MIDLANDS STATE UNIVERSITY

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF APPLIED MATHEMATICS AND STATISTICS

PREDICTION OF POSSIBLE LOAN DEFAULTS USING SUPERVISED MACHINE LEARNING ALGORITHMS

LESLEY MATHE

R196777F

SUPERVISOR: Mrs. CHANDIWANA

This Dissertation is submitted in Partial Fulfillment of the Requirements of the Bachelor of Science and Technology in Applied Statistics Degree

APPROVAL FORM

The undersigned certify that they read and recommend to Midlands State University for acceptance, a research project: "Prediction of possible loan defaults using supervised machine learning algorithms" submitted by Mathe Lesley in partial fulfilment of the requirements of the Bachelor of Science Honours Degree in Applied Statistics at Midlands State University 2022.

…… ……

Student. Date.

……. ……..

Supervisor. Date.

……. ……..

Chairperson. Date.

DECLARATION

I, Mathe Lesley, do hereby declare that this is my own work. This dissertation has not been submitted for any degree or examination in any university. All the citations I have quoted have been indicated and acknowledged by complete reference.

Student's Signature……... Date…….

Supervisor's Signature…….. Date……..

Dedication

I dedicate this project to my father and mentor for the great encouragement that kept me pushing till the end.

Acknowledgements

The golden thanks goes to the creator of the universe who gave me the strength and ability to think in terms of numbers and computers. It would not have been possible to come up with such a project if it was not for Him. I would like to thank Mrs. E. Chandiwana for been lenient with me till the end, your efforts did not go in vain ma'am. Lastly, I would like to thank my family and friends for encouraging me to keep pushing, it was not easy, but with your motivations I made it to the finishing line.

ABSTRACT

A key cause of worry for banks and other financial organizations is loan default. Because the loan portfolio is regarded an asset to the institution and has a direct impact on the firm's profitability, appropriate procedures must be put in place to keep default risk manageable. The ability of customers to pay back their loans within the predetermined time frame determines whether they are considered "good" or "bad" customers. Good customers routinely pay off their debts in accordance with the terms of the bank's agreement, and as a result, they gradually receive a higher credit limit, enabling them to access more credit, whereas bad accounts repeatedly miss payments and are branded as "bad" customers, or face harsher measures being taken against them. The latter results in losses for banks. Therefore, it is crucial for these financial institutions to enhance their credit management procedures in order to optimize their bad debt provisioning, reduce default losses, and increase revenue from "good" customers. Traditional credit scoring methods have been based on static loan default prediction models, which produce classifications of clients based on default risk but offer little insight into how loan conditions evolve over time. Given the predetermined time range, changes in loan states make for a good research topic because they may reveal a lot about the loans' terminal status. In this work, models were built using machine learning algorithms that learnt patterns from observed account transaction activity and then used that knowledge to forecast future credit loan situations. These techniques included stacked ensemble analysis, Naive Bayes, K-Nearest Neighbor, and Support Vector Machine. Successful predictions were generated with varying recall, accuracy, precision, f1 score, and an AUC-ROC, showing the value of considering the loan's status throughout time as well as the events that led up to terminal default or non-default state default.

Table of Contents

ABBREVIATIONS

ANN: ARTIFICIAL NEURAL NETWORK

KNN: K-NEAREST NEIGHBOR

NB: NAÏVE BAYES

SVM: SUPPORT VECTOR MACHINE

ML: MACHINE LEARNING

EDA: EXPLORATORY DATA ANALYIS

SKLERAN: SCIKITLEARN

CRM: CREDIT RISK MANAGEMENT

FICO: FAIR ISAAC CORPORATION

AUC-ROC: AREA UNDER THE RECEIVER OPERATION CURVE

LR: LOGISTIC REGRESSION

AI: ARTIFICIAL INTELLIGENCE

CHAPTER ONE: INTRODUCTION

1.1 INTRODUCTION.

Consumer spending has become a significant factor in the microeconomic conditions across the nation during the past few decades. In this instance, the gradual rise in consumer spending is strongly correlated with the changing monetary policies of numerous organizations that support consumers by providing loans to them. It has once again been abundantly evident that consumer behavior has played crucial roles at every stage following the disastrous inflation of 2001–2009, termed the biggest economic crisis since 1980. The consumer function has taken the legacy all the way to its conclusion, starting with planting the seeds of the calamity.

In order to enhance client behavior and forecast their attitudes, "credit risk management" has thus been given top attention. As time has gone on, "credit risk management" has emerged as a dependable means of ensuring that individuals or organizations can be relied upon to issue a loan for the benefit of the security of the loan-granting institution. So, in essence, 'credit risk management' is an exercise offered by financial institutions to help them limit the losses associated with loan giving. These programs frequently use a variety of techniques to reduce or eliminate the risks connected with loan lending. Many firms utilize statistical approaches to determine whether or not a client will default on their loan. Discriminant analysis, credit rating, and logistic regression are among the methods used.

The normal range of a person's credit score is 300 to 850. The person is deemed to be more financially reliable the higher the score. The Fair Isaac Corporation, better known as FICO, created the common credit score model that is utilized by financial institutions. The industry's

most popular approach is unquestionably the FICO score. There are now various classification models for assessing borrower trustworthiness, including logistic regression, gradient-boosted trees, support vector trees, and random forest, among others. Now comes the question of which model is best for predicting loan defaults.

The researcher's work here attempts to provide an outline of the methodologies that will be utilized to construct a model that will forecast credit default risks. In order to achieve the best results and thus be considered a pragmatic approach to credit risk default forecasting, the researcher attempted to incorporate machine learning techniques into the loan default model.

1.2 BACKGROUND OF THE STUDY.

Machine learning (ML) is transforming the finance sector, as a growing number of businesses begin to adopt machine learning technology to automate processes, increase their productivity, and improve decision-making (Roldós, 2020). In a more sensible way, financial institutions such as banks and insurance brokers use machine learning techniques to detect frauds, predict risks and loan defaults and automate repetitive tasks.

Clark (2019) explains the importance of credit risk management by stating that; for any lender the importance of credit risk measurement (CRM) is paramount. It is the basis for which a lender can calculate the likelihood of a borrower defaulting on a loan or meeting other contractual obligations. More broadly, credit risk management attempts to measure the probability that a lender will not receive the owed principal and accrued interest, which if allowed to happen, will lead to a loss and increase costs for collecting the debt owed. GiniMachine (2021) further employed that detecting a red-flag credit risk can be tricky. That’s why credit providers often employ a variety of credit risk management tools and procedures to ensure minimal risk to a business and maintain maximum profitability. One of the techniques being employed by many

credit providers is by analyzing the micro and macro loan trends. In 2021, GiniMachine argued that credit risk doesn't always come from one singular client. Instead, it might be a portfolio risk. By assessing the number of the micro (single loans) and macro (groups of loans), a company can detect risk or risky lending patterns that it may engage in the future. This helps the corporation to maintain a healthy debt-to-capital ratio and ensure competitively priced lending products. Using this technique, the company would have to go on their lending records one by one which would include checking if the client defaulted, paid on time, or had a late payment. Thus, this has raised interest in building models that estimate the probability of a customer defaulting. The most applied ones are Discriminant Analysis and Logistic Regression.

This research aims to investigate the use of numerous machine learning methods in credit risk management and test if they produce more reliable probabilities as compared to the traditional techniques currently being applied.

1.3 PROBLEM STATEMENT

Loans are granted to clients after their eligibility has been assessed. This has been made easy by the use of credit scores models that easily produce scores translating to the likelihood of a client defaulting on a loan. The other technique is to manually go through the clients' borrowing history and derive if they are eligible of getting the loan. Key factors that play a pivotal role are age, employment status, and salary of the client. This proved to be time consuming and prone to errors as the models are believed to be fixed, thus lacking the precise ability to predict the possibility of loan defaults accurately. Hence the need to explore flexible machine learning techniques that can enhance accurate loan default prediction.

1.4 JUSTIFICATION OF THE STUDY

Credit risk evaluation has become the most observed priority in financial institutions in Zimbabwe; this is because most of their revenues come from the lending department. Thus, this has led to many institutions coming up with more advanced techniques for evaluating credit risk. This has proven to be a more versatile initiative. The objective of this study is to explore how the most versatile machine learning algorithms can be used to predict loan default in an accurate manner.

1.5 RESEARCH QUESTIONS

i. Which machine learning model can be best suited for predicting possible loan default?

ii. How best can we compare different machine learning algorithms used in loan default prediction?

1.6 AIM OF THE STUDY

To apply and examine the influence of several supervised binary classification algorithms on default prediction.

1.7 ASSUMPTIONS

i. The research assumes readers are aware of what an algorithm is and are able to differentiate between different types of machine learning algorithms.

ii. The research assumes the reader has a general knowledge of what binary classification is.

1.8 LIMITATIONS

i. Some of the algorithms under study require more processing power during the model training phase as compared to other phases. For instance, the Support Vector Machine.

ii. The dataset used cannot be classified as big data. Thus, it does not completely suit the algorithms under study. However, it proved to be sufficient enough to demonstrate how algorithms can be applied in binary classification situations.

1.9 OBJECTIVES

i. To apply the GNB, KNN, SVM algorithms, and the Stacking Ensemble model on loan default predictions.

ii. To classify the data using GNB, KNN, the SVM, and the Stacking Ensemble models.

iii. To select the best model using evaluation metrics.

1.10 SIGNIFICANCE OF THE STUDY

The study is going to explore numerous machine learning algorithms that have the capacity to substitute traditional models that are currently used in credit risk management. The researcher believes that they have the quality of manipulating binary data and drawing noticeable conclusions based on the evaluation criterions which the traditional methods are incapable of applying. In addition, the algorithms have the capability of adjusting to current situations given that the data presents new customer variables. The study also takes into account the comparison of these algorithms in order to choose the one that best predicts loan defaults.

CHAPTER TWO: LITERATURE REVIEW

2.1 INTRODUCTION

The subject of this chapter is loan defaults. It also includes an examination of the factors that have the greatest influence on the chance of default. The multiple machine learning methods that have been used to predict the possibility of loan payment default are also taken into account.

2.2 EMPIRICAL REVIEW

2.2.1 METHODS AND TECHNOGIES USED FOR PREDICTING DEFAULTS

Every person considering whether or not to lend money to a beneficiary must make an educated guess as to whether the recipient will keep his or her promise to pay back the loan in full or in part. The goal is for the counterparty to repay (with interest for those whose terms necessitate it) and within the stipulated time limit. The lender evaluates how well they know the recipient based on their potential to repay, or the recipient's prior performance at loan repayment, or the history of similar borrowers in similar situations, in order to make such a judgment. The choice to lend then follows, which entails risk of loss should the counterparty fail to adhere to the agreed-upon terms. Even in automated (computerized) systems, this serves as the foundation for default prediction (Brown & Moles, 2014).

The modeling of the counterparty's default likelihood is necessary for credit risk assessment. This can be used to determine whether to initially extend credit or to modify the amount of credit extended, as in the case of loans. The objective is to weigh the prospective loss against the gain from credit extension (via interest payments). The following graph illustrates the judgment regarding credit evaluation given that the probability of default for a specific repayment cycle is p.

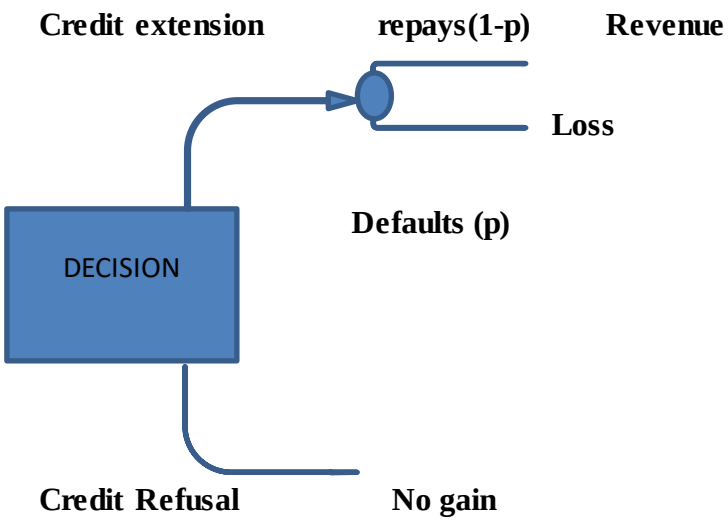


Fig 2.1: Assessment of Credit Risk (Brown & Moles, 2014).

One of the main determinants of whether loan is granted or denied is the chance of default (p). The likelihood of that happening is based on factors specific to the loan recipient and can also be inferred from prior transactions, as was previously stated in the sections. As a result, it is possible to look into the applicants' character traits and transactional habits to get a solid idea of the probability's value, or p, and use that information as the basis for making a credit decision.

The solution suggested in this study takes into account how the revolving credit principle applies to prior loan applications, which offers a novel way of tackling the issue and creating a tool for anticipating defaults. The ability to carry a debt over to the following billing period demonstrates a certain form of dependency of events occurring throughout succeeding billing cycles, such that the loan situation at the conclusion of one billing cycle is influenced by the end of the previous billing cycle. In this sense, a default event can be predicted by looking at the occurrences before it because the events are not independent of one another. One method to carry out this analysis is to employ a regression technique, which models a relationship between predictor and response variables so that unit changes in the values of one or more of the predictor variables result in an

observable change in the response variable. Discrete time events can be used to simulate loans, with event outcomes being determined after a billing cycle concludes. A series of events that took place over the loan's lifecycle might be seen as leading up to the default event. The consumer may have gone through a number of loan states over that time, some of which may have been solvent and others delinquent.

2.2.1.1 Linear Regression Analysis

In the context of credit risk modeling, Y is the observed status of the loan at the end of some predetermined period, and $X_1; X_2; \ldots; X_n$ are the covariates describing an account and which influence the outcome of Y during the observation period. This type of regression analysis establishes a linear model between a dependent variable, Y, and one or more independent variables, $X_1; X_2; \ldots; X_n$ (Pershad, 2000). Since the dependent variable Y produced by linear regression has a continuous value, a cut-off value must be established in order to categorize the accounts into those that will default and those that won't. Following is how linear regression is represented mathematically:

$$Y = \beta_0 + \beta_1 X_1 + \beta_0 X_1 + \cdots + \beta_n X_n + \varepsilon \quad \ldots\ldots\ldots\ldots\ldots (1)$$

Meyer and Pifer (1970) used this approach to forecast the failure of financial organizations based on four categories of explanatory factors: local economic conditions, general economic conditions, management quality, and personnel integrity. In order to get the appropriate cut-off value for the independent variable for determining failure, the researchers examined various choices of the cut-off value. At least 80% of the sample could be classified accurately, and predictions of future failures could be made with almost the same accuracy.

Since the problem requires that the outcome variable be a categorical variable (default or non-default), a situation that violates the linearity assumption in normal regression, this method is unpopular with studies involving loan default prediction and categorization. Utilizing logistic regression, which logarithmically modifies the output variable and enables the binary values that are required in these cases, has been the chosen method in many investigations (Agbemava, Nyarko, Adade & Bediako, 2016).

2.2.1.2 Logistic Regression Analysis

In Ghana's microfinance industry, Agbemava et al. (2016) developed a model using logistic regression to identify the traits that were statistically significant in predicting loan defaults by clients. Using binomial regression, the researcher used two levels to categorize clients as defaulters or non-defaulters. Only six predictor variables—marriage status, number of dependents, type of collateral, customer assessment, loan duration, and loan type—were found to be statistically significant for the study, and it was possible to predict an 86.67% default rate for Ghanaian borrowers of the microfinance institution's loans using these variables. The nature of the problem demanded a categorical dependent variable, and the predictor variables present were also categorical variables, so logistic regression analysis was appropriate for that study.

By using a linear function whose output represents the likelihood that the instance belongs to one of the classes of the dependent variable, the mathematical modeling technique known as logistic regression describes the relationship between one or more independent variables and one categorical dependent variable (Agbemava et al., 2016). When a dependent variable only has two possible values, it is referred to be a dichotomous or binomial dependent variable, and binomial logistic regression is used to do the analysis. Multinomial logistic regression is the analysis

technique utilized when the dependent variables are polytonomous or multinomial and contain more than two levels.

The mathematical representation of Logistic Regression is as follows:

$$Y = \frac{e^{(a + B_1X_1 + B_2X_2 + \cdots + B_nX_n)}}{1 + e^{(a + B_1X_1 + B_2X_2 + \cdots + B_nX_n)}} \quad \ldots\ldots\ldots (2)$$

Where Y denotes the likelihood of a customer belonging to either of the groups (i.e. defaulter or not). For example, where the customer is a defaulter, Y = P (1| x1, x2,..., xn) and where the customer is not a defaulter, Y = P (0| x1, x2,..., xn). This is based on the assumption that the values (0, 1) reflect defaulter and non-defaulter, respectively.

When values are plotted on a two-dimensional axis, the formula described above yields a sigmoid curve, with the inflexion point denoting the probability at which a customer switches from one class to the other.

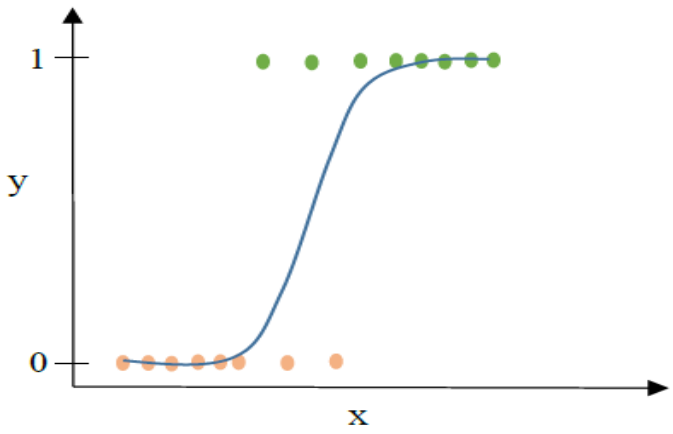


Fig 2.2: Logistic Sigmoid Function.

To transform the logistic regression model into a linear one, the Logit function is used where logit is defined as the logarithm of the odds of a default event occurring (Lando, 2004).

Odds of default are defined as follows:

Odds(Y = 1) = $\frac{p}{1-p}$, where p is the probability of defaulting.

The log of odds (logit) is hence defined as follows:

$((Y = 1 \mid x1, x2, \cdots, xn)) = log_e(\frac{p}{1-p})$, where p is the probability of defaulting.

The logit function transforms the logistic regression model into a linear model which takes the form:

$((Y = 1 \mid x1, x2, \cdots, xn)) = \alpha + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn$ ..... (3)

By entering the values of the independent variables (X-variables) into the equation, performing a mathematical summation of all the values, and then performing the inverse-logit of the result, it is now possible to obtain a prediction in a situation with two levels of outcomes (binomial logistic regression) or multiple levels of outcomes (multinomial logistic regression). The objective of logistic regression is to determine the constant and the coefficients of $\beta 1 \ldots \beta n$ of the independent variables.

Multiple models are created in Multinomial Logistic Regression (k-1 models, where k is the number of levels of the dependent variable). Here, the topic of the equation is a reference level, and calculations for all other levels are also provided (Williams, 2019). The logistic regression formula, for instance, will be as follows for a dependent variable with three levels, A, B, and C, and assuming that C is the reference level:

$$P(C) = \frac{1}{1 + \left(e^{(\alpha^A + \beta_1{}^A X_1 + \cdots + \beta_n{}^A X_n)}\right) + \left(e^{(\alpha^B + \beta_1{}^B X_1 + \cdots + \beta_n{}^B X_n)}\right)}$$

A log of the odds of A occurring with respect to C will be taken into account in one model, where C will be set at 1 - P(A) and B will not be taken into account, and a log of the odds of B occurring with respect to C will be taken into account in the other, where C will be set at 1 - P(B) and A will not be taken into account.

The level with the highest likelihood will be applied to the instance. The default levels in this study will be predicted using both categorical and continuous independent variables, and multinomial logistic regression is acceptable because the default levels are more than two. Furthermore, multinomial logistic regression uses a logarithmic transformation to enable the linear modeling of non-linear relationships (continuous and categorical variables to categorical outcomes); the inverse of this transformation yields a value indicating which category the outcome variable will fall into for each instance of the study.

2.2.1.3 Discriminant Analysis

Discriminant analysis makes two sets of cases, one consisting of cardholders who repeatedly default on their loans and the other consisting of customers who do not repeatedly default.

The assumption that instance attributes follow a normal distribution, which is implausible for many observations made in practice, is one of the drawbacks of discriminant analysis. Additionally, the model is static and does not provide data on how long a candidate remains current on payments, which would be helpful data for the financial institution granting the loan (Lando, 2004).

2.2.2 Machine Learning Approaches

In order to create models that computers can use to solve issues without explicit instructions, machine learning applies computer algorithms and statistical methodologies (Rokad, 2019). Different methods have been employed in the study of loan defaults in the past based on the nature of the issue and the desired results. These methods can generally be divided into two groups:

i. Classification issues: after the deadline for determining default, merely classifying accounts as defaulters or non-defaulters.

ii. Prediction issues - predicting a customer's loan status after a predetermined period of time using independent factors defining the customer These studies also cover intensity modeling of credit default, where a survival function is established to determine the likelihood that a customer's account "survives" (remains solvent) for a predetermined period of time, *t*.

These models are based on conditional probabilities of default (Lando, 2004).

In order to achieve desired results, machine learning techniques can be applied in a variety of dimensions and are well suited for predicting credit risk. Below is a description of common machine learning techniques that have been used in the past to analyze credit risk.

2.2.2.1 Artificial Neural Networks

Kumar, Jain, Singhal, and Goel (2018) used a deep neural network multilayer perceptron (MLP) model on a dataset of 9578 instances to achieve a prediction accuracy of 93%. The model was built with a multilayer perceptron that has two hidden layers of 20 nodes each and was trained over 1000 epochs. The model's input layer contained 18 perceptrons that were activated using the function

$y(V_1) = (\ \mathbf{1} + e^{-v_1})^{-1}.$ ..... (3)

This demonstrated that neural networks may be successfully employed in credit risk modeling. The classification of instances and accounts or the prediction of default events could be the application context.

Artificial neural networks are mathematical simulations of biological neural networks that are designed to mimic how human memories function (Bacham & Zhao, 2017). Input, output, and a hidden layer are all parts of a neural network. The hidden layer turns the inputs into a format that the output layer can use (Dormehl, 2019). Back propagation is a method they use to modify buried layers of neurons until the output is in line with what the programmer intended. The ANN then performs at its best for that training dataset as a result.

2.2.2.2 Support Vector Machines (SVM)

Comparing the logistic regression model with support vector machines for the modeling of credit risk for loan applicants of Equity Bank ltd, a Kenyan bank, was done by Obare and Muraya (2011). The accuracy levels for logistic regression, the radial SVM, and the linear SVM model were 73%, 78%, and 86%, respectively, according to the study, which also assessed the linear SVM kernel and the radial SVM kernel on the same set of data. According to this study, the SVM models match the data and the described problem better than the logistic regression model.

SVMs have been effective at modeling credit risk in part because they can accommodate linearly non-separable scenarios; in contrast, approaches like discriminant analysis and logit analysis only perform well when the data is linearly separable (Chen, Hardle & Moro, 2011). However, the latter are easier to analyze because it is feasible to determine the direct contribution of research factors to the results that were observed.

Given training data, a support vector machine (SVM) is a supervised learning model that generates an ideal hyperplane that classifies new cases (Patel, 2017). The hyperplane, which divides a plane into two sections in two-dimensional space, places each class's data points on either side of the line so that the distance between them is as great as possible (Gandhi, 2018). By maximizing the distance between data points, one can reinforce the classification of future data points with more confidence. Hyperplanes are decision boundaries that aid in data classification, and data that lies on either side of the hyperplanes can be attributed to different classes (Gandhi, 2018).

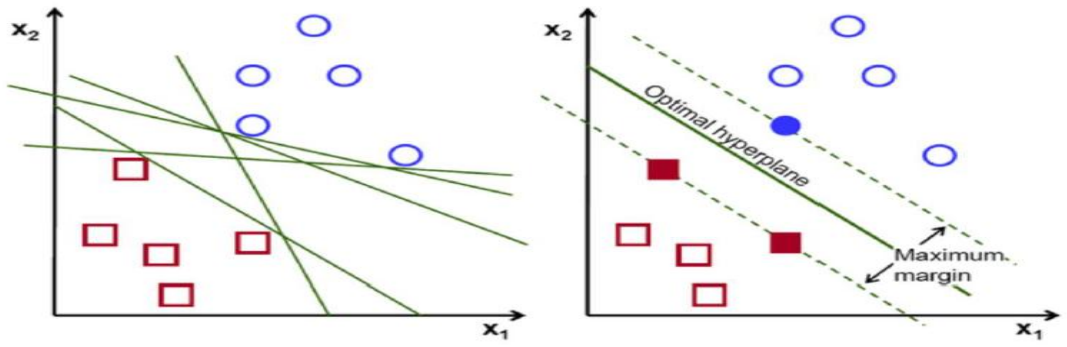


Fig 2.3: Support Vector Machine.

SVM identifies a hyperplane in an N-dimensional space, according to Gandhi (2018). The hyperplane is a line if there are two input characteristics. The hyperplane is a two-dimensional plane if there are three input features.

CHAPTER THREE: METHODOLOGY

3.0 INTRODUCTION

This chapter will take a look at the study design. It will review the methods of data collection used in this research, data preprocessing which includes dealing with the data's missing values, the outlier effects, and feature selection. It will then touch on data analysis stage, which explores two types of analysis used in this project. After data analysis, it will give a brief demonstration on model building using the chosen machine learning algorithms which are K-Nearest Neighbor, Naïve Bayes, and Support Vector Machines. This will also allow the researcher to describe the machine learning models to be used in this project. Model evaluation will be the last aspect the researcher will look at. This will give insights into the best model to use for prediction of possible loan defaults. Model evaluation will be based on different evaluation metrics applied to evaluate the effectiveness of each model. Evaluation metrics will include the confusion matrix, the F1_score, accuracy, precision, recall, and the ROC/AUC.

3.1 MODELS

The dataset was analyzed using several different supervised machine leaning algorithms which include NB, KNN, and the SVM.

3.1.1 NAÏVE BAYES

The NB classifier is one of the most common and powerful classification algorithms. If a dataset has millions of examples with multiple attributes, the NB classifier is recommended. The Bayes theorem serves as the foundation for the NB classifier. Bayes probability is used in the Bayes theorem. Class condition independence is a Naive Bayes classification assumption based on Bayesian probability.

$$\text{P(A/B)} = \frac{(P(B/A) * P(B))}{P(B)} \quad (1)$$

Where, P(A) is the prior probability of the class;

P (*A*|*B*) is the posterior probability of class given predictor

P (B) is the probability of the predictor

P (B|A) is the likelihood of the probability to normalize the result.

Naïve Bayes classifier predicts the probability of each instance of a class, and the class with highest probability is counted as most likely class, the process of determining the class with the highest probability is called Maximum A posteriori (Saxena et al., 2017).

MAP (A):

= max (*P* (*A*|*B*))

= max ((*P* (*B*|*A*) *P (B)) / P (B)))

= max ((*P* (*B*|*A*) *P (B))

Predicting loan default is a classification problem and it needs Gaussian Naïve Bayes Classification, Gaussian Naïve Bayes gave powerful output in classification.

$$P(x_i/y) = \frac{1}{\sqrt{2\pi\delta^2 y}} \exp\left(-\frac{x_i - \mu_y}{2\delta^2 y}\right)^2 \quad (2)$$

The parameters μ_y *and* δ_y can be by estimated using the maximum likelihood.

3.1.2 SUPPORT VECTOR MACHINES MODEL

The SVM is one of the best-known and most effective supervised machine learning techniques. SVMs typically beat decision trees and logistic regression when it comes to categorization. SVM is frequently used for classification difficulties, but it can also offer incredibly hopeful outcomes for regression issues. Both continuous and categorical data may be easily handled by the SVM. For the purpose of classifying various groups, the support vector machine builds a hyperplane in multidimensional space. In an iterative process, SVMs create the ideal hyperplane to reduce error. (Navlani & Avinash, 2018), SVM separates binary classified data by a hyperplane such that the marginal width between hyperplane. By maximizing the marginal width, the complexity of the model has been reduced. For, imbalanced dataset SVM detects accuracy by this equation.

$$\text{Accuracy} = \frac{(TP+TN)}{(FP+FN+TP+TN)} \quad (3)$$

Where,

TP are True Positives a model managed to correctly classify;

TN are True Negatives a model managed to classify;

FP is False Positives, meaning instances that the model classified as positives whilst they are meant to be negatives; and

FN are instances that a model misclassified as negatives when they are actually positives.

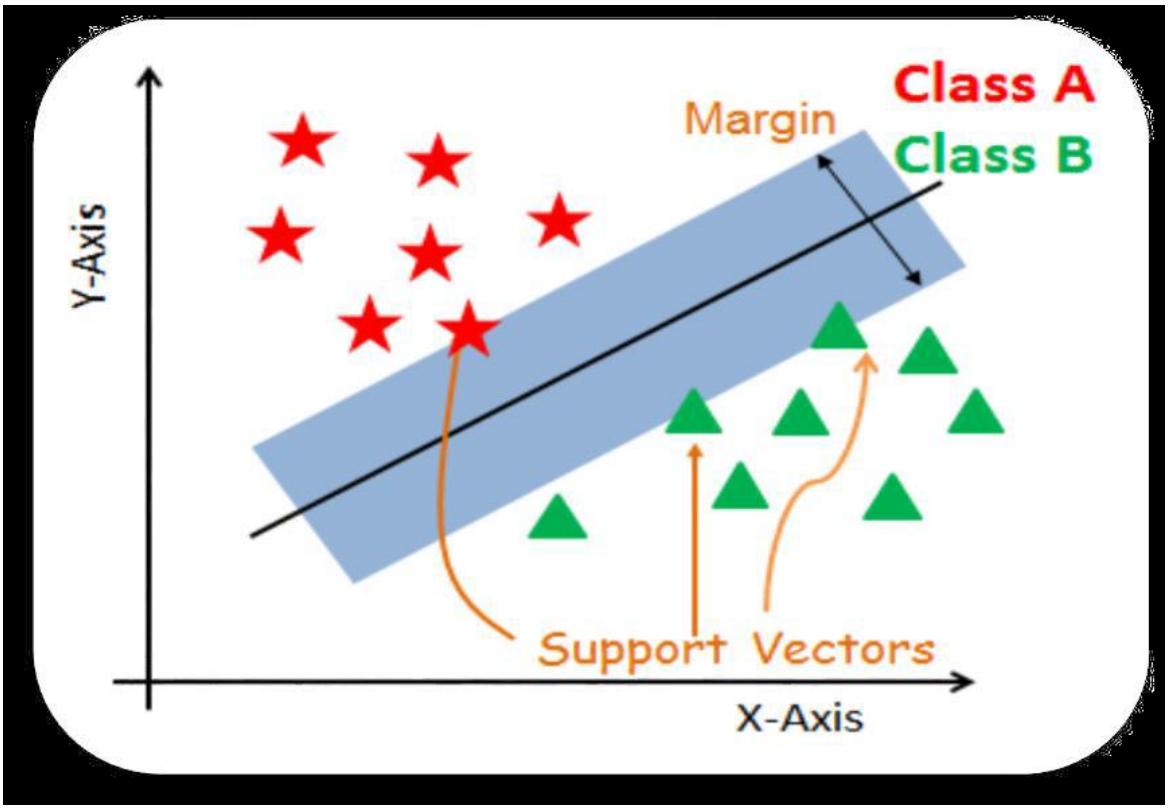


Fig 3.1: SVM Classification.

Figure 3.1 illustrates how the support vector machine algorithm is able to classify a new data point to a certain group by making use of the support vectors.

3.1.3 K-NEAREST NEIGHBORS MODEL

The KNN algorithm classifies new observations in the feature space using target values for the nearest neighbors. The hyperparameters include what is defined as a near neighbor. For instance, the number of neighbors could be set to ten. The new observation is then assigned to the class with the highest frequency of occurrence among the ten closest neighbors. Most often, the algorithm is weighted, such that the nearest neighbors have higher weights (Provost & Fawcett, 2015). The "distance" to a neighbor is also frequently defined as the Euclidean distance. The

number of neighbors to be chosen that best discriminates between classes is empirical, and several values should be investigated.

3.2 RESEARCH FRAMEWORK

The research framework is split into several sections, which includes looking at the tools used which includes the software of choice and its libraries. It also takes a look at study design which is further broken down into several parts which be looked at later on in this chapter. It will also review the performance of each model developed.

3.2.1 TOOLS USED

This section will highlight the analysis and programing tools used by the researcher throughout the project. It will touch on the software and the libraries used.

3.2.1.1 SOFTWARES

For this study, the researcher will employ Python. Every code and program will be run inside of a Jupyter notebook. Using live scripts, equations, narrative text, and visualizations, Jupyter Notebook is an interactive computing platform that is web-based and interactive.

Python has some of the most popular machine learning packages in today's world. Scikitlearn, one such package, presents users with several fundamental tools for creating neural networks and analyzing data (Educative, 2022). This makes python one of the versatile software to use for machine learning.

3.2.1.2 LIBRARIES

The python libraries imported helps in exploratory data analysis (EDA), data visualization, in data reduction scenarios, and for developing machine learning algorithms. Necessary information about the libraries is given below.

i. **Pandas** is an open source package in python used for data cleaning, data visualizations and exploratory data analysis.

ii. **NumPy** is an open source python package used for the provision of support for multi-dimensional arrays. The package is often seen as a replacement of Python List when we are working with an array of numbers. *NumPy* array is much faster for mathematical operation in array-like data (Gupta, 2020).

iii. **Matplotlib** is a visualization package. According to ActiveState (2022), *Matplotlib* is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

iv. **Math** gives access to mathematical functions.

v. **Seaborn** is a package built in matplotlib that performs the same task as *matplotlib* itself.

vi. **Sklearn** is a machine learning library including a variety of features such as classification, regression, and clustering techniques. *Scikit-learn (Sklearn)* is a Python computer language package that is commonly used in machine learning projects.

3.2.2 STUDY DESIGN

The major goal of this study is to use classification models to forecast whether or not a client would default on their loan. After specifying the algorithms and tools to be used in this research, the next thing is to look at how loan default will be predicted. Figure 3.2 visually illustrates the blueprint to be followed in loan default prediction. Each phase will be explained in detail in sections to follow.

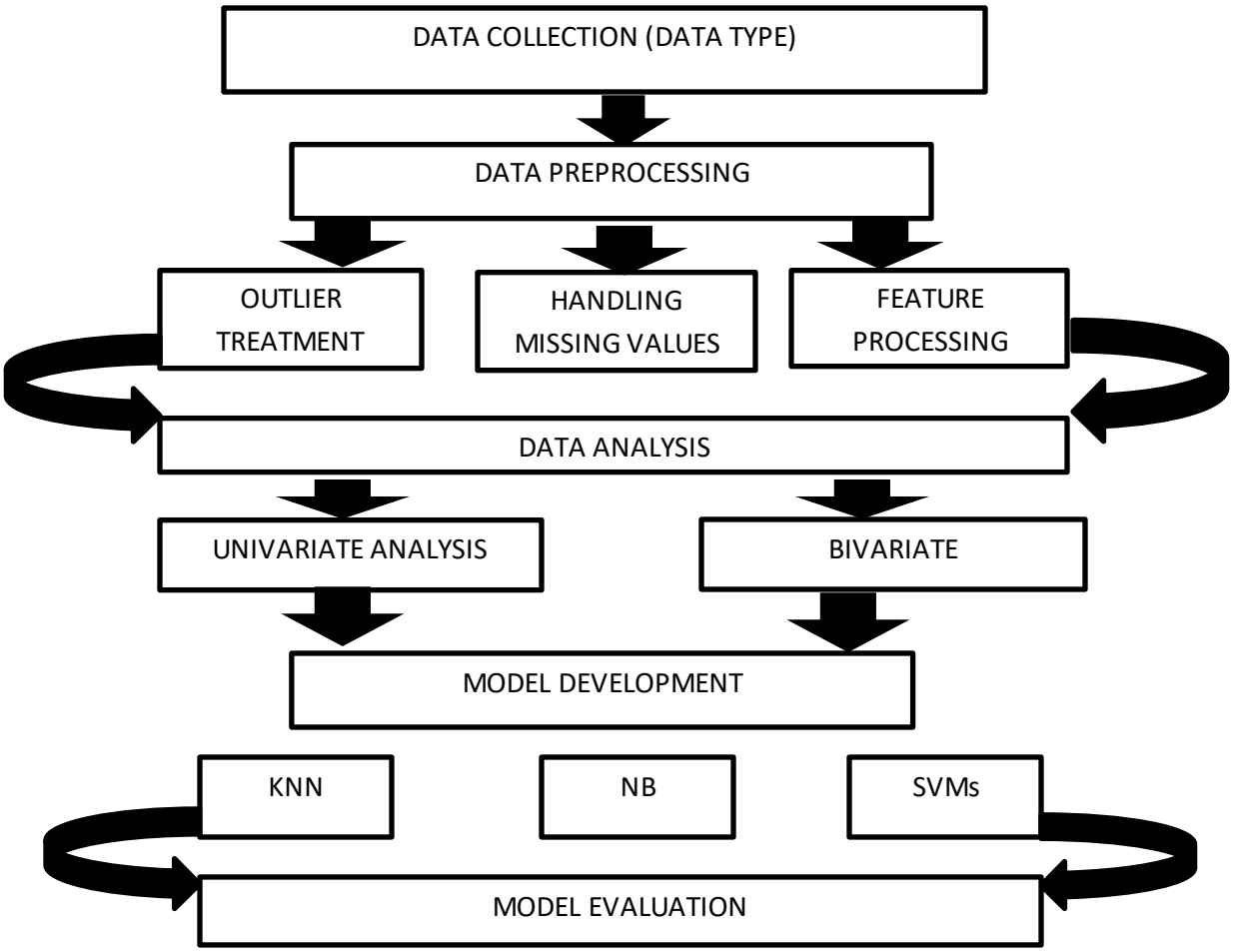


Fig 3.2: Study Design.

3.2.2.1 DATA COLLECTION

The dataset has been collected from a financial services firm. It was created by extracting information from the firm's data warehouse. (Serra, 2018), whereas first-hand data is intrinsically good, as it most likely provides trustworthiness and transparence about the phenomena researchers focus on, it is also likely to be fruit of hard work, expensive to obtain or gather and, overall, limited. Thus, the researcher opted for secondary dataset.

3.2.2.1.1 DATA TYPE

The institution is continuously reviewing the dataset which dates back from 2015 to 20222. It consists of 112 columns and 238,524 rows. The columns include some of the variables like the applicant's application identity number; this is used for easy location of the client's information given that some of the information is not well known. Age is another variable, it only provides the age of the client. The 'Default date' is one of the least variables mentioned. This variable consists of the actual date the client was deemed to have defaulted; this variable will help in determining the variable of interest. The whole dataset does not contain any binary information which may be vital in loan default prediction, thus during the analysis stage, some of the observations would be converted to binary format.

3.2.2.2 DATA PREPROCESSING

Data preprocessing includes the steps we need to follow to transform or encode data so that it may be easily parsed by the machine (Baheti, 2022). The main agenda for a model to be accurate and precise in predictions is that the algorithm should be able to easily interpret the data's features (Baheti, 2022). Data preprocessing may be divided into several subsections which in whole makes up the preprocessing phase of the data analytics life cycle. Figure 3.3 illustrates the subsections of preprocessing phase. The first subsection is data integration which may be defined as the process of merging data from different sources into a single, unified view. One may opt to transform the data they are using; hence data transformation may be implied. Data transformation is best defined as the process of converting from one format to the other format. Data reduction may be another process applied during data preprocessing and it may be defined as reducing the amount of capacity to store the information. This is important in data preprocessing phase as it can increase the storage efficiency and somehow reduce costs. The last subsection in data preprocessing is data cleaning. This is the process of removing incorrect,

corrupt, incorrectly formatted, and duplicate data within a dataset. The researcher intends to apply all the subsections of data preprocessing in the research.

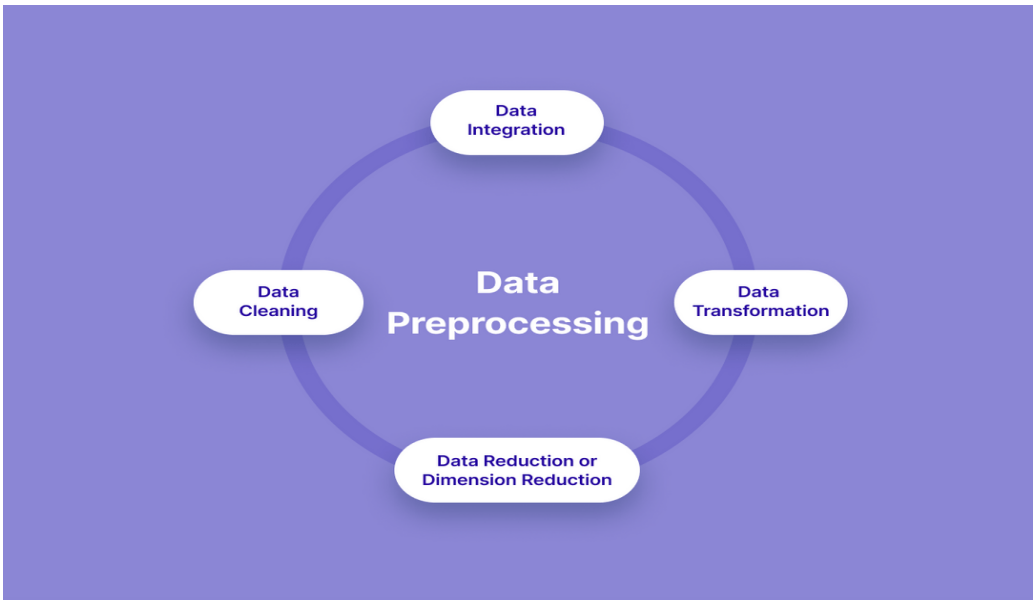


Fig 3.3: Data preprocessing phase (Baheti, 2022).

The lending dataset has 112 columns like loan ID, age, and working experience, only to mention, but a few. It also has 238,524 rows. The researcher aims at identifying and removing some of the duplicate variables that may decrease the storage space during the data analysis phase. Some of the variables include 'DateOfBirth' which is almost similar to 'Age', the other variable is 'LoanID' and 'CustomerId'. Either of the variables has to be removed during the data reduction subsection phase of data preprocessing.

Importing libraries for data preprocessing

Figure 3.4 illustrates the importation of python libraries that are mainly used for data analysis. *Pandas* is an open source package in python used for data cleaning, data visualizations and

exploratory data analysis. *Numpy* in some sense helps in creating data arrays which are easy to deal with especially in machine learning.

IMPORTING LIBRARIES FOR DATA PREPROCESSING

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import math
5 import seaborn as sns
```

Fig 3.4: Importing libraries

Figure 3.4 shows some lines of code that demonstrate data importation and renaming. This is followed by checking the actual structure of the data, that is checking the number of rows and columns the dataset has. The next library imported is *matplotlib* as shown in Figure 3.4; it is used for data visualizations. *Math* is another package or library imported. This library helps in executing math related functions and codes. The last data analysis package imported in the data analysis phase is the *seaborn* library. This library performs the tasks as *matplotlib* though in machine learning they are greatly used together.

```
#importing the dataset and assigning a new name to it.
data = pd.read_csv("LOANZIM.csv")
```

```
C:\Users\lesley\AppData\Local\Temp\ipykernel_7000\2035063265.py:1: DtypeWarning: Col
ypes. Specify dtype option on import or set low_memory=False.
  data = pd.read_csv("LOANZIM.csv")
```

```
#checking the number of rows and columns the dataset has.
pd.set_option("display.max_columns", 500)
data.shape
```

```
(238524, 112)
```

Fig 3.5: Importing the dataset.

Figure 3.5 best illustrates how data is imported in jupyter notebook. It is facilitated by the help of pandas library denoted in 'pd' in short when applying it. The dataset was to be named using a different name, for instance in our case it is now called 'data', which makes it easier for the researcher and readers to notice where we are coming from and going in terms of writing our codes. After data importation, the first thing to do under data preprocessing and analysis is to check the shape of our dataset. This is also illustrated in Figure 3.5.

3.2.2.2.1 DATA CLEANING

As stated earlier, the initial stage of data preprocessing is data cleaning. Data cleaning allows us to notice and get rid of several factors that may result in data over fitting or data under fitting. Thus, we check for missing values, check for outliers, and we go on to feature selection. All the stages will be discussed in sections to follow.

3.2.2.2.2 MISSING VALUES

Data cleaning involves tiding up the dataset by checking for several anomalies, for instance missing values. After missing values have been identified we go on to apply formal methods of solving that anomaly. Data imputation helps to fill up blank entries in a dataset; this is a vital action for model accuracy. The problem of missing values or entries may be solved by removing the column with blank entries, or replacing the zero entry with the column mean. In this case mean imputation is used for relevant columns with numerous missing values. Mean imputation is a type of data imputation that allows one to replace the missing values of a variable of interest by the mean of the non-missing values. Also dropping of columns which do not have any values will be applied since there are some variables in the dataset that do not have any value at all.

Figure 3.6 illustrates the code of identifying variables with missing values. This was represented in percentages. Variables with 0.0 indicate that they do not have any missing values, whilst the

ones with figures greater than one illustrate that they do contain missing values. For instance, the 'LoanId' variable had 0.0 meaning that it does not contain any missing value. Variable 'ContractEndDate' indicates that 61% of its values are missing. As stated earlier, some variables will be removed due to their irrelevancy in this research, whilst on some variables data imputation will be applied. This is being shown on the figure below.

```
#deriving the proportion of missing values on each column
pd.set_option("display.max_rows", None)
round(data.isnull().sum()/len(data.index), 2)*100
```

```
ReportAsOfEOD                          0.0
LoanId                                 0.0
LoanNumber                             0.0
ListedOnUTC                            0.0
BiddingStartedOn                       0.0
BidsPortfolioManager                   0.0
BidsApi                                0.0
BidsManual                             0.0
PartyId                                0.0
NewCreditCustomer                      0.0
LoanApplicationStartedDate             0.0
LoanDate                               0.0
ContractEndDate                       61.0
FirstPaymentDate                       0.0
MaturityDate_Original                  0.0
MaturityDate_Last                      0.0
ApplicationSignedHour                  0.0
ApplicationSignedWeekday               0.0
VerificationType                       0.0
```

```
missing_values1 = ['ContractEndDate', 'DateOfBirth', 'MonthlyPayment', 'County',
       'NrOfDependants', 'EmploymentDurationCurrentEmployer',
       'EmploymentPosition', 'WorkExperience', 'PlannedPrincipalTillDate',
       'PlannedInterestTillDate', 'LastPaymentOn', 'CurrentDebtDaysPrimary',
       'DebtOccuredOn', 'CurrentDebtDaysSecondary',
       'DebtOccuredOnForSecondary','PrincipalOverdueBySchedule', 'PlannedPrincipalPostDefault',
       'PlannedInterestPostDefault', 'EAD1', 'EAD2', 'PrincipalRecovery',
       'InterestRecovery', 'RecoveryStage', 'StageActiveSince', 'Rating',
       'EL_V0', 'Rating_V0', 'EL_V1', 'Rating_V1', 'Rating_V2',
       'ActiveLateCategory', 'WorseLateCategory', 'CreditScoreEsMicroL',
       'CreditScoreEsEquifaxRisk', 'CreditScoreFiAsiakasTietoRiskGrade',
       'CreditScoreEeMini', 'PrincipalWriteOffs',
       'InterestAndPenaltyWriteOffs', 'InterestAndPenaltyBalance',
       'NoOfPreviousLoansBeforeLoan', 'AmountOfPreviousLoansBeforeLoan',
       'PreviousRepaymentsBeforeLoan', 'PreviousEarlyRepaymentsBefoleLoan',
       'PreviousEarlyRepaymentsCountBeforeLoan', 'GracePeriodStart',
       'GracePeriodEnd', 'NextPaymentDate', 'NextPaymentNr',
       'NrOfScheduledPayments', 'ReScheduledOn', 'PrincipalDebtServicingCost',
       'InterestAndPenaltyDebtServicingCost', 'ActiveLateLastPaymentCategory']
data.drop(missing_values1, axis = 1, inplace = True)
data.shape
```

```
(238524, 59)
```

Fig 3.6: Removing columns with missing values.

3.2.2.2.3 DATA REDUCTION

Data reduction involves the total removal of duplicate variables. There are some variables like "country", "DateOfBirth", "Age" and "LoanId" that appear more than once in the dataset. For instance, "Age" and "DateOfBirth" are somehow one variable. Removing duplicate variables may increase the chances of model accuracy. Some of the variables are also deemed to be irrelevant in deriving our target variable of interest which will play an important role in loan default prediction.

```
#Removing irrelevant and duplicate columns that have nothing to do with loan default prediction
irr_variables = ['ReportAsOfEOD', 'LoanId', 'LoanDate', 'FirstPaymentDate', 'MaturityDate_Original',
       'MaturityDate_Last','Country', 'AppliedAmount','IncomeFromPrincipalEmployer', 'IncomeFromPension',
       'IncomeFromFamilyAllowance', 'IncomeFromSocialWelfare','IncomeFromLeavePay', 'IncomeFromChildSupport',
       'ActiveScheduleFirstPaymentReached','ModelVersion','Restructured', 'PrincipalPaymentsMade',
       'InterestAndPenaltyPaymentsMade']
data.drop(irr_variables, axis = 1, inplace = True)
data.shape
```

```
(238524, 35)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238524 entries, 0 to 238523
Data columns (total 35 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   LoanNumber                      238524 non-null  int64
 1   ListedOnUTC                     238524 non-null  object
 2   BiddingStartedOn                238524 non-null  object
 3   BidsPortfolioManager            238524 non-null  int64
 4   BidsApi                         238524 non-null  float64
 5   BidsManual                      238524 non-null  int64
 6   PartyId                         238524 non-null  object
 7   NewCreditCustomer               238524 non-null  bool
 8   LoanApplicationStartedDate      238524 non-null  object
 9   ApplicationSignedHour           238524 non-null  int64
 10  ApplicationSignedWeekday        238524 non-null  int64
 11  VerificationType                238524 non-null  int64
 12  LanguageCode                    238524 non-null  int64
 13  Age                             238524 non-null  int64
 14  Gender                          238524 non-null  int64
 15  Amount                          238524 non-null  int64
 16  Interest                        238524 non-null  int64
 17  LoanDuration                    238524 non-null  int64
 18  City                            238524 non-null  object
 19  UseOfLoan                       238524 non-null  int64
 20  Education                       238524 non-null  int64
 21  MaritalStatus                   238524 non-null  int64
 22  EmploymentStatus                238524 non-null  int64
 23  OccupationArea                  238524 non-null  int64
 24  HomeOwnershipType               238524 non-null  int64
 25  IncomeTotal                     238524 non-null  float64
 26  ExistingLiabilities             238524 non-null  int64
 27  LiabilitiesTotal                238524 non-null  float64
 28  ExpectedLoss                    238524 non-null  float64
```

Fig 3.7: Variable Reduction (Preprocessed dataset).

3.2.2.2.4 OUTLIER HANDLING

Outliers can have a negative effect on model training which can also reduce model accuracy. They also increase error variance and reduce the power of statistical tests. Outliers are usually caused by data entry, this can be an experiment measurement errors. Data manipulation or unwanted mutations in data collection can also be another possible cause of outliers. Fortunately, there are several suggested ways of dealing with outliers. The univariate method looks for data points with very high figures or values on one variable. The other method can be by the reduction of the contribution of the suspected outliers in the training process. This method is called the Minkowski error. The researcher proposed the visualization of data as a method of detecting outliers. Figure 3.8 shows how visualize were used to identify outliers. Variable 'Amount' proved to have large figures. Box plots and histogram was used to identify those outliers.

```
#Looking at amount and its outliers
data.boxplot(column = 'Amount')
```

```
<AxesSubplot:>
```

```
data['Amount'].hist(bins = 20)
```

```
<AxesSubplot:>
```

Fig 3.8: Outlier Detection.

3.2.2.2.5 DERIVATION OF TARGET VARIABLE

The preprocessed data does not contain the variable "Default" which could be used as the target variable which is supposed to be having two entry responses either "default" or "Non-default". Two variables that are much closer to the desirable target variable are "Status" and "DefaultDate". The "Status" variable has three unique values namely; current, repaid, and late. It is impossible to use the "Status" variable as our target variable alone because the predictor

models in machine learning only take two values, in this case “default” or “Not default”. The variable “DefaultDate” contains dates of when a certain client defaulted thus making it easier to tell that indeed the customer defaulted. To come up with the target variable, “Status” and “DefaultDate” will be merged together to create the variable of interest. This is because each of these variables has lower predicting power, so if used individually the model would produce inaccurate or undesirable results as compared to when they are merged together. After the derivation of the target variable, “Status” and “DefaultDate” variable will be dropped so as to avoid duplicate values or variables.

```
#Creating the predictor variable
data['Status'].value_counts()
```

```
Current    88274
Repaid     77981
Late       72269
Name: Status, dtype: int64
```

```
#Removing the 'current' status in 'Status' variable
data = data[data['Status'] != 'Current']
```

```
#Creating the target variable, incoperating "DefaultDate" into "Status"
#Where there is a date provided a 0 will be used to denote 'defaulted', and where there
#is no date, a 1 will denote "not defaulted"

data["Default"] = data["Status"].apply(lambda d: 1 if d == 'Repaid' else 0 )
data["Default"].value_counts()
```

```
1    77981
0    72269
Name: Default, dtype: int64
```

```
#dropping 'DefaultDate' and 'Status' variables
#To avoid duplicates

irr_variables2 = ['Status', 'DefaultDate']
data.drop(irr_variables2, axis = 1, inplace = True)
data.shape
```

```
(150250, 34)
```

Fig 3.9: Creating a Target (Predictor) Variable.

Figure 3.9 illustrates how the predictor variable is created. Initially, the ‘Status’ variable is examined on how many responses it contains. The next step is to remove the ‘current’ response; this is because it does not hold any value in the agenda of variable creation. This is done by creating a new dataset which contains variable ‘Status’ with only two responses which are

'Repaid' and 'Late'. The next step is to call upon the date of default and status variable and merge them, giving a certain condition. The condition simple implies that where there is a date provided in default date, that means the customer did default given that there appears a 'late' response in 'Status' variable. The repaid status is set at 1, given that late response is set at 0. In simpler terms this translates to 0 represents that the customer did default in the past, whereas 1 denotes non-default.

3.2.2.3 DATA ANALYSIS

Data analysis is a process of transforming preprocessed data into meaningful and insightful information that can easily be understood by everyone. This section will look at data analysis of the data that was preprocessed earlier on this chapter.

3.2.2.3.1 EXPLORATORY DATA ANALYSIS (EDA)

Exploratory data analysis is the initial stage of data analysis that includes performing critical and vital investigations on the dataset so as to identify any anomalies and trends, and to summarize the data based on graphical representations and summary statistics. Basically, there are three types of exploratory data analysis. Performing data analysis on one categorical or numerical variable so as to gain insights is called univariate analysis. Bivariate analysis is the statistical observation of two variables at the same time with an aim of getting insight about the variables. Observing at least three variables at the same time with an aim of understanding how they are related is called multivariate analysis. In this case, only univariate and bivariate exploratory data analysis will be reviewed by application of graphing visualizations. Bar graphs easily translate numerical information to visuals that everyone can understand, especially in the financial sector. They also help minimize data distortion.

i. UNIVARIATE ANALYSIS

Several categorical features which include 'Gender', 'Education', 'EmploymentStatus', 'NewCreditCustomer', 'Default Status', 'IncomeTotal', and 'Age' will be observed. The observation of the features assists in understanding the actual statistics behind the borrowers. It helps to answer the questions of; the most common gender in a loan application, the education level attained by most borrowers, the employment status of each borrower, and to answer the question of most common borrowers, if the process is most dominated by new borrowers or old ones. With reference from figure 3.10, the code shown is for generating numerous bar graphs under the univariate analysis phase. The library Ggplot was imported to provide basic, but stylish and clear visuals in the form of bar graphs.

```
#Univariate Analysis

#DEFAULT# 0:Defaulted, 1:Did Not Default
#GENDER# 0:Male, 1: Female, 2:Unspecified
# EDUCATION# 1:Primary education, 2:Secondary Education, 3:Vocational Training, 4:Higher Education
# MEWCREDITCUSTOMER# True: New Clients, False: Old clients
#EmploymentStatus

plt.style.use('ggplot')
figure, ax1 = plt.subplots(1,6)
data['Default'].value_counts(normalize = True).plot(ax = ax1[0],figsize=(20, 5),kind =
'bar',title = 'Default', color = "b", rot = 0)
data['Gender'].value_counts(normalize = True).plot(ax = ax1[1],kind =
'bar',title = 'Gender', color = "g", rot = 0)
data['Education'].value_counts(normalize = True).plot(ax = ax1[2], kind = 'bar', title =
'Education', color = "r", rot = 0)
data['NewCreditCustomer'].value_counts(normalize = True).plot(ax = ax1[3],kind = 'bar', title =
'NewCreditCustomer', color = "b", rot = 0)
data['Age'].value_counts(normalize = True).plot(ax = ax1[4], kind = 'bar',title =
'Age', color = "g", rot = 0)
data['EmploymentStatus'].value_counts(normalize = True).plot(ax = ax1[5], kind = 'bar',title =
'EmploymentStatus', color = "r", rot = 0)
figure.tight_layout()
```

Fig 3.10: Univariate Analysis.

The bar graphs are incorporated into a single visualization object. Each figure size influenced by the default bar graph is set at 20 inches of height and 5 inches of length. The whole idea behind it was to fit all six visuals in one matrix of visuals.

ii. BIVARIATE ANALYSIS

Features observed in the univariate analysis will be reviewed again in bivariate analysis to observe their interrelation with the target variable (Default). Bivariate analysis helps to gain insights on two features at the same time. It does not consider causes, but it just considers the relationship between the two variables of interest under study. This is to provide us with the gender that has the highest defaulters, the type of education each defaulter attained, their employment status relating to defaulting, and if new customers default more than the old customers.

```
#Bivariate Analysis
#Independent Variables vs. Dependent(Target Variable)
#EmploymentStatus vs. Default

sns.countplot(y = 'EmploymentStatus', hue = 'Default', data = data)
```

```
#NewCreditCustomer vs. Default
sns.countplot(data = data, hue = 'Default', y = 'NewCreditCustomer')
```

```
<AxesSubplot:xlabel='count', ylabel='NewCreditCustomer'>
```

Fig 3.11: Bivariate Analysis.

The earlier imported seaborn library is now applied to visualize the features in a bivariate manner.

Figure 3.11 is a visualization code, which explains how two variables are compared at the same time. For instance, the first block of code illustrates the relationship between the newly formed

variable ‘Default’ and ‘EmploymentStatus’, this is expected to produce some statistics which clarifies the total number of those employed and not employed on how frequently they default.

3.2.2.3.2 CATEGORICAL FEATURE CONVENTION TO NUMERICAL VARIABLE

Model accuracy and preciseness depends on how well a model was trained. For a model to be accurate it has to be trained on numerical data (input), in return it will produce numerical results (output) as well. Machine learning algorithms in the same sense cannot handle categorical variables thus the need to first convert those variables to numerical variables. The variables ‘PartyId’, ‘NewCreditCustomer’, and some will be converted to numerical features. *Scikit-learn or SKlearn* is the package that contains all machine learning algorithms, thus it is important to import it before the convention process. This is illustrated in figure 3.12.

```
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
varr = ['NewCreditCustomer', 'LoanApplicationStartedDate','ListedOnUTC',
        'PartyId','BiddingStartedOn','City']
for vari in varr:
    le = preprocessing.LabelEncoder()
    data[vari] = le.fit_transform(data[vari].astype('str'))

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150250 entries, 159 to 238523
Data columns (total 34 columns):
```

Fig 3.12: Categorical Variable Convention.

3.2.2.4 MODEL DEVELOPMENT

After data preprocessing and analysis, the next stage is to develop models that will predict loan defaults. The main purpose of the research is to build a model that can easily predict if a client will default or not if they are granted some loan. This will basically reduce pressure from the available staff at the institution; it will also ascertain that the rightful client gets the loan based on his past borrowing experiences. The researcher decided to build three models based on SVMs,

Gaussian NB, and KNN algorithms, and the one with the highest predicting power after evaluation will be chosen as the desirable model for loan default prediction.

3.2.2.4.1 PROPOSED MODEL

The desirable model should be able to fulfill the task at hand by producing accurate results. The model's task is to be able to predict if a borrower will default on a loan or not. Figure 3.13 illustrate a logical flow chart of how the proposed model should be developed, deployed and produce results.

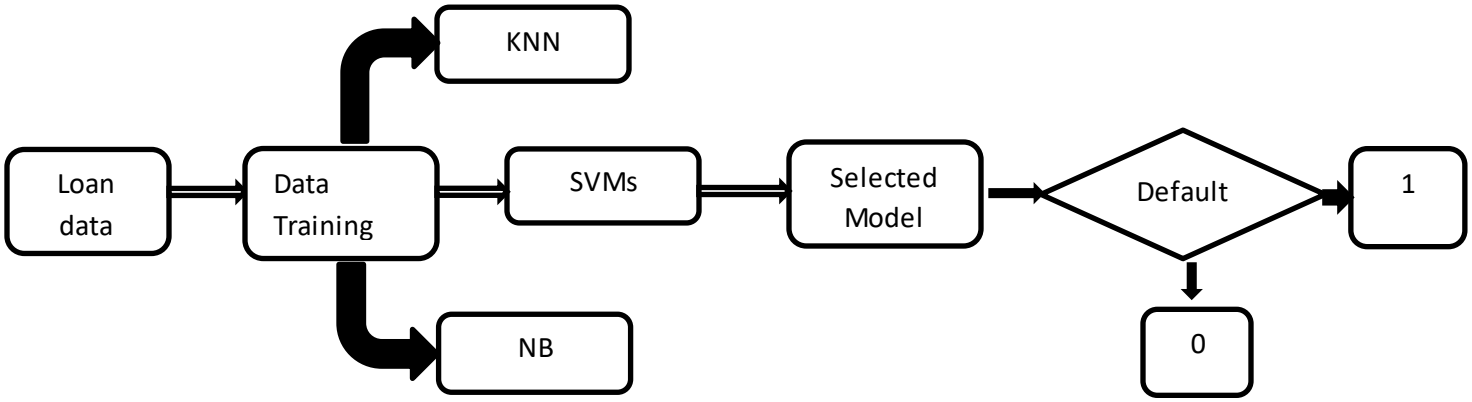


Fig 3.13: Proposed Model.

3.2.2.4.2 THEORETICAL PROCEDURE

The theoretical procedure gives a summary of each stage of model development. Below is a brief explanation of what will be happening in each phase.

i. **Model development**.

Model development includes the formulation of a model computationally. The use of *scikitlean* or *sklearn* library proves to be of vital importance as this is the library that contains all machine learning algorithms if imported. The models being built include NB Gaussian, KNN, and the

SVMs. After the development of the model(s), they are then trained using the loan application dataset that is acquired from the lending institution.

ii. **The loan application data is fed into the trained model**.

Given that our models are already trained, we are then able to ingest another dataset that we want to analyze or draw conclusions from. In our case, we will stick to the loan default prediction. The loan application dataset is fed into the models.

iii. **Classification and prediction**.

Now that the dataset was ingested in either of the built and trained models, we then move on to binary classification and prediction of our results. The objective is to have insights on whether an applicant will default in their loan given that they did default or did not default in the past. Thus, we make use of the loan dataset to determine that. Those that will default are classified as 0 and those that will not are classified as 1. This is because of binary data points created during the data preprocessing phase of our research.

iv. **Model Evaluation**.

The trained models are then evaluated based on their performance in loan default prediction. All three models were trained thus, being evaluated individually. The evaluation metrics applied are the confusion matric, the accuracy rate, the precision, the recall, the f1_score, and the ROC/AUC metrics. These will be discussed further in sections to follow.

3.2.2.4.3 VARIABLE DECLARATION

The initial step in machine learning model building is to declare the feature into some categorical distinguishing variables first. According to figure 3.14, the categorical (independent) features

which include 'Gender', 'NewCreditCustomer', 'Education', 'EmploymentStatus', and 'Restructured' will be declared an **X** variable. This is because these variables are correlated to loan default. The dependent feature 'Default' will be declared **Y** and an array created as well. The 'iloc' function is defined in *Pandas* which helped the researcher to specifically select the columns to be incorporated in each of the newly formed variables.

```
#VARIABLE DECLARATION

X = data.iloc[:, np.r_[7,14,20,22]].values
Y = data.iloc[:,33].values
```

Fig 3.14: Variable Declaration.

3.2.2.4.4 DATA TRAINING

Data splitting is the process of splitting the cleaned dataset under study into two parts; train and test data. Model accuracy and preciseness is the backbone of every model built, thus training it before deployment is an important step. The model requires train and data in order to be developed appropriately.

The splitting ratio of train data to test data used is 70%**:** 30%. This is because model accuracy is mostly determined by the amount of data it was trained on. Thus, the data is split into two using *sklearn* library. First, the data was divided into features X and labels Y. The dataframe after dividing the dataset is further divided into X_train, X_test, Y_train, and Y_test datasets. The train datasets are used to train the models to predict based on the past events, whilst the tests sets are used to test the models if its predicting power is efficient and sufficient. Figure 3.15 illustrates the lines of code that split data into train and test datasets. They are split into 70% and 30% train and test datasets respectively.

```
#data splitting to train and test sets

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.3, random_state = 0)
```

Fig 3.15: Splitting Data.

3.2.2.4.5 DATA STANDARDIZATION

Data standardization is a process with an application of statistical methods or any other data processing workflow that allows the convention of different data structures into one common format of data. This is usually done by turning the data into a dataset with the distribution of mean 0 and standard deviation of 1. Data standardization is so important in data preprocessing because it allows the data (after cleaning) to be in one format, thus making it easier to apply machine learning algorithms and derive accurate observations. In figure 3.16, *SKlearn* was once again applied to data standardization process. The standard scalar package in *SKlearn* is used to remove the mean in the dataset and scaling to unit variance. The centering (mean removal) and scaling (obtaining variance 1) happens independently for every variable in the training and testing datasets after the data has been split into two. This is illustrated in figure 3.16:

```
#stndardizing the dataset to a set with mean 0 and variance 1

from sklearn.preprocessing import StandardScaler
da = StandardScaler()
X_train = da.fit_transform(X_train)
X_test = da.transform(X_test)
```

Fig 3.16: Data Standardization.

3.2.2.4.6 MODEL TRAINING AND TESTING

The next step is to train and test the model after splitting the dataset into two sets of data (train and test), each with two datasets (train: X train and Y train, test: X test and Y test). This section will go over how the model is generally trained and tested. The test and train subsets will be used to independently train and test NB, KNN, and SVMs.

```
# Model building

#NAIVE BAYES CLASSIFIER (NBGAUSSIAN)

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, sensitivity_score
from sklearn.metrics import classification_report, confusion_matrix
NBClassifier = GaussianNB()
NBClassifier.fit(X_train, Y_train)
```

Fig 3.17: Naïve Bayes Model Building (Training).

Figure 3.17 illustrates the development of the NB model. *Sklearn (scikitlearn)* is the default library used in machine learning algorithm development. The model built is the Gaussian format of the NB. The researcher also included the building of the evaluation metrics which will be discussed later on in this chapter. In model training, only the train datasets of the dependent variable (Default Y) and independent variables (merged variable X) were used.

```
#K-NEAREST NEIGHBOR CLASSIFIER (KNN)
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, sensitivity_score
from sklearn.metrics import classification_report, confusion_matrix
KNN = KNeighborsClassifier()
KNN.fit(X_train, Y_train)
```

Fig 3.18: K-Nearest Neighbor Model Building (Training).

Figure 3.18 illustrates the development of the KNN model. *Sklearn (scikitlearn)* is the default library used in machine learning algorithm development. The researcher also included the building of the evaluation metrics which will be discussed later on in this chapter. In model training, only the train datasets of the dependent variable (Default Y) and independent variables (merged variable X) were used.

```
#SUPPORT VECTOR MACHINE (SVM)
from sklearn import svm
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, sensitivity_score
from sklearn.metrics import classification_report, confusion_matrix
svm.SVC(kernel = 'linear', gamma = 'auto', C = 2)
SVM = svm.SVC(kernel = 'linear', gamma = 'auto', C = 2)
SVM.fit(X_train, Y_train)
```

Fig 3.19: Support Vector Model Building (Training)

Figure 3.19 illustrates the development of the SVM model. *Sklearn (scikitlearn)* is the default library used in machine learning algorithm development. The *Kernel* function is a method used to take data as input and transform it into the required form of processing data. It is used due to a set of mathematical functions used in SVM providing the window to manipulate the data. So, the *kernel* function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces (GeeksforGeeks, 2022). (Pedregosa et al., 2011), intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and the high values meaning 'close'. They can be seen as the inverse of the radius of influence of samples selected by the model as support vector. The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points accurately. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words C behaves as a regularization parameter in the Support Vector Machine (Pedregosa et al., 2011). The researcher also included the building of the evaluation metrics which will be discussed later on in this chapter. In model training, only the train datasets of the dependent variable (Default Y) and independent variables (merged variable X) were used.

3.2.2.4.7 LOAN DEFAULT PREDICTION

In the previous sections of this chapter, the researcher looked at data preprocessing, model building, and model training. This section will then look at how the developed models are used to predict loan defaults. As loan default prediction is the main priority of this research, the researcher intends to look at each and every aspect carefully.

```
Y_pred = NBClassifier.predict(X_test)
Y_pred
```

Fig 3.20: Loan default prediction.

Figure 3.20 illustrates how the developed models are then used to predict loan defaults. This is done by calling a new variable in each of the models developed (Y_pred, pred meaning predict). This is done by using the X test dataset which is the test data of the merged variables which form our independent variable.

3.3 MODEL EVALUATION

The researcher proposes to evaluate the models used in default predictions. This will allow the main objective of the study to be met, which is to come up with a more versatile and accurate model that can easily predict loan defaults.

On a high level, Machine Learning is the union of statistics and computation. The crux of machine learning revolves around the concept of algorithms or models which are in fact statistical estimations on steroids. However, any given model has several limitations depending on the data distribution. None of them can be entirely accurate since they are just estimations (Ghosh, 2022). Thus, the need to choose the one with the highest accuracy rate compared to others. According to (Ghosh, 2022), model evaluation is a method of assessing the correctness of

models on test data. The test data consists of data points that have not been seen by the model before.

Several evaluation methods will be applied in the process of evaluating the models used to predict loan defaults. These methods include the confusion matrix, the accuracy rate, precision, recall, and the ROC/AUC. More detailed information about the methods is shared in the sections to follow.

3.3.1 CONFUSION MATRIX IN BINARY CLASSIFICATION

An overview of the results of predictions on a classification task is provided by a confusion matrix. It gives a tallied count of the right and wrong guesses. An evaluation approach is used to assess the effectiveness of classification models for a specific set of test data. The confusion matrix can be computed if the actual values of the test data are known. Because of the following benefits, it is frequently preferred to alternative evaluation techniques:

- It delivers information on errors made by the classifier and the sort of errors that are being created.
- It is used when there is a severe imbalance in the classification problem and one class dominates the others.
- It exhibits the disarray and confusion that characterize a classification model's prediction process.
- It can be used to compute the ROC-AUC curve as well as Recall, Precision, Specificity, and Accuracy.

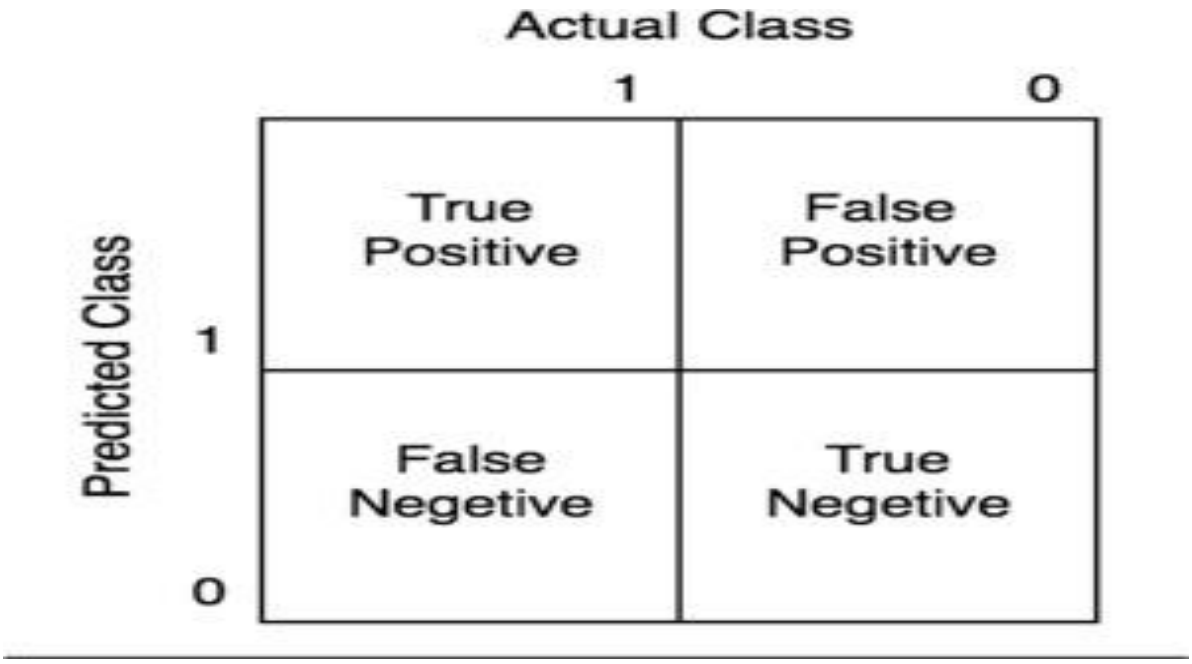


Fig 3.21: Confusion Matrix for Binary Classification.

Figure 3.21 illustrates the composition of a confusion matrix. It has four sub-boxes represented in binary format (1s and 0s) on both sides; the model is evaluated based on if it actually got the results correct or incorrect.

True Positive is a scenario whereby the observation is predicted positively and is actually positive. This literally means that the model got the prediction correct based on the actual fact that it was a true instance.

False Positive (also known as type one error) is a scenario where the observation is predicted positive, whilst it is actually negative. The model might have predicted the observation true, but in actual fact it turns out it is false.

False Negative (also known as type two error) on the other hand, is a scenario whereby an observation is predicted negative, whilst it is actually positive.

True Negative is when the predicted observation is negative, and is actually negative.

Within the confusion matrix, there are several metrics (parameters) that justify the figures that we get in the confusion matrix. A clear illustration on how we derive these metrics is provided below.

3.3.2 ACCURACY

This is a metric that evaluates the model performance across all classes. It is generally calculated as the ratio between the number of correct predictions to the number of total predictions. Being one of the most common classification metrics, accuracy is very intuitive and easy to understand and implement. It ranges from 0 to 100 percent (altexsoft, 2022). The formula is given below:

$$\text{Accuracy} = \frac{True_{positives} + True_{negatives}}{True_{positives} + True_{negatives} + False_{positives} + False_{negatives}}$$

3.3.3 PRECISION

Precision is calculated as the ratio between the number of positive observations correctly classified to the total observations classified as positive. Precision does well in cases when one needs to or can avoid False Negatives but cannot ignore False Positives (altexsoft, 2022). It is highly recommended when dealing imbalanced data.

$$\text{Precision} = \frac{True_{positives}}{True_{positives} + True_{negatives}}$$

Usually when the precision is small it is because of the model making too many incorrect positive classifications which tends to increase the denominator and as a result lower the precision percentage. On other hand, if the precision is high it will be because of the model making too many correct positive classifications.

3.3.4 RECALL

It is calculated as the ratio between the number of positive observations correctly classified as positives to the total number of positive observations.

$$\text{Recall} = \frac{True_{positives}}{True_{positives} + True_{negatives}}$$

3.3.5 F1-SCORE

The F1_score is a more intricate metric that allows you to get results closer to reality on imbalanced classification problems (altexsoft, 2022). F1-score is the harmonic mean of precision and recall. It is given by the following formula;

$$\text{F1-score} = 2 * \frac{precision * recall}{precision + recall};$$

Where,

Precision is the evaluation metric described in section 3.3.4,

Recall is also an evaluation metric described in section 3.3.5.

3.3.7 AUC-ROC CURVE

ROC curves, or receiver operating characteristic curves, are one of the most common evaluation metrics for checking a classification model's performance (Agarwal, 2021). It is a probability curve that plots two parameters, the True Positive Rate (TPR) against the False Positive Rate (FPR), at different threshold values and separates a so-called 'signal' from the 'noise'. If the user lowers the classification threshold, more items get classified as positives, which increase both the False Positives and the True Positives (Terra, 2022).

AUC is short for ‘Area Under the ROC Curve’, which measures the whole two dimensional area located underneath the entire ROC curve from (0, 0) to (1, 1). The AUC measures the classifier’s ability to distinguish between classes. It is used as a summary of the ROC curve. The higher the AUC, the better the model can differentiate between positive and negative classes. AUC supplies an aggregate measure of the model’s performance across all possible classification thresholds (Terra, 2022).

Some of the reasons why the researcher intends to use AUC as an evaluation method are stated below:

- AUC is a scale-invariant. It swiftly measures how well the predictions were ranked instead of measuring their absolute values.
- AUC is classification-threshold-invariant. This means that it measures the quality and uniqueness of the model’s predictions regardless of the classification threshold.

Figure 3.22 explains the AUC-ROC visually. As the model (denoted by a red dotted line as a random classifier) shoots up towards the curves, it means that it is a good model with a high performance rate.

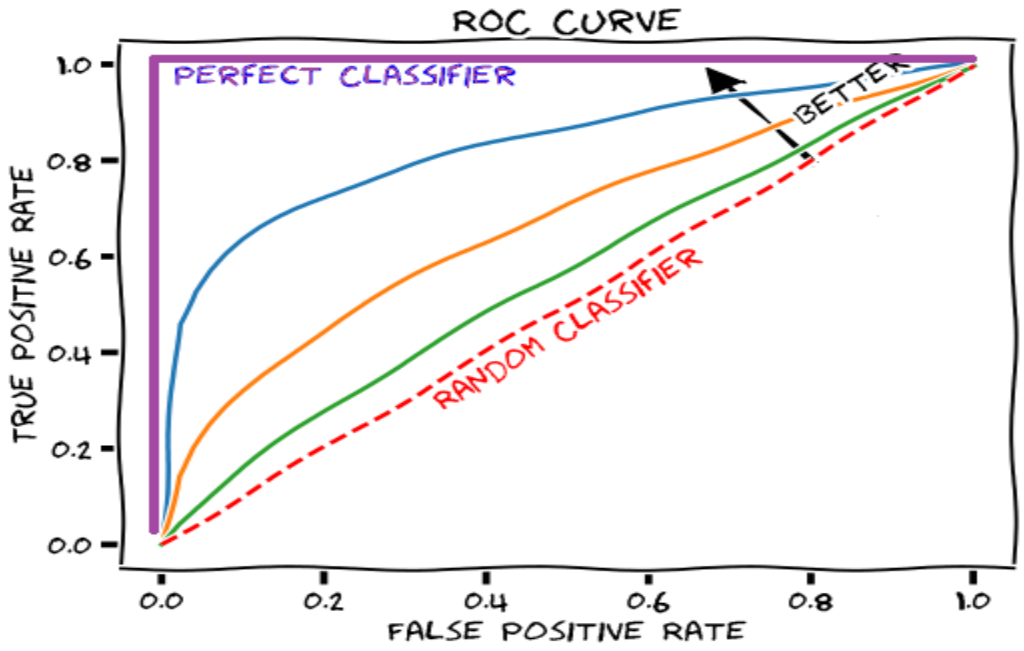


Fig 3.22: AUC-ROC curve.

3.4 SUMMARY

The chapter highlighted several stages in data analysis and model building as a whole. It covered the descriptive statistics part and data visualization which intends to bring out understandable results. On the aspect of model building; it touched on how data is reduced to a desirable set using sklearn which is also used to call up machine learning algorithms and other machine learning techniques. It also reviewed the aspect of data splitting, how data is split into train and test data soon after deriving the feature of interest (Default). Loan default prediction is another aspect that was looked at, where different models or algorithms that where built are used to predict defaults. These include SMVs, which use the method of choosing the extreme points or vectors that help in creating the hyperplanes. KNN is also implemented, it uses different distance metrics (such as the Euclidean distance method) to derive the distance between the new data point and the existing data class given that they have similarities. Lastly, Gaussian NB is to be

implemented as well. This looks at numerous conditions so that it can classify new data points accurately. As discussed in chapter 2, the algorithms implemented are supervised machine learning algorithms. Machine learning and statistics link well in the area of predictions. They function differently, but obtain the same goal at last.

CHAPTER FOUR: DATA ANALYSIS AND RESULTS INTERPRETATION

4.1 INTRODUCTION

Data analysis and result interpretation are the main topics of this chapter. Several Python libraries, including *matplolib* and *seaborn*, were employed throughout the data analysis phase. This is due to the fact that the majority of the results are presented visually, such as in bar graphs. It also emphasizes how to analyze the outcomes of the created loan default prediction models. The proposed models, which include SMVs KNN, and NB, were applied in accordance with the goals of the study. Additionally, each model's results are examined in relation to how they connect to the methodology and problem statement.

4.2 DETAILED STATISTICAL ANALYSIS OF THE LOAN DATASET

Descriptive statistics, according to Trochim (2022), are used to describe the fundamental characteristics of data in a study. They provide concise summaries of the sample and measurements. They are the foundation of almost every quantitative data analysis, along with simple graphics analysis. The researcher started by performing descriptive analysis on the loan dataset, this was to gain insights on categorical as well as on numerical features of the data. The analysis takes a direct look at variables that play a pivotal role in the client's default. In previous chapters, age was categorized as an independent variable that contributes to potential loan default. Annual salary (IncomeTotal) and employment status are also thought to play a role in loan default. Other variables of interest, such as gender, marital status, and education status, were also analyzed in relation to the formed dependent variable of interest (default) to determine which gender defaults the most, which gender defaults the most, and which educational criterion many defaulters fall under. In machine learning, it is advisable to carry out data preprocessing before any data analysis is carried out. Thus, in this section, data analysis (exploratory data analysis) of preprocessed data will be considered.

4.2.1 EXPLORATORY DATA ANALYSIS

4.2.1.1 UNIVARIATE ANALYSIS

This section looks at the descriptive information of the suggested features of interest, which include independent variables such as Age, Gender, Education Status, and Income Total. It also looks at the predictor variable (Default). Summary statistics about the variables are given below:

Descriptive Statistics	**Default**	**Age**	**Gender**	**Education Status**	**Income Total**	**Employment Status**
Max	1	70	N/A	N/A	5000	N/A
Min	0	18	N/A	N/A	200	N/A
Mean	N/A	39.9	N/A	N/A	1531.1	N/A
Mode	1	31	0	N/A	1200	N/A
Std.	N/A	12.3	N/A	N/A	829.3	N/A
25%	0	30	N/A	N/A	976.7	N/A
50%	N/A	38.4	N/A	N/A	1254.5	N/A
75%	1	49	N/A	N/A	2000	N/A

Table 4.1: Summary Statistics.

4.2.1.1.1 INTERPRETATION OF UNIVARIATE ANALYSIS RESULTS

The summary statistics for the independent variables and the dependent variable are displayed in Table 4.1. Since the variables marked with "N/A" in summary statistics are thought to be

categorical variables, they are unable to yield useful numerical statistics. Simpler explanations of the statistics are provided below:

i. Default

The default variable, which is also the predictor variable, has a maximum value of 1, which simply means that the greatest value in default is 1. It also has a minimum value of 0. These values represent the binary notation, with 1 meaning the customer did not default and 0 meaning the customer indeed did default. This part of the analysis was done to see if there were any error values other than 1 and 0. The variable did not have any mean, this is due to the fact that we cannot derive a mean for the binary variable, and even if we did, it would not give us any relevant insights about the data or the variable itself.

The mode appeared to be 1. Mode helps us understand the figure that appears the most in that variable, for instance, in our case, the mode is 1. This means that the figure that appears the most is 1, which further translates to the decision that the company has many non-defaulters as compared to defaulters, which are represented by the value 0. According to figure 4.1, twenty-five percent were discovered to be defaulters, whereas seventy-five percent were believed to be non-defaulters. This was determined by the quartile ranges, which were derived from the box plot shown below.

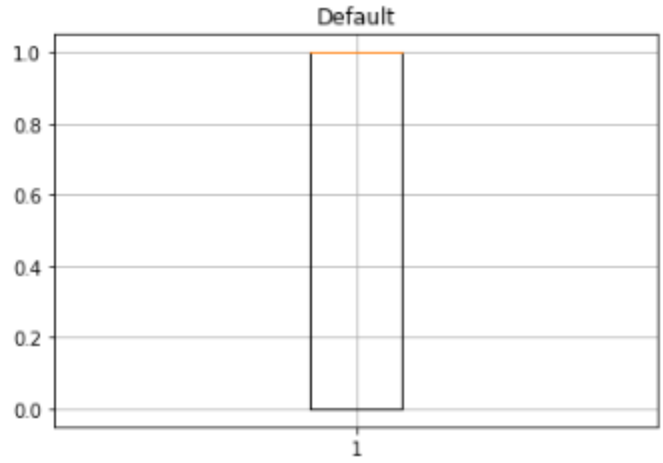


Fig 4.1: Default boxplot.

ii. AGE

The maximum age is 70, which means we had 70-year-old customers. The company also has an 18-year-old client. The average age of the applicants is 39.9 years. Most age that appears the most in loan applications is 31. This means that we have many clients who are over 31. The standard deviation is 12.3. According to figure 4.2, many of the applicants are under 30 years old, putting them in the lower percentile. The average number of applicants is 38.4.

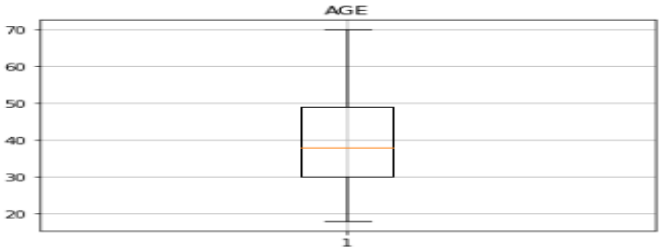


Fig 4.2: Age Box plot.

iii. GENDER

The researcher only managed to get the gender with the highest number of applicants. From the analysis, it was derived that males were the most likely applicants. Males are denoted by a 0 in the dataset, and code snippets are to be provided.

iv. INCOME

The dataset mainly considers the ones who are earning United States dollars instead of Zimbabwean dollars or RTGS. The highest earner seemed to be earning $5000 USD, with the lowest earner taking home $200 USD. Figure 4.3 provides a boxplot for the income variable.

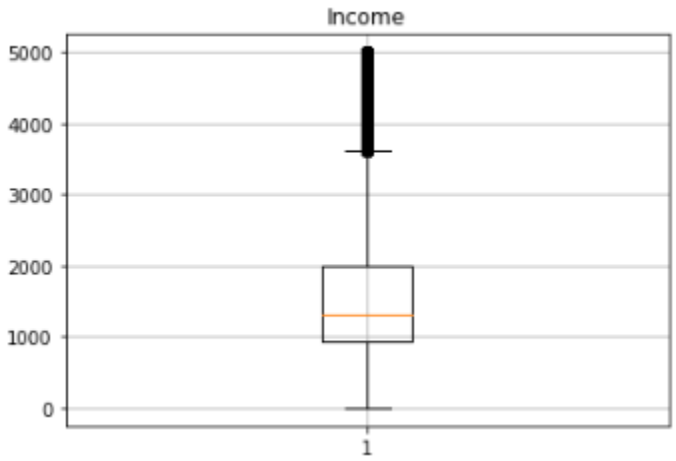


Fig 4.3: Income Box plot.

4.2.1.2 DATA VISUALISATION AND REPORTING

This section will go deeper into the analysis of several categorical and numerical variables, as done above in the summary statistics. The only difference between this section and the one above is that; this section will analyze the data visually as compared to summary statistics. Also, there will be some independent variables considered as well. "Gender," "new credit customer," "marital status," "employment status," and "education" are among them.

```
#Univariate Analysis
#DEFAULT# 0:Defaulted, 1:Did Not Default
#GENDER# 0:Male, 1: Female, 2:Unspecified
# EDUCATION# 1:primary education, 2:Basic Education, 3:Vocational Training, 4:Secondary Education
# 5: Higher Education.

plt.style.use('ggplot')
figure, ax1 = plt.subplots(1,3)
data['Default'].value_counts(normalize = True).plot(ax = ax1[0],figsize=(22, 7),kind =
'bar',title = 'Default', color = "b", rot = 0)
data['Gender'].value_counts(normalize = True).plot(ax = ax1[1],kind =
'bar',title = 'Gender', color = "g", rot = 0)
data['Education'].value_counts(normalize = True).plot(ax = ax1[2], kind = 'bar', title =
'Education', color = "r", rot = 0)
figure.tight_layout()
```

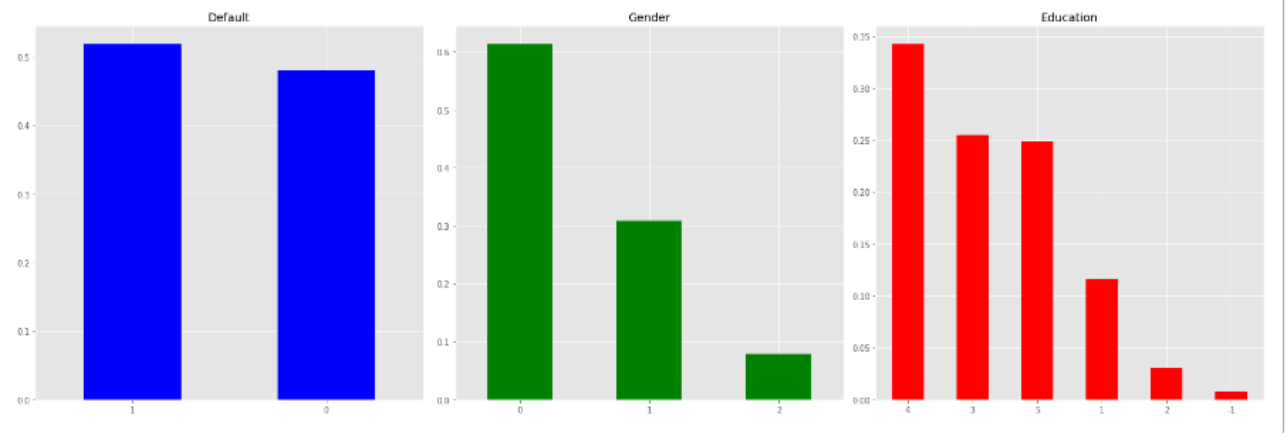


Fig 4.4: Univariate Analysis

4.2.1.2.1 OBSERVATIONS

Figure 4.4 Results Interpretations:

51.9% are believed to be non-defaulters, while 48.1 % are loan defaulters.

61% are male applicants (denoted by 0 in the bar graph), 30% are female (denoted by a 1) applicant, and 9% are believed to be of unspecified gender (denoted by a 2).

34% of the applicants are believed to have attained only secondary education, while 26% attained vocational training education, 25% got higher education, 11% attained primary education, 3% attained basic education, and 1% are unspecified as to which education they attained.

```
# MEWCREDITCUSTOMER# True: New Clients, False: Old clients
#EmploymentStatus# -1: unspecified, 1: employed, 2: partially employed
# 3:  unemployed, 4: self-employed, 5: Entreprenuer, 6: retiree.
#Marital Status# -1:unspecified, 1: married, 2:Cohabitant, 3: Single, 4: Divorced, 5: Widow


plt.style.use('ggplot')
figure, ax2 = plt.subplots(1,3)
data['EmploymentStatus'].value_counts(normalize = True).plot(ax = ax2[0],figsize = (22,7),
kind = 'bar',title =
'EmploymentStatus', color = "r", rot = 0)
data['NewCreditCustomer'].value_counts(normalize = True).plot(ax = ax2[1], kind = 'bar',title =
'NewCreditCustomer',rot = 0)
data['MaritalStatus'].value_counts(normalize = True).plot(ax = ax2[2], kind = 'bar',title =
'MaritalStatus',rot = 0)
figure.tight_layout()
```

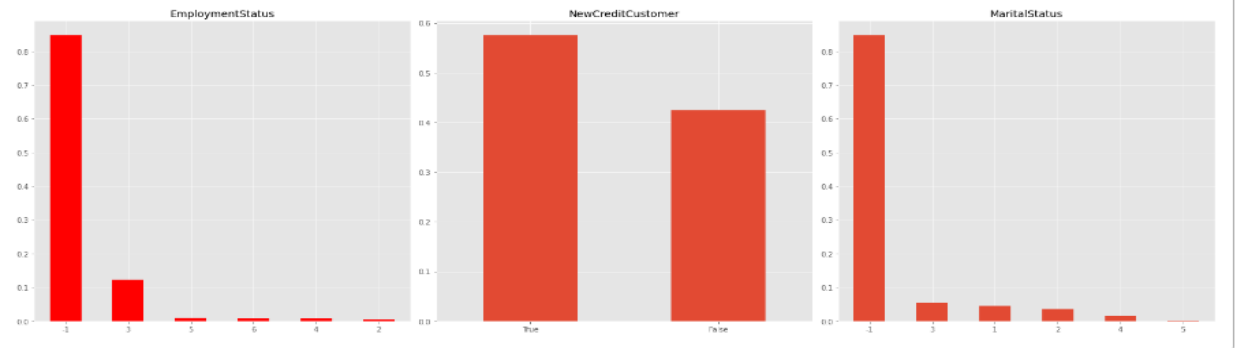


Fig 4.5: Univariate Analysis.

Figure 4.5 Results Interpretations:

85% of the loan applicants are believed to be fully employed, 11% are unemployed, 1% of the applicants are entrepreneurs, 1% are retired, 1% is self-employed, and the other 1% is partially employed.

58% of the credit customers are believed to be new customers, as indicated by the binary notation of "true" or "false." True indicated new customers, while False indicated old customers. 42% of the customers are believed to be old customers.

85% of the loan applicants have an unspecified marital status, while 5% are believed to be single. 4% are married applicants. 3% are cohabitants, 2% are divorced, and 1% are widows.

4.2.1.2.2 BIVARIATE ANALYSIS

The analysis of two variables at the same time is called bivariate analysis. In this section of the analysis, categorical features that were analyzed in the univariate analysis section will be reviewed again. In this case, they shall be compared to the variable of interest "default.".

4.2.1.2.2.1 OBSERVATIONS.

i. From figure 4.6 it is noted that those with unspecified employment status default the most as compared to the rest of the applicants.

```
#Bivariate Analysis
#Independent Variables vs. Dependent(Target Variable)
#EmploymentStatus vs. Default

sns.countplot(y = 'EmploymentStatus', hue = 'Default', data = data)
```

```
<AxesSubplot:xlabel='count', ylabel='EmploymentStatus'>
```

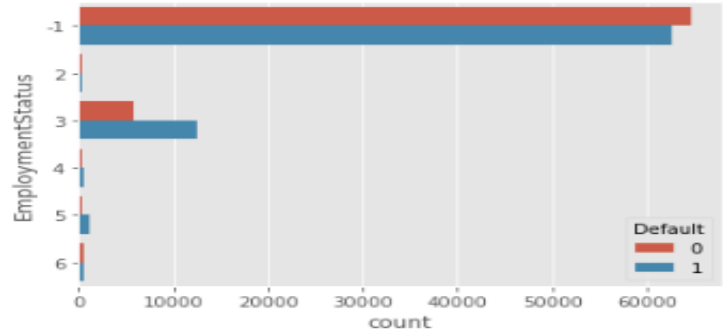


Fig 4.6: Employment Status vs. Default.

ii. Male applicants default more than females and the unspecified gender. Also unspecified gender has more loan defaulters as compared to males and females who have less. This is illustrated visually in figure 4.7.

```
#Gender vs. Default

sns.countplot(data = data, y = 'Gender', hue = 'Default')
```

```
<AxesSubplot:xlabel='count', ylabel='Gender'>
```

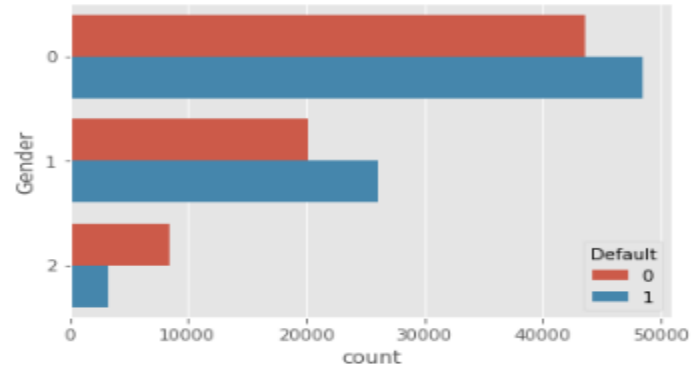


Fig 4.7: Gender vs. Default.

iii. According to figure 4.8, those with unknown marital status are believed to have more defaulters as compared to married, widowed, and divorced.

```
#MaritalStatus vs. Default
sns.countplot(data = data, y ='MaritalStatus', hue = 'Default')
```

```
<AxesSubplot:xlabel='count', ylabel='MaritalStatus'>
```

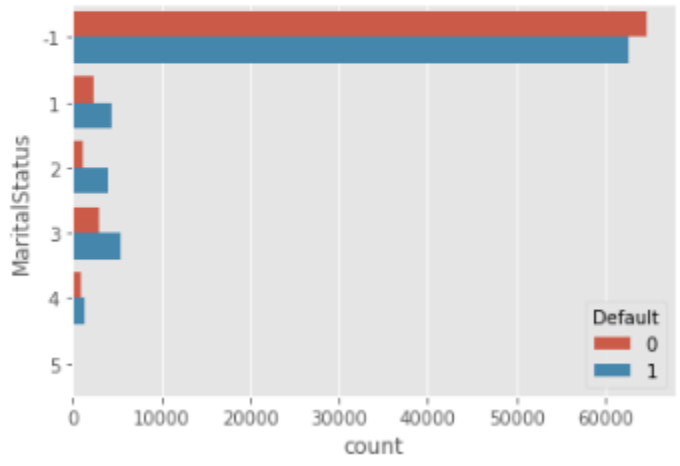


Fig 4.8: Marital Status vs. Default.

iv. With reference from figure 4.9, Secondary education certificate holders default the most as compared to other education statuses.

```
#Education vs. Default
sns.countplot(data= data, y = 'Education', hue = 'Default')
```

```
<AxesSubplot:xlabel='count', ylabel='Education'>
```

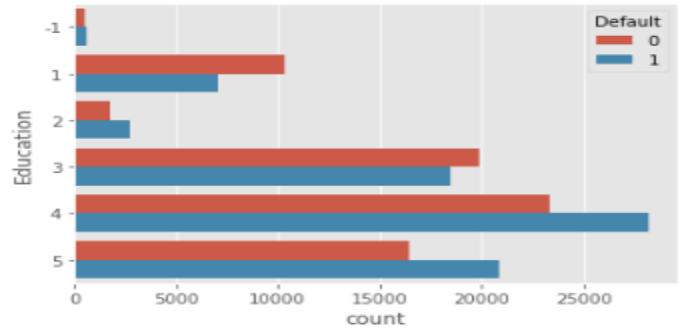


Fig 4.9: Education vs. Default.

v. According to figure 4.10, new credit customers default the most as compared to old customers.

```
#NewCreditCustomer vs. Default
sns.countplot(data = data, hue = 'Default', y = 'NewCreditCustomer')
```

```
<AxesSubplot:xlabel='count', ylabel='NewCreditCustomer'>
```

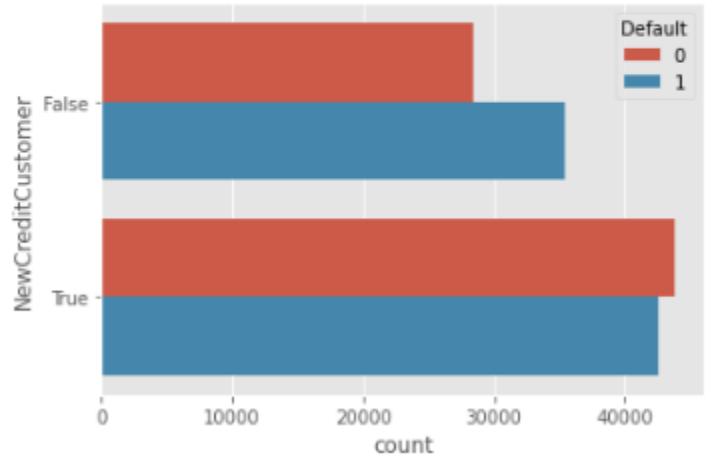


Fig 4.10: NewCreditCustomer vs. Default.

4.3 MODEL EVALUATION RESULTS

The goal of the study is to improve a loan lending organization's ability to predict loan defaults with greater accuracy. The model that has been created is utilized to classify defaulted and non-defaulting loans using supervised machine learning techniques. To build the model from historical loan data, the categorical variable of a borrower's loan default status is used. The algorithm was successful in spotting borrower behavior patterns and forecasting the chance of default for brand-new loan applications. The best approach for determining the chance of default had to be chosen, and the performance metrics used to evaluate the model's performance were extremely important. These metrics include the confusion matrix, accuracy, precision, and the F1-score in addition to the AUC-ROC.

4.3.1 MODEL EVALUATION USING THE CONFUSION MATRIX

The confusion matrices were computed separately for each model. Results obtained are explained in the following sections.

4.3.1.1 NAÏVE BAYES CLASSIFIER (NB-GAUSSIAN) ALGORITHM

The confusion matrix (Table 4.2), which shows the proportion of correctly classified and misclassified items in each category, provides a complete picture of the test results. The results are rounded to the nearest integer. The distribution showed that 56% of the data was correctly classified, while 44% was misclassified. Misclassifications can be categorized into two groups: type one error (false positive errors) and type two error (false negative errors). The distribution also showed that 19% (8432) of the applicants were misclassified to be defaults, whereas they did default. This leads to type one error, thus the company may incur losses when the very same customers are granted loans. The distribution further showed that 25% (11304) fell under type two error where they were misclassified to default if granted loan whereas they did not default in

their last loan grants. This does not necessarily lead to a loss, but will definitely not increase the institution's revenue.

Table 4.2 shows the confusion matrix of the NB Gaussian algorithm:

		ACTUAL	
PREDICTED	**N = 45,075**	**Default**	**Non-Default**
	Default	13,227 (***29% classified correct***)	8,432 (***Type I error***) (***19% misclassified***)
	Non-Default	11,304 (***Type II error***) (***25% misclassified***)	12,112 (***27% classified correct***)

Table 4.2: NB Confusion Matrix.

4.3.1.2 K NEAREST NEIGHBOR CLASSIFIER (KNN) ALGORITHM

According to Table 4.3, the confusion matrix stating the proportion of the correctly classified as well as those misclassified for each category gives a fulfilling picture of the test result. The results are rounded off to the nearest integer. The distribution showed that 57% of the data was correctly classified whilst 43% was misclassified. Misclassifications can be categorized into two groups; type one error (false positive errors) and type two error (false negative errors). The distribution also showed that 22% (10,158) of the applicants were misclassified to be non-defaults, whereas they did default. This leads to type two error, thus the company may incur losses when the very same customers are granted loans. The distribution further showed that 21% (9,325) fell under type two error where they were misclassified to default if granted loan

whereas they did not default in their last loan grants. This does not necessarily lead to a loss, but will definitely not increase the institution's revenue.

ACTUAL

	N = 45,075	**Default**	**Non-Default**
PREDICTED	**Default**	11,501 (***26% classified correct***)	10,158 ***Type I error*** (***22% misclassified***)
	Non-Default	9,325 ***Type II error*** (***21% misclassified***)	14,091 (***31% classified correct***)

Table 4.3: KNN confusion matrix.

4.3.1.3 SUPPORT VECTOR MACHINE (SVM) ALGORITHM

The confusion matrix in table 4.4 stating the proportion of the correctly classified as well as those misclassified for each category gives a fulfilling picture of the test result. The results are rounded off to the nearest integer. The distribution showed that 55% of the data was correctly classified whilst 45% was misclassified. Misclassifications can be categorized into two groups; type I error (false positive errors) and type II error (false negative errors). The distribution also showed that 31% (13,750) of the applicants were misclassified to be non-defaults, whereas they did default. This leads to type I error, thus the company may incur losses when the very same customers are granted loans. The distribution further showed that 14% (6,437) fell under type II error where they were misclassified to default if granted loan whereas they did not default in

their last loan grants. This does not necessarily lead to a loss, but will definitely not increase the institution's revenue.

		ACTUAL	
PREDICTED	**N = 45,075**	**Default**	**Non-Default**
	Default	7,909 (***18% classified correctly***)	13,750 **Type I error** (***31% misclassified***)
	Non-Default	6,437 **Type II error** (***14% misclassified***)	16,979 (***37% classified correctly***)

Table 4.4: SVM confusion matrix.

4.3.2 MODEL EVALUATION USING ACCURACY METRIC

Dividing the total number of correct predictions an algorithm can obtain by the total number of predictions all multiplied by 100, one can derive the accuracy rate. According to Table 4.5, the NB got 56% of its predictions correct while the KNN had 57% and the SVM had an accuracy rate of 55%.

NB	**KNN**	**SVM**
ACCURACY RATE	***ACCURACY RATE***	***ACCURACY RATE***
56%	**57%**	**55%**

Table 4.5: Accuracy rates.

4.3.3 MODEL EVALUATION USING RECALL METRIC

Recall is the ratio of default predictions (true positives) to all the potential default samples. The recall rate shows the probability of the model being able to predict the same outcome in future. Table 4.6 shows the Recall rates of the three models. The NB has a recall rate of 61%, followed by the KNN which has a rate of 57% which is much higher than that of the SVM which stands at 55%.

NB	**KNN**	**SVM**
RECALL RATE	***RECALL RATE***	***RECALL RATE***
61%	**57%**	**55%**

Table 4.6: Recall rates.

4.3.4 MODEL EVALUATION USING PRECISION METRIC

Precision is the ratio of actual default predictions to all samples predicted as default. Table 4.7 provides the precision rates for the three models. The NB has a precision rate of 54% which is lower than that of the KNN. The SVM managed to obtain 55% in its precision evaluation.

NB	**KNN**	**SVM**
PRECISION RATE	***PRECISION RATE***	***PRECISION RATE***
54%	**57%**	**55%**

Table 4.7: Precision rates.

4.3.5 MODEL EVALUATION USING F1_SCORE

F1_score is a harmonic mean between recall and precision. According to Table 4.8, the NB and KNN obtained the same harmonic mean of 57% making the SVM the model with the least F1_score of 54%.

NB	**KNN**	**SVM**
F1_SCORE	***F1_SCORE***	***F1_SCORE***
57%	**57%**	**54%**

Table 4.8: F1_SCORE rates.

4.3.6 MODEL EVALUATION USING THE AUC-ROC METRIC

The researcher noticed that the models that were implemented in loan default prediction had their strengths and weaknesses in terms of classification abilities, so a more clearer conclusion could not be drawn using the above implemented evaluation metrics such as accuracy, recall, F1_score, precision. One model proved to be dominant in accuracy, whilst in other metrics used it was deemed to perform poorly, for instance KNN had the highest Accuracy rate, but proved to under-perform in terms of recalling. Thus, the introduction of Area Under ROC curve.

Table 4.9 illustrates the frequently used AUC standards in terms of classifying the model as a best performer. According to the mentioned table, a model is said to have an excellent prediction if its AUC rate is greater than 85%. A model is termed to be a very good performer if its AUC rate is greater than 80%, whilst it is said to be a good model if its AUC rate is greater than 75%. Lastly, the model is said to be acceptable or fairly performing if its AUC rate is greater or equal to 60%. One may refer to table 4.8 for the AUC standards.

PREDICTIVE POWER	AREA UNDER ROC
EXCELLENT	**>85**
VERY GOOD	**>80**
GOOD	**>75**
ACCEPTABLE	**>=60**

Table 4.9: Standards of AUC.

4.3.6.1 RESULTS INTERPRETATION OF AUC-ROC

According to Table 4.10, the three implemented models seemed to produce different AUC rates which are justified. This was mainly because of their performances in classifying and being able to predict loan defaults. The table shows that the NB obtained a rate of 60% (After rounding of 59.8% to whole number) which signifies an acceptable or fairly performing model. The KNN showed to perform below the required threshold of 60%. SVM also obtained a lower rate of 52%.

NB: All models used in the research can still be used, but the objective of this study is to obtain the best performing one.

Naïve Bayes	K-Nearest Neighbor	Support Vector Machine
60%	58%	52%

Table 4.10: AUC-ROC curve results.

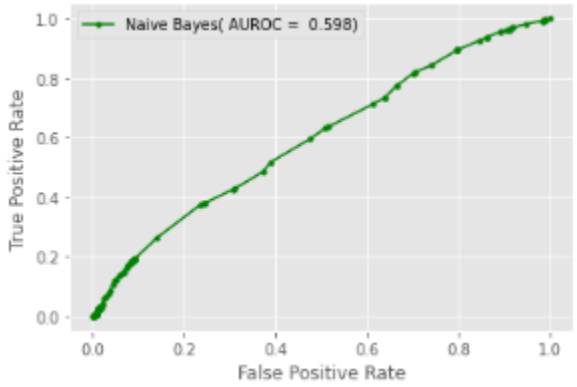
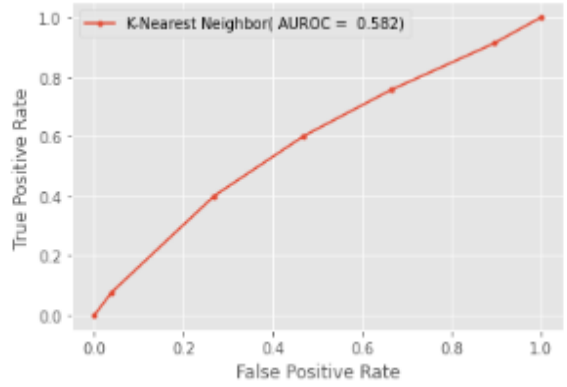
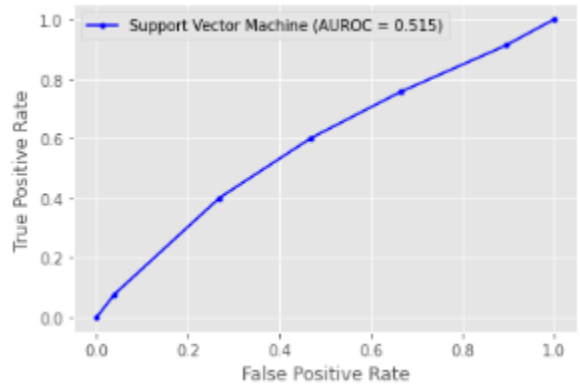
MODEL AUC/ROC curve	INTERPRETATION
 Fig 4.11: NB AUC/ROC curve.	The Naïve Bayes Algorithm has an AUC of 59.8% which if rounded off to the nearest whole number gives us 60%. This AUC falls under the acceptable AUC standards. Thus, we cannot disregard the model.
 Fig 4.12: KNN AUC/ROC curve.	Figure 4.12 shows the AUC/ROC curve of a K Nearest Neighbor. It shows a AUC of 58.2% which rounds off to 58%. This AUC rate falls under the acceptable AUC standard. Thus, we tend to disregard the model, though it can be used to classify.
 Fig 4.13: SVM AUC/ROC curve	According to figure 4.13, the Support Vector Machine also falls under the required AUC grading standard, so it is also disregarded. This model can still be used for classification purposes, though its rate fails to meet the required threshold.

Table 4.11: Overall AUC/ROC curves evaluation.

Table 4.11 further explains the AUC-ROC rates of each model. It provides curves for each model separately so as to make it easy to understand. All the explanations are provided in the table.

4.3.7 STACKED ENSEMBLE MODEL

Stacking ensemble algorithm is a machine learning technique that incorporates multiple best performing models by combining their predictions to come up with a highest accuracy rate. After all the models were evaluated, the researcher came to a conclusion that the models were under or fairly performing. Thus, the proposal of a stacked ensemble model which could take up the predictions of the two best performing algorithms (NB and KNN) and combining them with a logistic regression algorithm which becomes our default estimator. NB and KNN were combined together to produce a semi-stacked model which is then combined with a Logistic regression estimator. The logistic regression estimator is chosen because it satisfies the scaled data requirements.

Table 4.12 shows the accuracy, precision, recall, and f1_score of the Stacked model. The model has an accuracy rate which is above the three implemented models (NB, KNN, & SVM). Its precision is also higher than that of the KNN which is deemed to be the highest precision rate as compared to NB and SVM. Recall and F1_Score were also higher as compared to individually implemented algorithms.

ACCURACY	***PRECISION***	***RECALL***	***F1_SCORE***
70%	58%	70%	71%

Table 4.12: Stacked Ensemble Algorithm

Table 4.13 shows the Area Under Receiver Operation Curve of the Stacked model. The model has an AUC-ROC rate of 72%.

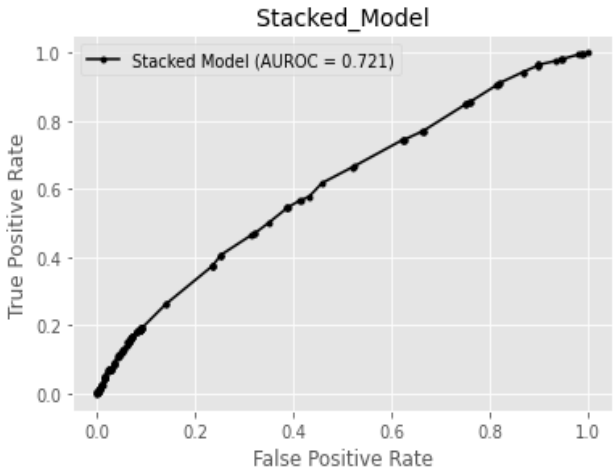
MODEL AUC/ROC curve	INTERPRETATION
 Fig 4.14: Stacked Model AUC-ROC curve.	According to Figure 4.14, the Stacked Model has an AUC-ROC curve with an accuracy rate of 72%. According to the AUC-ROC standards, this is a good model, which is by far better than the Naïve Bayes and the K Nearest Neighbor accuracy rates. Thus, it may be a good model to implement.

Table 4.13: Stacked Model AUC-ROC

4.4 DISCUSSION OF RESULTS

Following the confusion matrices (given in section 4.3.1) distribution in evaluating the performance of the KNN, it was determined that it correctly classified 56% of the data samples while misclassifying 43%. NB managed to correctly classify 56% of the data samples, while it failed to classify the remaining 44%. The SVM got 55% of its classifications correct and 45% of its classifications wrong. The correct classifications were split into two categories, which are true positives and true negatives. True positives state that the customer deemed a defaulter was indeed predicted to default by the algorithms. True negatives simply meant that a customer termed as a "non-defaulter" was indeed predicted not to default. Thus, NB managed to predict that 29% of the correctly classified samples were defaulters, while the remaining 27% were

predicted to be non-defaulters. KNN predicted that 26% of the correctly classified samples were going to default, while the remaining 31% were not going to default. The SVM classified 18% of its predictions as defaulters, while the remaining 37% of the correctly classified samples were predicted to be non-defaulters.

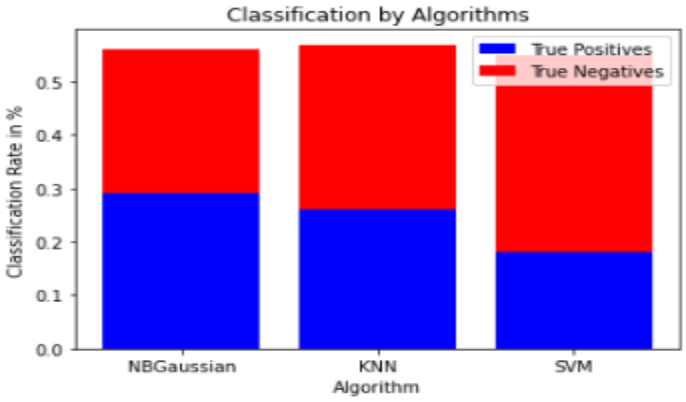


Fig 4.15: Classification by algorithms.

Figure 4.15 visually illustrates how the classification percentages are split so as to suit both true positives and true negatives.

By looking at the distribution of the confusion matrices provided in section 4.3.1, some samples of the data or data points were misclassified. For instance, some customers were termed non-defaulters based on the historical information received by the researcher, but the algorithms classified them as defaulters. This raises a lot of concerns as some customers might be rejected for loan applications due to incorrect predictions by the algorithms. Misclassifications are classified as either type one or type two errors. Type one errors refer to a customer as a "non-defaulter," whereas the actual facts label them as defaulters. Type two errors term a client as a defaulter, while the historical data terms them as a non-defaulter. Thus, the SVM misclassified a total of 45% of data points. It classified 31% of its misclassifications as false positives, meaning

that it classified some data points as "non-defaults" when they were defaults. The other 14 percent were misclassified as false negatives, meaning that non-defaulters were classified as defaulters. The first instance of misclassifying the 31% is called a type one error, while the second instance is called a type two error. NB has the second-highest rate of misclassification. It recorded a total of 44% misclassifications, with 19% being termed as false positives which are deemed type one error and 25% being false negatives which are deemed type two error. Misclassifications in KNN were 43%. False positives accounted for 22% of the total, while false negatives accounted for 21%. Figure 4.16 illustrates this graphically.

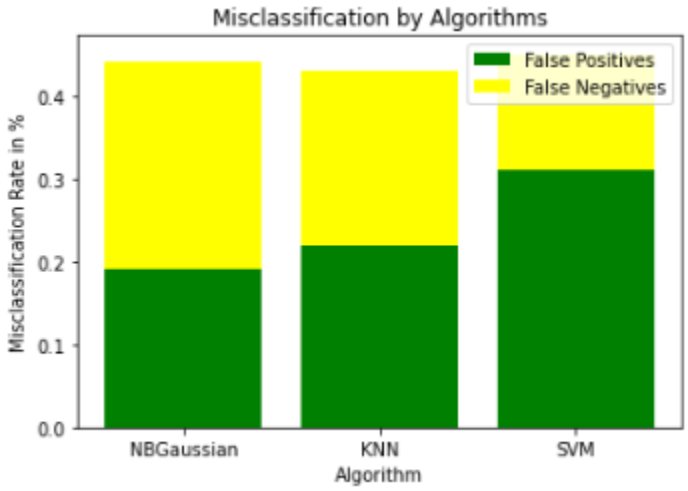


Fig 4.16: Misclassification by algorithms.

Each model was evaluated using the same evaluation metrics. The confusion matrices were computed and provided in section 4.3.1. The other metrics included accuracy (section 4.3.2), recall (section 4.3.3), precision (section 4.3.4), f1_score (section 4.3.5), and AUC-ROC (section 4.3.6).

In section 4.3.3, NB had the highest recall of 61%, whilst KNN managed to obtain 57% and the SVM with 55%. Recall plays an important role in minimizing risk as it produces the ratio of all

predicted true positives to all potential data samples that could have been classified as true positives. Recall is widely used to determine the best model though other metrics are not discouraged as well.

Section 4.3.4 showed that the KNN had 57% precision, with the SVM lagging behind with 55% and NB obtaining 54%.

NB and KNN had an F1 score of 57% while the SVM had 54% making it the least score (refer to section 4.3.5). The models' performances were then cross validated to examine the future prediction power. This is based on the results each model got on the applied evaluation metrics. The SVM managed to have validation rate of 68%, NB had 56% and KNN had 52% making it the lowest. Figure 4.17 provides visualizations that demonstrate the comparison performance of each model which is determined by the above mentioned and explained evaluation metrics.

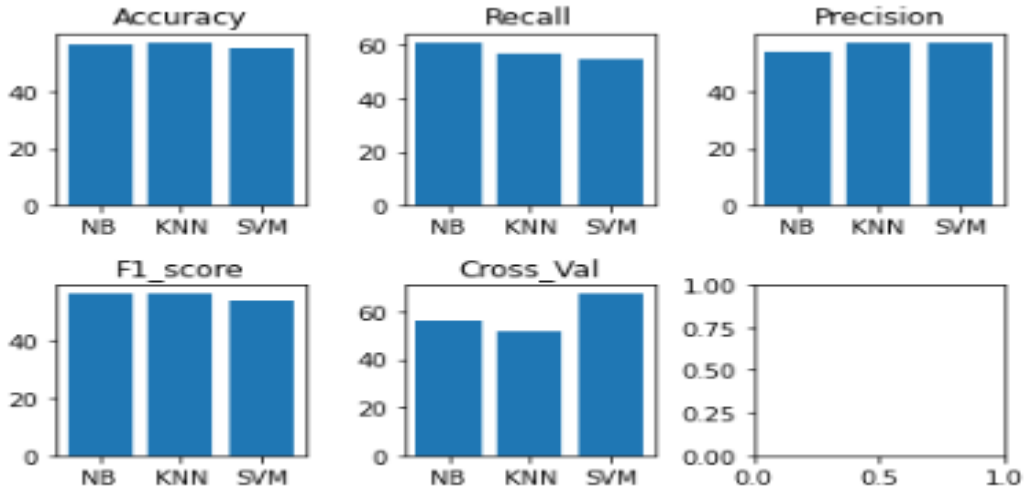


Fig 4.17: Performance Evaluation.

Section 4.3.6 provided the Area under ROC curve (AUC-ROC) for each model separately and these were accounted for in Table 4.13. The NB was termed the best model as it had a rate of 60%. Table 4.9 illustrated the AUC-ROC standards and this simply translates to choosing the NB

as the acceptable model. The KNN had an AUC-ROC rate of 58%. This was slightly below the minimum threshold of 60% thus, KNN cannot be deemed as an acceptable model under the AUC-ROC standards. Table 4.11 visually illustrates the AUC-ROC rate of the SVM. It is evident that the model cannot be deemed as an ideal model because of its rate which stood at 51% given that the threshold is 60%.

After the evaluation of the models using different evaluation metrics, it was noted that the ideal model had an AUC-ROC rate of 60%. This is an acceptable threshold though it is the least according to the AUC-ROC evaluation standards. Thus, the researcher implemented the a Stacked ensemble model which had a capability of combining the two best performing models' (NB & KNN) predictions. The model managed to yield an accuracy rate of 72%. The model proved to be efficient enough as compared to the individually implemented algorithms.

4.5 CONCLUSION

In this chapter results were presented, analyzed, and interpreted. The proposed models were also examined on how best they fit the loan data. Since the objective of this study was to develop a competitive model that can minimize the misclassification of bad customers as good, precision proved to be a better indicator of model performance in that regard. KNN had the highest precision of 57%, with SVM recording the second highest of 55% and NB recording the least precision of 52%. All this summarized to model evaluation, where the best model is chosen based on its performance. The researcher also noticed that determining the model that performs best in loan default prediction by using recall, the confusion matric, accuracy, precision, and f1_score proved to generate little information, thus he proposed AUC-ROC implementation. Overall, the Stacked ensemble model proved to perform better than the individually implemented

models, which makes it the versatile model in predicting possible loan defaults. The researcher also considered the models' ability to generalize. In other words, this means that the model was examined if it overfits. This is was to be sure of its future predicting power based on new data; hence a 10-fold cross validation method was implemented. The researcher noticed that the more data we have, the more accuracy we derive from the algorithms. The following chapter will consider the recommendations based on the research carried out by the researcher, it will also present the conclusion of this study.

CHAPTER FIVE: CONCLUSIONS & RECOMMENDATIONS

5.1 CONCLUSION

The development of a loan default prediction model was the main purpose of this study, which was focused on achieving three other goals. The following are the three goals:

i. To classify the data using NB, KNN, and the SVM models.

- Since the machine learning algorithms applied were supervised and classification types of algorithms, they were able to classify the observations in the dataset to default and non-default. This was because of model training and testing, which was later shown by the confusion matrix.

ii. To apply the NB, KNN, and the SVM algorithms on loan default predictions.

- In this, three supervised machine learning classification and predicting algorithms were examined on their possibilities to predict loan defaults. The Stacking model appeared to have a high predictive power of 72% as compared to individually implemented algorithms.

iii. To evaluate the performance of each model in predicting loan default.

- The main objective and main aim of this research was to derive an efficient machine learning algorithm that has the greatest predictive power that is able to predict possible loan defaults. Three algorithms were examined and evaluated, Stacked model proved to be an algorithm that can best predict loan default as it attained a success rate of 72%, which was derived from the Area Under the ROC curve. NB had a predictive power of 60%, the KNN managed to attain a success rate of 58% and the SVM had a success rate of 52%.

5.2 RECOMMENDATIONS

Based on the findings of the study, the following recommendations can be made by the researcher:

- Since it was evident from the previous studies carried out by other researchers that traditional models tend to be insufficient when applied to loan default prediction based on their several assumptions that were continuously violated, the researcher recommends the engagement of supervised machine learning algorithms in credit risk management as these tend to have fewer or no assumptions at all as compared to the traditional models.
- The research recommends the application of a stacked algorithm in predicting possible loan defaults. This is because of the research's findings that concluded that the stacked model is believed to have a higher success rate as compared to other algorithms. The algorithm proved too flexible in both large and small dataset thus, making the best choice.
- The research recommends the application of machine learning techniques in other departments of finance, such as in customer care services where an Artificial intelligence model is able to provide answers to customers' queries in a shortest possible time.

5.3 FUTURE WORK

Further research on assessing the connection between statistics and machine learning is suggested by the researcher. Additionally, while conducting the study, it would be helpful to conduct more research on adding more micro-economically focused factors to the dataset. Microeconomic factors have an impact on all customers, but different customers react differently to varying amounts of these variables, and a time-variant consideration alters the results when the analysis is conducted at different times of day.

REFERENCES

Inés Roldós. (2020). *Machine learning in finance*. Retrieved August, 25, 2022 from https://monkeylearn.com/blog/machine-learning-in-finance

Austin Clark**. (**2019). *Banking Risk Management, Credit Risk, and Treasury Risk Management*. Retrieved August, 25, 2022, from https://www.theglobaltreasure.com/author/austineclark

Ginimachine. (2021). *Measuring and Managing Credit Risk: Tools, Techniques & Best Practices*. Retrieved August, 25, 2022, from https://ginimachine.com/blog/credit-risk-management-best-practices

Brown, K., & Moles, P. (2014). *Credit Risk Management*. Retrieved November 11, 2022, from

http://www.ebsglobal.net/EBS/media/EBS/PDFs/Credit-Risk-Management.pdf

Pershad, R. (2000). *A Bayesian Belief Network for Corporate Risk Assessment*. Master's thesis.

University of Toronto. Retrieved November 8, 2022 from https://tspace.library.utoronto.ca

Agbemava, E, Nyarko I. K., Adade, T. C., & Bediako, A. K. (2016, January). Logistic Regression Analysis of Predictors of Loan Defaults by Customers of Non-Traditional Banks in Ghana. *European Scientific Journal* 12(1), 175-189.

Rokad, B. (2019, August 1). *Machine Learning Approaches and its Application*. Retrieved

November 8, 2022, from https://medium.com/datadriveninvestor/machine-learningapproaches-and-its-applications-7bfbe782f4a8

Kumar, M., Goel, V., Jain, T., Singhal, S., & Goel, L. M. (2018, April). Neural Network

Approach to Loan Default Prediction. *International Research Journal of Engineering and Technology (IRJET)*. 5(4), 4231-4234

Bacham, D., & Zhao, J. (2017, July). Machine Learning: Challenges, Lessons, and Opportunities in Credit Risk Modeling. Retrieved November 11, 2022, from *https://www.moodysanalytics.com/risk-perspectives-magazine/managingdisruption/ spotlight/machine-learning-challenges-lessons-and-opportunities-in-creditrisk-modeling*

Dormehl, L. (2019, January 5). *What is an artificial neural network? Here's everything you need to know*. Retrieved November 10, 2022, from https://www.digitaltrends.com/cooltech/ what-is-an-artificial-neural-network/

Obare, D. M. & Muraya, M. M. (2018). Comparison of Accuracy of Support Vector Machine Model and Logistic Regression Model in Predicting Individual Loan Defaults. *American Journal*

of Applied Mathematics and Statistics. 6(6), 266-271 Chen, S., Hardle, W. K., & Moro, R. A. (2011). Modeling Default Risk with Support Vector Machines. *Quantitative Finance*. 11(1), 135-154

Gandhi, R. (2018). *Support Vector Machines – Introduction to Machine Learning Algorithms*.

Retrieved November 10, 2022, from https://towardsdatascience.com/support-vector-machineintroduction- to-machine-learning-algorithms-934a444fca47

Patel, S. (2017, May 3). *Machine learning 101*. Retrieved November 11, 2022, from https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machinetheory-f0812effc72

Saxena, Rahul. (February 6, 2017). How the Naive Bayes Classifier works in Machine Learning. Retrieved November 12, 2022, from http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning

H. Zhang (2004). The optimality of Naive Bayes. Proc. FLAIRS. Retrieved from November 12, 2022, from http://scikit-learn.org/stable/modules/naive_bayes.html

Navlani, Avinash. (2018). Support Vector Machines with Scikit-learn. Datacamp Community. Retrieved from November 13, 2022 https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python.

Professor W.M.K Trochim. (2022). *Descriptive Statistics*. Retrieved August, 25, 2022, from https://conjointly.com/kb/descriptive-statistics