defaulted or not). This dataset forms the basis for training and evaluating a range of machine-learning models.

The study serves several purposes, including:

- Constructing machine learning models capable of accurately predicting loan default rates for applicants.
- Developing models such as K-Nearest Neighbors, Logistic Regression, Artificial Neural Networks, Decision Trees, and Support Vector Machines.
- Conducting a comparative analysis of these models to identify the most suitable one using various evaluation metrics.
- Employing exploratory data analysis techniques to understand the dataset and explore internal correlations.
- Addressing class imbalance issues during the training and testing of historical data to enhance model performance.

This research enhances loan default prediction by providing insights for informed decisions in financial institutions. The project introduces a machine learning-based early approval system to reduce errors, minimize loan defaults, and improve institution profitability.

Required Resources

Dataset: The analysed data comes from Kaggle, a popular platform known for diverse datasets used in data science and machine learning education. Kaggle is a collaborative hub where users share

datasets across domains, providing valuable resources for learning and practical skill development. It significantly contributes to the growth of data science and machine learning communities.

Software and Libraries: The study will use various software tools: R Studio for statistical analytics and Machine Learning, SQL for database access, Microsoft Excel for data manipulation, and Power BI for data visualization.

Pre-requisite Knowledge/Skills Required

With a background in Data Analysis and prior involvement in sales and marketing, I'm well-equipped to understand industry complexities and tackle statistical challenges. Proficient in data science tools like SQL, Power BI, Python, and R, I'm poised to handle complex data analysis, statistical modelling, and machine learning effectively, ensuring valuable insights and solutions for the study's objectives**.**

Nature and Sources of Data

The dataset is extensive and includes borrower details like income, gender, loan purpose, and term. Yet, it has multicollinearity and missing values. It holds 148,671 records and 34 variables. Due to incomplete data, diverse cleaning methods are needed. The financial data, gathered from the bank's 2019 customers, is suitable for classification analysis with binary classes.

CHAPTER TWO

2.0 Literature Review

2.1 Introduction

All over the world, the transaction volume of personal credit loans has been tremendous. Over the past few decades, there has been significant research interest in credit scoring techniques (Fay,2017). The banking sector generates and accumulates vast quantities of data each day, encompassing collateral details, transaction records, credit card information, risk profiles, loans, and customer data. This critical information forms the foundation for numerous decision-making processes within the banking environment (Xu *et al*.,2015). Researchers in this field have recognized the significance of incorporating these attributes to obtain valuable insights for assessing financial lending. The aim is to mitigate credit risk and support the retail banking system by leveraging these insights. (Arun *et al*.,2016)

A literature review involves thoroughly analysing and combining published research and scholarly articles pertaining to a specific topic or research question. Its purpose is to offer a comprehensive understanding of the existing literature, pinpoint areas of limited knowledge, and emphasize the significance of prior studies in relation to ongoing research. In this chapter, an examination of the theoretical literature review, empirical literature review, and conceptual framework is presented, focusing on the phenomenon of personal loan default.

2.2 Personal Loan

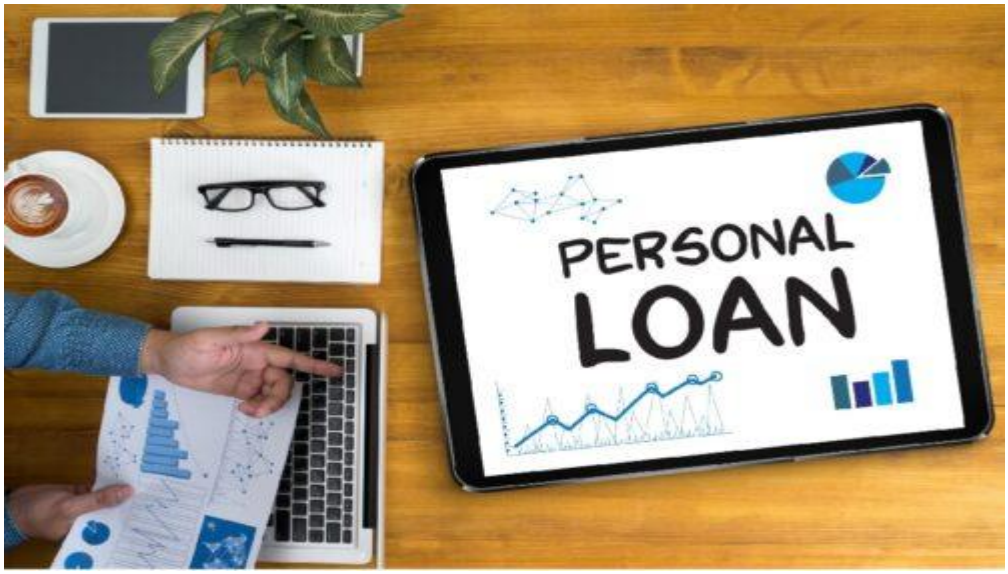


Fig 2.1: Diagram of personal loan (https://timebusinessnews.com/benefits-of-personal-loans/ **)**

A loan entails the transfer of funds from one party to another, under a mutual agreement for eventual repayment. The recipient of the funds, commonly known as the borrower, undertakes a financial responsibility and is generally required to pay interest as a form of compensation for utilizing the borrowed amount. (). This can be a **Commercial loan** or a **Personal loan** (Amin*et al.,2015).*

According to (Guttentag, 2007), Personal loans are a type of loan that can be utilized for various personal purposes, such as funding a wedding, covering vacation expenses, or acquiring consumer goods. With their versatile nature, personal loans cater to the diverse needs of individuals, making them a convenient financial solution. Personal loans, which are offered by banks, credit unions, and other lending institutions, are financial products that grant individuals a lump sum of money for personal purposes. These loans are either **Secure and Unsecured** (Amin*et al.,2015*). A secured loan is one that necessitates the provision of collateral as a prerequisite for borrowing (Hamid and Ahmed, 2016). For instance, you might secure a personal loan with assets like a savings account or a certificate of deposit, or with tangible possessions such as your car or boat. In the event of loan default, the lender could retain your collateral to settle the outstanding debt. On the other hand, an unsecured loan does not require any collateral for borrowing money. Both secured and unsecured personal loans can be offered to eligible borrowers by banks, credit unions, and online lenders (Li et al., 2018). Generally, banks perceive unsecured loans as riskier than secured ones due to the absence of collateral for recovery. Consequently, this could lead to higher interest rates on unsecured personal loans.

The flexibility of personal loans allows them to be utilized for a wide range of purposes, including debt consolidation, home renovations, medical expenses, and financing significant life events like weddings or vacations. (Vadya, 2017) The specific loan amount, interest rate, and repayment terms are determined by considering factors such as the borrower's creditworthiness, income level, and financial history.

2.2.1 Personal loan default

Loan default refers to the failure of a borrower to meet their contractual obligation of repaying a loan in accordance with the agreed-upon terms and conditions. It occurs when the borrower is unable or unwilling to make the necessary payments within the specified timeframe. Loan default carries substantial implications for both the borrower and the lender, making it a significant and concerning financial situation (Signoriello,1991). Prior to being considered in default, a borrower may enter a period of delinquency. Delinquency refers to the status of being overdue on one or more loan

payments. Lenders typically allow for a grace period, which is a specific period after the due date during which late payments can be made without penalty or default being triggered.

2.2.2 Predictor variables for personal loans default

Predictor variables, also known as independent variables or features, are factors used to predict or explain the values of a dependent variable. They capture relevant information and influence the outcome of interest in statistical or predictive modelling (Aslam *et al., 2019).* predictor variables can be numerical or categorical and are used to build models that estimate or predict the dependent variable based on known relationships and patterns in historical data.

Several predictor variables should be considered in analysing variables that may influence personal loan defaults. These variables should include Debt-income- Ratio, Credit score, Debt-to-income ratio, Lifetime value, Credit worthiness.

- **Credit Score:** The assessment of potential risk for new customers and prediction of future behaviour for existing customers is facilitated using credit scoring. [Lee *et al.,* 2017]. This process involves employing statistical models to convert relevant data into numerical measures that provide guidance for credit decisions (Abdou and Pointon, 2011). Traditionally, credit scoring has relied on the financial history of consumers to generate a credit score, which serves as an indicator of their credit risk (Wei *et al.,* 2016)
- **Debt-to-Income Ratio:** This is a vital financial metric indicating how well an individual manages debt compared to income. It is calculated by dividing total debt payments by income, it reflects repayment capacity and financial stability (Gale et al., 2016). It is crucial for borrowers, it guides responsible borrowing choices, prevents excessive debt, and aids lenders in assessing creditworthiness and risk (Gale et al., 2016). Chase Bank (2023) recommends a DTI ratio of 43% or lower, with lower ratios favoured by lenders for stronger financial positions.
- **Life Time Value:** This is a metric used in business and marketing to estimate the total value that a customer will bring to a company over the duration of their relationship. It considers factors such as customer acquisition costs, revenue generated from purchases, and the customer's retention rate. (Reinartz *et al.,* 2018)
- **Credit Worthiness:** Credit worthiness is vital in lending, and it lets lenders evaluate lending risk. It's about borrowers meeting financial commitments and repaying loans on time. Lenders assess creditworthiness through credit history, income, job stability, and finances**.** (Altman, 2018). Highly creditworthy borrowers are regarded as dependable with positive credit

histories marked by timely payments, low debt, and responsible credit management. Those with stable jobs and higher incomes are considered more creditworthy, as they're better equipped to fulfil financial obligations. Conversely, low creditworthiness due to late payments, high debt, or limited credit history implies greater default risk. Lenders employ creditworthiness assessments to define loan terms, encompassing interest rates, loan sizes, and repayment periods. (Lee *et al.,* 2017).

2.3 Machine Learning

Machine learning is defined as the field of study that empowers computers to learn and improve from data without being explicitly programmed. It involves the development and application of algorithms that enable machines to process and analyse data to make informed predictions, decisions, or take specific actions (Alpaydin, 2010).

The fundamental concept behind machine learning is to create models or systems that can automatically learn and adapt from experience or training data (Marsland, ,2015). By utilizing various algorithms and statistical techniques, these models can identify patterns, extract meaningful insights, and make accurate predictions or decisions based on new or unseen data (Keller, 1985).

Machine learning algorithms can be categorized into different types, such as supervised learning, unsupervised learning, and reinforcement learning. Each type of algorithm has its own characteristics and applications based on the nature of the available data and the desired learning objectives.

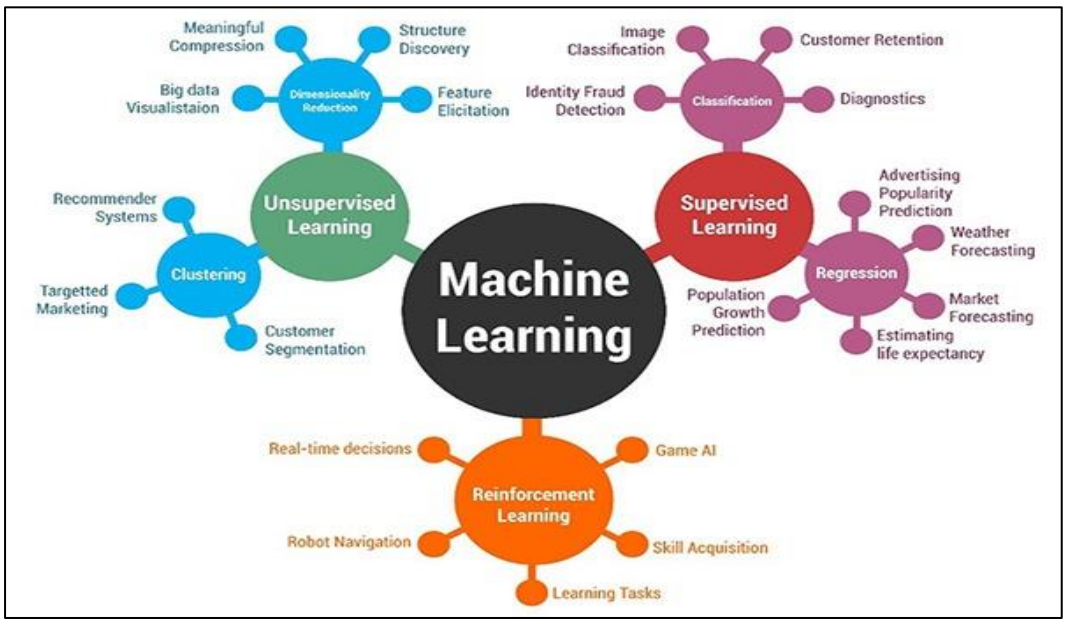


Fig 2.2 : Branches of Machine Learning (https://askdatascience.com/13/what-are-the-main-branches-of-machine-learning)

2.3.1 Types of Machine Learning

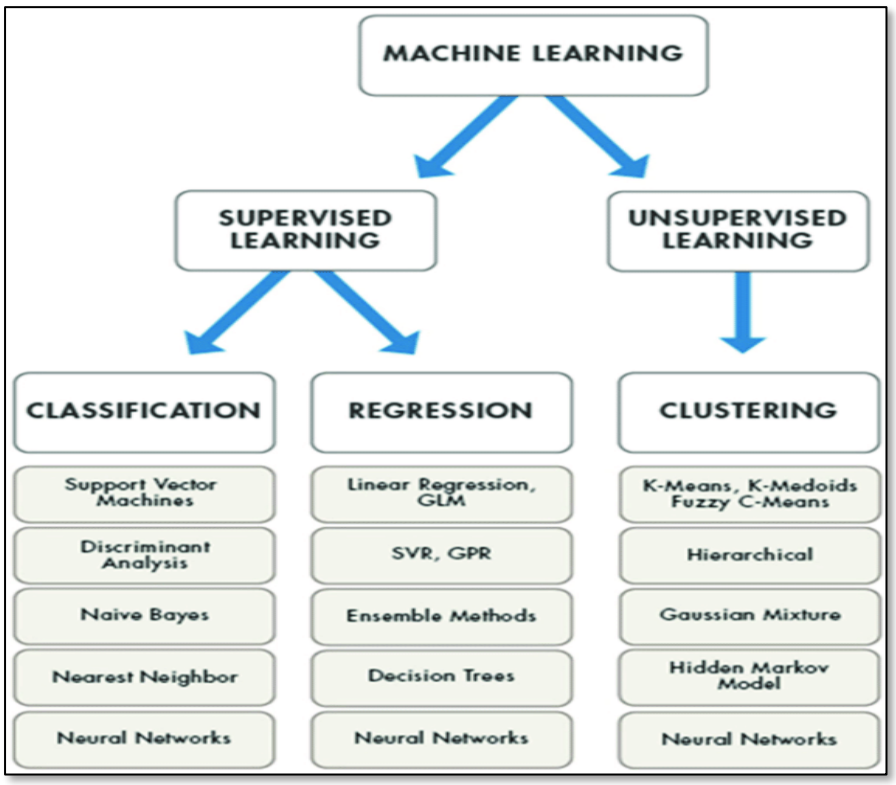


Fig 2.3: Branches of Machine Learning (https://medium.com/@jorgesleonel/supervised-learning-c16823b00c13)

Supervised Learning: Supervised learning trains a machine learning model to learn the connection between inputs and outputs using labelled training data. The model forms a rule to predict new outputs based on unseen inputs, akin to recognizing patterns and making predictions from correct examples. (Sutton, 1992). Supervised Learning is divided into Classification and Regression. **Classification** is a task where the goal is to categorize input data into predefined classes or categories while **Regression** is a task where the goal is to predict a continuous or numerical value based on input data. It involves finding a mathematical relationship between the input features and the output values.

In supervised learning, data is divided into training and testing sets. For classification, various algorithms like linear classifiers, Logistics regression, Naïve Bayes, K Nearest Neighbors, decision trees, random forests, and Bayesian networks are available. These algorithms categorize data into classes using different techniques and decision rules. The choice of algorithm depends on the problem and data characteristics. (Taiwo, 2010).

2.3.2 Unsupervised learning:

This refers to a category of machine learning where there are no predetermined correct answers or a guide the algorithms. In this approach, the algorithms have the freedom to independently explore and discover meaningful structures within the data. They learn to identify important patterns and extract relevant features from the data without explicit guidance or supervision. When presented with new data, unsupervised learning algorithms utilize the knowledge gained from the previous exploration to classify or identify patterns in the data. (Alpaydin, 2010). They leverage the learned features to make sense of the new information without requiring explicit labels or guidance from a teacher.

2.3.3 Reinforcement learning:

This is a foundational concept in machine learning where agents learn by trial and error through interactions with their environment. Agents receive rewards that guide their learning process (Harrington, 2012), aiming to develop optimal strategies for decision-making that maximize cumulative rewards over time. Unlike supervised and unsupervised learning, reinforcement learning centres on learning through interactions and feedback from the environment. This approach is utilized in areas like robotics, game playing, and resource management, where sequential decision-making is vital (Bkassiny et al., 2012).

2.5 Related Works in Personal loan prediction Using Machine Learning

Author Name	Article Title	Data Source	Methods Used	Result and Discussion	Significance and relevance
Dutta, 2021	A study on Machine learning Algorithm for Enhancement of Loan prediction	The dataset used in this project, obtained from Kaggle, comprises two sets: one for training and another for testing. It includes 13 labeled columns: Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Applicant Income, Co-applicant Income, Loan Amount, Loan_Amount_tearm, Credit_History, Property_Area, and Loan_Status.	Logistic regression, Decision tree and Random forest.	In the evaluation of three models, Logistic Regression had an accuracy of 89.7059%, followed by Decision Tree (85.4054%) and Random Forest (77.4566%).	Logistic Regression is considered the most suitable algorithm for loan prediction using Machine Learning. It is observed that applicants with a low credit history have a higher chance of loan rejection.
Alaba *et al.,* 2021	Data mining algorithm for	Dataset which contains a total of 875 loan	Decision Trees	In this analysis, Decision tree model	The predictive model was implemented using

	development of a predictive model for mitigating loan risk in Nigerian banks.	applicant records were collected from Access Bank Plc over a ten-year period (2010 to 2019). The data includes 27 information. After processing the data, attribute selection procedures were conducted to identify the eight most crucial attributes (input variables): Age, Gender, Purpose, Credit_history, Credit_amount, Housing, and Job. The class variable was the loan.	classification , BayesNet and NaiveBayes.	had and accuracy of 99.2%,BayesNet model had an accuracy of 98.74% and NaiveBay had an accuracy of 98.06%.	WEKA software. After evaluating various algorithms, it was discovered that the best one for loan risk classification is Decision Tree. This algorithm demonstrated high accuracy and low mean absolute error, making it the most effective choice for the task.
Mohankumar *et al.,* 2016	Comparative analysis of Decision tree algorithm for the prediction of eligibility of a man for availing bank loan	The dataset contains 17 attributes. The information was provided by loan applicants through dynamic webpages where the users to input authenticated information and the values of the attributes are captured.	Decision tree (ID3-Iterative Dichotomiser 3, CART-Classification and Regression Trees, and C4.5	In this study, the accuracy of ID3 had and accuracy of 54%, C4.5 had an accuracy of 73.3% and CART had an accuracy of 70%.	Three different classification algorithms, namely ID3, C4.5, and CART, were evaluated for their predictive accuracy in a specific task. The reported accuracies for each algorithm reflect how well they performed in classifying instances correctly based on the given dataset.
Smith and Johnson, 2017	Loan default prediction using logistic regression.	The dataset comprises financial data related to consumer loans and a concise social description of clients from a Portuguese banking institution. The data covers the period between January 2008 and December 2009, and the currency used is Euro. It consists of 14 variables, with eight being quantitative and six being qualitative in nature.	Logistic regression	In this analysis, the accuracy of logistic regression which was not compared with another model was 89.9%.	The analysis revealed that the risk of default rises with the loan spread, loan term, and customer's age. However, the risk decreases when the customer owns more credit cards.

Chen, 2018	Loan default prediction using decision trees and random forest	The dataset was obtained from a publicly available Lending Club dataset on Kaggle, containing information about around 2.2 million loans that were funded by the platform from 2007 to 2015.	Decision tree, Random forest	This analysis shows that Decision tree had an accuracy of 73% while Random Forest had an accuracy of 80%.	The Random Forest model seems more suitable for this type of data. Nevertheless, we should investigate the issue of the algorithm misclassifying some non-defaulters as defaulters to improve the model's ability to accurately predict creditworthy borrowers.
Nitesh *et al.,* 2021	Loan approval prediction using machine learning algorithms approach.	The dataset was gotten form Kaggle.	Random Forest, Support Vector Machine, Decision Tree, Logistic regression.	In this experiment, Random Forest showed an accuracy of Random Forest 77%, Support Vector Machine showed 80%, Decision tree had 70% and Logistic regression had 76% accuracy.	Upon evaluating the model on the test dataset, all of these algorithms achieved precision rates ranging from 70% to 80%. However, it is evident that the Support Vector Machine model stands out as highly efficient and outperforms the other models, delivering superior results.
Mehul *et al.,2021*	Loan default prediction using decision trees and random forest: A comparative study	A publicly accessible Lending Club dataset sourced from Kaggle was used for this analysis. This dataset encompasses approximately 2.2 million loans that were funded through the Lending Club platform between the years 2007 and 2015.	Decision tree and Random Forest.	In this study, the Random Forest exhibited an 80% accuracy rate, whereas the Decision Tree achieved an accuracy of 73%.	The Random Forest model appears better suited for this data, but the limited loan default cases over the initial eight-year period (2007-2015) present a drawback. To overcome this, we could enhance the dataset with data from the following three years (2015-2018) to improve predictions and train more accurate models.
Hafiz *et al., 2019.*	Loan Default Prediction Model Using Sample, Explore, Modify, Model, and Assess (SEMMA)	The datasets consist of 13 attributes, including seven categorical and four continuous variables, along with an applicant's ID and a class variable. Additionally, the training dataset	Decision tree, Logistic Regression and Neural Network.	When assessing three models, Logistic Regression achieved an accuracy of 80.9%, the Decision Tree yielded 79.8%, and the Neural Network outperformed with	The proposed model development employed three distinct data mining techniques, each evaluated across different parameters. Based on these assessments, the most effective approach was

		comprises 614 observations, while the testing dataset contains 367 observations. Each observation captures the financial and socio-economic characteristics of the respective loan applicant.		an accuracy of 83.07%.	selected, described, and suggested for its noteworthy ability to predict loan defaults in the financial sector.
Luca *et al.,* 2021	Forecasting Loan Default in Europe with Machine Learning	The dataset from European Datawarehouse covers diverse loan types like residential mortgages, credit cards, and more. It includes both dynamic performance data updated quarterly and static information recorded during loan origination, such as loan amount and borrower's income.	Logistic Regression, Gradient tree boosting and Extreme GB.	In the analysis, Extreme GB had the accuracy of 90.62%, Gradient tree Boosting had the accuracy of 89.58% and Logistic regression had an accuracy of 63.54%.	The findings indicate that XGB achieves notably superior out-of-sample performance in predicting accuracy. The pivotal variables for predicting default occurrences are the current interest rate and Loan-to-Value (LTV) ratio, in conjunction with local economic conditions.

These studies highlight various methodologies and datasets used for loan default prediction, showcasing the effectiveness of deep neural networks, ensemble learning, hybrid models, and interpretable machine learning techniques in improving prediction accuracy and understanding the key factors influencing loan defaults.

CHAPTER THREE

3.0 METHODOLOGY

3.1 Research Methodology

The methodology acts as a guiding framework that outlines the step-by-step procedures undertaken in the study. It covers key aspects such as data collection, exploration, visualization, and the subsequent implementation and evaluation of machine learning models. The research commences with the acquisition of data, which is followed by a critical phase known as data preprocessing. Data cleaning, data mining, and data wrangling form essential components of the data preprocessing stage, ensuring the accuracy and reliability of the data. Data mining involves employing various techniques to extract patterns and insights from the dataset, while data wrangling focuses on transforming the data to make it suitable for analysis and modelling purposes. These stages lay the foundation for the subsequent implementation of machine learning models and facilitate the generation of meaningful conclusions based on the collected data.

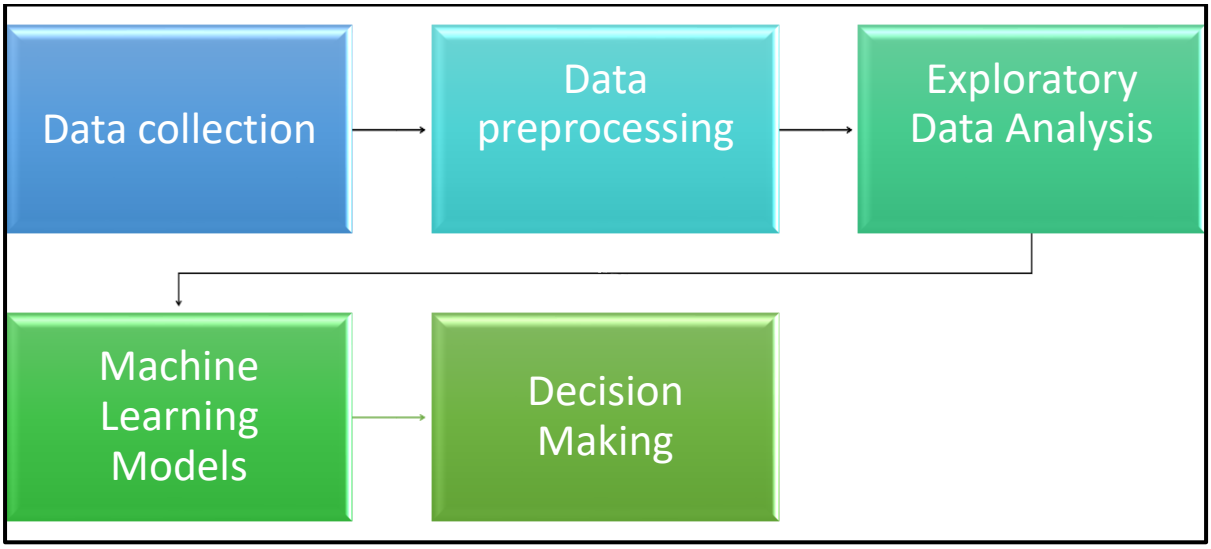


Fig 3.1 Study Flowchart (self, 2023)

3.2 Data Source and Description

The data being analysed to gain insights is obtained from Kaggle; an open-source platform widely recognized as a prominent provider of datasets for learning purposes. Kaggle serves as a popular online hub for data science and machine learning, offering a vast array of datasets contributed by the community. It provides a collaborative space for data scientists and researchers to engage in cooperation, participate in competitions, and exchange knowledge.

The dataset is extensive and encompasses various deterministic factors such as borrower's income, gender, loan purpose, and term, among others. It exhibits notable multicollinearity and contains

empty values. The dataset comprises a total of 148,671 records and 34 variables. Some variables have missing values, which will be addressed using various data cleaning techniques. The financial data was extracted from the bank's existing customers in the year 2019 and is well-suited for classification analysis with predefined binary classes.

Source of data : https://www.kaggle.com/datasets/yasserh/loan-default dataset/download?datasetVersionNumber=1

The following table illustrates a sample representation of the dataset structure.

Attribute Name	Attribute Description	Data type
ID	A unique ID is assigned for each borrower	Integer
year	Loan disbursement year	Number
loan limit	the maximum available amount of the loan allowed to be taken	Integer
Gender	Gender of each borrower	Character
Approved in adv	Is the loan pre-approved or not	Character
loan type	Type of loan	Character
loan purpose	the reason you want to borrow money	Character
Creditworthiness	how worthy you are to receive new credit based on your credit score.	Character
Open credit	the unlimited number of repeated withdrawals up to a certain limit.	Character
Business or commercial	Usage type of the loan amount	Character
loan amount	The exact loan amount requested by the borrower	Integer
Rate of interest	Interest rate on the loan amount disbursed	Number
Interest_rate_spread	Interest rate difference between the borrowers' and depositors'	Number
Upfront charges	The fee paid to a lender by a borrower as consideration for making a new loan	Number
term	the loan's repayment period in months	Number
Neg_ammortization	refers to a situation when a loan borrower makes a payment less than the standard instalment	Character
Interest only	amount of interest only without principal (loan amount)	Character
Lump sum payment	repayment paid in a single payment rather than in instalments.	Character
Property value	value of the collateral used in securing the loan	Integer
Construction type	the type of collateral used in securing the loan	Character
Occupancy type	Housing condition of the borrower. E.g, Rentage, Mortgage	Character
Secured by	Collateral used in securing the loan	Character
Total units	Number of units in the building	Character
income	money that is earned from doing work or received from investments	Integer
Credit type	type of credit being run by the borrower	Character
Credit Score	a 3-digit number that shows how likely you are to be accepted for credit	Number
co-applicant credit type	is an additional person involved in the loan application process?	Character
age	applicant's age	Number
Submission of application	Ensure the application is complete or not	Character
LTV	life-time-value (LTV) of the loan.	Number
Region	applicant's place	Character
Security Type	Type of collateral Used	Character
Status	Loan status (Approved/Declined)	Number
dtir1	debt-to-income ratio	Number

3.2.1 Data Pre-processing

Data preprocessing is a crucial step in the data mining process, involving the manipulation or elimination of data to improve performance and ensure reliable results. It encompasses various techniques for preparing the data before applying data mining algorithms or models. These techniques may include cleaning the data to handle errors, inconsistencies, or missing values, as well as transforming the data to meet the requirements of specific algorithms. Data preprocessing plays a vital role in improving the quality and usability of the data, enhancing the effectiveness and efficiency of subsequent data mining tasks. (Pyle, 1999)

3.2.2 Data Cleaning

Data cleaning, also referred to as data cleansing or scrubbing, involves the process of removing noise, addressing missing values, and resolving inconsistencies in the dataset. The presence of dirty data in a database can arise from incorrect data entry, updates, or transmission errors (Garcia, 2015). Additionally, this phase encompasses the identification and elimination of outliers. Data cleaning is an essential and pivotal step in guaranteeing the accuracy and reliability of the dataset. It is uncommon to encounter completely clean data due to the presence of noise or errors introduced during manual or automated data collection processes. Practical datasets often contain inconsistencies and inadequacies that may arise from malfunctioning sensors or human errors. These imperfections can potentially influence the performance of machine learning models trained on such datasets. Therefore, data cleaning becomes crucial in mitigating these issues and ensuring the quality of the data for subsequent analysis and modelling (Han, 2011).

3.2.3 Data Transformation

Data transformation aims at converting data into a format that is usable and comprehensible. This process is essential for enhancing the efficiency of data mining operations (Han *et al.*, 2011). Various strategies are employed for data transformation, including data smoothing, feature construction, normalization, discretization, generalization, and concept hierarchy generation, particularly for nominal data. These techniques facilitate the optimization of data analysis and modelling tasks, enabling more effective and insightful exploration of the dataset. (Brown, 2014)

3.2.4 Data Reduction

Data reduction is a crucial step in data analysis and management, particularly in the context of big data. It entails the process of reducing the number of variables or attributes in a dataset while striving to maintain the accuracy and quality of the resulting model. The primary objective of data reduction

is to streamline data storage, improve data mobility within networks, and optimize computational resources. (Bing, 2012)

There are several techniques employed in data reduction that serve different purposes. One common approach is dimensionality reduction, which aims to reduce the number of variables while preserving the most informative aspects of the data. This technique helps in mitigating the curse of dimensionality, where high-dimensional data poses challenges in analysis and computational efficiency. Data reduction techniques also address the issues of duplicate and redundant data. (Xian, 2021) By identifying and eliminating redundant entries or records, the dataset becomes more streamlined, concise, and efficient. This not only saves storage space but also enhances data retrieval and processing speed.

3.2.5 Data Integration

Data integration is a process that involves combining and merging datasets from multiple sources to create a unified and comprehensive dataset. In situations where data is collected from diverse sources, such as different databases, files, or systems, data integration enables analysts to create a single dataset that contains information from all these sources. (Rahm & Do, 2000)

The goal of data integration is to ensure that data from various sources can be effectively and efficiently analysed or used for further processing. This process typically involves identifying common data elements or key variables that can serve as matching criteria for combining datasets (Doan *et al.,* 2000). By aligning and merging the datasets based on these common elements, analysts can create a consolidated dataset that incorporates information from multiple sources.

3.3 Methods and Algorithms

The choice of algorithms for model development is determined by the dataset, study objectives, and challenges. In this study, Machine Learning classification would be applied. Classification is a method that predicts group membership for data instances based on structured data. The study will use five machine learning algorithms, each described below:

3.3.1 Logistic Regression

Logistic regression is a statistical technique employed to model the connection between a binary outcome variable (also called the dependent variable) and one or more independent variables (also termed predictor variables or features). It finds frequent application when the outcome variable is categorical, presenting two potential results that could be labelled as "success" or "failure," "yes" or

"no," or "1" or "0" (Taiwo, 2010). This is a classification method that employs a single multinomial logistic regression model with a single estimator. Logistic regression is utilized to determine the boundary between classes and estimate the class probabilities based on the distance from the boundary (Bkassiny *et al.,* 2012). This approach allows for the modelling of the relationship between variables and the prediction of class membership in classification tasks (Hormozi et al., 2012. Logistic regression will be selected as the method of choice for this study due to its suitability for classification analysis. Additionally, its well-established nature and abundance of learning resources make it a favourable option.

3.3.2 Decision Trees (DT)

A decision tree is a visual tool in machine learning and data analysis that breaks down data using a tree structure based on input feature values, with internal nodes reflecting decisions and branches indicating potential outcomes, while leaf nodes represent final predictions or classifications (Hormozi *et al., 2012)*. These are hierarchical structures used for classifying instances by sorting them according to their feature values. Each node in a decision tree represents a feature from the instance to be classified, and each branch corresponds to a value that the node can take. This process of recursively dividing the data based on feature values allows decision trees to make informed decisions and classify data points accurately (Kotsiantis, 2007). The reason to employ the decision tree method in this study lies in its versatility across diverse datasets. Moreover, its ease of interpretation and visual representation contribute to a clearer understanding of the model's decision logic.

3.3.3 Support Vector Machines

Support Vector Machine (SVM) is a supervised machine learning technique utilized for classification and regression, especially proficient with intricate high-dimensional data, adept at establishing distinct boundaries between diverse classes (Alex & Vishwanathan, 2008). The latest advancement in supervised machine learning is the Support Vector Machine (SVM) technique. SVM models share close connections with classical multilayer perceptron neural networks. (Marsland, 2015). The foundation of SVMs is based on the concept of a "margin," which represents the space on either side of a hyperplane that separates two data classes. This margin is essential in SVMs for effectively classifying data points into their respective categories. (Setiono and Loew, 2000) The reason for selecting this algorithm for the study is its capability to manage high-dimensional data effectively while also demonstrating reduced susceptibility to overfitting.

3.3.4 K- Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised algorithm for classification and regression. It identifies the "k" closest training examples to a test point and uses their labels/values to predict for the test point. Distance metrics, commonly Euclidean distance, define the nearest neighbors (Tapas, 2002). kNN is a versatile machine learning algorithm used for classification and regression tasks. It classifies or predicts a data point based on the majority class or average value of its k-nearest neighbors in the feature space. The choice of distance metric and k value is crucial for its effectiveness. kNN is non-parametric and widely used in various applications, but it can be computationally expensive for large datasets Sutton, 1992). KNN is utilized in this study due to its capacity to handle both classification and regression tasks, along with its versatility and ability to handle various types of datasets.

3.3.5 Artificial Neural Networks

An Artificial Neural Network (ANN) mimics the brain's structure and is essential in deep learning, a subset of machine learning. Artificial Neural Networks (ANNs) are machine learning algorithms inspired by the human brain's neural networks. They consist of interconnected nodes organized in layers that process input data and produce output predictions (Harrington, 2012). ANNs learn by adjusting weights during training to optimize performance. They are used for various tasks, such as classification, regression, image recognition, and natural language processing. Deep Learning, a subset of ANNs with multiple hidden layers, has achieved significant success in solving complex problems (Yan *et al*., 2001). ANN is also utilized in this study because of its ability to handle large amounts of data.

3.4 Model Evaluation

Model evaluation is a fundamental step in machine learning that evaluates the performance of a trained model on new, previously unseen data. For classification, various techniques such as accuracy, precision, recall, F1 score, and confusion matrix are used, as well as mean squared error and R-squared for regression. Cross-validation ensures that the results are reliable. (Leah, 2021) The evaluation metrics chosen are tailored to the specific problem at hand, allowing for model refinement and informed decision-making in real-world applications (Iqbal *et al.,* 2019).

A. **Cross-validation** is a technique in machine learning used to assess a model's performance and its ability to generalize to unseen data. It involves partitioning the dataset into subsets for both training and testing purposes, with each partition being used multiple times to get a more robust evaluation (Narkhede, 2018). This method helps prevent overfitting and aids in selecting the best

model parameters by providing more reliable performance estimates than a single train-test split.

B. **The confusion matrix** is a valuable evaluation tool for classification models. It compares the model's predicted labels with the actual labels in the test dataset, generating four metrics: True Positive, True Negative, False Positive, and False Negative (Nandacumar, 2020). From these metrics, performance measures like accuracy, precision, recall, and F1 score can be derived, offering a comprehensive assessment of the model's performance on various aspects. Particularly useful for tasks with imbalanced classes, the confusion matrix aids in fine-tuning the model and improving its overall effectiveness. (Saito and Rehmeismeier, 2017)

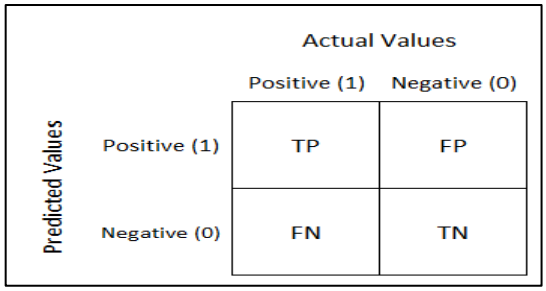


Fig 3.2: Diagram of a confusion table (https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62)

The AUC of ROC curve: is a metric used to evaluate the performance of binary classification models. It quantifies the area under the Receiver Operating Characteristic curve, with values ranging from 0 to 1. (Saito and Rechmsmeier, 2015) Higher AUC values indicate better model performance, where 1 represents a perfect classifier and 0.5 denotes a random classifier. This metric is especially valuable for assessing the model's ability to distinguish between positive and negative classes, making it suitable for imbalanced datasets. It provides a single value that measures the model's discrimination power and assists in model selection and comparison. (Iqbal *et al.,* 2019).

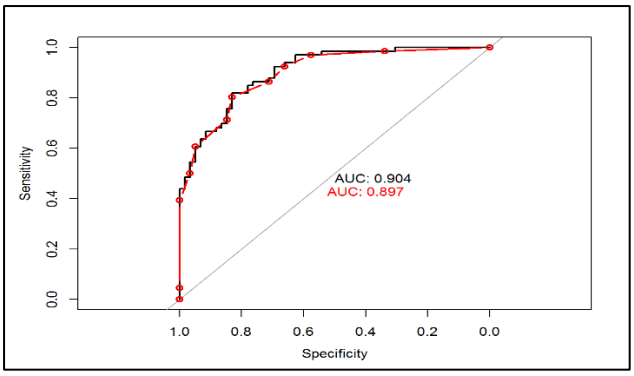


Fig 3.3: Diagram of AUC of ROC curve (https://blog.revolutionanalytics.com/2016/11/calculating-auc.html)

3.5 Performance metrics

Precision is a binary classification metric that measures the model's accuracy in predicting positive instances. It focuses on the proportion of true positive predictions out of all positive predictions made. A high precision value indicates fewer false positives, making it valuable in applications like medical diagnoses or fraud detection. (Auckland, 2017)

Precision = $\frac{TP}{TP+FP}$

Accuracy is a classification metric that represents the ratio of correctly predicted instances to the total instances in a dataset. While accuracy provides an overall indication of the model's performance, it might not be suitable for imbalanced datasets. In such cases, metrics like precision, recall, F1 score, or AUC-ROC are often employed to offer a more comprehensive assessment of the model's effectiveness. (Mishra, 2018)

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

Recall is a classification metric that evaluates the model's ability to correctly identify positive instances. It measures the proportion of true positive predictions out of all actual positive instances. High recall is crucial in minimizing the number of missed positive instances in certain applications (Vujović, 2020). Along with other metrics, recall provides a comprehensive evaluation of the model's performance.

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

Specificity is a classification metric that evaluates the model's accuracy in identifying negative instances. It calculates the proportion of true negative predictions out of all actual negative instances. High specificity is valuable in reducing false positives, particularly in applications like medical diagnoses or safety-critical systems (J.Sim and Wright, 2005). Together with other metrics, specificity provides a comprehensive assessment of the model's performance in binary classification tasks.

Specificity =$\frac{TN}{TN+FP}$

F1 score The F1 score, a performance metric for binary classification, offers a balanced assessment by combining precision and recall. This evaluation considers both positive and negative instances, providing a comprehensive view of the model's performance. With a range from 0 to 1, higher F1 score values signify better model effectiveness. This metric proves especially beneficial for imbalanced

datasets, achieving equilibrium between minimizing false positives and false negatives. (Delgrado *et al.,* 2019).

F1-Score = $\frac{TP}{TP+\frac{1}{2}(FP+FN)}$

CHAPTER FOUR

4.0 Practical Implementation

Introduction

The project implementation involves several initial steps, including data preparation, exploration, and model development with critical features. The process is divided into two sections. The first section focuses on fundamental tasks like data collection, addressing missing values, and rectifying errors to ensure clean data for analysis. This prepares the data for in-depth exploration, helping to identify key attributes for the subsequent model-building phase. The second stage involves machine learning, where final features are used to create multiple models using specified algorithms.

The analysis was conducted within R Studio, a programming language that offers diverse tools and functions designed for exploratory analysis purposes. This was divided into different sections.

4.1 Data Exploration in Power BI

The dataset's dashboard visualization was created using Microsoft Power BI, a software suite developed by Microsoft. Power BI encompasses a collection of software systems, applications, and plugins that collaborate to convert data into interactive insights.

Creating a Dashboard in Microsoft Power BI

- Load the file on PowerBI.
- Connect to data sources using power query and import the data.
- Defining measures and calculated columns using dax.
- Building the report page containing Visuals such as charts, cards, pie charts and tables.
- Creating a dashboard and arranging the tiles.
- Applying filters and slicers.

4.2 Data Loading

4.2.1 Setting a working directory and data loading.

```
#-----------------Section 01-----------------------
# Set working directory
setwd(dirname(file.choose()))
getwd()

# read file
loan.dat <- read.csv("Loan_Default.csv", stringsAsFactors = FALSE)

```

The code sets the R working directory using a chosen file's path. It displays the current directory using `getwd()`. It reads a CSV file named "Loan_Default.csv" using `read.csv()`, with `stringsAsFactors` set to `FALSE` to prevent automatic string conversion to factors.

4.3 Data Exploration

4.3.1 Data Summary statistics and Data cleaning.

This section explores data by viewing, summarizing, and visualizing. It removes unnecessary columns, deals with missing data, and cleans the dataset, reducing observations from 148,670 to 124,408.

```
#------------------Section 02--------------------
# examine the structure of the loan.data data frame
head(loan.dat)
summary(loan.dat)
#Drop column 1 & 2 not needed
loan.dat <- loan.dat[,-1:-2]
str(loan.dat)

#Remove variables for undisbursed loan.data
loan.dat <- loan.dat[,-10:-12]


# check for missing data
library(Amelia)
library(tidyr)
apply(loan.dat, MARGIN = 2, FUN = function(x) sum(is.na(x)))
missmap(loan.dat, col = c("red1", "aquamarine4"), legend = FALSE)
loan <- na.omit(loan.dat)
missmap(loan, col = c("red2", "aquamarine4"), legend = FALSE)

```

The code previews data with `head()` and summarizes with `summary()`. Unneeded columns are removed. Data structure is checked using `str()`, and missing data is managed through calculation, visualization, and handling.

4.3.2 Data Visualization

Data visualization simplifies complex information through charts and graphs, aiding pattern recognition. This study employs diverse visualizations (bar charts, scatter plots, etc.) using tools like R's ggplot2 and plotly for insightful analysis and communication.

```
#------------------Section 03--------------------
#------------------Data Visualizations--------------------

# table of status
table(loan$Status)
# Percentage distribution status
round(prop.table(table(loan$Status)) * 100, digits = 2)
#Barplot visualization
barplot(table(loan$Status), col = c("darkgreen", "red"), legend = TRUE, main="Loan Status Distribution")
#Simple Pie Chart from Loan Status Proportions
slices <- c(104092, 20316)
lbls <- c("Repaid", "Defaulted")
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices, labels = lbls,col = c("green4", "red"), main="Pie Chart of Loan Status")

# table of Age by group
table(loan$age)
round(prop.table(table(loan$age)) * 100, digits = 2)
#Barplot visualization
barplot(table(loan$age), main="Age Distribution")

#Gender Distriution
table(loan$Gender)
# Percentage distribution Gender
round(prop.table(table(loan$Gender)) * 100, digits = 2)
#Simple Pie Chart from Gender Proportions
slices <- c(23562, 34828, 35279, 30739)
```

The code employs R for detailed data visualization, including loan status, age distribution, gender, pre-approval rates, loan types, loan purposes, and creditworthiness represented via tables, percentages, bar plots, and pie charts. It also uses multi-variable bar plots to illustrate relationships with loan status and employs advanced visualization using the `ggplot2` library to show age groups, income, and credit score relationships through a line plot.

4.3.3 Data Transformation and Boxplot Visualization

Data transformation involves modifying the structure or content of a dataset to make it more suitable for analysis or modelling. Data transformation is a crucial step to ensure accurate and effective analysis, as well as to prepare the data for machine learning algorithms or other statistical modelling techniques.

```
#----------------------------Section 04-------------------------
#-----------------Data Transformation--------------------
loan$loan_limit <- as.factor(loan$loan_limit)
loan$approv_in_adv <- as.factor(loan$approv_in_adv)
loan$loan_purpose <- as.factor(loan$loan_purpose)

#Transform the data frame into  matrix (Except Target Variable that was encoded)
loan.data <- data.matrix(loan[,-28])
loan.data <- as.data.frame(loan.data)
loan.data$Status <- loan$Status
str(loan.data)
summary(loan.data)


#------------Boxplots for Transformed Data----------
boxplot(loan.data, col="slategray4", main = "Boxplot of all Variables")
# Boxplot of the dependent variable
boxplot(loan.data$dtir1, col="slategray4", main="Boxplot of Loan's DebtToIncome Ratio")


# Boxplot of all variables for random visualization
boxdata <- boxplot(loan.data$income , col="coral4", main = "Boxplot of Income Spread")
boxdata <- boxplot(loan.data$age, col="coral4", main = "Boxplot for Age")
boxdata <- boxplot(loan.data$property_value, col="coral4", main = "Boxplot for Property Value")
boxdata <- boxplot(loan.data$loan_amount, col="coral4", main = "Boxplot for Loan Amount")
boxdata <- boxplot(loan.data$LTV, col="coral4", main = "Boxplot for Life-Time Value (LTV)")
boxdata <- boxplot(loan.data$Credit_Score, col="coral4", main = "Boxplot for Credit Score")
```

The code first converts categorical variables to factors, transforms the data for analysis, and then visualizes the data using boxplots. The boxplots provide insights into the distribution and spread of various variables within the dataset.

4.3.4 Correlation Analysis

Correlation analysis entails investigating the statistical connection between variables in a dataset to establish how they vary collectively. This analysis reveals if correlations are positive (variables increase together), negative (one variable rises as the other falls), or near zero (minimal relationship).

```
#----------------------------Section 05-------------------------
#-----------------Correlation Analysis--------------------

#Check correlations between all variables using Matrix
library(corrplot)
loan.data.cor = cor(loan.data)
loan.data.cor
#Order by alphabet and arranged in order of correlation
corrplot(loan.data.cor, method = 'number',  diag=FALSE)

#check correlations between all independent variables
library(polycor)
Loan.cor <- hetcor(loan.data[-29])
Loan.cor <- round(Loan.cor$correlations, 2)
#Mixture of number and bubble chart in alphabetical order
corrplot.mixed(Loan.cor, order = 'alphabet')

#check correlations between Independent attributes
corMatrix <- cor(loan.data [,1:28])
corrplot(corMatrix, method="color",
         # Show the coefficient value
         addCoef.col="Black",
         # "FIRST PRINCIPAL ORDER" is to enable it order OF their correlation
         order="FPC" ,
         # Show only the matrix bottom and avoid the diagonal of ones.
         type="lower", diag=FALSE,
         # Cross the values that are not significant
         sig.level=0.05, main = "Correlation Matrix of Independent Variables")

attach(loan.data)
#-------Calculate partial correlation------
library(ppcor)
pcor.test(Status, Gender, age)
pcor.test(Status, Credit_Score, Credit_Worthiness)
pcor.test(Status, property_value, Region)
pcor.test(Status, dtir1, LTV)
pcor.test(Status, loan_amount, term)
pcor.test(Status, Security_Type, approv_in_adv)
pcor.test(Status, loan_type, loan_purpose)
```

The code conducts thorough correlation analysis involving calculations, visualizations, and tests to understand variable relationships. It uses libraries like `corrplot` and `polycor` to compute and visualize correlations within the `loan.data` dataset. It calculates partial correlations and performs individual correlation tests to reveal relationships between the dependent variable (`Status`) and selected independent variables.

4.3.5 Removing Outliers

The code conducts thorough correlation analysis involving calculations, visualizations, and tests to understand variable relationships. variables.

```
289
290 #----------------------------Section 08-------------------------
291 #------------Remove Outliers-----------
292 library(rstatix)
293 library(dplyr)
294 #Identify the outliers in Target Variable
295 df2 <- identify_outliers(Loan.new, c(property_value))
296
297 #define the outliers function using The Interquartile Range (IQR) method
298 get_outliers = function(x){which(x > quantile(x)[4] + 1.5*IQR(x) | x < quantile(x)[2] - 1.5*IQR(x))}
299 outliers <- get_outliers(Loan.new$property_value)
300 Loan.outlier <-Loan.new[-outliers,]
301
302 boxplot(Loan.outlier, col="coral4", main = "Boxplot of Variables after removing Outliers")
303 #Rename the dataframe
304 LoanML = Loan.outlier
305
```

This code utilizes statistical techniques to detect and eliminate outliers from the "property_value" variable within the "Loan.new" data frame. Following this, a boxplot is crafted to depict how the distribution of variables changes after outlier removal. The resultant data frame, "LoanML," is prepared for forthcoming machine learning analysis.

4.3.6 Factor Analysis

The code analyses correlations using calculations, visualizations, and tests. It employs `corrplot` and `polycor` to compute and visualize correlations in `loan.data`. It tests relationships between `Status` and selected variables.

```r
#---------------------------Section 06-------------------------
#-----------------Factor Analysis-------------------
library(psych)
# The Kaiser-Meyer-Olkin (KMO) test to measure suitable attributes for my analysis
KMO(loan.data)

# The Bartlett's test of sphericity from Correlation Analysis
cortest.bartlett(loan.data.cor)

#------Eigen Value Analysis-------------
# Determine Number of Factors to Extract
library(nFactors)

# get eigenvalues: eigen() uses a correlation matrix
ev <- eigen(cor(loan.data))
ev$values
head(ev$values, n = 15)
zz <- as.data.frame(ev$values)
yy <- as.data.frame(ev$vectors)
# plot a scree plot of eigenvalues
plot(ev$values, type="b", col="blue", xlab="variables", lwd=2, main = "Eigen Value Plot")


# calculate cumulative proportion of eigenvalue and plot
ev.sum<-0
for(i in 1:length(ev$value)){
  ev.sum<-ev.sum+ev$value[i]
}
ev.list1<-1:length(ev$value)
for(i in 1:length(ev$value)){
  ev.list1[i]=ev$value[i]/ev.sum
}
ev.list2<-1:length(ev$value)
ev.list2[1]<-ev.list1[1]
for(i in 2:length(ev$value)){
  ev.list2[i]=ev.list2[i-1]+ev.list1[i]
}
plot (ev.list2, type="b", col="red", xlab="number of components", ylab ="cumulative proportion", lwd =2,
      main = "Eigen Value Cumulative Proportion Plot")


#-----Principal Components Analysis-------
# retaining 'nFactors' components
library(GPArotation)
library(psych)
# principal() uses a data frame or matrix of correlations
pca <- principal(loan.data[-29], nfactors=6, rotate="varimax", method = "maximum likelihood")
pca
pca$Structure #Examine important variables by Rotating components
pca$r.scores  #Create 6 variables (RC1 - RC6) to represent the rotated components
```

The code assesses data suitability for factor analysis using the KMO. It measures how well variables can form factors. High KMO values indicate good correlation for factor analysis. Eigenvalues indicate factors' variance capture while Scree plot helps choose meaningful factors. Cumulative proportion plot also aids in selecting factors for satisfactory variance capture. The code guides factor selection using these analyses.

4.3.7 Normalization

Normalization is a method in data preparation that aims to place numerical variables on a common scale. Its purpose is to ensure that various features carry equal weight during analyses and model building, especially in cases where the data's scale could influence the results.

```r
#---------------------Section 09----------------------------
#------------Standardization/Normalization---------------
library(DMwR2)
#-------Normalization Using Soft Max------
Loan.sm <- as.data.frame(LoanML[c(1:15)])
# soft max [edit lambda]
Loan.sm <- apply(Loan.sm, MARGIN = 2, FUN = function(x) (SoftMax(x,lambda = 15, mean(x), sd(x))))
#--Investigate the result of the normalization
boxplot (Loan.sm, main = "Soft Max, lambda = 15", col="Bisque2")

#Fit in the dependent variable in the transformed data frame
Loan.sm <- as.data.frame(Loan.sm)
Loan.sm$Status<- as.factor(LoanML$Status)
boxplot(Loan.sm, main = "Soft Max, lambda = 15", col="Bisque")
describe(Loan.sm)

#Export new data frame to testing and training of all Models.
write.csv(Loan.sm, "Loan.SM.csv")
```

The code performs SoftMax normalization on a subset of the "LoanML" dataset, visualizes the results through boxplots, includes the dependent variable, generates descriptive summaries, and exports the normalized data for use in model testing and training.

4.4 Machine Learning Algorithms

4.4.1 Training and Testing

Training and testing a dataset is a fundamental step in machine learning. It involves splitting the dataset into two parts: one for training the model and another for testing its performance.

```
#-------------------------Section 10----------------------------------
#--------------Training & Testing--------------------
# Split into train and test (at mother ID = 119255) by 70-30
library(caret)
set.seed(12345)
#Shuffle the data set
Loan.sm = Loan.sm[sample(nrow(Loan.sm)), ]

#***********Splitting DataSet***************_
#*Ratio 70/30
training_set <- Loan.sm[1:83478, ]
test_set <- Loan.sm[83479:119255, ]

#checking if balanced data set or not using their proportions
table(training_set$Status) #Training Set
prop.table(table(training_set$Status))

table(test_set$Status) #Testing Set
prop.table(table(test_set$Status))

#-----Class Imbalance-----------
library("ROSE")
#Under Sampling Technique
train_dat<- ovun.sample(Status~., data=training_set,
                        method="under", N=27120, seed=12345)$data
table(train_dat$Status)

#-----Randomize By Shuffling-----------
train_dat = train_dat[sample(nrow(train_dat)), ]
```

This code prepares the dataset for training and testing a machine learning model. It shuffles the data, splits it into training and testing sets (70:30) and addresses class imbalance in the training set, and shuffles the training set for further processing.

4.4.2 Algorithm Building for Logistic Regression

```
#-------------------------Section 11----------------------------------
#--------------ALGORITHM BUILDING--------------
library(class)
library(pROC)
library(ROCR)
library(caTools)
library(vip)
#*********************Logistic Regression*****************************
set.seed(12345)
# Fitting Logistic Regression to the Training set
classifier = glm(formula = Status ~ ., family = binomial, data = train_dat)
summary(classifier)

# Predicting the Test set results
prob_pred = predict(classifier, type = 'response', newdata = test_set[-16])
Loan_Logistic = ifelse(prob_pred > 0.5, 1, 0)

# evaluating the models Performances with CrossTabulation & Confusion Matrix
library(gmodels)
library(caret)
CrossTable(x = test_set[,16], y = Loan_Logistic, prop.chisq=FALSE)
confusionMatrix(as.factor(Loan_Logistic), as.factor(test_set[,16]), mode  ="everything")
auc(test_set$Status, Loan_Logistic)

#Plotting ROC Curve for the Models with AUC
library(pROC)
LR.Perf <- roc(test_set$Status, Loan_Logistic)
plot(LR.Perf,plot=TRUE, print.auc=TRUE,
     col="red", lwd=4, legacy.axes=TRUE,main="Logistic Regression ROC Curve")

# Construct variable importance plot
vip(classifier, aesthetics = list(colour="brown", fill="darkgreen"),
    Main = "Variance of Importance for Variables")
vi(classifier)
```

In this segment, libraries for model evaluation, visualization, and classification are imported. A logistic regression model is constructed, incorporating all selected variables. Variables that don't contribute to the model's effectiveness are excluded for model improvement. The model's performance will be assessed using Cross-Tabulation, Confusion Matrix, and AUC metrics.

4.4.3 Algorithm Building for K-Nearest Neigbhour

```
#-------------------------Section 12-------------------------------
#*******************KNN*****************************
sqrt(23122) #Find the Square root of training dataset
#using K value as 152
set.seed(12345)
knn_pred = knn(train = train_dat[, -16], test = test_set[-16], cl = train_dat[, 16],
               k = 153)
#using K value as 23
knn_pred3 = knn(train = train_dat[, -16], test = test_set[-16], cl = train_dat[, 16], k = 3)

library(gmodels)
library(caret)
# evaluating the models Performances with CrossTabulation & Confusion Matrix
CrossTable(x = test_set[,16], y = knn_pred3, prop.chisq=FALSE)
confusionMatrix(as.factor(knn_pred3), test_set[,16], mode = "everything")
auc(test_set$Status, as.numeric(knn_pred3))

CrossTable(x = test_set[,16], y = knn_pred, prop.chisq=FALSE)
confusionMatrix(as.factor(knn_pred), test_set[,16], mode = "everything")
auc(test_set$Status, as.numeric(knn_pred))

#Plotting ROC Curve for the Models with AUC
KNN_Perf <- roc(test_set[,16]~ as.numeric(knn_pred))
plot(KNN_Perf,plot=TRUE, print.auc=TRUE, col="red", lwd=4, legacy.axes=TRUE,main="ROC Curve of kNN, k=153")
```

The performance of two kNN models with different k values using various evaluation metrics are shown in this section, including Cross-Tabulation, Confusion Matrix, AUC, and ROC curves. KNN prediction is made by considering the training dataset and the corresponding class labels. This is performed using the `knn` function.

Cross-Tabulation and Confusion Matrix are used to assess the model performance against the actual class labels. The Area Under the Curve (AUC) are calculated to further evaluate the models' performance using the `auc` function. ROC curves are plotted for both KNN models using the `roc` function from the `pROC` library. The resulting plots shows the performance of the KNN models in terms of true positive rate and false positive rate, and the AUC values are printed on the plots.

4.4.4 Algorithm Building for Support Vector Machine

```
#-------------------------Section 13-----------------------------------
library(kernlab)
library(e1071)
#**********************Support Vector Machine************************
set.seed(12412345)
# run initial model "svm0" ( Laplace = Non-Linear kernel)
svm0 <- ksvm(Status ~ ., data = train_dat, kernel = "laplacedot", type = "C-svc")
svm0 #Examining the Model's information

# apply the model to make predictions
svm_pred <- predict(svm0, test_set[-16])
table(svm_pred, test_set[,16])

# evaluating the models Performances with CrossTabulation & Confusion Matrix
library(caret)
library(gmodels)
CrossTable(x = test_set[,16], y = svm_pred, prop.chisq=FALSE)
confusionMatrix(svm_pred, test_set[,16], mode = "everything")
auc(test_set[,16], as.numeric(svm_pred))

#Plotting ROC Curve for the Models with AUC
library(pROC)
svm0.roc <- roc(test_set[,16], as.numeric(svm_pred))
plot(svm0.roc, avg= "threshold", col= "red", print.auc=TRUE,
     lwd=3, main="SVM ROC Curve (LaPlace Kernel)")


#---------------# Radial Basis-Gaussian (RBFDot Kernel string)-----------------------
# explore improvements of the model byusing a non-linear kernel function
set.seed(12345)
svm1 <- ksvm(Status ~ ., data = train_dat, kernel = "rbfdot", type = "C-svc")
svm1 # look at basic information about the model

# apply the model to make predictions
svm_pred1 <- predict(svm1, test_set[-16])
table(svm_pred1, test_set[,16])

# evaluating the models Performances with CrossTabulation & Confusion Matrix
library(caret)
library(gmodels)
CrossTable(x = test_set[,16], y = svm_pred1, prop.chisq=FALSE)
confusionMatrix(svm_pred1, test_set[,16], mode = "everything")
auc(test_set[,16], as.numeric(svm_pred1))

#Plotting ROC Curve for the Models with AUC
library(pROC)
svm1.roc <- roc(test_set[,16], as.numeric(svm_pred1))
plot(svm1.roc, avg= "threshold", col= "red", print.auc=TRUE,lwd=3, main="SVM~2 ROC curve (RBFDot)")
```

Support Vector Machines (SVM) are implemented for classification tasks using the `kernlab` and `e1071` libraries. This code demonstrates the implementation and evaluation of Support Vector Machine models with different kernel functions (non-linear and Radial Basis-Gaussian) for classification tasks. It shows the process of fitting, predicting, evaluating, and visualizing the performance of SVM models.

4.4.5 Algorithm Building for Decision Tree

```
#----------------------------Section 14---------------------------------
library(C50)
library(rpart)
library(rpart.plot)
#**********************Decision Tree************************
set.seed(12345)
#Fitting the Decision Tree for the Training set
fit <- rpart(Status~., data = train_dat, method = 'class')
rpart.plot(fit, box.palette="RdBu", shadow.col="gray", nn=TRUE,
           main="Decision Tree Schematic of Trained Model")

#identify best cp used by the model
fit
best <- fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"]
best # CP = Complexity Parameter of the tree

# build the simplest decision tree
Loan.DT <- C5.0(train_dat[-16], train_dat[,16])
Loan.DT # display simple facts about the tree
summary(Loan.DT) # display detailed information about the tree
plot(Loan.DT)

# apply the model to make predictions
DT_pred <- predict(Loan.DT, test_set)

# evaluating the models Performances with CrossTabulation & Confusion Matrix
library(gmodels)
library(caret)
CrossTable(test_set[,16], DT_pred, prop.chisq = FALSE, prop.c = FALSE)
confusionMatrix(DT_pred, test_set[,16], mode = "everything")
auc(test_set$Status, as.numeric(DT_pred))

# Construct variable importance plot
vip(Loan.DT, aesthetics = list(colour="brown", fill="darkgreen"),
    Main = "Variance of Importance for Variables")
vi(Loan.DT)

# improving model performance BY pruning the tree to simplify and/or avoid over-fitting
set.seed(12345)
DT.prune <- C5.0(train_dat[-16], train_dat$Status,
                 control = C5.0Control(minCases = 19))
DT.prune
summary(DT.prune)
plot(DT.prune)
# apply the model to make predictions
DTprune.pred <- predict(DT.prune, test_set)
```

This code section showcases the development and assessment of Decision Tree models for classification purposes. It begins by fitting a Decision Tree to training data, visualizing its structure, and identifying optimal complexity parameters. The code then builds a simpler tree, evaluates its performance, and generates variable importance plots. Additionally, it prunes the tree to prevent overfitting, evaluates the pruned model, and presents an ROC curve with AUC values. This comprehensive approach provides insights into model behavior, performance metrics, and the impact of predictors on decisions.

4.4.6 Algorithm Building for Artificial Neural Network

```
#-------------------------Section 15---------------------------------
library(NeuralNetTools)
library(nnet)
#*********************Artificial Neural Network (ANN)************************
set.seed(12345)
# create ann model
Loan.ANN <-nnet(Status ~ .,data = train_dat, size=5, decay=5e-4, maxit=100)
plotnet(Loan.ANN)
summary(Loan.ANN) #Summary of the model

#Fitting the model to make prediction
ann.pred1 <- predict(Loan.ANN, test_set)
ANN.prediction <- as.numeric(ann.pred1 > 0.5)

# evaluating the models Performances with CrossTabulation & Confusion Matrix
library(gmodels)
library(caret)
CrossTable(test_set[,16], ANN.prediction)
confusionMatrix(as.factor(ANN.prediction), test_set[,16], mode = "everything")
auc(test_set$Status, as.numeric(ANN.prediction))

#Plotting ROC Curve for the Model with AUC
library(pROC)
library(ROCR)
ANN_Perf <- roc(test_set[,16]~ as.numeric(ANN.prediction))
plot(ANN_Perf,plot=TRUE, print.auc=TRUE,
     col="red", lwd=4, legacy.axes=TRUE,main="ROC Curve of Artificial Neural Network (ANN)")

# Construct variable importance plot
vip(Loan.ANN, aesthetics = list(colour="brown", fill="darkgreen"),
    Main = "Variance of Importance for Variables")
vi(Loan.ANN)
```

This code section constructs and assesses an Artificial Neural Network (ANN). It loads libraries, creates an ANN model with "nnet" function, visualizes the network using "plotnet," and displays the model summary. The model predicts test data, evaluated via CrossTabulation, Confusion Matrix, and AUC metrics. An ROC curve shows AUC value. Variable importance plots demonstrate predictor significance. The code comprehensively analyzes the ANN's performance, effectiveness, and variable impact.

CHAPTER FIVE

5.0 RESULT INTERPRETATION

5.1 Data Exploration

Exploration of the dataset was conducted using both Microsoft Power BI and R-Studio. Data exploration serves as a pivotal phase within the data analysis journey, entailing the investigation and comprehension of a dataset's traits, patterns, and interconnections. This process yields insights, spotlights trends, and empowers well-informed decision-making.

Data Exploration in PowerBI

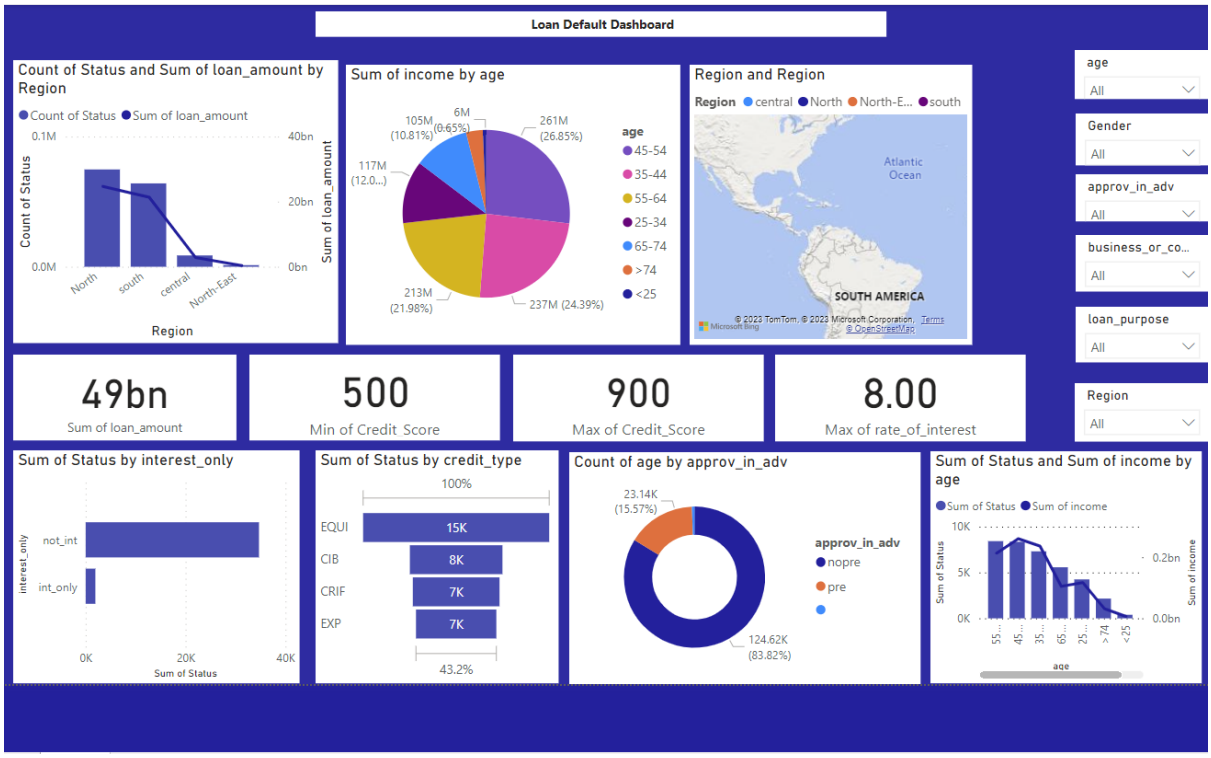


Fig 5.1: A Powerbi dashboard showing loan default datasets.

Several observations can be gleaned from the presented dashboard:

- The cumulative disbursed amount reached 49 billion.
- The credit scores ranged from a minimum of 500 to a maximum of 900.
- The Northern region exhibited the highest loan approval rate, closely trailed by the Southern region. In contrast, the Central and Northeastern regions showcased comparatively lower loan approval rates.

5.2 Data Exploration in R

5.2.1 Data Summary statistics and Data cleaning

Checking for missing Values

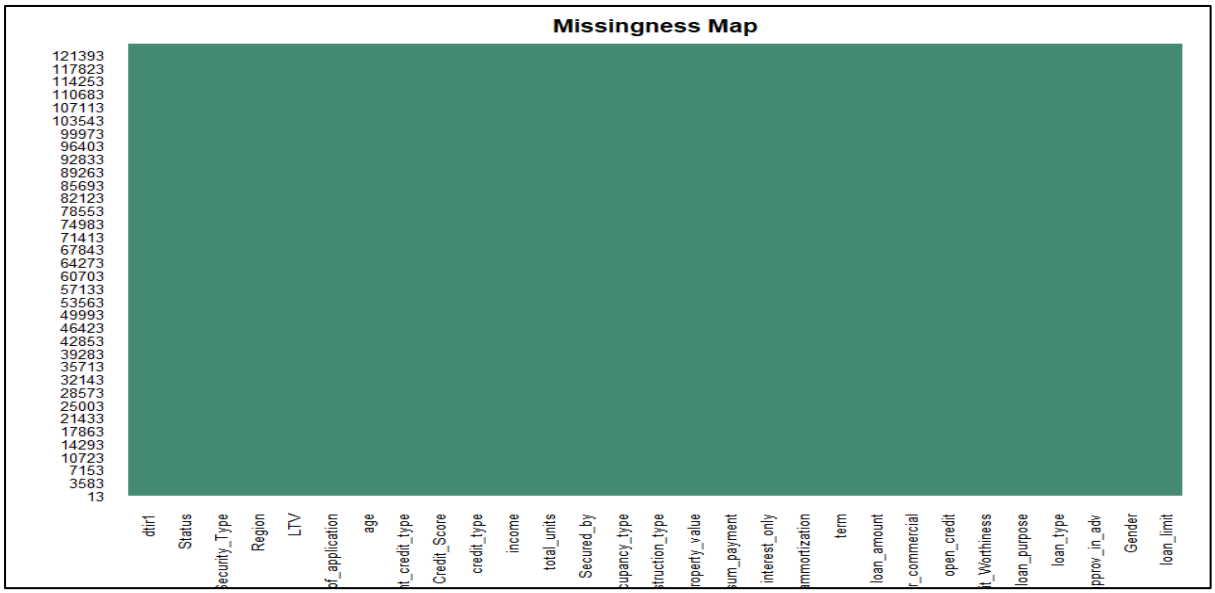


Fig 5.2: missingness map showing all observations.

Missing data is a common issue encountered in large datasets (Honaker & King, 2010). The dataset used for this analysis had a lot of missing data which was cleaned on R studio by running "loan <- na.omit(loan.dat)"The utilization of a library called Amelia aided in the identification of missing data. This library facilitates the visualization of the complete dataset and offers insights into both the present and absent data. After reading the data using Amelia, the map shows that there were no missing data.

5.2.2 Data Visualization

Class Distribution

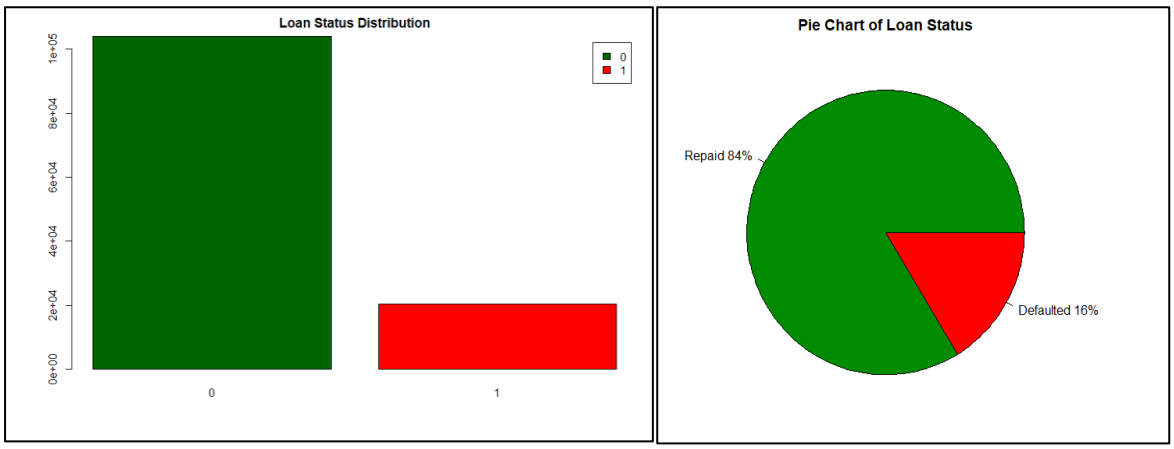


Fig 5.3: Repaired and Defaulted distribution of Class in R. (a) Bar chart (b) Pie Chart

This analysis provides a perspective on the distribution of the "status" variable, illustrating that the dataset contains a larger portion of instances with a status value of "0" in comparison to "1". The result reveals that 83.67% (104,092) of applicants successfully repaid their loans, while 16.33% (20,316) defaulted on their payments. Such analytical insights are essential for comprehending data composition and have a significant impact on decision-making and the development of model training strategies.

Distribution of Variables

Age

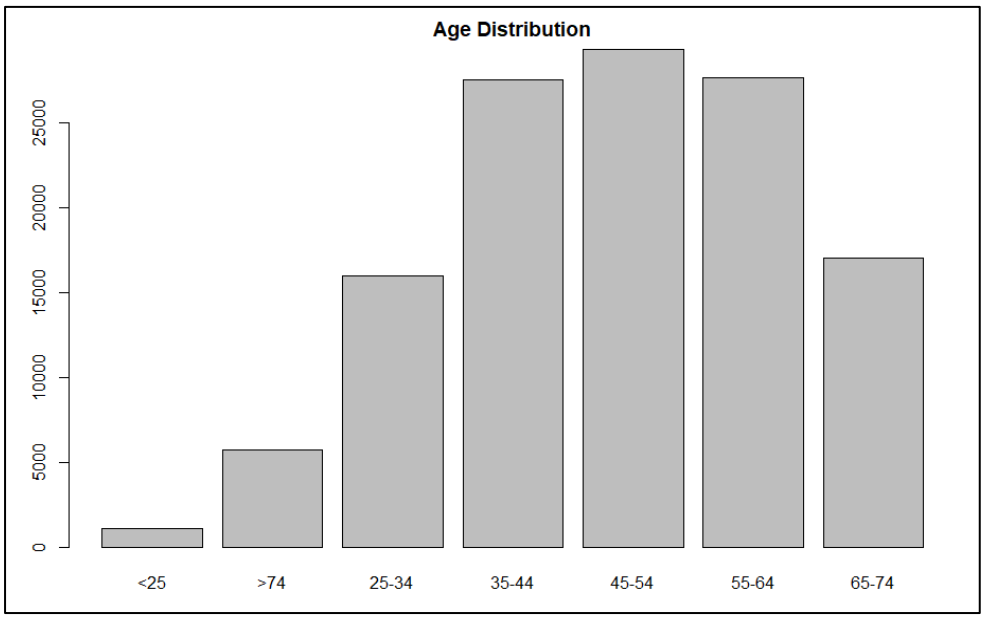


Fig 5.3: A bar plot of Age distribution in R

From the statistical breakdown within the dataset, it is evident that different age groups exhibit varying application patterns for personal loans. Specifically, the distribution reveals that:

- Individuals below 25 years old (<25) constitute 0.91% (1,128) of the applicants.
- The age range of 25-34 captures 12.83% (15,962) of the applicants.
- Those between 35-44 years old account for 22.12% (27,524) of applicants.
- The group aged 45-54 represents 23.59% (29,345) of applicants.
- The 55-64 age bracket comprises 22.25% (27,680) of applicants.
- Applicants aged 65-74 make up 13.71% (17,051) of the total.
- Individuals above 74 years old (>74) encompass 4.60% (5,718) of applicants.

This distribution analysis signifies distinct application tendencies among different age groups. Specifically, the younger age group (<25), which typically includes young adults, exhibits fewer personal loan applications. Similarly, the older age group (≥75), typically characterized as elderly individuals, also displays relatively fewer personal loan applications. Conversely, the age range of 45-54 emerges as the group with the highest rate of personal loan applications within the dataset

Gender

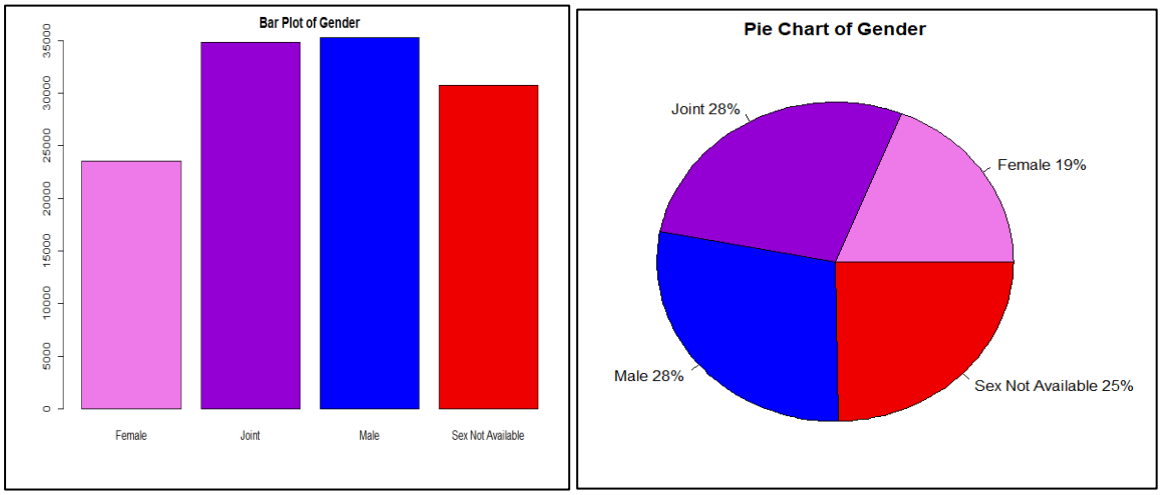


Fig 5.4: Gender distribution in R (a) Bar chart (b) Pie Chart.

The dataset illustrates diversity in the distribution of gender categories, encompassing Female, Male, Joint, and instances where gender information is unavailable. The outcomes indicate that 18.94% (23,562) were identified as females, 28.36% (35,279) were males, 27.99% (34,828) were categorized as Joint applicants, and 24.71% (30,739) had unspecified gender information. This analysis demonstrates that the highest proportion of loan applicants were males, while females constituted the lowest percentage. Additionally, 28% of applicants reported joint applications, while 25% of applicants did not provide gender information.

Loan approval in advance and Credit worthiness

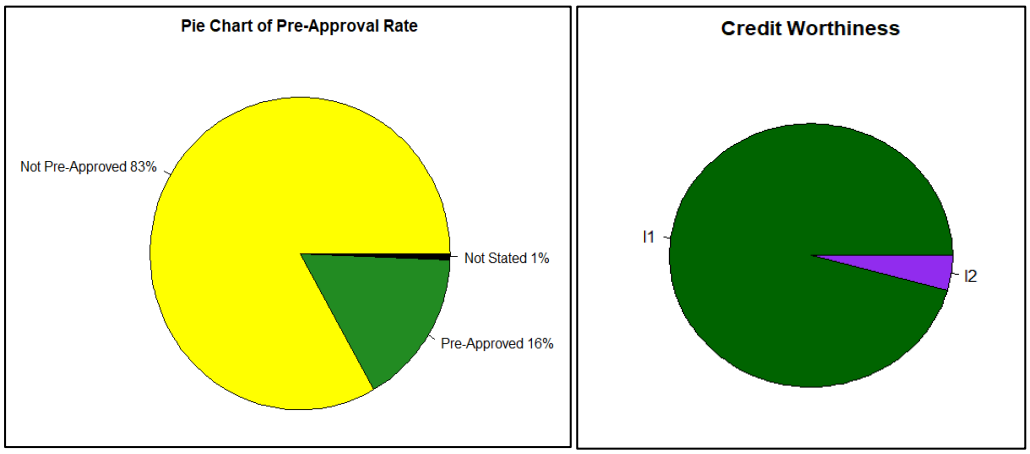


Fig 5.5: A pie chart representation of loan approval in advance and credit worthiness.

The analysis indicates that 16.19% (20,139) of applicants were granted pre-approval for the loan, highlighting a significant portion of successful pre-approvals. Conversely, a notable proportion of applicants of 83.19% (103,489) did not receive pre-approval, accounting for the majority. Additionally, a small portion of 0.63% (780) did not specify whether pre-approval was obtained or not. This analysis holds importance in comprehending pre-approval rates among loan applicants, offering insights into the efficacy of pre-approval procedures and their influence on application results.

Accordingly, based on the dataset, credit worthiness is indicated by credit scores for the majority, constituting 95.72% (119,088) of loan applicants, while a smaller portion of 4.28% (5,320) lacks credit worthiness due to their credit scores.

Distribution of Regions by Loan Status

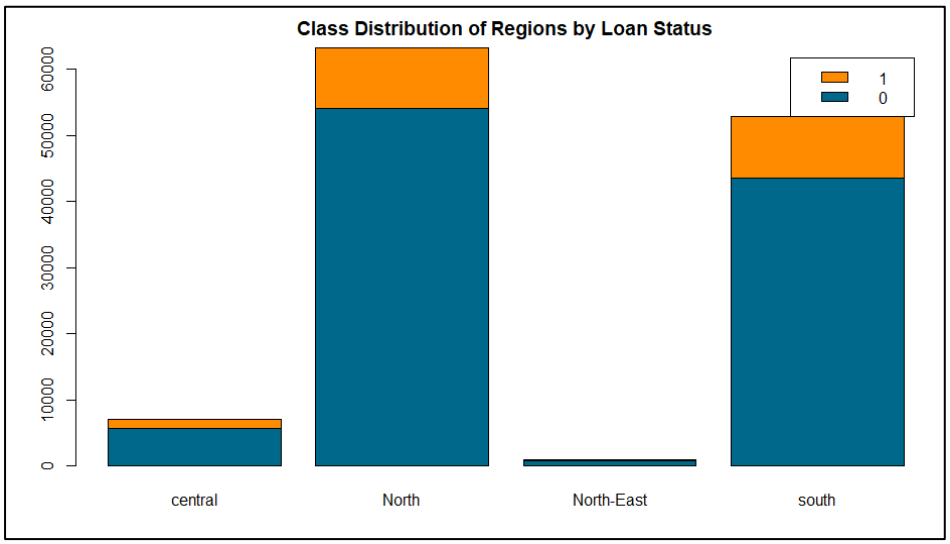


Fig 5.6: A bar chart showing the class distribution of regions by Loan status

As depicted by the bar plot, the "North" region stands out with the highest count of applicants (63,297). Among them, the "North" region also records the highest number of loan repayments (54,081) and defaults (9,216). The "South" region follows closely with 52,960 applicants, of which 43,528 successfully repaid their loans, and 9,432 defaulted. The "Central" region accounts for 7,147 applicants, among whom 5,699 repaid while 1,448 defaulted. In contrast, the "North-East" region reports the lowest number of applicants (1,004), with 784 applicants repaying and 220 applicants defaulting on their loans.

Distribution of Loan with Interest by Status

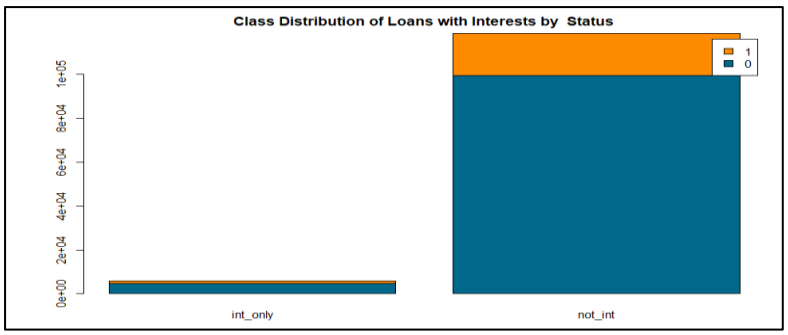


Fig5.7: Bar chart showing the distribution of loan by loan status

The bar plot indicates a lower frequency of loans with interest compared to loans without interest. Among loans without interest, there are 118,574 instances, of which 99,254 were repaid and 19,320 defaulted. Additionally, there are 5,834 instances of loans with interest, out of which 4,838 were repaid and 996 defaulted.

Age group by their income

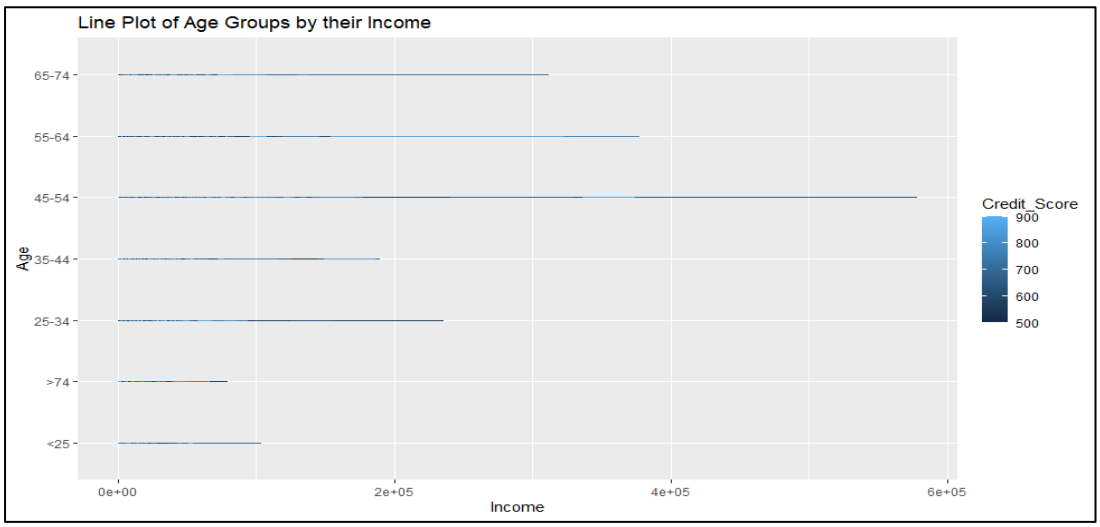
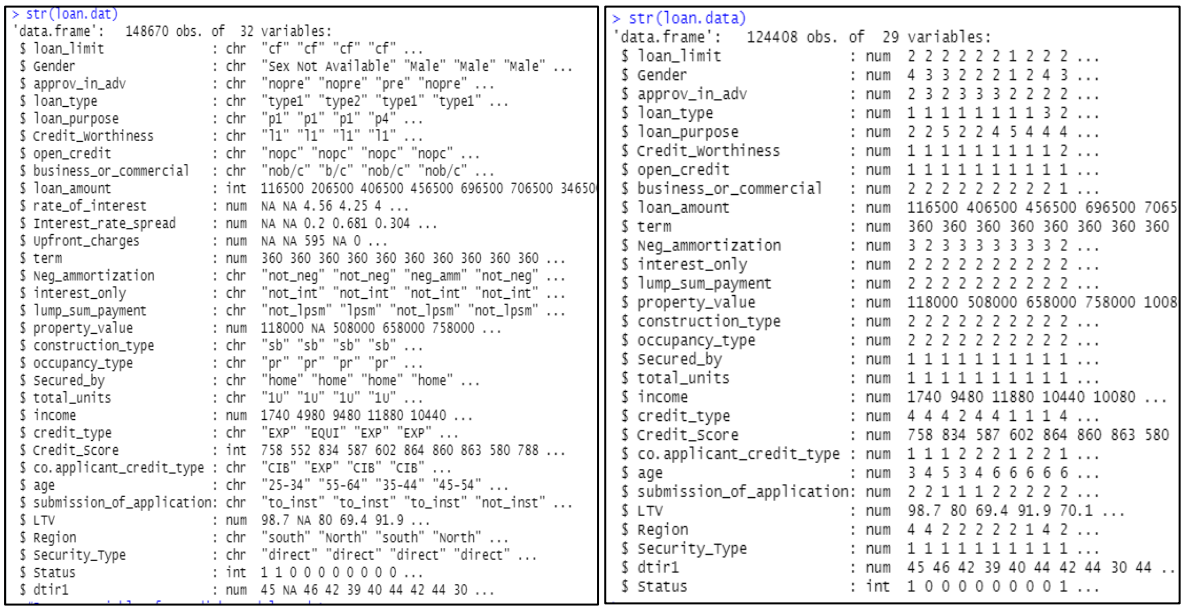


Fig 5.8: A line plot showing various age groups by their income.

The provided line plot depicts the relationship between the income of loan applicants and their age groups, as measured by their credit scores. According to the plot, individuals aged 45 to 54 exhibited the highest income relative to their credit scores, with the next highest income observed among those aged 55 to 64. In contrast, applicants aged above 74 displayed the lowest income based on their credit scores, and this was followed by applicants below 25 years old.

5.2.3 Data Transformation and box plot Visualization



```
> str(loan.dat)
'data.frame':   148670 obs. of  32 variables:
 $ loan_limit               : chr  "cf" "cf" "cf" "cf" ...
 $ Gender                   : chr  "Sex Not Available" "Male" "Male" "Male" ...
 $ approv_in_adv            : chr  "nopre" "nopre" "pre" "nopre" ...
 $ loan_type                : chr  "type1" "type2" "type1" "type1" ...
 $ loan_purpose             : chr  "p1" "p1" "p1" "p4" ...
 $ Credit_Worthiness        : chr  "l1" "l1" "l1" "l1" ...
 $ open_credit              : chr  "nopc" "nopc" "nopc" "nopc" ...
 $ business_or_commercial   : chr  "nob/c" "b/c" "nob/c" "nob/c" ...
 $ loan_amount              : int  116500 206500 406500 456500 696500 706500 34650
 $ rate_of_interest         : num  NA NA 4.56 4.25 4 ...
 $ Interest_rate_spread     : num  NA NA 0.2 0.681 0.304 ...
 $ Upfront_charges          : num  NA NA 595 NA 0 ...
 $ term                     : num  360 360 360 360 360 360 360 360 360 360 ...
 $ Neg_ammortization        : chr  "not_neg" "not_neg" "neg_amm" "not_neg" ...
 $ interest_only            : chr  "not_int" "not_int" "not_int" "not_int" ...
 $ lump_sum_payment         : chr  "not_lpsm" "lpsm" "not_lpsm" "not_lpsm" ...
 $ property_value           : num  118000 NA 508000 658000 758000 ...
 $ construction_type        : chr  "sb" "sb" "sb" "sb" ...
 $ occupancy_type           : chr  "pr" "pr" "pr" "pr" ...
 $ Secured_by               : chr  "home" "home" "home" "home" ...
 $ total_units              : chr  "1u" "1u" "1u" "1u" ...
 $ income                   : num  1740 4980 9480 11880 10440 ...
 $ credit_type              : chr  "EXP" "EQUI" "EXP" "EXP" ...
 $ Credit_Score             : int  758 552 834 587 602 864 860 863 580 788 ...
 $ co.applicant_credit_type : chr  "CIB" "EXP" "CIB" "CIB" ...
 $ age                      : chr  "25-34" "55-64" "35-44" "45-54" ...
 $ submission_of_application: chr  "to_inst" "to_inst" "to_inst" "not_inst" ...
 $ LTV                      : num  98.7 NA 80 69.4 91.9 ...
 $ Region                   : chr  "south" "North" "south" "North" ...
 $ Security_Type            : chr  "direct" "direct" "direct" "direct" ...
 $ Status                   : int  1 1 0 0 0 0 0 0 0 0 ...
 $ dtir1                    : num  45 NA 46 42 39 40 44 42 44 30 ...
```

```
> str(loan.data)
'data.frame':   124408 obs. of  29 variables:
 $ loan_limit               : num  2 2 2 2 2 2 2 1 2 2 2 ...
 $ Gender                   : num  4 3 3 2 2 2 2 1 2 4 3 ...
 $ approv_in_adv            : num  2 3 2 3 3 3 3 2 2 2 2 ...
 $ loan_type                : num  1 1 1 1 1 1 1 1 3 2 ...
 $ loan_purpose             : num  2 2 5 2 2 4 5 4 4 4 ...
 $ Credit_Worthiness        : num  1 1 1 1 1 1 1 1 1 2 ...
 $ open_credit              : num  1 1 1 1 1 1 1 1 1 1 ...
 $ business_or_commercial   : num  2 2 2 2 2 2 2 2 2 1 ...
 $ loan_amount              : num  116500 406500 456500 696500 7065
 $ term                     : num  360 360 360 360 360 360 360 360
 $ Neg_ammortization        : num  3 2 3 3 3 3 3 3 3 2 ...
 $ interest_only            : num  2 2 2 2 2 2 2 2 2 2 ...
 $ lump_sum_payment         : num  2 2 2 2 2 2 2 2 2 2 ...
 $ property_value           : num  118000 508000 658000 758000 1008
 $ construction_type        : num  2 2 2 2 2 2 2 2 2 2 ...
 $ occupancy_type           : num  2 2 2 2 2 2 2 2 2 2 ...
 $ Secured_by               : num  1 1 1 1 1 1 1 1 1 1 ...
 $ total_units              : num  1 1 1 1 1 1 1 1 1 1 ...
 $ income                   : num  1740 9480 11880 10440 10080 ...
 $ credit_type              : num  4 4 4 2 4 4 1 1 1 4 ...
 $ Credit_Score             : num  758 834 587 602 864 860 863 580
 $ co.applicant_credit_type : num  1 1 1 2 2 2 1 2 2 1 ...
 $ age                      : num  3 4 5 3 4 6 6 6 6 6 ...
 $ submission_of_application: num  2 2 1 1 1 2 2 2 2 2 ...
 $ LTV                      : num  98.7 80 69.4 91.9 70.1 ...
 $ Region                   : num  4 4 2 2 2 2 2 1 4 2 ...
 $ Security_Type            : num  1 1 1 1 1 1 1 1 1 1 ...
 $ dtir1                    : num  45 46 42 39 40 44 42 44 30 44 ..
 $ Status                   : int  1 0 0 0 0 0 0 0 0 1 ...
```

Fig 5.9: Structure of the data before and after data transformation in R (a)before (b) after

The structures of the data in the fig above shows variables and their different data types. Transforming a data frame from character and string formats to numeric values provides crucial benefits across data-related activities. Numeric data facilitates efficient analysis, supports machine learning algorithms, improves visualization, aids in feature engineering, and ensures compatibility with modelling frameworks. Numeric representation enhances efficiency, simplifies dimensionality reduction, and promotes interpretability, leading to well-informed decision-making. This conversion is fundamental for harnessing data-driven insights and achieving enhanced outcomes in fields such as data science, machine learning, and business analysis.

Boxplot

The boxplot method, as detailed in the book "Exploratory Data Analysis" published by Addison-Wesley in 1977, presents a graphical technique for detecting outliers. It's notable for its simplicity and its exclusion of extreme potential outliers when calculating dispersion measures. The inner and outer fences are determined based on hinges (or quartiles), ensuring they remain unaffected by a small

number of extremely high or low values. This safeguards against the problem of "masking," which could lead to the oversight of certain outliers.

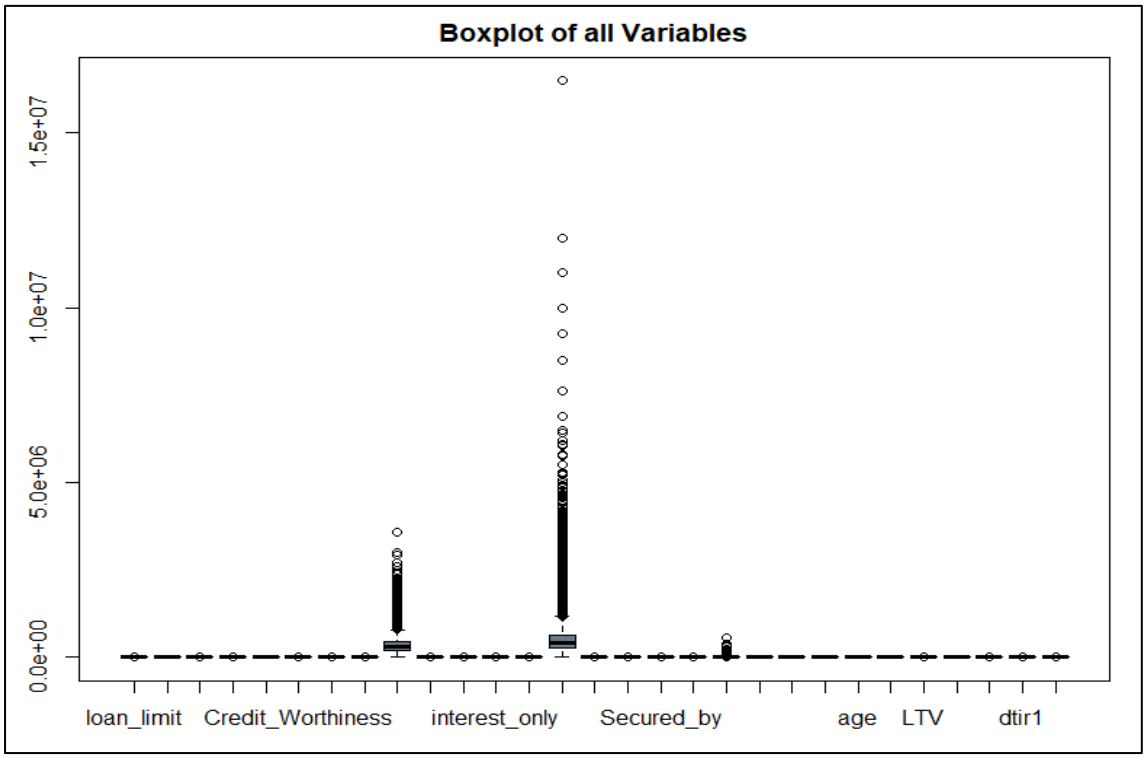


Fig 5.10: Boxplot of all variables of the data frame in R.

The provided plot illustrates the distribution of data points in the data frame, showcasing quartiles, mean, median, and maximum values. Notably, the boxplot's scale is impacted by the relatively larger values of loan_amount and property_value compared to other variables. To accommodate all variables within the boxplot's range, scaling or normalization of the variables is necessary.

Boxplot of Selected variables

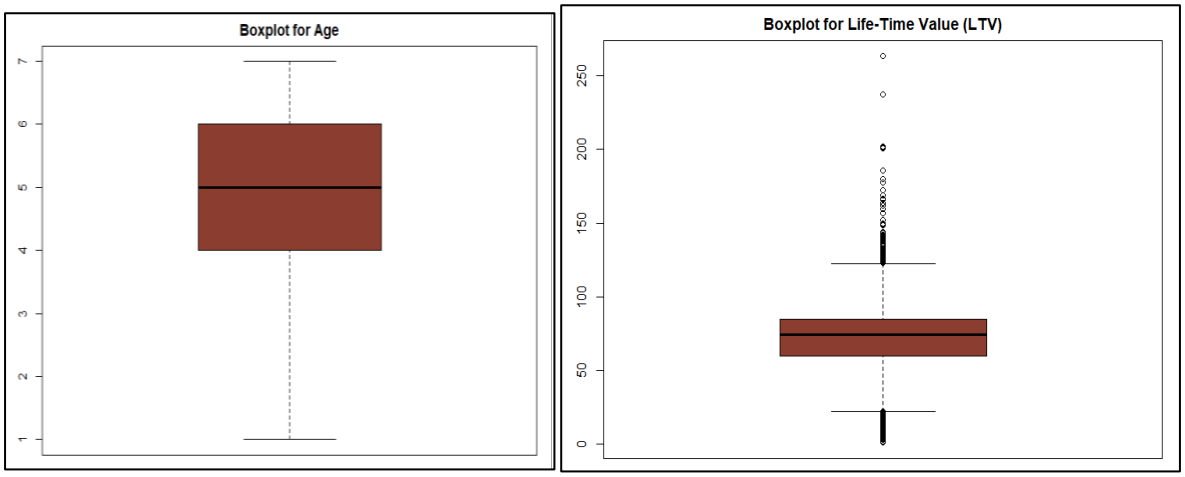


Fig 5.9: Boxplots of Age and Life-Time Value. (a)Age (b) Life-Time Value.

Above are boxplots representing the distribution of age, suggesting a normal distribution pattern. Additionally, the boxplot for Life-Time Value reveals the presence of outliers.

5.2.4 Correlation Analysis

Correlation, also referred to as correlation analysis, refers to the examination of the association between two or more quantitative variables. (Aggarwal & Ranganathan, 2016). This analysis assumes of a linear relationship between these variables. The outcome of correlation analysis is represented by a correlation coefficient, which ranges from -1 to +1. A correlation coefficient of +1 signifies a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and a coefficient of zero suggests no linear relationship between the studied variables.

```
#Check correlations between all variables using Matrix
library(corrplot)
loan.data.cor = cor(loan.data)
loan.data.cor
#Order by alphabet and arranged in order of correlation
corrplot(loan.data.cor, method = 'number',  diag=FALSE)
```

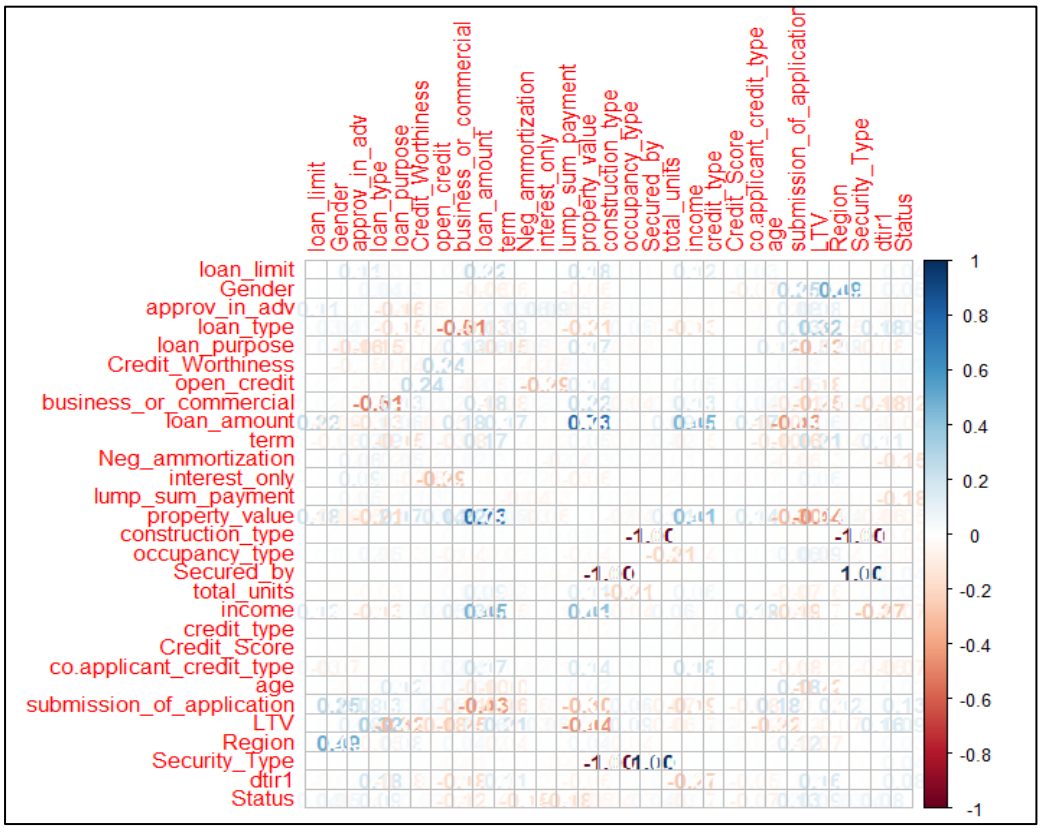


Fig 5.11: (b)Correlation matrix plot of independent variables in R.

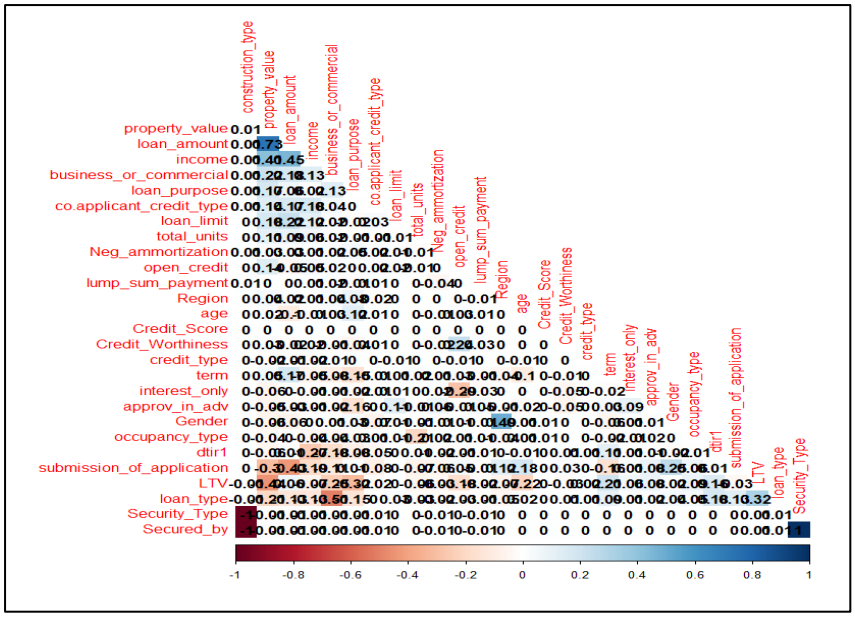


Fig 5.11: (b)Correlation matrix plot of independent variables in R.

While bearing in mind that correlation does not necessarily imply causation, the plots illustrates certain relationships among variables. For instance, the dependent variable (status) shows a weak negative correlation with "business_or_commercial" (-0.119) and "Neg_ammortization" (-0.147), while a weak positive correlation with "submission_of_application" (0.129). Conversely, no apparent association with the dependent variable is seen in the other independent variables.

The plot also reveals some weak negative and positive correlations between various independent variables. Notably, a strong positive correlation of 0.73 exists between "Loan_amount" and "Property_value." Additionally, "income" demonstrates correlations of 0.40 and 0.45 with "property_value" and "loan_amount," respectively. An intriguing finding is the strong negative correlation of -1 between "Secured_by" and "Construction_type," which extends to "Secured_type" and "Construction_type" as well. This suggests a consistent linear relationship where an increase in one variable corresponds to a proportional decrease in the other.

Conversely, a strong positive correlation between "Secured_by" and "Secured_type" implies that an increase in one variable is associated with a proportional increase in the other. It's crucial to remember that these correlations offer insights into relationships but not necessarily causal connections.

Internal Correlation

Internal correlation refers to an evaluation of interdependence within a collection of variables, encompassing canonical correlations, multiple correlations, and product moment correlations.(George and Jorge, 1989)

The function "cor.test" was employed to conduct Pearson's correlation analysis, aiming to examine the relationship between the independent variables and the dependent variable "Status." This analysis shows both the p-value of the correlation and the level of significance of the correlation coefficient. The confidence level of 95% and P value of 0.05 was used.

```
> cor.test(Status, Gender)

        Pearson's product-moment correlation

data:  Status and Gender
t = 17.717, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.04462185 0.05570748
sample estimates:
       cor
0.05016621
```

```
> cor.test(Status, age)

        Pearson's product-moment correlation

data:  Status and age
t = 8.7776, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.01932413 0.03043085
sample estimates:
       cor
0.02487825
```

```
> cor.test(Status, loan_amount)

        Pearson's product-moment correlation

data:  Status and loan_amount
t = -14.386, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.04630055 -0.03520540
sample estimates:
        cor
-0.04075423
```

```
> cor.test(Status, Credit_Score)

        Pearson's product-moment correlation

data:  Status and Credit_Score
t = 1.3369, df = 124406, p-value = 0.1813
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.001766501  0.009346938
sample estimates:
        cor
0.003790336
```

```
> cor.test(Status, term)

        Pearson's product-moment correlation

data:  Status and term
t = 3.0926, df = 124406, p-value = 0.001984
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.003211172 0.014323916
sample estimates:
        cor
0.008767814
```

```
> cor.test(Status, Credit_Worthiness)

        Pearson's product-moment correlation

data:  Status and Credit_Worthiness
t = 8.5412, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.01865431 0.02976139
sample estimates:
       cor
0.0242086
```

```
> cor.test(Status, LTV)

        Pearson's product-moment correlation

data:  Status and LTV
t = 32.107, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.08514031 0.09616258
sample estimates:
       cor
0.09065422
```

```
> cor.test(Status, credit_type)

        Pearson's product-moment correlation

data:  Status and credit_type
t = 2.1897, df = 124406, p-value = 0.02854
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.0006513939 0.0117645640
sample estimates:
        cor
0.006208171
```

```
> cor.test(Status, dtir1)

        Pearson's product-moment correlation

data:  Status and dtir1
t = 27.559, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.07237118 0.08341734
sample estimates:
       cor
0.07789665
```

```
> cor.test(Status, Region)

        Pearson's product-moment correlation

data:  Status and Region
t = 9.7098, df = 124406, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.02196497 0.03307016
sample estimates:
        cor
0.02751841
```

Fig 5.12: A table showing the selected variables after internal correlation.

Variable	Rho's Coefficient	P-Value	Interpretation	Null Hypothesis
Gender	0.05016621	< 2.2e-16	Weak positive correlation	Reject Null hypothesis
Age	0.02487825	< 2.2e-16	Weak positive correlation	Reject Null hypothesis

Loan_amount	-0.04075423	< 2.2e-16	Weak negative correlation	Reject Null hypothesis
Credit_Score	0.003790336	0.1813	Weak positive correlation	Accept Null hypothesis
Term	0.008767814	0.001984	Weak positive correlation	Reject Null hypothesis
Credit_worthiness	0.0242086	< 2.2e-16	Weak positive correlation	Reject Null hypothesis
LTV	0.0.9065422	< 2.2e-16	Weak positive correlation	Reject Null hypothesis
Credit_type	0.006208171	0.02854	Weak positive correlation	Reject Null hypothesis
Dtir1	0.07789665	< 2.2e-16	Weak positive correlation	Reject Null hypothesis
Region	0.02751841	< 2.2e-16	Weak positive correlation	Reject Null hypothesis

Partial Correlation

Partial correlation quantifies the degree of association between two variables, factoring in the influence of one or more additional variables. (Epskamp & Fried, 2018)This aids in uncovering the exclusive linkage between the primary variables, taking into account the joint influence exerted by the control variables. Partial correlation helps us understand how two variables are related when other factors are involved. Sometimes, these other factors can make it seem like the two variables are connected, even if they're not. Partial correlation helps us see the true connection between the two variables by removing the effects of these other factors.

```
> pcor.test(Status, Gender, age)
    estimate      p.value statistic      n gp  Method
1 0.04997879 1.235107e-69  17.65012 124408  1 pearson
```

```
> pcor.test(Status, Credit_Score, Credit_Worthiness)
     estimate   p.value statistic      n gp  Method
1 0.003782666 0.1821416  1.334197 124408  1 pearson
```

```
> pcor.test(Status, property_value, Region)
     estimate      p.value statistic      n gp  Method
1 -0.04785502 5.455557e-64 -16.89835 124408  1 pearson
```

```
> pcor.test(Status, dtir1, LTV)
    estimate       p.value statistic      n gp  Method
1 0.06476083 1.017447e-115   22.8899 124408  1 pearson
```

```
> pcor.test(Status, loan_amount, term)
     estimate      p.value statistic      n gp  Method
1 -0.04293899 7.348469e-52 -15.15903 124408  1 pearson
```

```
> pcor.test(Status, Security_Type, approv_in_adv)
     estimate      p.value statistic      n gp  Method
1 0.03509294 3.293282e-35  12.38529 124408  1 pearson
```

Fig 5.13: A table showing the results of the Pearson correlation test.

According to the table above, most of the variables have a p-value of less than 0.05 which makes us reject the null hypothesis in the test and conclude that there is a significant correlation between the variables. The figure above demonstrates no indication of a spurious association among the variables.

When the p-value of an internal correlation is less than 0.5, it indicates that the observed correlation likely isn't due to chance alone. This suggests a genuine relationship between the studied variables. Consequently, in some cases, highly correlated variables might be excluded from the analysis. This is because strongly correlated variables essentially convey the same information and, therefore, having both might not provide additional insights, therefor, omitting it will simplify the analysis and will avoid overfitting. For instance, if variables like Credit Score and Credit Worthiness are closely related, it's reasonable to remove Credit Score as it duplicates the information already captured by Credit

Worthiness Certain variables will be excluded prior to constructing the models, resulting in the utilization of 16 remaining variables to enhance model efficiency.

5.2.5 Removing Outliers

Outliers in data refer to data points that significantly differ from most of the values in a dataset. These exceptional points stand out due to their unusual characteristics. Detecting outliers is facilitated by using a graphical representation called a boxplot. When examining the boxplot, outliers can be identified as points that lie beyond the "whiskers" of the box, signifying their distinctness from the bulk of the data.

Addressing outliers is particularly important, as their presence can have a notable impact on the accuracy and effectiveness of classification models. Outliers can lead to skewed or misleading results, which is why they need to be properly managed.

In this context, the Interquartile Range (IQR) method was employed to identify outliers. This involves calculating the difference between the upper and lower quartiles of the data distribution. Data points falling below the lower quartile or exceeding the upper quartile are considered outliers. By removing these outlier data points, the dataset underwent a change in its structure. This led to a reduction in the number of observations within the new dataset, diminishing it from an initial count of 124,408 observations to a revised total of 119,255 observations.

5.2.6 Factor Analysis

Factor Analysis is a statistical method used when a researcher wants to identify distinct groups of variables within a single dataset that are relatively independent of each other. This technique helps to uncover underlying patterns and relationships among variables (Tabachnick *et al.,* 2013). Factor analysis proves to be highly advantageous in pinpointing the underlying factors that drive the variables by grouping together related variables within the same factor (Verma *et al.,* 2019). Although factor analysis and principal component analysis (PCA) share some similarities, they serve different purposes. PCA is primarily a descriptive method that focuses on transforming data to reduce dimensionality while preserving variance. On the other hand, factor analysis is used for exploration, aiming to reveal underlying latent factors that explain observed variable relationships.

Kaiser-Meyer-Olkin Test (KMO)

The KMO test is designed to assess whether data is appropriate for factor analysis. In simpler terms, it checks if the sample size is sufficient. This test evaluates how well the variables in the model and

the entire model itself are suited for analysis by measuring sampling adequacy (Pituch ,2016). Typically, a dataset is deemed appropriate for factor analysis when its adequacy value reaches 0.6 or above. Conversely, if the value falls below 0.5, it is generally seen as unsuitable.

KMO measure	Interpretation
KMO ≥ 0.90	Marvelous
0.80 ≤ KMO < 0.90	Meritorious
0.70 ≤ KMO < 0.80	Average
0.60 ≤ KMO < 0.70	Mediocre
0.50 ≤ KMO < 0.60	Terrible
KMO < 0.50	Unacceptable

Fig 5.13; Kaiser Meyer Olkin (KMO) level of acceptance by the adequacy value.

The illustration provided above showcases a scale that portrays the degree of acceptance evident in a KMO test outcome. This scale commonly spans from 0 to 1.

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = loan.data)
Overall MSA =  0.5
MSA for each item =
               loan_limit                    Gender             approv_in_adv                 loan_type
                      0.5                       0.5                       0.5                       0.5
             loan_purpose          Credit_Worthiness               open_credit     business_or_commercial
                      0.5                       0.5                       0.5                       0.5
              loan_amount                      term          Neg_ammortization             interest_only
                      0.5                       0.5                       0.5                       0.5
         lump_sum_payment            property_value          construction_type            occupancy_type
                      0.5                       0.5                       0.5                       0.5
               Secured_by               total_units                    income               credit_type
                      0.5                       0.5                       0.5                       0.5
             Credit_Score  co.applicant_credit_type                       age submission_of_application
                      0.5                       0.5                       0.5                       0.5
                      LTV                    Region             Security_Type                     dtir1
                      0.5                       0.5                       0.5                       0.5
                   Status
                      0.5
>
```

Fig 5.14: KMO adequacy result for all the attributes

The outcome of the KMO test above reveals an overall Measure of Sampling Adequacy (MSA) of 0.5. This value falls within the moderate range, indicating that the suitability for factor analysis is not optimal but could be taken into consideration if needed. Each of these values is also 0.5, suggesting that all variables are showing similar levels of adequacy for factor analysis. Typically, for dependable factor analysis, it's preferable to have a KMO value surpassing 0.6. However, to reduce overfitting and enhance the effectiveness of the model, certain variables will be eliminated, using the insight from correlation analysis.

5.2.7 Normalization

The primary objective of Normalization is to ensure uniformity in the scales of diverse features present in a dataset. Scaling variables to a uniform range is vital for enhancing model accuracy. This practice involves excluding the dependent variable. Through normalization, all variables receive equal importance, safeguarding against any inadvertent impact from a single variable on the model's performance. To enhance the dataset through normalization, the variables were scaled using the SoftMax function.

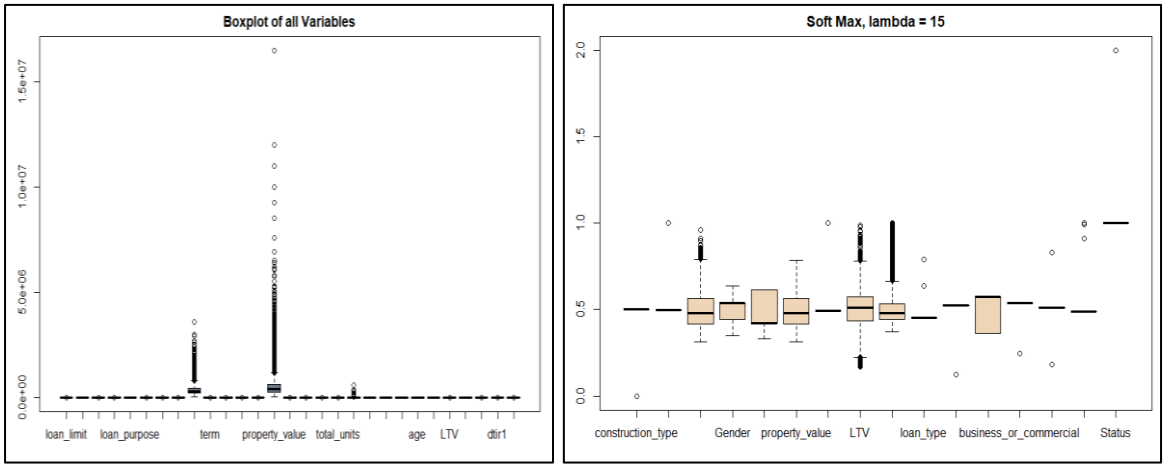


Fig 5.18: Boxplots of the transformed dataset (a) before normalizing; (b) Soft Max function.

Despite the normalization process, a few data outliers persistently influenced the boxplot visualizations of certain variables.

5.3 Model Building

Training and Testing

The process of training and testing involves the division of your dataset into two distinct segments, facilitating the evaluation of your model's accuracy and its ability to apply knowledge to new and unfamiliar data. During the training phase, a machine learning system is exposed to a substantial volume of accurately labelled data. This data is employed to instruct the model in recognizing patterns within the dataset and making predictions. The testing and training procedure serves to avert problems like model overfitting or underfitting. Overfitting arises when a model becomes overly complex, capturing noise from the training set and leading to subpar performance on the testing set. Conversely, underfitting occurs when a model is too simplistic, failing to capture the underlying data patterns and resulting in inadequate performance on both the training and testing sets.

The dataset was partitioned using a ratio of 70:30, allocating 70% for training models and 30% for testing models. The training set comprises 83,478 observations, while the testing set consists of 35,777 observations. To ensure uniformity in performance, a seed of "12345" was set for all models. In general, the separation of a dataset into training and testing sets involves using most of the data for training purposes, with a smaller portion allocated for testing.

Class Imbalance

Class imbalance refers to a in a classification problem where the classes you are trying to predict have significantly different numbers of instances. In other words, some classes have a much larger number of examples compared to others. This can potentially pose challenges for machine learning algorithms because they might become biased towards the majority class, leading to poor performance in predicting the minority class.

To counteract class imbalance, under-sampling was employed. This technique entails decreasing the number of instances in the majority class within an imbalanced dataset, with the goal of establishing a more equitable class distribution. The approach chosen for this purpose involved utilizing the 'ROSE()' function in R. In the new training set, there is an even distribution of 13,560 samples for both classes (0) and (1).

```
> #-----Class Imbalance-----------
> library("ROSE")
> #Under Sampling Technique
> train_dat<- ovun.sample(Status~., data=training_set,
+                         method="under", N=27120, seed=12345)$data
> table(train_dat$Status)

    0     1
13560 13560
```

Fig 5.19: The result of Under-Sampling in R

5.4 Machine Learning Algorithms

5.4.1 Logistic Regression (LR)

A logistic regression analysis was conducted after setting a seed of '12345'. The focus of this analysis was to examine the relationship between the dependent attribute (Status) and the independent attributes present in the dataset. The logistic regression model was constructed using the Generalized Linear Model (glm) approach, utilizing a binomial family distribution and a logit link function.

```
set.seed(12345)
# Fitting Logistic Regression to the Training set
classifier = glm(formula = Status ~ ., family = binomial, data = train_dat)
summary(classifier)

# Predicting the Test set results
prob_pred = predict(classifier, type = 'response', newdata = test_set[-16])
Loan_Logistic = ifelse(prob_pred > 0.5, 1, 0)
```

Fig 5.20: LR model with prediction in R

After conducting multiple regression models, this specific model displayed the highest accuracy, although the results were similar between the models. The True Positive (TP): 17,807, False Positive (FP): 12,172 False Negative (FN): 2,055 and True Negative (TN): 3,743. The model has an accuracy of 60.23%, which is the ratio of correctly predicted instances to the total number of instances while it has a specificity of 64.56%. Additionally, the model's AUC score was 0.62, indicating its competence in performance.

```
> CrossTable(x = test_set[,16], y = Loan_Logistic, prop.ch

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  35777

               | Loan_Logistic
test_set[, 16] |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             0 |     17807 |     12172 |     29979 |
               |     0.594 |     0.406 |     0.838 |
               |     0.897 |     0.765 |           |
               |     0.498 |     0.340 |           |
---------------|-----------|-----------|-----------|
             1 |      2055 |      3743 |      5798 |
               |     0.354 |     0.646 |     0.162 |
               |     0.103 |     0.235 |           |
               |     0.057 |     0.105 |           |
---------------|-----------|-----------|-----------|
  Column Total |     19862 |     15915 |     35777 |
               |     0.555 |     0.445 |           |
---------------|-----------|-----------|-----------|
```

```
> confusionMatrix(as.factor(Loan_Logistic),
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 17807  2055
         1 12172  3743

               Accuracy : 0.6023
                 95% CI : (0.5972, 0.6074)
    No Information Rate : 0.8379
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1406

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5940
            Specificity : 0.6456
         Pos Pred Value : 0.8965
         Neg Pred Value : 0.2352
              Precision : 0.8965
                 Recall : 0.5940
                     F1 : 0.7146
             Prevalence : 0.8379
         Detection Rate : 0.4977
   Detection Prevalence : 0.5552
      Balanced Accuracy : 0.6198

       'Positive' Class : 0
```

Fig. 5.21: (a) Cross table analysis (b) Confusion matrix for Logistic Regression.

5.4.2 K-Nearest Neighbours (KNN)

The k-Nearest Neighbors (KNN) technique is a straightforward yet impactful approach for classification. Various k values were tested during the model's training and prediction in However, the highest accuracy was achieved with the square root of the training dataset, where k was set to 153.

```
sqrt(23122) #Find the Square root of training dataset
#using K value as 153
set.seed(12345)
knn_pred = knn(train = train_dat[, -16], test = test_set[-16], cl = train_dat[, 16],
               k = 153)
```

Fig 5.22: KNN model in R with k=153

After experimenting with different iterative values for k, the models produced outcomes that were quite like each other. The counts are as follows: True positive = 20,444 True negatives = 3,462 False positive = 9,535 and False Negative = 2,336.

The model achieved an accuracy of around 66.82%, displaying a balanced performance between sensitivity at 68.19% and specificity at 59.73%. Additionally, the model's AUC score was 0.64, indicating its proficient performance.

```
> CrossTable(x = test_set[,16], y = knn_pred, prop.chis

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  35777

               | knn_pred
test_set[, 16] |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             0 |     20444 |      9535 |     29979 |
               |     0.682 |     0.318 |     0.838 |
               |     0.897 |     0.734 |           |
               |     0.571 |     0.267 |           |
---------------|-----------|-----------|-----------|
             1 |      2336 |      3462 |      5798 |
               |     0.403 |     0.597 |     0.162 |
               |     0.103 |     0.266 |           |
               |     0.065 |     0.097 |           |
---------------|-----------|-----------|-----------|
  Column Total |     22780 |     12997 |     35777 |
               |     0.637 |     0.363 |           |
---------------|-----------|-----------|-----------|
```

```
> confusionMatrix(as.factor(knn_pred), test
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 20442  2335
         1  9537  3463

               Accuracy : 0.6682
                 95% CI : (0.6633, 0.673)
    No Information Rate : 0.8379
    P-Value [Acc > NIR] : 1

                  Kappa : 0.186

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6819
            Specificity : 0.5973
         Pos Pred Value : 0.8975
         Neg Pred Value : 0.2664
              Precision : 0.8975
                 Recall : 0.6819
                     F1 : 0.7750
             Prevalence : 0.8379
         Detection Rate : 0.5714
   Detection Prevalence : 0.6366
      Balanced Accuracy : 0.6396

       'Positive' Class : 0
```

Fig. 5.23: (a) Cross table analysis (b) Confusion matrix for KNN

5.4.3 Support Vector Machine

A Support Vector Machine (SVM) is a versatile machine learning method that identifies the best possible line or boundary to separate different classes in data. Two Support Vector Machine (SVM) algorithms were employed for analysis, utilizing two distinct SVM kernel functions: SVM0, referred to as "vanilladot," and SVM1, referred to as "rbfdot." These kernels were constructed for modeling purposes through the utilization of the kernlab() function within the R programming environment. In terms of predictive modeling, the linear "rbfdot" kernel, which is the SVM1 exhibited better performance.

```
#--------------# Radial Basis-Gaussian (RBFDot Kernel string)--------------------
# explore improvements of the model byusing a non-linear kernel function
set.seed(12345)
svm1 <- ksvm(Status ~ ., data = train_dat, kernel = "rbfdot", type = "C-svc")
svm1 # look at basic information about the model

# apply the model to make predictions
svm_pred1 <- predict(svm1, test_set[-16])
table(svm_pred1, test_set[,16])
```

```
> svm1 # look at basic information about the model
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.08269101935893

Number of Support Vectors : 20265

Objective Function Value : -19642.1
Training error : 0.335324
```

Fig 5.24: SVM0 model summary with prediction in R.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  35777


                 | svm_pred1
test_set[, 16] |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             0 |     19878 |     10101 |     29979 |
               |     0.663 |     0.337 |     0.838 |
               |     0.907 |     0.729 |           |
               |     0.556 |     0.282 |           |
---------------|-----------|-----------|-----------|
             1 |      2036 |      3762 |      5798 |
               |     0.351 |     0.649 |     0.162 |
               |     0.093 |     0.271 |           |
               |     0.057 |     0.105 |           |
---------------|-----------|-----------|-----------|
  Column Total |     21914 |     13863 |     35777 |
               |     0.613 |     0.387 |           |
---------------|-----------|-----------|-----------|
```

```
> confusionMatrix(svm_pred1, test_set[,16], mode =
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 19878  2036
         1 10101  3762

               Accuracy : 0.6608
                 95% CI : (0.6558, 0.6657)
    No Information Rate : 0.8379
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1998

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6631
            Specificity : 0.6488
         Pos Pred Value : 0.9071
         Neg Pred Value : 0.2714
              Precision : 0.9071
                 Recall : 0.6631
                     F1 : 0.7661
             Prevalence : 0.8379
         Detection Rate : 0.5556
   Detection Prevalence : 0.6125
      Balanced Accuracy : 0.6560

       'Positive' Class : 0
```

Fig. 5.25: (a) Cross table analysis (b) Confusion matrix for SMV

The model demonstrates True Positives of 19,878 and False Positives of 10,101, alongside True Negatives amounting to 3,762 and False Negatives totalling 2,036. The model achieves an accuracy rate of 66% and exhibits a specificity value of 64%.

5.4.4 Decision Tree

Decision tree is a graphical representation of a series of decisions and their potential outcomes in a tree-like structure. Decision trees are used to model decisions and their consequences in a way that can be easily understood and interpreted. Multiple models were built, which also included pruning. The model performed better with higher accuracy after the model was pruned.

```
# improving model performance BY pruning the tree to simplify
set.seed(12345)
DT.prune <- C5.0(train_dat[-16], train_dat$Status,
                 control = C5.0Control(minCases = 19))
DT.prune
summary(DT.prune)
plot(DT.prune)
# apply the model to make predictions
DTprune.pred <- predict(DT.prune, test_set)
```

Fig 5.26: Pruned Model for decision tree in R

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  35777


                | DTprune.pred
test_set[, 16] |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             0 |     20342 |      9637 |     29979 |
               |     0.679 |     0.321 |     0.838 |
               |     0.569 |     0.269 |           |
---------------|-----------|-----------|-----------|
             1 |      2141 |      3657 |      5798 |
               |     0.369 |     0.631 |     0.162 |
               |     0.060 |     0.102 |           |
---------------|-----------|-----------|-----------|
  Column Total |     22483 |     13294 |     35777 |
---------------|-----------|-----------|-----------|
```

```
> confusionMatrix(DTprune.pred, test_set[,16
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 20342  2141
         1  9637  3657

               Accuracy : 0.6708
                 95% CI : (0.6659, 0.6757)
    No Information Rate : 0.8379
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2033

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6785
            Specificity : 0.6307
         Pos Pred Value : 0.9048
         Neg Pred Value : 0.2751
              Precision : 0.9048
                 Recall : 0.6785
                     F1 : 0.7755
             Prevalence : 0.8379
         Detection Rate : 0.5686
   Detection Prevalence : 0.6284
      Balanced Accuracy : 0.6546

       'Positive' Class : 0
```

Fig. 5.27: (a) Cross table analysis (b) Confusion matrix for Decision tree

The model was trained using 15 predictors and a class on 27,120 samples. This classifier produced 68 different tree sizes. The pruned Decision Tree model had an average tree size of 68 and a training error of 32.6%. It achieved an accuracy of 67%, with a sensitivity and specificity of 67.85% and 63%, respectively. The model's predictions included True Positives of 20,342, True Negatives of 3,657, False Positives of 9,637, and False Negatives of 2,141. Notably, pruning the model improved its performance.

The model's AUC score is 0.655, suggesting that the model is performing competently.

5.4.5 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computer-based model that takes inspiration from the way biological neural networks work in the human brain. It's a tool in machine learning used to analyze and comprehend patterns within data. ANNs are composed of interconnected artificial neurons that are structured in layers.

```r
#**********************Artificial Neural Network (ANN)************************
set.seed(12345)
# create ann model
Loan.ANN <-nnet(Status ~ .,data = train_dat, size=5, decay=5e-4, maxit=100)
plotnet(Loan.ANN)
summary(Loan.ANN) #Summary of the model

#Fitting the model to make prediction
ann.pred1 <- predict(Loan.ANN, test_set)
ANN.prediction <- as.numeric(ann.pred1 > 0.5)
```

Fig. 5.28: ANN model with prediction in R.

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  35777


                | ANN.prediction
test_set[, 16] |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             0 |     21231 |      8748 |     29979 |
               |   119.256 |   228.501 |           |
               |     0.708 |     0.292 |     0.838 |
               |     0.903 |     0.713 |           |
               |     0.593 |     0.245 |           |
---------------|-----------|-----------|-----------|
             1 |      2277 |      3521 |      5798 |
               |   616.624 |  1181.481 |           |
               |     0.393 |     0.607 |     0.162 |
               |     0.097 |     0.287 |           |
               |     0.064 |     0.098 |           |
---------------|-----------|-----------|-----------|
  Column Total |     23508 |     12269 |     35777 |
               |     0.657 |     0.343 |           |
---------------|-----------|-----------|-----------|
```

```
> confusionMatrix(as.factor(ANN.prediction)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21231  2277
         1  8748  3521

               Accuracy : 0.6918
                 95% CI : (0.687, 0.6966)
    No Information Rate : 0.8379
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2176

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7082
            Specificity : 0.6073
         Pos Pred Value : 0.9031
         Neg Pred Value : 0.2870
              Precision : 0.9031
                 Recall : 0.7082
                     F1 : 0.7939
             Prevalence : 0.8379
         Detection Rate : 0.5934
   Detection Prevalence : 0.6571
      Balanced Accuracy : 0.6577

       'Positive' Class : 0
```

Fig. 5.29: (a) Cross table analysis (b) Confusion matrix for ANN.

The artificial neural network design resulted in a 15-5-1 network configuration comprising 86 weights. The model achieved an accuracy rate of 69.18% and a specificity of 60.73%. The model's predictions encompassed values such as True Positives at 21,231, True Negatives at 3,521, False Positives at 8,748, and False Negatives at 2,277.

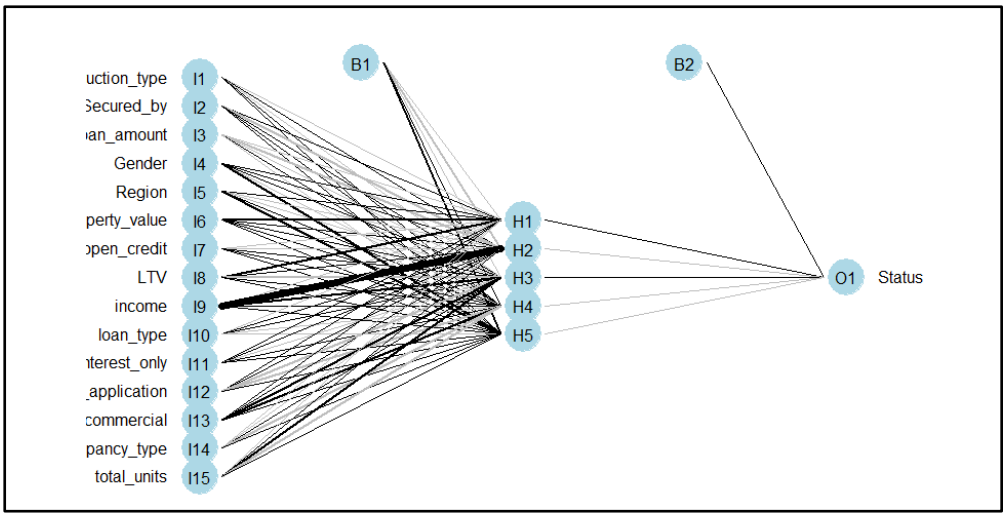


Fig 5.30 : schematic of the ANN model used for prediction in R.

CHAPTER 6

6.0 Model Evaluation

Model evaluation is the process of assessing the performance, accuracy, and effectiveness of a machine learning model using various metrics and techniques. It's a critical step in the machine learning workflow to determine how well the model is performing and whether it's ready to be deployed in real-world scenarios (Taylor, 2000). By comparing the predicted and actual results of each model, their respective performance can be determined after evaluating them against the test dataset.

Model	Accuracy	Recall	Precision	F1 score	Specificity	AUC
LG	60.23	59.40	89.65	71.46	64.56	61.98
KNN	66.82	68.19	89.75	77.50	59.73	63.96
SVM	66.08	66.31	90.71	76.61	64.88	65.60
ANN	69.18	70.82	90.31	79.39	63.07	65.80
Decision tree	67.08	67.85	90.48	77.55	60.73	65.50

Fig 6.1 : Evaluation and comparison of the machine learning models

Each of the models demonstrated strong performance, achieving accuracies spanning from 60% to nearly 70%. All five algorithms - Logistic Regression, Artificial Neural Network (ANN), Decision Tree, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) - underwent randomized training on the provided dataset. Remarkably, all of them yielded their optimal outcomes in terms of accuracy, recall, precision, F1 score, and specificity.

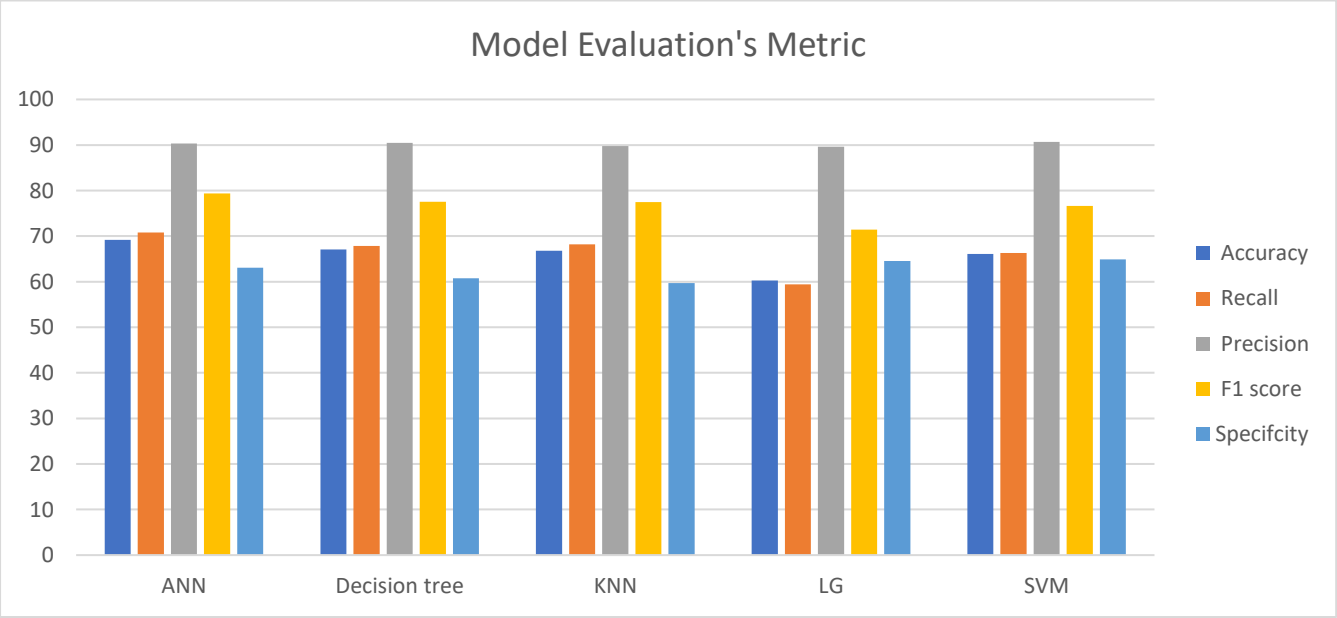


Fig 6.2: bar chart representing the performance of all algorithms.

Based on the provided metrics, the "Artificial Neural Network (ANN)" algorithm appears to have performed the best among the listed options. It achieved the highest accuracy (69.18%), recall (70.82%), precision (90.31%), and F1 score (79.39%) compared to the other models. The AUC score (65.80) is also relatively high, suggesting good overall performance.

Nevertheless, the algorithm that exhibited the lowest performance is Logistic Regression, achieving an accuracy of 60.23%. Additionally, it holds the smallest percentages for Recall, Precision, F1 score, and AUC among the considered metrics.

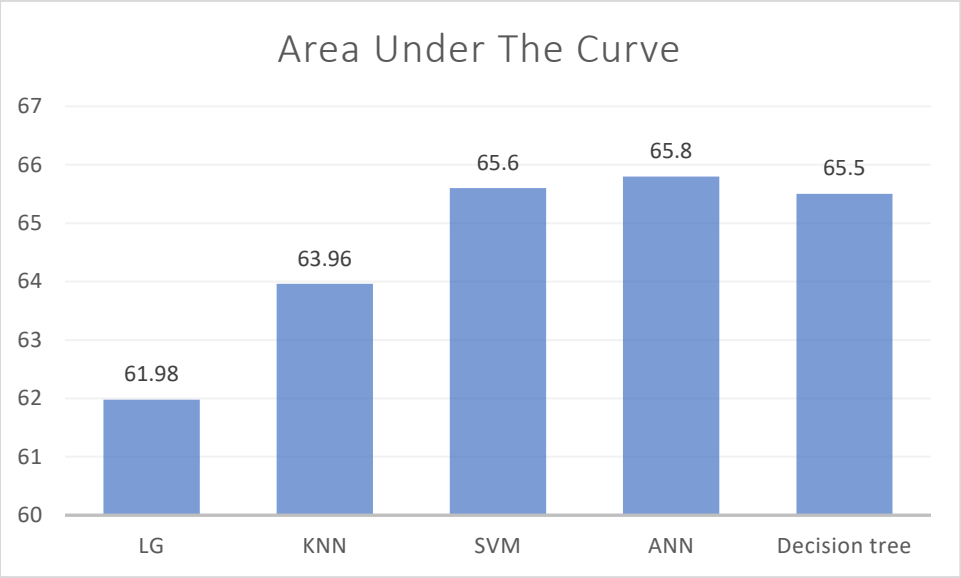


Fig 6.3: Bar chart representing all models.

Every algorithm displays an AUC value spanning from 60% to 65%, indicating a comparable capacity to differentiate between individuals who defaulted on their payments and those who successfully repaid their loans.

MODELS	TRUE POSITIVE	TRUE NEGATIVE	FALSE POSITIVE	FALSE NEGATIVE
LG	17,807	3,743	12,172	2,055
KNN	20,444	3,462	9,535	2,336
SVM	19,878	3,762	10,101	2,036
ANN	21,231	3,521	8,748	2,277
DECISION TREE	20,342	3,647	9,637	2,141

Table 6.3. Predictions for each algorithm based on test data.

Among the algorithms, the performance of ANN stood out in terms of predictions. ANN precisely predicted 21,231 applicants as "Repaid" (True Positives) and 3,521 applicants as "Default" (True Negatives). Conversely, there were 8,748 applicants inaccurately predicted as "Repaid" (False Positives) and 2,277 applicants inaccurately predicted as "Default" (False Negatives).

CHAPTER SEVEN

7.1 Critical Evaluation

I initially encountered a hurdle in selecting the right topic. Since I was keen on working with financial data, drawn from my previous role's familiarity with it, I reached out to my former organization's data department to obtain financial data. Unfortunately, I faced an obstacle due to data protection regulations that prevented me from acquiring the desired data. Subsequently, I proceeded to gather data from Kaggle. Among the various datasets related to financial information, the first dataset I stumbled upon was about Loan Approval Prediction. However, after exploring different datasets, I ultimately decided to work with the Loan Default Dataset on Kaggle, as none of the other datasets managed to capture my interest.

Finding journals related to Loan Default prediction proved to be a significant task. I initiated my research by utilizing platforms such as Google Scholar, SpringerLink, ScienceDirect, Academia.edu, and ResearchGate. However, accessing many of these sites demanded subscriptions and logins, which presented a challenge. Nonetheless, despite these obstacles, I managed to successfully download approximately twenty-five journals focused on Loan Default.

As I embarked on my first dissertation writing experience, I began by immersing myself in numerous articles, written pieces, and video tutorials cantered around writing a dissertation. This endeavour provided me with valuable insights into the structure and essence that a well-constructed dissertation should encompass. Another challenge I came across was cleaning up the data which was quite dirty and scattered. I initially started the cleaning process by removing the empty columns and empty cells which eventually affected my dependant variable. I ended up using R studio to remove the empty cells and blank columns.

However, a substantial challenge arose during this dissertation journey when my laptop encountered technical issues. It necessitated sending the laptop back to Lenovo for repairs, a process that extended for approximately 10 days. This unexpected setback significantly impeded the pace of my implementation efforts.

Based on my experience in a fintech B2B tech company that offers loan services to customers, a substantial number of customers tend to default on their loan repayments, posing significant challenges for the company's operations. Ensuring fair and successful loan distribution while minimizing the loan defaulter ratio is a crucial and prevalent issue in the financial lending industry. Currently, the method employed for predicting loan defaults relies on manual screening, which hinders accurate predictions of customer default likelihood. Having firsthand experience with this system has motivated me to explore machine learning approaches for loan default prediction.

I opted for this topic due to my prior involvement with a financial institution. Through personal encounters, I've witnessed the implications of loan defaulters on the company's operations, influencing my choice in selecting this subject. Recognizing the potential, I believe that Machine Learning Models can play a pivotal role in identifying potential loan defaulters, offering the organization enhanced tools for future detection.

7.2 Discussion

The process of prediction starts with data cleaning and processing, addressing missing values, conducting experimental analysis on the dataset, followed by model building, evaluation, and subsequent testing on test data. The following conclusions are reached after analysis that applicants with lower Life Time Value and Income are more likely to encounter loan approval challenges, as there exists a heightened probability of them not meeting the loan repayment obligations.

In accordance with the analysis, the Artificial Neural Network (ANN) demonstrates the most effective performance in predicting Loan Default. The Artificial Neural Network (ANN) showcased an accuracy rate that nearly reaches 70%, marking it as a robust model. Additionally, its recall score of 70%, reflecting the model's capability to predict actual positive cases accurately, further underlines its effectiveness. This model also demonstrates high precision and F1 scores, indicating its strong predictive power. However, its AUC score, which gauges the model's distinction between positive and negative classes, is considered reasonable but not reasonably high.

Given the dataset's substantial size and high dimensionality, the correlation analysis unveiled associations between certain variables. As part of dimensionality reduction, it became apparent that specific variables should be eliminated. During the analytical process, the count of variables used for model construction was curtailed using Eigenvalue analysis and Principal Component Analysis.

Possibly, some inaccurate predictions might have arisen due to the dataset's origin from an unverified third party. Such datasets from external sources tend to carry biases, outliers, and errors, potentially leading to erroneous predictions. Consequently, in the context of forecasting loan defaults, it becomes crucial to evaluate the impact of these erroneous predictions and determine the appropriate level of associated risk.

In summary, financial institutions can reap several benefits from using machine learning for loan default prediction. These advantages encompass heightened accuracy, expedited decision-making, cost savings, tailored financing strategies, automated verdicts, and an enhanced risk management infrastructure. Machine learning algorithms assess a spectrum of applicant indicators, such as credit history, income, and expenses, aiding financial institutions in crafting more personalized loan approval frameworks.

7.3 Conclusion

The objective of this study was to build, examine, and construct a machine learning algorithm capable of accurately determining whether an individual, based on specific attributes, possesses a likelihood of defaulting on a loan. Such a model holds the potential for financial institutions to discern certain financial characteristics of prospective borrowers that may indicate a risk of failing to repay the loan within the stipulated timeframe. By applying various performance metrics for comparative assessment, the Artificial Neural Network (ANN) yielded an accuracy rate of 69.18%, while the Decision Tree approach produced an accuracy of 67.08%. Consequently, the ANN model emerges as a more favourable choice compared to the other algorithms for predicting potential loan defaulters among consumers.

According to the insights provided by the ANN model, financial institutions should exercise caution when determining potential borrowers who align with specific criteria. For instance, individuals seeking loans should pay close attention to indicators such as Lifetime Value and Income during the pre-approval stage. Neglecting these aspects could potentially lead to instances of borrowers defaulting on their loans.

However, one of the major drawbacks is Asymmetric information. This could apply to failure of applicants to fully disclose their financial situations resulting to lenders not having a complete information about the applicant’s ability to repay the loan. This could be fixed through Disclosure and Transparency. That is, providing clear, accurate and comprehensive information about the transaction. Financial institutions should carefully assess these models' performance and update them frequently to adjust for changes in the market, regulatory, and policy requirements.

7.4 Future Work

Numerous organizations stand to gain financially by applying the methods used in this dissertation to identify defaults. This study has the potential for further exploration on extended loan term data and loans with lower risk profiles, such as commercial loans, mortgages, and student loans. Furthermore, it's worth noting that early repayment can affect loan profitability by reducing the loan interest.

Integrating the concept of expected profit into this analysis could provide a more explicit prediction. An interesting avenue for future research could involve utilizing classification trees to forecast the time until default. However, it's observed that as the prediction increases, the model's accuracy tends to decrease. Exploring the concept of multiple binary predictions could also be a potential next step.

This approach could involve an initial prediction of default or non-default, given its low error rate. Subsequently, a separate model could be employed to forecast the time until default based on the output from the first prediction model.

REFERENCES

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: a review of the literature. Intelligent Systems in Accounting, Finance and Management, 18(2/3), 59-88.

Aggarwal, R., & Ranganathan, P. (2016). Common Pitfalls in Statistical Analysis: The Use of Correlation Techniques. Perspectives in Clinical Research, 7, 187-190.

Aggarwal, R., & Ranganathan, P. (2016). Common Pitfalls in Statistical Analysis: The Use of Correlation Techniques. Perspectives in Clinical Research, 7, 187-190.

Alaba, O. B., Taiwo, E. O., & Abass, O. A. (2021). Data mining algorithm for development of a predictive model for mitigating loan risk in Nigerian banks. Journal of Applied Sciences and Environmental Management, 25(9), 1613–1616. https://doi.org/10.4314/jasem.v25i9.11

Alex S.& Vishwanathan, S.V.N. (2008). Introduction to Machine Learning. Published by the press syndicate of the University of Cambridge, Cambridge, United Kingdom. Copyright © Cambridge University Press 2008. ISBN: 0- 521-82583-0. Available at KTH website: https://www.kth.se/social/upload/53a14887f276540ebc81ae c3/online.pdf Retrieved from website: http://alex.smola.org/drafts/thebook.pdf

Altman, E. I. (2018). Assessing and Managing Credit Risk: The Current State of Practice. Journal of Applied Corporate Finance, 30(4), 73-86. doi: 10.1111/jacf.12298.

Amin, R. K., Indwiarti, & Sibaroni, Y. (2015). Implementation of decision tree using C4.5 algorithm in decision making of loan application by debtor (case study: bank pasar of Yogyakarta special region). The 3rd Int. Conf. on Information and Communication Technol. (ICoICT), pp. 75-80.

Arun, K., Ishan, G., & Sanmeet, K. (2016). Loan approval prediction based on machine learning approach. IOSR Journal of Computer Engineering (IOSR-JCE), 5, 18-21.

Aslam, U., Aziz, H. I. T., Sohail, A., & Batcha, N. K. (2019). An empirical study on loan default prediction models. Journal of Computational and Theoretical Nanoscience, 16, 3483–8.

Auckland, S. (2017). Precision-recall curves - what are they and how are they used. Acutecuretesting.

Bkassiny, M., Li, Y., & Jayaweera, S. K. (2012). A survey on machine learning techniques in cognitive radios. IEEE Communications Surveys & Tutorials, 15(3), 1136–1159.

Brazdil, P., Soares, C., & da Costa, J. (2003). Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. Machine Learning, 50(3), 251–277. doi:10.1023/A:1021713901879. Available at Springer website: https://link.springer.com/content/pdf/10.1023%2FA%3A1021713901879.pdf.

Campbell-Verduyn, M., Goguen, M., & Porter, T. (2017). Big data and algorithmic governance: the case of financial practices. New Political Economy, 22(2), 219-236.

Chase Bank. (2023). How to Calculate Debt-to-Income Ratio | Chase. © JPMorgan Chase & Co. Available at: https://www.chase.com/personal/credit-cards/education/basics/what-is-debt-to-income-ratioand-why-it-is-important. Accessed: 16 March 2023.

Chen, L., et al. (2018). Loan default prediction using decision trees and random forests. Expert Systems with Applications, 95, 205-214.

Delgrado, R., Tibau, X-A. (2019). Why Cohen's Kappa should be avoided as a performance measure in classification. PLoS One, 14(9).

Domingos, P. & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning Volume 29, pp. 103–130 Copyright © 1997 Kluwer Academic Publishers. Manufactured in The Netherlands. Available at University of Trento website: http://disi.unitn.it/~p2p/RelatedWork/Matching/domingos9 7optimality.pd

Dutta, P. (2021). A STUDY ON MACHINE LEARNING ALGORITHM FOR ENHANCEMENT OF LOAN PREDICTION. In International Research Journal of Modernization in Engineering Technology and Science www.irjmets.com @International Research Journal of Modernization in Engineering (Vol. 160). www.irjmets.com

Epskamp, S., & Fried, E.I. (2018). A Tutorial on Regularized Partial Correlation Networks. Psychological Methods, 23(4), 617–634. https://doi.org/10.1137/met0000167

Fay, B., (2017). What is a Credit Score and How is it Calculated? Available at: https://www.debt.org/credit/report/scoring -models [Accessed: 12 June 2018].

Gale, W. G., & Levine, B. J. (2016). Debt-to-Income Ratios as an Indicator of Borrower Default. Journal of Economic Perspectives, 30(1), 195-212. doi: 10.1257/jep.30.1.195

George, W., & Mendoza L. J. (1989). The Internal Correlation: Its Applications in Statistics and Psychometrics. Journal of Educational Statistics, 14(3), 211-226.

Guttentag, Jack (October 6, 2007). "The Math Behind Your Home Loan". The Washington Post

Hafiz, I., Asim, S., Uzair, A., & Nowshath, K. (2019). Loan Default Prediction Model Using Sample, Explore, Modify, Model, and Assess (SEMMA). Journal of Computational and Theoretical Nanoscience, 16, 3489–3503.

Hamid, A. J., & Ahmed, T. M. (2016). Developing prediction model of loan risk in banks using data mining. Machine Learning and Applications: An Int. Journal (MLAIJ), 3, 1–9.

Harrington, P. (2012). Machine Learning in Action. Manning Publications Co., Shelter Island, New York.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer Verlag.

Honaker J, King G (2010). "What to Do about Missing Values in Time Series Cross-Section Data." American Journal of Political Science, 54(2), 561–581

Honaker, J., & King, G. (2010). What to Do about Missing Values in Time Series Cross-Section Data. American Journal of Political Science, 54(2), 561–581.

Hormozi, H., Hormozi, E. & Nohooji, H. R. (2012). The Classification of the Applicable Machine Learning Methods in Robot Manipulators. International Journal of Machine Learning and Computing (IJMLC), Vol. 2, No. 5, 2012 doi: 10.7763/IJMLC.2012.V2.189pp. 560 – 563. Available at IJMLC website: http://www.ijmlc.org/papers/189-C00244-001.pdf

https://en.wikipedia.org/wiki/Instance-based_learning

Investopedia. (2021, May 13). Personal Loan. Retrieved from https://www.investopedia.com/terms/p/personal-loan.asp

Iqba, A., Aftab, S., Ali, U., Nawaz, Z., Sana, L., Ahmad, M., & Husen, A. (2019). Performance Analysis of Machine Learning Techniques on Software Defect Prediction using NASA Datasets. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5).

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica 31 (2007). Pp. 249 – 268. Retrieved from IJS website: http://wen.ijs.si/ojs2.4.3/index.php/informatica/article/download/148/140.

Kumar, A., & Bhattacharya, S. (2022). Loan default prediction using interpretable machine learning techniques. Journal of Risk Research, 25(7), 881-902.

Leal, V. (2021). How to Build a Confusion Matrix for a Multiclass Classifier? CrossValidated, StackExchange Inc.

Lee, I., & Shin, Y. J. (2017). Fintech: Ecosystem, Business Models, Investment Decisions, and Challenges. Business Horizons.

Li, Y., et al. (2021). Loan default prediction using hybrid models: A comparative study. Decision Support Systems, 145, 113505.

Luca, B., Sebastiano, M., & Elisa, T. (2021). Forecasting Loan Default in Europe with Machine Learning. Journal of Financial Econometrics, 21(2), 569–596.

Marsland, S. (2015). Machine Learning: An Algorithmic Perspective. CRC Press.

Mehul, M., Aniket, K., Chirag, K., Rachna, J., & Preeti, N. (2021). Loan default prediction using decision trees and random forest: A comparative study. IOP Conference Series: Materials Science and Engineering, 1022, 012042.

Mishra, A. (2018). Metrics to Evaluate your Machine Learning Algorithm.” Towards Data Science.

Mohankumar, M., Amuthakkani, S., Jeyamala, G., Professor, A., of CSE, D., Student, U., & Vidya, S. (2016). COMPARATIVE ANALYSIS OF DECISION TREE ALGORITHMS FOR THE PREDICTION OF ELIGIBILITY OF A MAN FOR AVAILING BANK LOAN. In Online) International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST) (Vol. 2).

Nandacumar, S. (2020). Confusion Matrix - are you confused? (Part I and Part II). Medium.

Narkhede, S. (2018). Understanding Confusion Matrix. Towards Data Science.

Newsom, I. (2015). Data Analysis II: Logistic Regression. Available at: http://web.pdx.edu/~newsomj/da2/ho_logistic.pdf

Ranjan Jena, S. (2022). PREDICTION OF MODERNIZED LOAN APPROVAL SYSTEM BASED ON MACHINE LEARNING APPROACH. International Research Journal of Modernization in Engineering Technology and Science, 2582–5208. www.irjmets.com

Reinartz, W., & Kumar, V. (2018). The Customer Lifetime Value Concept: Past, Present, and Future. Journal of the Academy of Marketing Science, 46(3), 402-427. doi: 10.1007/s11747-017-0523-9

Saito, T., & Rechmsmeier, M. (2015). The Precision-Recall Plot is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLoS ONE, doi: 10.1371/journal.phone.0118432.

Saito, T., & Rehmeismeier, M. (2017). Basic Evaluation Measures from the Confusion Matrix. WordPress.

Saladi, J. S. (2019). What Are The Stages of Liver Failure?. Healthline.

Setiono R. and Loew, W. K. (2000), FERNN: An algorithm for fast extraction of rules from neural networks, Applied Intelligence.

Signoriello, Vincent J. (1991), Commercial Loan Practices and Operations, ISBN 978-1-55520- 134-0

Sim, J., & Wright, C. C. (2005). The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. Physical Therapy, 85(3), 257-268. https://doi.org/10.1093/ptj/85.3.257

Smith, J., & Johnson, A. (2017). Loan default prediction using logistic regression. Journal of Finance and Economics, 42(2), 153-168.

Sutton, R. S. (1992). Introduction: The Challenge of Reinforcement Learning. Machine Learning, 8, 225-227. Kluwer Academic Publishers, Boston.

Tabachnick, B.G. & Fidell, L.S. (2013). Using Multivariate Statistics. Pearson.

Taiwo, O. A. (2010). Types of Machine Learning Algorithms, New Advances in Machine Learning, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech, University of Portsmouth United Kingdom. Pp 3 – 31. Available at InTech open website: http://www.intechopen.com/books/new-advances-inmachine-learning/types-of-machine-learning-algorithms.

Tapas Kanungo, D. M. (2002). A local search approximation algorithm for k-means clustering. Proceedings of the eighteenth annual symposium on Computational geometry. Barcelona, Spain: ACM Press.

Taylor, K.E. (2000). Summarizing Multiple Aspects of Model Performance in a Single Diagram. PCMDI Report No 65. Program for Climate Model Diagnosis and Intercomparison, Lawrence Livermore National Laboratory, University of California, Livermore, CA.

Tejaswini, J., Mohana Kavya, T., Devi, R., Ramya, N., Triveni, P. S., & Rao Maddumala, V. (2020). ACCURATE LOAN APPROVAL PREDICTION BASED ON MACHINE LEARNING APPROACH. 11. www.jespublication.com

Timothy Jason Shepard, P. J. (1998). Decision Fusion Using a Multi-Linear Classifier. In Proceedings of the International Conference on Multisource-Multisensor Information Fusion.

Tukey, J.W. (1977). Exploratory Data Analysis. Addison-Wesley, Reading, MA.
Tukey, J.W. (1977). Exploratory Data Analysis. Addison-Wesley, Reading, MA.

Vaidya, A. (2017). Predictive and Probabilistic Approach Using Logistic Regression: Application to Prediction of Loan Approval. The 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1, 1–6.

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. (2nd ed.). Springer Verlag. Pp. 1 – 20. Retrieved from website: https://www.andrew.cmu.edu/user/kk3n/simplicity/vapnik2000.pdf

Verma, J., & Abdel-Salam, A. (2019). Testing Statistical Assumptions in Research. John Wiley & Sons Inc.

Vujović. Z (2020). The Big Data and Machine Learning. Journal of Information Technology and Multimedia Systems, 19(7), 11-19. DOI: 10.5281/zenodo.427923.Zhang, H., et al. (2019). Loan default prediction using deep neural networks. Expert Systems with Applications, 128, 32-43.

Wang, X., et al. (2020). Loan default prediction using ensemble learning and alternative data sources. Journal of Banking and Finance, 45, 165-175.

Wei, Y., Yildirim, P., Van den Bulte, C., & Dellarocas, C. (2016). Credit Scoring with Social Network Data. Marketing Science, 35(2), 234-258.

Wikipedia. (2023, July 4). Loan. In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Loan

Wyman, O. (2015). The Role of Financial Services in Society: Statement in Support of Macroprudential Policies. Retrieved from http://www3.weforum.org/docs/WEF_The_Role_of_Financial_Services_in_Society_report_2015.pdf

Xu, J. J., Lu, Y., & Chau, M. (2015). P2P Lending Fraud Detection: A Big Data Approach. In Pacific-Asia Workshop on Intelligence and Security Informatics (pp. 71-81). Springer, Cham.

Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., & Ma, W. (2005). OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization. In 28th Annual International Conference on Research and Development in Information Retrieval (pp. 122-129). ACM SIGIR.

Zurada, J. (2007). Using Memory-Based Reasoning For Predicting Default Rates On Consumer Loans. Review of Business Information Systems-First Quarter, 11(1), 1-16.

APPENDIX

R Script – One Drive

https://uelacmy.sharepoint.com/:u:/r/personal/u2328860_uel_ac_uk/Documents/Loan%20default.R?csf=1&web=1&e=EXhGoG

Dataset from Kaggle

https://www.kaggle.com/datasets/yasserh/loan-defaultdataset/download?datasetVersionNumber=1

Excel File – One Drive

https://uelacmy.sharepoint.com/:x:/r/personal/u2328860_uel_ac_uk/_layouts/15/Doc.aspx?sourcedoc=%7B3C038E11-5FAD-42C0-9E8A-4D1D63A00721%7D&file=Loan_Default.csv&action=default&mobileredirect=true

Receiver Operating Characteristic Curve Plot of all Algorithms