



UNIVERSITY OF NAIROBI

COLLEGE OF BIOLOGICAL AND PHYSICAL SCIENCES

SCHOOL OF COMPUTING AND INFORMATICS

MASTER OF SCIENCE IN COMPUTATIONAL INTELLIGENCE

LOAN DEFAULT PREDICTION USING MACHINE LEARNING :
A CASE OF MOBILE BASED LENDING

AUTHOR: GOLDA TRACEY KISUTSA

REG NO: P52/35489/2019

SUPERVISOR: DR. WANJIKU NG'ANGA

A project report submitted in partial fulfillment of the Master of Science degree in
Computational Intelligence.

2021

DECLARATION

I hereby declare that this research project is my own work, and has to the best of my knowledge, not been submitted to any other institution of higher learning.

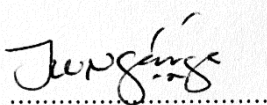
Name: Golda Kisutsa

Registration: P52/35489/2019

Signature:  Date: 17.08.2021

This research project has been submitted as partial fulfilment of the requirements for the award of the Degree of Master of Science in Computational Intelligence at the University of Nairobi with my approval as the faculty supervisor.

Supervisor: Dr. Wanjiku Ng'anga

Signature:  Date: 17.08.2021

ABSTRACT

Over the last decade, digital credit has been the fastest growing financial innovation in Kenya. This has largely been attributed to by technological innovations and mobile phone penetration enabling expanded access to financial services to individuals who were previously unbanked. Overall access to formal financial services now stands at 83%, up from 67% in 2016, and 88% of the adult population has access to a mobile money account (KFSD , 2019).

Precise credit risk assessment also known as loan default prediction is crucial to the functioning of lending institutions. Traditional credit score models are constructed with demographic characteristics, historical payment data, credit bureau data and application data. In online mobile based lending, borrower's fraudulent risk is higher. Hence, credit risk models based on machine learning algorithms provide a higher level of accuracy in predicting default.

The main objective of this project is to predict loan default by applying machine learning algorithms. The proposed methodology involves data collection , data pre-processing , data analysis , model selection and performance evaluation . This project takes data of previous customers to whom on a set of parameters loan were approved. The machine learning model is then trained on that record to get accurate results. The main machine learning algorithms applied are logistic regressions, naïve bayes and decision trees. The performance of the machine learning models are then compared using performance metrics and the best machine learning algorithm is selected to predict the loan default.

TABLE OF CONTENTS

Declaration	i
Abstract	ii
Table of Contents	iii
List of Figures	vi
List of Tables	vii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Significance of Study	2
1.4 Research Objectives	2
1.5 Justification	2
Chapter 2: Literature Review	3
2.1 Introduction	3
2.2 Traditional Credit Risk Assessment	3
2.2.1 Linear Regression	3
2.2.2 Discriminant Analysis	3
2.2.3 Probit Analysis and Logistic Regression	4
2.2.4 Judgment-Based Models	4
2.3 Machine Learning approaches in Credit Scoring	4
2.3.1 Decision Trees	5
2.3.2 Random Forest	5
2.3.3 Logistic Regression	5
2.3.4 Neural Network	6

2.3.5 Naïve Bayes	6
2.4 Related Research Work.....	7
2.4.1. Prediction For Loan Approval Using Machine Learning Algorithm.....	7
2.4.2 Loan Prediction Using Decision Tree and Random Forest	7
Chapter 3: Methodology	8
3.1 Introduction.....	8
3.2 Research Design.....	8
3.3 Data Collection	9
3.4 Conceptual Design	9
3.5 Proposed Model	10
3.5.1 Design Requirements.....	11
3.6 Data Preprocessing.....	12
3.6.1 Data Cleaning.....	13
3.6.2 Data Reduction.....	14
3.6.3 Feature Engineering	15
3.6.4 Exploratory Data Analysis.....	16
3.6.5 Converting Categorical Variables.....	20
3.6.6 Standard Scaler	20
3.6.7 Handling Outliers.....	21
3.6.8 Modelling.....	24
3.6.9 Model Testing	25
3.7 Performance Metrics.....	27
3.7.1 Confusion Matrix	27
3.7.2 Accuracy	27
3.7.3 Precision.....	28

3.7.4 Recall	28
3.7.5 Specificity	28
3.7.6 F1 Score	28
Chapter 4: Results and Discussions	29
4.1 Introduction.....	29
4.2 Results.....	29
4.2.1 Decision Tree	30
4.2.2 Logistic Regression.....	32
4.2.3 Naïve Bayes	34
4.3 Discussion.....	36
Chapter 5: Conclusions and recommendations.....	38
5.1 Introduction.....	38
5.2 Conclusions.....	38
5.3 Limitations of the Research	39
5.4 Recommendations and Future Work	39
References.....	40
Appendices.....	43
Appendix A: Gantt Chart.....	43

LIST OF FIGURES

Figure 2.1 Neural Networks.....	6
Figure 3.1 Research Design	8
Figure 3.2 CRISP-DM Methodology.....	9
Figure 3.3 Proposed Model.....	10
Figure 3.4 Importing Python Libraries	12
Figure 3.5 Train Data.....	12
Figure 3.6 Removing Missing Values	13
Figure 3.7 Preprocessed Data	14
Figure 3.8 Creating Target Variable	15
Figure 3.9 Univariate analysis	17
Figure 3.10 Gender vs Default.....	18
Figure 3.11 Education vs Default	18
Figure 3.12 New Credit Customer vs Default	19
Figure 3.13 Employment Status vs Default	19
Figure 3.14 Marital Status vs Default.....	19
Figure 3.15 Converting Categorical Variables	20
Figure 3.16 Scaling data	20
Figure 3.17 Normalized Income Total.....	22
Figure 3.18 Normalized Amount	23
Figure 3.19 Variable Declaration.....	24
Figure 3.20 Splitting Data.....	24
Figure 3.21 Test Data Preprocessing	25
Figure 3.22 Handling Outliers in Test Data.....	26
Figure 4.1 Decision Tree Results.....	31
Figure 4.2 Logistic Regression Results	33
Figure 4.3 Naïve Bayes Results	35

LIST OF TABLES

Table 3.1 Confusion Matrix	27
Table 4.1 Dataset	29
Table 4.2 Decision Tree Confusion matrix.....	30
Table 4.3 Decision Tree Performance	31
Table 4.4 Logistic Regression Confusion matrix	32
Table 4.5 Logistic Regression Performance	33
Table 4.7 Naive Bayes Confusion matrix	34
Table 4.8 Naïve Bayes Performance.....	35
Table 4.10 Confusion Matrix Comparison	36
Table 4.11 Performance Comparison	37

CHAPTER 1: INTRODUCTION

1.1 Background

Kenya has made tremendous progress in electricity connectivity , internet penetration and mobile network coverage. Mobile cellular subscriptions per 100 people has significantly grown from 0.4 in 2000 to 80.4 in 2016. (World Development Indicators , 2016). This rapid telecommunications and infrastructure development coupled with the global decline in cellphone prices has been harnessed by companies to provide value-added services such as digital credit.

Credit providers have traditionally required interaction between agents and clients, risk assessment based on previous financial history, and loans delivered into a bank account. This excluded those without a bank account or access to a bank branch and those with undocumented financial histories, This hurdle was readily overcome by digital credit, which refers to loans that are supplied and repaid digitally, generally using a cell phone. Digital credit is instant , loan decisions are automated based on a set of rules applied on available data and it is managed remotely. (CGAP, 2016).

Digital credit has evolved to incorporate a number of different business models. The first model is a bank and mobile network operator partnership such as M-Shwari by NCBA Bank and Safaricom. The second model is a non-bank lender and mobile network operator such as Kopa Cash by Jumo and Airtel Kenya. The third model is a bank utilizing mobile network operator channels such as MCo-op Cash by Co-operative bank that uses USSD. The fourth model is non-bank mobile internet applications which involves non-bank lenders disbursing loans through smartphone mobile application such as Branch and Tala.

Mobile lending platforms use predictive analytics like transaction history , call logs, text messages, contact lists , age , education, and income to arrive at a credit worthiness score and limit. When analyzing first-time borrowers, alternative digital data is especially significant, whereas repayment-based credit history becomes more important for subsequent loan applications. This research project aims to evaluate the application of machine learning technique to improve the predicted loan default rates.

1.2 Problem Statement

Financial mobile lending institutions use credit scoring models to evaluate potential loan default risks. These models generate a score that translates the likelihood of defaulting on a loan, making lending decisions easier. Developing credit scoring model is time consuming. These models are also fixed and do not easily evolve with changing customer behavior to predict default more accurately. Machine learning approaches can help enhance the accuracy of loan default prediction.

1.3 Significance of Study

Credit risk assessment is crucial to the success of lending institutions since customer credit risk affects profitability directly. Traditional procedures are inefficient and time-consuming. The goal of this study is to investigate the use of machine learning approaches in loan prediction that is more dynamic and adaptable to changing client data. These techniques will also provide higher accuracy in predicting loan default.

1.4 Research Objectives

- i. To review the existing literature on techniques applied in loan default prediction.
- ii. To design a machine learning model that can be forecast loan default.
- iii. To train and test the machine learning model to predict loan default.
- iv. To evaluate the performance of the model in predicting loan default.

1.5 Justification

The importance of credit risk evaluation has significantly increased with digital credit. Financial institutions have developed advanced systems to assess the credit worthiness of their customers. The objective of this research is to explore the use of machine learning algorithms to predict loan default and improve the accuracy of default prediction.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter reviews the existing literature on the traditional credit risk assessment methods, the machine learning algorithms that are increasingly being used to evaluate credit risk and the relevance of the literature to the problem statement.

2.2 Traditional Credit Risk Assessment

The type of data used in traditional credit scoring is historical data which includes bank transactional data such as past credit, records of late payment, credit bureau checks and commercial data such as financial statements and length of credit history (World Bank, 2019).

2.2.1 Linear Regression

Linear regression is used to determine the linear relationship between explanatory and target variables. The assumption in a general linear regression issue is that there is a dependent or response variable y_i that is impacted by independent variables. $x_{i1}, x_{i2}, \dots, x_{iq}$. A regression model can express this relation: $y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_q x_{iq} + \epsilon_i$ where $\beta_1, \beta_2, \dots, \beta_q$ are fixed regression parameters and ϵ_i is a random error or noise parameter. Before attempting to construct a linear model, it is critical to assess whether there is a link between the variables of interest.

2.2.2 Discriminant Analysis

Discriminant analysis is a credit scoring approach developed by Sir Ronald Fisher in 1936 to distinguish between two groups (Fisher, 1936). The most basic form is a two-category label, such as default versus nondefault. Linear discriminant analysis was the first statistical tool used to systematically explain which firms went bankrupt using accounting ratios and other financial indicators in default prediction.

2.2.3 Probit Analysis and Logistic Regression

The inverse standard normal distribution of the probability is modelled as a linear combination of characteristics in the probit (probability unit) model. The log of odds is used by the logit (logistic unit) function. The log of the label's odds ratio is described as a linear combination of attributes in the logit model. The following formula is used to predict log odds ratios:

$$\text{logit}(p) = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + \dots + \beta_n \times x_n \quad (5)$$

2.2.4 Judgment-Based Models

Multiple strategies are used to create judgment-based models. The analytic hierarchy process (AHP), which is a structured approach for organizing and analyzing complex decisions, is one such method. The decision makers break down their choice problem into a hierarchy of easier-to-understand subproblems, each of which may be studied individually. Human judgments, not just the underlying data, are used to complete the evaluations in the AHP, according to Bana e Costa, Barroso, and Soares (2002)

2.3 Machine Learning Approaches in Credit Scoring

Machine learning is based on the development of algorithms that can take in data and apply statistical analysis to anticipate an output, as well as update outcomes when new data becomes available. There are three different forms of machine learning. The purpose of supervised learning is to present a computer with labeled data and estimate the mapping function to the point that you can predict the output variables (y) for that data when you have new input data (x). Clustering and association problems are examples of unsupervised learning, in which a computer is supplied with unlabeled, uncategorized data and the system's algorithms act on the data without prior training. Reinforcement learning is a type of learning algorithm that learns by interacting with its surroundings. When the agent performs successfully, he is rewarded, and when he performs wrong, he is penalized. By maximizing its reward and reducing its penalty, the agent learns without the need for human intervention.

2.3.1 Decision Trees

A supervised learning approach used to tackle classification and regression problems is decision trees. To tackle the prediction problem, decision trees use tree representation, in which the external node and leaf node of a tree represent attribute and class labels, respectively.

The categorical variable decision tree contains categorical target variables that are separated into categories, such as yes or no. A continuous variable decision tree is one that has a continuous target variable, such as an individual's income, which can be forecasted using information such as their occupation, age, and other continuous factors.

2.3.2 Random Forest

Random forest is a machine learning algorithm that is supervised. The bagging approach is used to train a forest, which is a collection of decision trees. Random forest generates a large number of decision trees and then combines them to get a consistent and accurate classification. The Random Forest algorithm has the advantage of being able to be used for both classification and regression analysis.

2.3.3 Logistic Regression

A classification procedure called logistic regression is used to describe data and explain the relationship between one dependent binary variable and one or more independent variables.

Binary logistic regression has three major assumptions:

- i. The dependent variable must be either binary or dichotomous (e.g. yes or no).
- ii. The data should be free of outliers, which can be determined by converting continuous predictors to standardized scores.
- iii. The predictors should not have a lot of strong correlations (multicollinearity). The independent variables must be unrelated to one another. A correlation matrix among the predictors can be used to examine this.

2.3.4 Neural Network

Artificial Intelligence includes neural networks, which are a type of learning model affected by the activity of organic neurons. The neural network is made up of nodes that process the data provided to them and send the results to other nodes. Each node's output is known as the activation or node value. Weights are assigned to the nodes, which can be changed to assist the network learn. The magnitude of an input's influence on an output is represented by these weights. A linear, ramp, move sigmoid, hyperbolic, or Gaussian activation function is used to perform the net linear calculation. Because it can recognize non-linear regions, the Multilayer Perceptron Model is utilized to detect fraud.

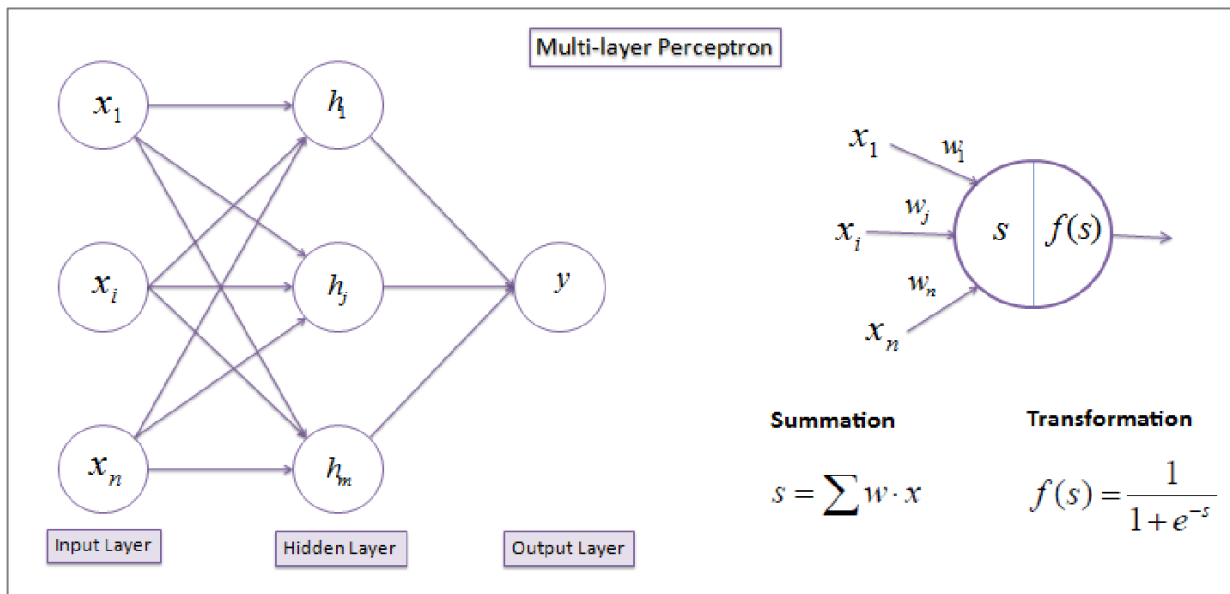


Figure 2.1 Neural Networks

2.3.5 Naïve Bayes

The Bayes Theorem is used to build a collection of classification algorithms known as Naive Bayes classifiers. They're employed in a variety of fields for things like prediction and anomaly detection. Each node represents a variable, and the arcs reflect the relationship between them, and they are represented by a graph with nodes and directed linkages between them. Although no information on the Nave Bayes is accessible in fraud detection, the set of variables that cause the frauds can be computed using the same theorem.

2.4 Related Research Work

2.4.1. Prediction for Loan Approval using Machine Learning Algorithm

This study examines banking systems and how banks might reduce non-performing loans or defaulting loans by forecasting loan defaulters (Kadam, Nikam, Aher, Shelke & Chandgude, 2021). The project's major goal is to use support vector machines and the Nave Bayes method to forecast loan safety. The subject of forecasting loan defaulters is studied using a critical predictive analytics technique. Data collection, data pre-processing, model selection, model evaluation, classification, and result are all part of the suggested model. This project gathers information from prior clients of various banks who were accepted for loans based on a set of criteria. To generate reliable results, the machine learning model is then trained on that record.. When comparing the Nave Bayes algorithm to the support vector machine, it was found that the Nave Bayes algorithm produced the most accurate predictions.

2.4.2 Loan Prediction using Decision Tree and Random Forest

Every year, the number of people or organizations seeking for a loan in India grows. (Madaan, Kumar, Keshri, Jain, & Nagrath, 2021). Banks must put in a lot of effort to determine whether a customer can pay back the loan amount on time or not (defaulter or non-defaulter). The purpose of this paper is to determine the nature, background, and credibility of the client who is seeking a loan. To cope with the challenge of granting or rejecting a loan request, or in short loan prediction, we apply exploratory data analysis. The purpose of this paper is to see if a loan made to a specific individual or organization will be approved. Feature analysis, data cleaning, exploratory data analysis modeling, and testing are all part of the research's recommended methodology. Random forest and decision trees were the machine learning methods used. The decision tree approach had a 73 percent accuracy whereas the random forest classifier had an accuracy of 80 percent.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter reviews the research design and its justification , the sources of data obtained and its relevance to the problem statement , the methods for data collection , the data analysis methods applied and their justifications.

3.2 Research Design

This research project considers the problem of selecting the relevant predictor variables in a classification. Based on the factors collected during the loan application process, the classification goal is to forecast if a specific borrower is likely to fail on their loan. The design involves data collection , data preprocessing , data analysis , model building by applying decision trees, logistic regression and naive bayes algorithms. Evaluating the performance of the models.

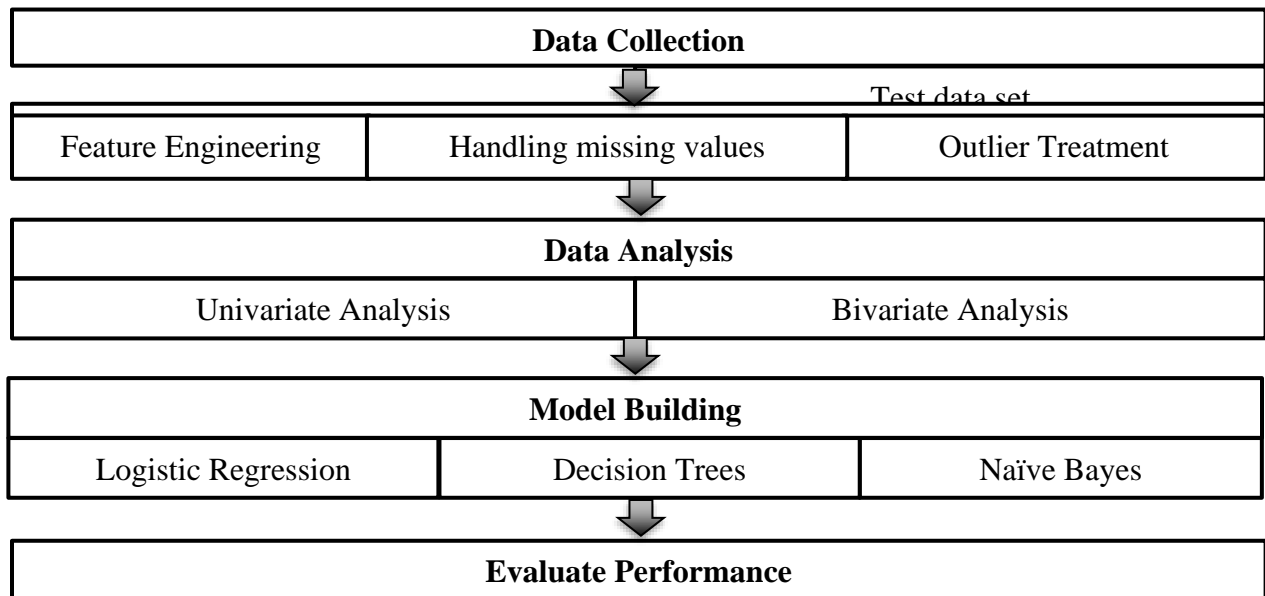


Figure 3.1 Research Design

3.3 Data Collection

The data collection method is secondary data from Bondora which is a financial lending institution based in Estonia. The data was obtained from the company website on their public reports page. The dataset included loan data from the year 2009 to 2021 which is 176,680 rows and 62 columns. The amount of training data required for a machine learning algorithm varies according to the complexity of the model, the pattern in the data and the correlation between attributes. The rule of 10 states the amount of training data needed for a well performing model should be 10x the number of parameters in the model. Hence, the dataset for the year 2020 is extracted for the purpose of this study which contains 17,933 rows and 62 columns. The data preprocessing includes extracting the relevant features, handling missing values, and handling outliers. Finally, the data is split into a train and test set with 70% training data and 30% test data.

3.4 Conceptual Design

The conceptual design used to develop the model is the CRISP-DM methodology that incorporates six design phases.



Figure 3.2 CRISP-DM Methodology

- i. **Business understanding.** The first step is to comprehend the research's background, the problem description, and how the proposed project will achieve the goals.
- ii. **Data understanding.** The second stage requires collection of data listed in the project resources. This involves describing the data requirements and exploring key data attributes.
- iii. **Data preparation.** The third stage involves cleaning the data to handle any missing values.
- iv. **Modelling.** This involves determining the modelling technique and testing the design.
- v. **Evaluation.** This is an evaluation of the results achieved to determine the performance of the model with the best accuracy.
- vi. **Deployment.** The last stage is the implementation of the model.

3.5 Proposed Model

The model will be able to predict whether a loan applicant will default on a given loan. The system architecture is as below.

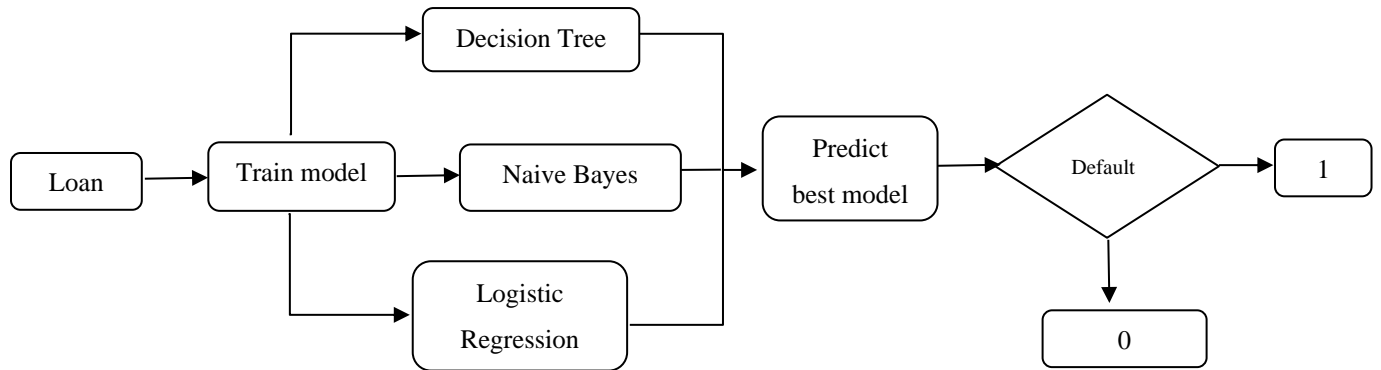


Figure 3.3 Proposed Model

Step 1 : The Loan application goes through the trained model where the three classification algorithms are applied.

Step 2: The machine learning with the best performance in accuracy is selected.

Step 3 : The machine learning algorithm is applied to the loan application.

Step 4: The machine learning algorithm determines the probability of default. 1, being true and 0 being false.

3.5.1 Design Requirements

Hardware and Software Requirements

The hardware requirements for the project include a laptop with at least 4GB ram running windows or Linux operating system. The software requirements include a code editor. This project uses Microsoft Visual Studio which is a code editor redefined and optimized for building and debugging modern web and cloud applications.

Python Libraries

The machine learning models are implemented using python version 3.7 on a Jupyter notebook with the listed libraries: numpy, pandas , matplotlib, seaborn , and sklearn.

- i. **Jupyter notebooks** are a web-based interface in which you can write, visualize, and execute python code in cells. It is good for exploratory analysis and enable to run individual code cells.
- ii. **Numpy** is a Python library that may be used to work with multi-dimensional arrays, linear algebra, the Fourier transform, and matrices.
- iii. **Pandas** is a data manipulation and analysis package written in Python.
- iv. **Matplotlib** is a Python package that allows you to create static, animated, and interactive visualizations.
- v. **Seaborn** is a matplotlib-based python data visualization package. It has a high-level interface for creating visually appealing and instructive statistics visuals.
- vi. **Sklearn** is a Python toolkit that allows you to create machine learning and statistical models including clustering, classification, and regression.

3.6 Data Preprocessing

Data preprocessing entails converting raw data into a comprehensible format that a machine learning model can understand. The data is loaded on a Jupyter notebook in Microsoft Visual Studio and the python libraries numpy, pandas, matplotlib, seaborn and sklearn are imported.

The dataset contains 17,933 rows and 62 columns before preprocessing. The data preprocessing involves data cleaning which involves handling missing values, data transformation which involves normalizing the data and data reduction which involves using only relevant features and discarding duplicate values of less relevant attributes.

```
#Import Libraries
import numpy as np #linear algebra
import pandas as pd #data analysis
import matplotlib.pyplot as plt # data visualizations
%matplotlib inline
import seaborn as sns #data visualization
✓ 0.4s
```

Figure 3.4 Importing Python Libraries

```
DATA PREPROCESSING

dataset = pd.read_csv("train.csv")
pd.set_option('display.max_columns', 500)
dataset.shape
✓ 0.2s
(17933, 62)
```

Figure 3.5 Train Data

3.6.1 Data Cleaning

The first step of preprocessing is data cleaning by checking and eliminating any missing values because they affect the accuracy of the model. This is achieved by either filling the missing values with a mean or mode function or by dropping all missing values. In this case the missing values are dropped.

```
# Percentage of missing values
pd.set_option('display.max_rows', None)
round(dataset.isnull().sum()/len(dataset.index), 2)*100
✓ 0.9s
```

ReportAsOfEOD	0.0
LoanId	0.0
LoanNumber	0.0
UserName	0.0
NewCreditCustomer	0.0
LoanDate	0.0
ContractEndDate	34.0
FirstPaymentDate	0.0
MaturityDate_Original	0.0
MaturityDate_Last	0.0
Age	0.0

```
#Removing columns having missing values
missing_columns = dataset.columns[100*(dataset.isnull().sum()/len(dataset.index)) > 0]
print(missing_columns)
✓ 0.8s
```

```
Index(['ContractEndDate', 'DateOfBirth', 'County', 'City', 'NrOfDependants',
      'WorkExperience', 'LastPaymentOn', 'CurrentDebtDaysPrimary',
      'DebtOccuredOn', 'CurrentDebtDaysSecondary',
      'DebtOccuredOnForSecondary', 'DefaultDate',
      'PrincipalOverdueBySchedule', 'PlannedPrincipalPostDefault',
      'PlannedInterestPostDefault'],
      dtype='object')
```

```
miss_col=['ContractEndDate', 'DateOfBirth', 'County', 'City', 'NrOfDependants',
          'WorkExperience', 'LastPaymentOn', 'CurrentDebtDaysPrimary',
          'DebtOccuredOn', 'CurrentDebtDaysSecondary', 'DebtOccuredOnForSecondary',
          'PrincipalOverdueBySchedule', 'PlannedPrincipalPostDefault',
          'PlannedInterestPostDefault']
dataset.drop(miss_col, axis=1, inplace=True)
dataset.shape
✓ 0.7s
```

```
(17933, 48)
```

Figure 3.6 Removing Missing Values

3.6.2 Data Reduction

The next step of the data preprocessing is data reduction. This is used to remove duplicate features e.g., 'LoanId' when there's 'LoanNumber', 'DateofBirth' when the feature 'Age' is present. Features relating to dates excluding 'DefaultDate' are deleted. The multiple values of income are also deleted since they are already aggregated in 'IncomeTotal'. The data is reduced to 17,933 rows and 20 columns.

```
#Removing features that have no role in default prediction. These include duplicate
# values such as LoanID vs LoanNumber. Income values when there is Income Total.
cols_del=['ReportAsOfEOD','LoanId', 'UserName','LoanDate','FirstPaymentDate',
          'MaturityDate_Original','MaturityDate_Last','Country','AppliedAmount',
          'IncomeFromPrincipalEmployer','IncomeFromPension','IncomeFromFamilyAllowance',
          'IncomeFromSocialWelfare','IncomeFromLeavePay','IncomeFromChildSupport','IncomeOther',
          'MonthlyPaymentDay','ActiveScheduleFirstPaymentReached','PlannedInterestTillDate',
          'PrincipalPaymentsMade', 'InterestAndPenaltyPaymentsMade','PrincipalBalance',
          'InterestAndPenaltyBalance', 'PlannedPrincipalTillDate','PrincipalWriteOffs',
          'InterestAndPenaltyWriteOffs', 'PreviousEarlyRepaymentsBefoleLoan']
dataset.drop(cols_del, axis=1 , inplace=True)
dataset.shape
✓ 0.7s
(16046, 20)
```

```
dataset.info()
✓ 0.1s
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16046 entries, 0 to 17932
Data columns (total 20 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   LoanNumber                               16046 non-null  int64
1   NewCreditCustomer                        16046 non-null  bool
2   Age                                       16046 non-null  int64
3   Gender                                   16046 non-null  int64
4   Amount                                   16046 non-null  int64
5   Interest                                 16046 non-null  float64
6   LoanDuration                             16046 non-null  int64
7   MonthlyPayment                           16046 non-null  float64
8   Education                                 16046 non-null  int64
9   MaritalStatus                            16046 non-null  int64
10  EmploymentStatus                         16046 non-null  int64
11  EmploymentDurationCurrentEmployer        16046 non-null  object
12  IncomeTotal                              16046 non-null  float64
13  DebtToIncome                             16046 non-null  float64
14  Restructured                             16046 non-null  bool
15  NoOfPreviousLoansBeforeLoan              16046 non-null  int64
16  AmountOfPreviousLoansBeforeLoan          16046 non-null  float64
17  PreviousRepaymentsBeforeLoan             16046 non-null  float64
18  PreviousEarlyRepaymentsCountBeforeLoan   16046 non-null  int64
19  Default                                  16046 non-null  int64
dtypes: bool(2), float64(6), int64(11), object(1)
memory usage: 2.4+ MB
```

Figure 3.7 Preprocessed Data

3.6.3 Feature Engineering

When utilizing machine learning to create a predictive model, feature engineering is the act of choosing and modifying variables in a dataset. The 'Status' and 'DefaultDate' variables will be used to create the target variable, 'Default'. The 'Status' variable cannot be used since it has three unique values current, late and repaid. Late can also not be treated as default since in some records the loan status is late however the default date is null which implies the loan was not defaulted but was only late. The 'DefaultDate' informs when a borrower defaulted. Combining both the 'Status' feature and 'DefaultDate' feature will enable to create the target variable 'Default'. This is achieved by filtering the loan status to current and checking the default dates to create a new target variable called 'Default' that will have the values 0 if default and 1 if loan is not default. The 'Status' and 'DefaultDate' features are removed once the target variable is created.

```
#Create Categorical Variable
dataset['Status'].value_counts()
✓ 0.4s
Late      8961
Repaid    7085
Current   1887
Name: Status, dtype: int64

# filtering Current Status records
dataset= dataset[dataset['Status'] != 'Current']
✓ 0.4s

#Creating a new target variable in which 1 will be assigned when default date is null
#means borrower has never defaulted while 0 in case default date is present.
dataset["Default"] = dataset['Status'].apply(lambda x: 1 if x=='Repaid' else 0)
dataset['Default'].value_counts()
✓ 0.5s
0      8961
1      7085
Name: Default, dtype: int64
```

Figure 3.8 Creating Target Variable

3.6.4 Exploratory Data Analysis

There are two types of independent variables in the data set. Categorical features which include 'Gender', 'Education', 'MaritalStatus' and numerical features which include 'IncomeTotal' and 'Amount'.

Univariate Analysis

This term refers to data that consists solely of observations on a single attribute. The basic goal of univariate analysis is to characterize the data and discover patterns within it. The data is visually shown using graphing. The primary goal of graphs is to convey data, summarize data, enhance verbal descriptions, describe and explore data, facilitate comparisons, avoid distortion, and stimulate thought about the data. The bar graph is the specific graph that has been used. On the y (vertical) and x (horizontal) axes, the graph is labeled (horizontal axis). The categorical and ordinal features explored include 'Gender', 'Education', 'Marital Status', 'EmploymentStatus', 'EmploymentDurationCurrentEmployer', 'NewCreditCustomer'.

Observations

- 55% (8,961) of the loans are default.
- 70% of the loan applicants are male.
- Nearly 40% have a secondary education while approximately 30% have a higher education.
- Nearly 25% are fully employed.
- 50% are new credit customers.

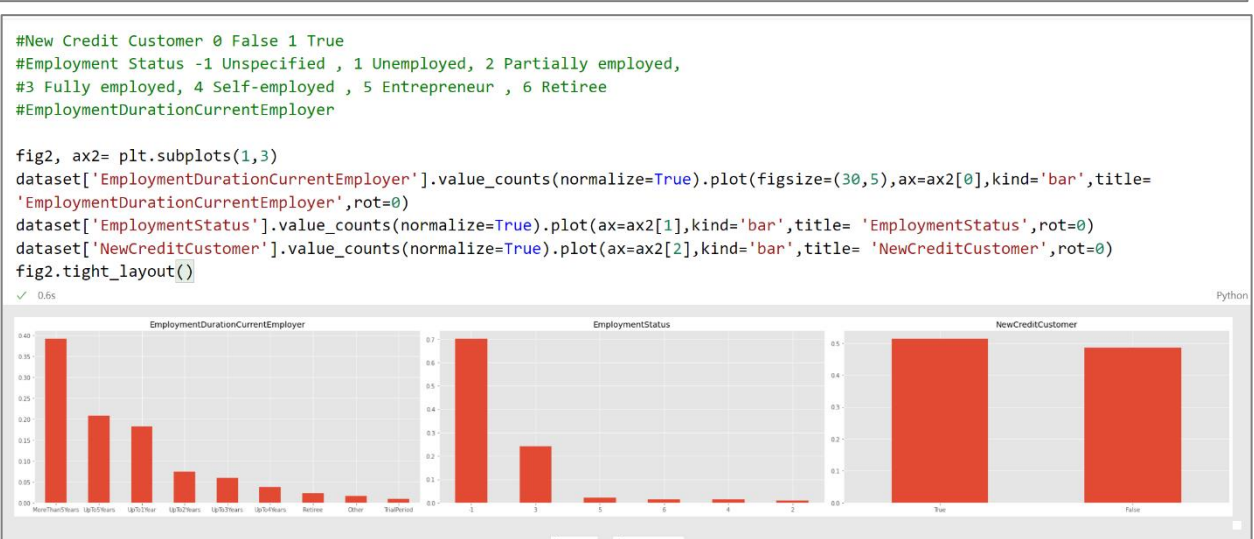
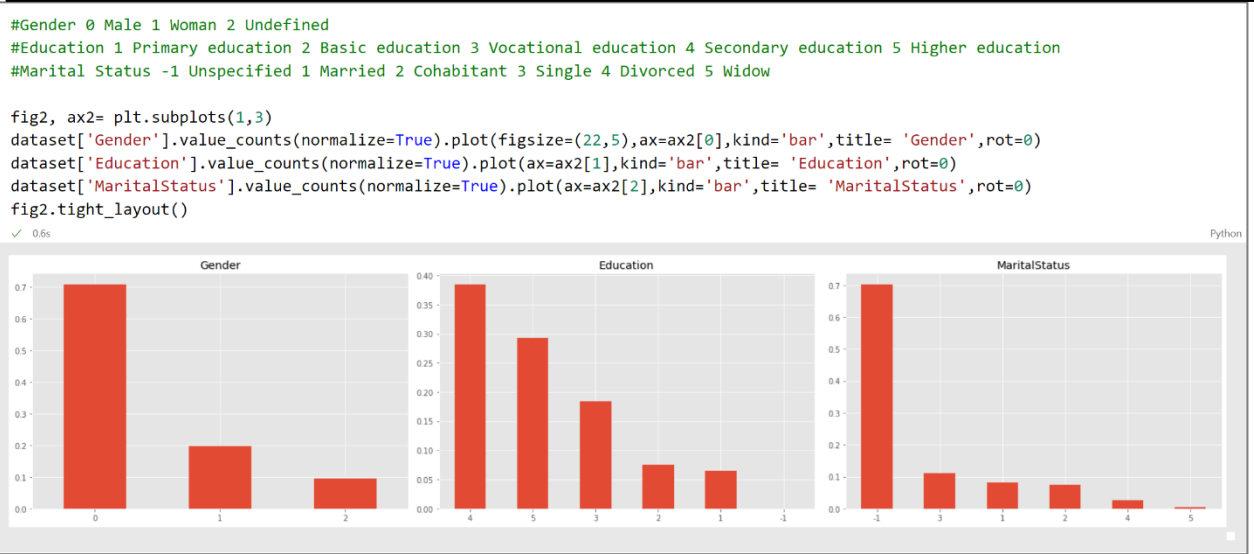
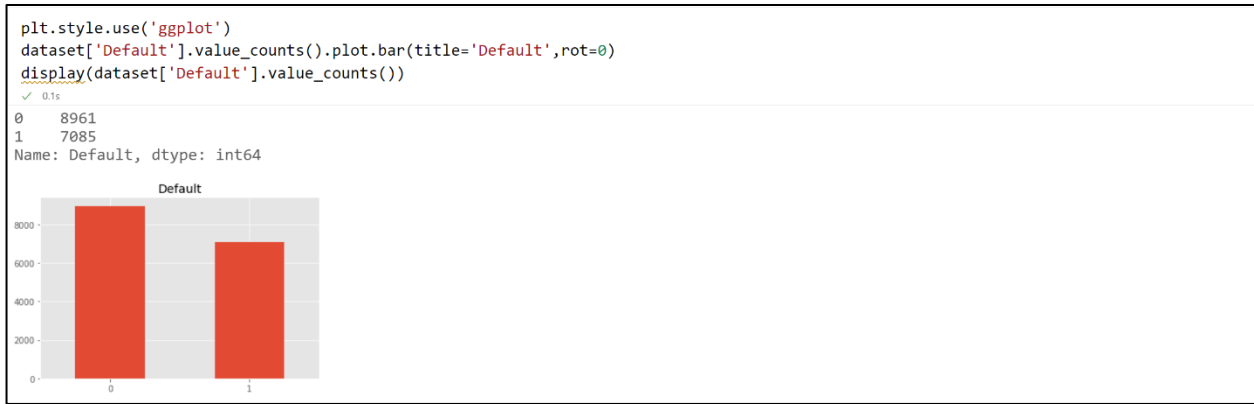


Figure 3.9 Univariate analysis

Bivariate Analysis

The analysis of two variables with the goal of identifying the empirical link is known as bivariate analysis. The following categorical variables: 'Gender', 'Education', 'EmploymentStatus', 'MaritalStatus', 'New credit customer' will be compared to the dependent variable 'Default'.

Observations

- i. Male loan applicants default more than female.
- ii. Those with Secondary education default most than the other education status.
- iii. New Credit Customers default more than existing credit customers.
- iv. Those who have more than 5 years employment duration default more.



Figure 3.10 Gender vs Default

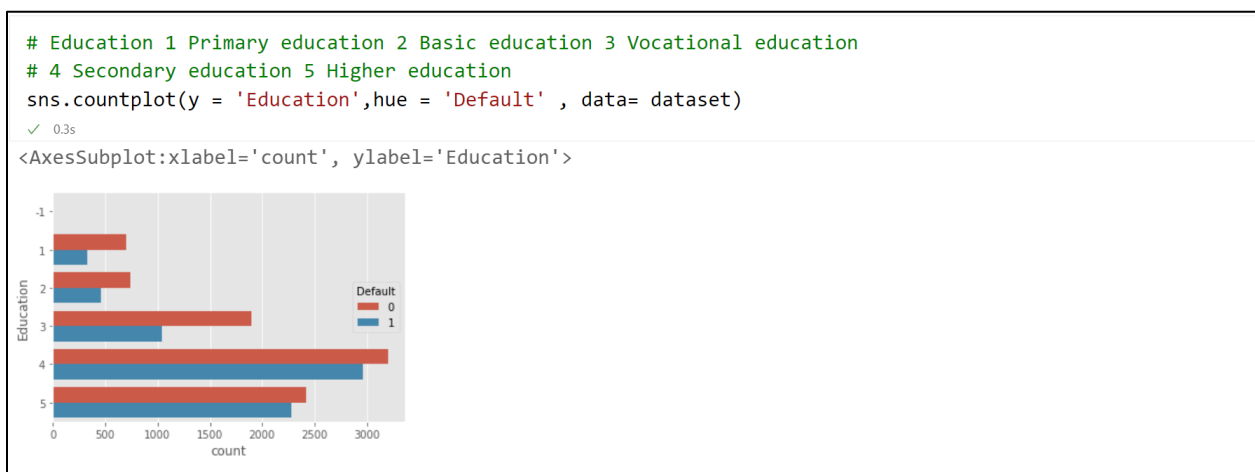


Figure 3.11 Education vs Default

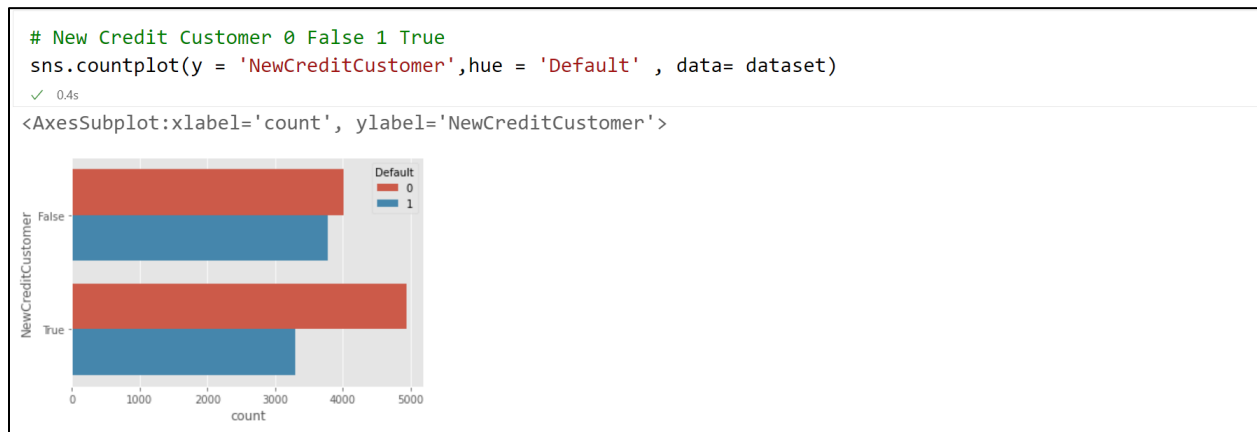


Figure 3.12 New Credit Customer vs Default



Figure 3.13 Employment Status vs Default

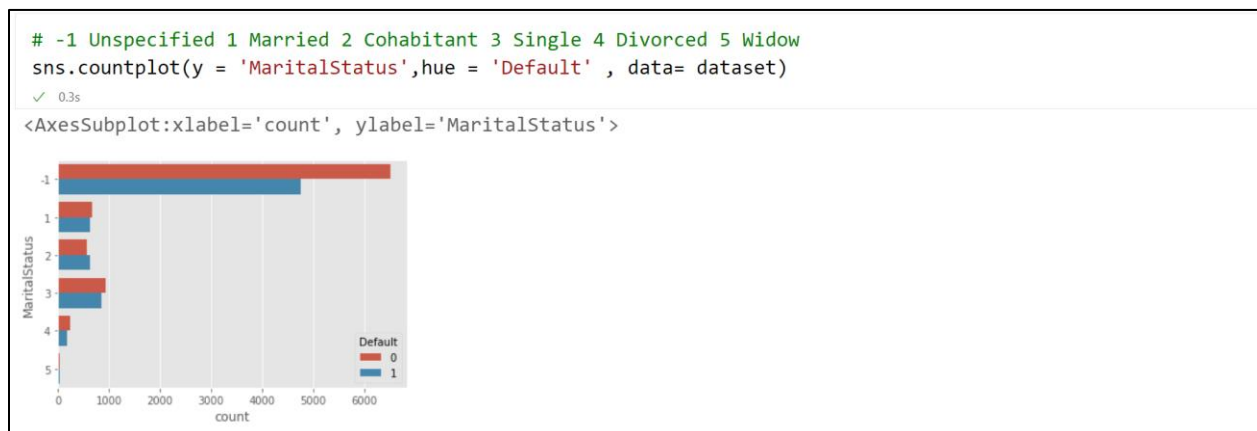


Figure 3.14 Marital Status vs Default

3.6.5 Converting Categorical Variables

Sklearn requires all inputs to be numeric. The categorical variables are converted to numerical variables using label encoder. The values 'NewCreditCustomer' , 'Restructured' , 'EmploymentDurationCurrentEmployer' will be converted to numerical values.

```
#Convert non numeric variables to numeric
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
cat=['NewCreditCustomer','Restructured','EmploymentDurationCurrentEmployer']
for var in cat:
    le = preprocessing.LabelEncoder()
    dataset[var]=le.fit_transform(dataset[var].astype('str'))
```

dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16046 entries, 0 to 17932
Data columns (total 22 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   LoanNumber                            16046 non-null  int64
1   NewCreditCustomer                    16046 non-null  int32
2   Age                                   16046 non-null  int64
3   Gender                                16046 non-null  int64
4   Amount                                16046 non-null  int64
5   Interest                              16046 non-null  float64
6   LoanDuration                          16046 non-null  int64
7   MonthlyPayment                        16046 non-null  float64
8   Education                              16046 non-null  int64
9   MaritalStatus                         16046 non-null  int64
10  EmolvmentStatus                       16046 non-null  int64
```

Figure 3.15 Converting Categorical Variables

3.6.6 Standard Scaler

To turn data into a distribution with a mean of 0 and a standard deviation of 1, use a standard scaler.

```
#Scale dataset
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
X_train = ss.fit_transform(X_train)
X_test = ss.transform(X_test)
```

Figure 3.16 Scaling data

3.6.7 Handling Outliers

Outliers are data points that are far apart from other similar points, which could be due to measurement variability or experimental errors. The range and distribution of attribute values are particularly important to machine learning algorithms. Outliers in the data might cause the training process to be misled, resulting in longer training times, fewer accurate models, and inferior outcomes. To analyze the data and detect any outliers, data visualization is employed.

There are four methods for dealing with outliers in a dataset. Remove the outlier records entirely to get rid of them. By setting a value range, you can limit the data of outliers. If the data is out of scope for the intended variable, assign a new value. Using techniques such as log transformation, data can be transformed.

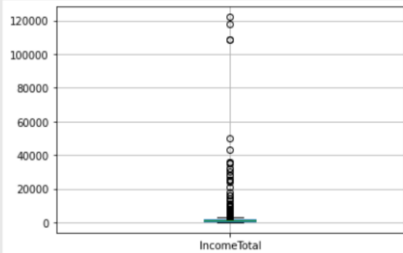
The 'IncomeTotal' and 'Amount' variables include some outliers and are skewed, as can be seen in the dataset. The log transformation is used to normalize the data. The log transformation is used to skewed data in order to approximate normality. Because the dataset has a log-normal distribution, the log-transformed data will have a normal or near-normal distribution as well, reducing skewness.

Normalizing Income Total Variable.

```
# Box Plot for understanding the distributions and to observe the outliers.  
dataset.boxplot(column='IncomeTotal')
```

✓ 0.3s

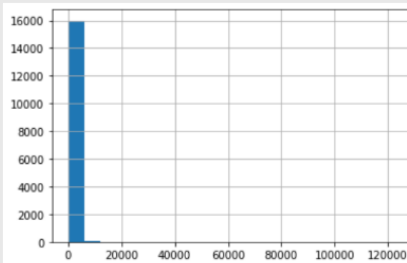
<AxesSubplot:>



```
dataset['IncomeTotal'].hist(bins=20)
```

✓ 0.2s

<AxesSubplot:>



```
#Handling Outliers of IncomeTotal using log transformation  
dataset['IncomeTotal_log']=np.log(dataset['IncomeTotal'])  
dataset['IncomeTotal_log'].hist(bins=20)
```

✓ 0.2s

<AxesSubplot:>

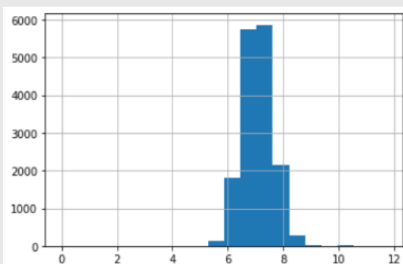


Figure 3.17 Normalized Income Total

i. Normalizing Amount Variable

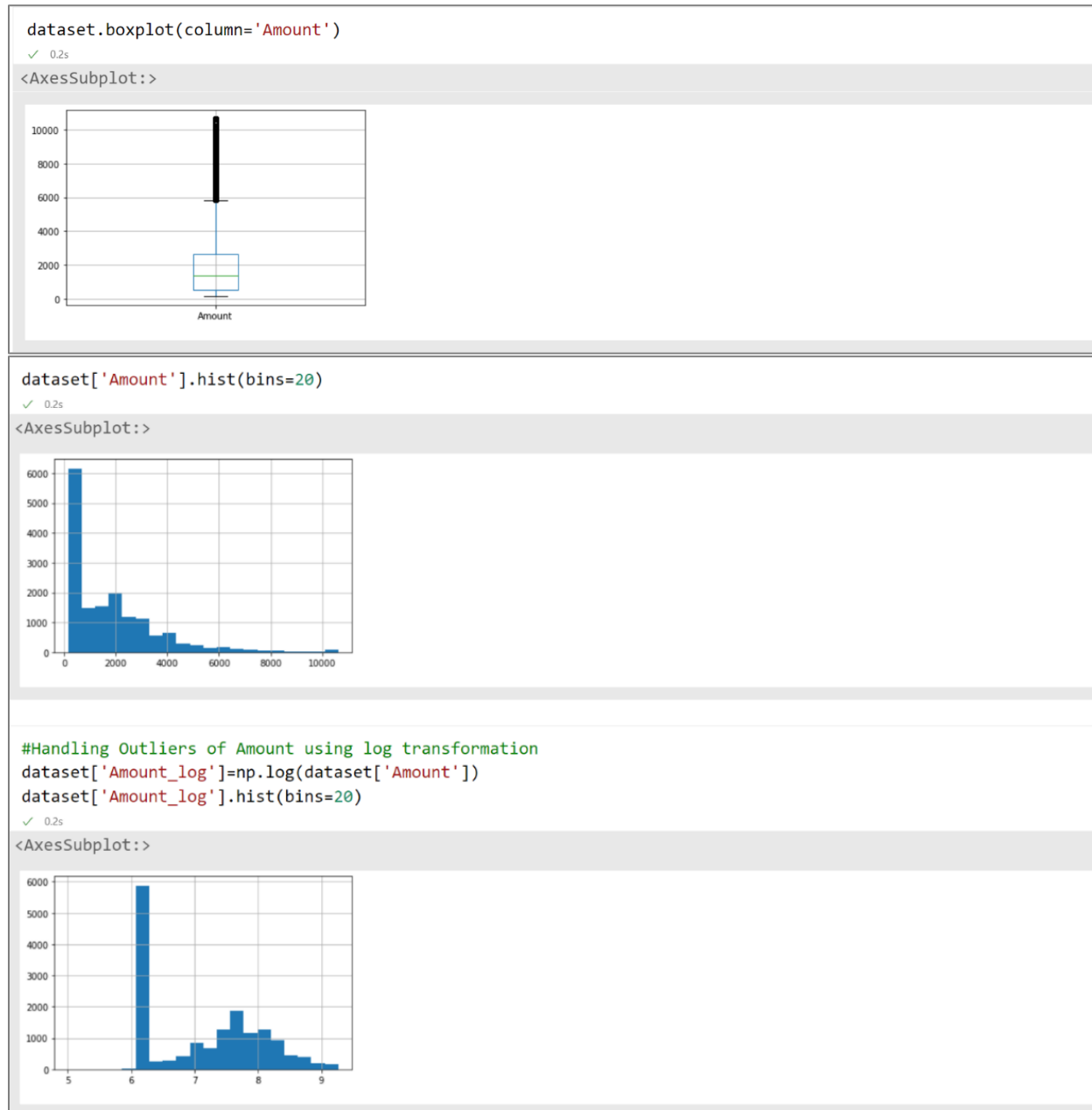


Figure 3.18 Normalized Amount

3.6.8 Modelling

i. Declaration of Variables

The independent variables are declared in x. These include the features ‘NewCreditCustomer’, ‘Gender’, ‘Education’, ‘EmploymentStatus’ and ‘Restructured’. These features are selected since they are categorical variables. The dependent value is declared in y which is ‘Default’.

```
X= dataset.iloc[:,np.r_[1,3,8,10,14]].values
y= dataset.iloc[:,19].values
✓ 0.6s

X
✓ 0.1s
array([[ 1,  2,  3,  3,  0],
       [ 1,  2,  2,  3,  0],
       [ 1,  2,  3,  3,  0],
       ...,
       [ 0,  0,  3, -1,  1],
       [ 0,  0,  1, -1,  0],
       [ 0,  0,  4, -1,  0]], dtype=int64)

y
✓ 0.6s
array([0, 1, 0, ..., 1, 1, 1], dtype=int64)
```

Figure 3.19 Variable Declaration

ii. Splitting Data into Train and Test Set

```
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)
✓ 0.6s

X_train
✓ 0.1s
array([[ 1,  0,  3, -1,  0],
       [ 0,  0,  4, -1,  0],
       [ 0,  0,  4, -1,  0],
       ...,
       [ 1,  0,  5, -1,  0],
       [ 1,  0,  3, -1,  0],
       [ 1,  0,  5,  3,  1]], dtype=int64)

y_train
✓ 0.1s
array([1, 1, 1, ..., 1, 1, 1], dtype=int64)
```

Figure 3.20 Splitting Data

3.6.9 Model Testing

- i. **Preprocessing.** This entails loading the test data, dealing with missing values, and deleting features that won't help predict loan default. Feature alignment as well to ensure that the features in the test data align with the features in the model.

```
testdata= pd.read_csv("test.csv")
testdata.shape
✓ 0.2s
(10514, 62)
```

```
# Percentage of missing values
pd.set_option('display.max_rows', None)
round(testdata.isnull().sum()/len(testdata.index), 2)*100
✓ 0.1s
```

```
#Removing columns having missing values
missing_columns = testdata.columns[100*(testdata.isnull().sum()/len(testdata.index)) > 0]
print(missing_columns)
✓ 0.1s
Index(['ContractEndDate', 'DateOfBirth', 'County', 'City', 'NrOfDependants',
      'LastPaymentOn', 'CurrentDebtDaysPrimary', 'DebtOccuredOn',
      'CurrentDebtDaysSecondary', 'DebtOccuredOnForSecondary', 'DefaultDate',
      'PrincipalOverdueBySchedule', 'PlannedPrincipalPostDefault',
      'PlannedInterestPostDefault'],
      dtype='object')
```

```
miss_col=['ContractEndDate', 'DateOfBirth', 'County', 'City', 'NrOfDependants',
          'LastPaymentOn', 'CurrentDebtDaysPrimary', 'DebtOccuredOn',
          'CurrentDebtDaysSecondary', 'DebtOccuredOnForSecondary', 'DefaultDate',
          'PrincipalOverdueBySchedule', 'PlannedPrincipalPostDefault',
          'PlannedInterestPostDefault']
testdata.drop(miss_col, axis=1, inplace=True)
testdata.shape
✓ 0.1s
(10514, 48)
```

```
#Removing features that have no role in default prediction. These include duplicate
# values such as LoanID vs LoanNumber. Income values when there is Income Total.
cols_del=['ReportAsOfEOD', 'LoanId', 'UserName', 'LoanDate', 'FirstPaymentDate',
          'MaturityDate_Original', 'MaturityDate_Last', 'Country', 'AppliedAmount',
          'IncomeFromPrincipalEmployer', 'IncomeFromPension', 'IncomeFromFamilyAllowance',
          'IncomeFromSocialWelfare', 'IncomeFromLeavePay', 'IncomeFromChildSupport', 'IncomeOther',
          'MonthlyPaymentDay', 'ActiveScheduleFirstPaymentReached', 'PlannedInterestTillDate',
          'PrincipalPaymentsMade', 'InterestAndPenaltyPaymentsMade', 'PrincipalBalance',
          'InterestAndPenaltyBalance', 'PlannedPrincipalTillDate', 'PrincipalWriteOffs',
          'InterestAndPenaltyWriteOffs', 'PreviousEarlyRepaymentsBefoleLoan', 'Status', 'WorkExperience']
testdata.drop(cols_del, axis=1, inplace=True)
testdata.shape
✓ 0.1s
(10514, 19)
```

Figure 3.21 Test Data Preprocessing

ii. Handling Outliers

The data in the columns 'IncomeTotal' and 'Amount' are skewed to the right, indicating that majority of the data is skewed to the right due to outliers. Outliers affect the mean and standard deviation. This can be removed using log transformation to reduce the larger values and normalize.

```
testdata['IncomeTotal_log']=np.log(testdata['IncomeTotal'])
✓ 0.5s

testdata['Amount_log']=np.log(testdata['Amount'])
✓ 0.3s

#Convert non numeric variables to numeric
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
cat=['NewCreditCustomer','Restructured','EmploymentDurationCurrentEmployer']
for var in cat:
    le = preprocessing.LabelEncoder()
    testdata[var]=le.fit_transform(testdata[var].astype('str'))
✓ 0.5s
```

Figure 3.22 Handling Outliers in Test Data

ii. Loan Default Prediction

Selecting the independent categorical variables same as the features.

```
test
✓ 0.1s
array([[0, 1, 4, 6, 0],
       [1, 0, 4, 5, 0],
       [1, 0, 2, 3, 0],
       ...,
       [0, 1, 4, 3, 1],
       [0, 0, 4, 3, 0],
       [0, 0, 3, 3, 0]], dtype=int64)

test = ss.fit_transform(test)
✓ 0.6s

pred= DTClassifier.predict(test)
pred
✓ 0.1s
array([0, 1, 0, ..., 0, 1, 0], dtype=int64)
+ Code + Markdown

testdata["Default"] = pred
testdata['Default'].value_counts()
✓ 0.1s
0    7874
1    2640
Name: Default, dtype: int64
```

Figure 3.23 Loan Default Prediction

3.7 Performance Metrics

3.7.1 Confusion Matrix

This produces a matrix that describes the model's overall performance.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Table 3.1 Confusion Matrix

True Positives are instances where the forecast is YES and the actual output is YES.

True Negatives: When the prediction is NO and the actual output is also NO.

False Positives are instances where the prediction is YES but the actual outcome is NO.

False Negatives: When the predicted outcome is NO but the actual outcome is YES.

3.7.2 Accuracy

This is the ratio of number of right predictions to the total number of input samples.

Accuracy for the matrix is calculated as:

True Positive + True Negative

Total Sample

3.7.3 Precision

When compared to the total number of positively predicted values, this metric shows the number of True Positives that are truly positive.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

3.7.4 Recall

The Recall Metric shows how many True Positives the model has classified out of the total number of samples that should have been positive.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}.$$

3.7.5 Specificity

The number of True Negatives classified by the model out of the total number of samples that should have been classified as negative is called specificity.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}.$$

3.7.6 F1 Score

The Harmonic Mean of precision and recall is the F1 Score. F1 Score has a range of [0, 1]. It demonstrates how exact your classifier is, i.e. how many instances it correctly classifies, as well as how robust it is, i.e. how many examples it does not miss. The higher the F1 Score, the better our model's performance.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1 Introduction

This Chapter discusses the results obtained from the developed model of loan default prediction. The results are analyzed with respect to the research objectives and how they relate to the problem statement and outlined methodology.

4.2 Results

The research addresses the problem of improving the accuracy of predicting loan default in mobile lending. The developed model is used to classify default and non-default loans by applying machine learning algorithms. The model is developed from historical loan data of borrowers using the categorical variable of their loan default status. The model was able to successfully recognize the behavior patterns of the borrowers and predict the probability of default of new loan applications. The performance metrics applied to evaluate the performance of the model were instrumental in determining the best algorithm to determine the probability of default. These metrics include confusion matrix, precision, recall and F1 score. The decision tree algorithm had the highest accuracy and precision of 0.64, however it is also equally important to avoid the misclassification of default loans as non-default loans as this results in loss. The comparison of the classification results obtained from the three machine learning algorithms which are decision trees logistic regression and Naive bayes shows the efficiency of using machine learning algorithms in predicting loan default. The results presented also accomplish the objectives of the research which is to design a machine learning loan prediction model , train and test the model and evaluate the performance of the model. The dataset includes 16,046 samples. The dataset distribution is 44% is non default and 56% default.

	Non-default (Positive)	Default (negative)	Samples
	1 = 7,085(44%)	0 = 8,961 (56%)	16,046
Training data (70%)	4,959	6,273	11,232
Test data (30%)	2,126	2,688	4,814

Table 4.1 Dataset

4.2.1 Decision Tree

The confusion matrix stating the proportion of correctly classified as well as those misclassified for each category gives a fulfilling picture of the test result. The result is rounded off to nearest integer.

- The distribution shows 63% of the data was correctly classified while 27% was misclassified.
- Misclassifications can be split into type 1 (false positive) and type 2 (false positive errors). The model contains 11% type 1 errors which means 513 loans were classified as non-default, but they actually default. This is a problem because the mobile lending institution will incur a loss when they customers are issued a loan.
- Type 2 error contains 1,233 cases (26%) where the model classifies a borrower will default but they actually did not default. This doesn't result in a loss however limits the mobile lender from issuing loans to customers who would have fulfilled their obligations and repaid.

Confusion Matrix N=4,814		Actual	
		Non-default (Positive)	Default (Negative)
Predicted	Non- default (Positive)	Predicted correct (True positive) 2186 (46%)	Type 1 error (False positive) 513 (11%)
	Default (Negative)	Type 2 error (False negative) 1233 (26%)	Predicted correct (True negative) 882 (17%)

Table 4.2 Decision Tree Confusion matrix

Accuracy	Precision	Recall	F1score
0.64	0.64	0.64	0.62

Table 4.3 Decision Tree Performance

The model is also evaluated across accuracy , precision , recall and F1 score. The model obtained a 64% result in accuracy and recall , 63% in precision and 62% in F1 Score.

- **Accuracy** is the total number of correct predictions made to determine if a loan would be default or non-default from the entire sample.
- **Precision** shows the actual number of non-default predictions out of all the all the values predicted as non-default. This is achieved by:
- **Recall** shows the percentage of non-default predictions are correct over the total number of samples that should have been non-default.
- **F1 Score** is the harmonic mean between precision and recall.

In the case of loan default prediction, the recall metric is most important to determine the percentage of non-default predictions from all values that should have been predicted as default.

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
from sklearn import metrics
DTClassifier= DecisionTreeClassifier (criterion= 'entropy', random_state=0)
DTClassifier.fit(X_train, y_train)
y_pred= DTClassifier.predict(X_test)
y_pred
confusion = confusion_matrix(y_test, y_pred)
print('Confusion Matrix\n')
print(confusion)
print ('The accuracy of the decision tree is:', metrics.accuracy_score(y_pred,y_test))
print('Weighted Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='weighted')))
print('Weighted Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='weighted')))
print('Weighted F1-score: {:.2f}'.format(f1_score(y_test, y_pred, average='weighted')))
✓ 0.1s
Confusion Matrix
[[2186  513]
 [1233  882]]
The accuracy of the decision tree is: 0.6373078520980474
Weighted Precision: 0.64
Weighted Recall: 0.64
Weighted F1-score: 0.62

```

Figure 4.1 Decision Tree Results

4.2.2 Logistic Regression

The confusion matrix stating the proportion of correctly classified as well as those misclassified for each category gives a fulfilling picture of the test result. The result is rounded off to nearest integer.

- The distribution shows 63% of the data was correctly classified while 27% was misclassified.
- Misclassifications can be split into type 1 (false positive) and type 2 (false positive errors). The model contains 12% type 1 errors which means 555 loans were classified as non-default, but they actually default. This is a problem because the mobile lending institution will incur a loss when they customers are issued a loan.
- Type 2 error contains 1,221 cases (25%) where the model classifies a borrower will default but they actually did not default. This doesn't result in a loss however limits the mobile lender from issuing loans to customers who would have fulfilled their obligations and repaid.

Confusion Matrix N=4,814		Actual	
		Non-default (Positive)	Default (Negative)
Predicted	Non- default (Positive)	Predicted correct (True positive) 2144 (45%)	Type 1 error (False positive) 555 (12%)
	Default (Negative)	Type 2 error (False negative) 1221 (25%)	Predicted correct (True negative) 894 (18%)

Table 4.4 Logistic Regression Confusion matrix

The model is also evaluated across accuracy , precision , recall and F1 score. The model obtained a 63% result in accuracy , recall and precision and 62% in F1 Score.

- **Accuracy** is the total number of correct predictions made to determine if a loan would be default or non-default from the entire sample.
- **Precision** shows the actual number of non-default predictions out of all the all the values predicted as non-default.
- **Recall** shows the percentage of non-default predictions are correct over the total number of samples that should have been non-default.
- **F1 Score** is the harmonic mean between precision and recall.

Accuracy	Precision	Recall	F1score
0.63	0.63	0.63	0.62

Table 4.5 Logistic Regression Performance

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred= model.predict(X_test)
y_pred
confusion = confusion_matrix(y_test, y_pred)
print('Confusion Matrix\n')
print(confusion)
print ('The accuracy of the Logistic Regression is:', metrics.accuracy_score(y_pred,y_test))
print('Weighted Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='weighted')))
print('Weighted Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='weighted')))
print('Weighted F1-score: {:.2f}'.format(f1_score(y_test, y_pred, average='weighted')))
✓ 0.1s
Confusion Matrix

[[2144  555]
 [1221  894]]
The accuracy of the Logistic Regression is: 0.6310760282509348
Weighted Precision: 0.63
Weighted Recall: 0.63
Weighted F1-score: 0.62

```

Figure 4.2 Logistic Regression Results

4.2.3 Naïve Bayes

The confusion matrix stating the proportion of correctly classified as well as those misclassified for each category gives a fulfilling picture of the test result. The result is rounded off to nearest integer.

- The distribution shows 61% of the data was correctly classified while 29% was misclassified.
- Misclassifications can be split into type 1 (false positive) and type 2 (false positive errors). The model contains 23% type 1 errors which means 1109 loans were classified as non-default, but they actually default. This is a problem because the mobile lending institution will incur a loss when they customers are issued a loan.
- Type 2 error contains 780 cases (16%) where the model classifies a borrower will default but they actually did not default. This doesn't result in a loss however limits the mobile lender from issuing loans to customers who would have fulfilled their obligations and repaid.

Confusion Matrix N=4,814		Actual	
		Non-default (Positive)	Default (Negative)
Predicted	Non- default (Positive)	Predicted correct (True positive) 1590 (33%)	Type 1 error (False positive) 1109 (23%)
	Default (Negative)	Type 2 error (False negative) 780 (16%)	Predicted correct (True negative) 1335 (28%)

Table 4.7 Naive Bayes Confusion matrix

The model is also evaluated across accuracy , precision , recall and F1 score. The model obtained a 63% result in accuracy , recall and precision and 62% in F1 Score.

- **Accuracy** is the total number of correct predictions made to determine if a loan would be default or non-default from the entire sample.
- **Precision** shows the actual number of non-default predictions out of all the all the values predicted as non-default.
- **Recall** shows the percentage of non-default predictions are correct over the total number of samples that should have been non-default.
- **F1 Score** is the harmonic mean between precision and recall.

Accuracy	Precision	Recall	F1score
0.61	0.62	0.61	0.61

Table 4.8 Naïve Bayes Performance

```

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
NBClassifier = GaussianNB()
NBClassifier.fit(X_train,y_train)
y_pred = NBClassifier.predict(X_test)
y_pred
confusion = confusion_matrix(y_test, y_pred)
print('Confusion Matrix\n')
print(confusion)
print ('The accuracy of the Naive Bayes is:', metrics.accuracy_score(y_pred,y_test))
print('Weighted Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='weighted')))
print('Weighted Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='weighted')))
print('Weighted F1-score: {:.2f}'.format(f1_score(y_test, y_pred, average='weighted')))
✓ 0.1s
Confusion Matrix

[[1590 1109]
 [ 780 1335]]
The accuracy of the Naive Bayes is: 0.6076028250934774
Weighted Precision: 0.62
Weighted Recall: 0.61
Weighted F1-score: 0.61

```

Figure 4.3 Naïve Bayes Results

4.3 Discussion

The distribution shows decision trees and logistic regression classified 63% of the data was correctly classified while 27% was misclassified. Naives Bayes classified 61% of the data was correctly classified while 29% was misclassified. Misclassifications can be split into type 1 (false positive) and type 2 (false positive errors).

Type 1 error refers to the loans that were classified as non-default, but they default. This is a problem because the mobile lending institution will incur a loss when they customers are issued a loan. Decision trees had the lowest type 1 error at 11% followed by Logistic regression at 12% and lastly Naïve Bayes at 33%. This is measured by precision.

Type 2 error refers to the loans that were classified as default, but they did not default. This doesn't result in a loss however limits the mobile lender from issuing loans to customers who would have fulfilled their obligations and repaid. Decision trees had the highest type 2 error at 26% followed by Logistic regression at 25% and lastly Naïve Bayes at 16%. This is measured by recall.

Confusion Matrix N=4,814		Actual	
		Non-default (Positive)	Default (Negative)
Predicted	Non- default (Positive)	Predicted correct (True positive)	Type 1 error (False positive)
	Decision Tree	2186 (46%)	513 (11%)
	Logistic Regression	2144 (45%)	555 (12%)
	Naïve Bayes	1590 (33%)	1109 (23%)
	Default (Negative)	Type 2 error (False negative)	Predicted correct (True negative)
	Decision Tree	1233 (26%)	882 (17%)
	Logistic Regression	1221 (25%)	894 (18%)
	Naïve Bayes	780 (16%)	1335 (28%)

Table 4.10 Confusion Matrix Comparison

The machine learning algorithms have various assumptions which affect the performance results.

Decision Trees	Logistic Regression	Naïve Bayes
The variables are categorical	The variables are categorical	The variables are categorical
At the beginning, the whole training set is considered as the root.	The predictor variables are independent	The predictor variables are independent
Records are distributed recursively on the basis of attribute values	There is no multicollinearity (this occurs when two or more exploratory variables are highly correlated).	
	There are no extreme outliers	

Decision tree has the highest accuracy , recall leading with a minimal margin in comparison with Logistic regression. Naives Bayes has the least accuracy , precision , recall and F1 score. The performance metrics are important in giving a wholistic view of the algorithms. In the case of loan default prediction that has a direct impact on revenue, recall has a significance impact. Recall shows how many True Positives the model has classified from the total number of all samples that should have been identified as positive. The slightest margin makes a significance as they determine the overall accuracy and the best lending decision to reduce the risk of default.

	Accuracy	Precision	Recall	F1 Score
Decision Trees	0.64	0.64	0.64	0.62
Logistic Regression	0.63	0.63	0.63	0.62
Naïve Bayes	0.60	0.62	0.61	0.61

Table 4.11 Performance Comparison

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This Chapter discusses the value of the research , the limitations of the research , the conclusion in accordance with the evidence presented and recommendations for further study.

5.2 Conclusions

The application of machine learning techniques in the financial sector with the goal of profit maximization has seen a rising interest over the last few years. There has been increasing number of research conducted in the areas of credit scoring, risk management and bankruptcy prediction using machine learning approaches. Rapid telecommunications and infrastructure development in Kenya coupled with the global decline in cellphone prices has led to an increase in mobile lending. Customers who were previously unbanked can now access digital credit from their mobile phones. This new dynamic creates an opportunity and a challenge to mobile lending institutions on how best to make lending decisions to determine whether a customer will default on a loan.

This research proposes machine learning as a method to improve the accuracy of loan default predictions. This better understanding of customer behaviors to improve the prediction of loan default will contribute to tremendous financial benefit in the mobile lending sector. This research successfully explores the features of loan data that contribute to the risk of loan defaults. Exploratory data analysis shows the correlation of various features with loan default to select the most appropriate features to train the machine learning model. The train and test data set are then applied to three machine learning algorithms to determine the one with the most accurate results. Key performance metrics which include confusion matrix, accuracy , precision and recall and applied to evaluate the best machine learning technique in loan default prediction.

5.3 Limitations of the research

Machine learning algorithms are limited to the dataset used to train and test the model. Data is governed by data protection laws making it challenging to access primary data for the purpose of research. This research was conducted using secondary open data from bondora.com that is available to the public. This limits the generalization of the model as it is specific towards the dataset used to train and test the model. It would be beneficial to look comprehensively at the main features that are relevant to the characteristic that drive default and can be applied. Further limitation is in reference to the variables provided in the dataset, although the dataset is open due to data protection laws some factors may not be available to the public and these may have had an impact on the predictions of the probability of default. Lastly, the research focused on the probability of default in a default state however default loans may still be recovered during the recollection process,

5.4 Recommendations and Future Work

This research explores using machine learning algorithms to improve the accuracy of predicting loan default. This model will be instrumental to mobile lending institutions in evaluating their customer credit risk. The best performing model in the research which is decision trees achieves an accuracy of about 64%. This is a fair performance and can further be improved through different methods of parameter tuning and feature selection which may possibly yield improvements in the model performance. It may also be beneficial to do a cross validation with other sources of open dataset as they become more accessible to compare the performance of the model. Since the research is also limited to the probability of default in a default state , further exploration may be made in determining the expected return of the loan based on borrower's characteristics , loan characteristics the recollection of loans processes.

REFERENCES

- [1]. Oates, B. (2006). *Researching information systems and computing*. London: SAGE.
- [2]. Gwer, F., Odero, J., & Totolo, E. (2020). Digital credit audit report: Evaluating the conduct and practice of digital lending in Kenya – Financial Sector Deepening Kenya. Retrieved 18 October 2020, from <https://fsdkenya.org/publication/digital-credit-audit-report-evaluating-the-conduct-and-practice-of-digital-lending-in-kenya/>
- [3]. Credit scoring approaches guidelines - World Bank. (n.d.). Retrieved October 19, 2020, from <http://pubdocs.worldbank.org/en/935891585869698451/CREDIT-SCORING-APPROACHES-GUIDELINES-FINAL-WEB.pdf>
- [4]. Central Bank of Kenya, Kenya National Bureau of Statistics and FSD Kenya. 2016. The 2016 FinAccess household survey.
- [5]. Sousa, M., Gama, J. and Brandão, E., 2016. A new dynamic modeling framework for credit risk assessment. *Expert Systems with Applications*, 45, pp.341-351.
- [6]. Arun, K., Ishan, G., & Sanmeet, K. (2016). Loan Approval Prediction based on Machine Learning Approach. *IOSR Journal Of Computer Engineering*, 18(3), 79-81.
- [7]. Kadam, A., Nikam, S., Aher, A., Shelke, G., & Chandgude, A. (2021). Prediction for Loan Approval using Machine Learning Algorithm. *International Research Journal of Engineering and Technology*, 08(04), 4089-4092.
- [8]. Blessie, E., & Rekha, R. (2019). Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 2714-2719.
- [9]. Aithal, V., & Jathanna, R. (2019). Credit Risk Assessment using Machine Learning Techniques. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 3482-3486. doi: 10.35940/ijitee.a4936.119119

- [10]. Mohammadi, N., & Zangeneh, M. (2016). Customer Credit Risk Assessment using Artificial Neural Networks. *International Journal of Information Technology and Computer Science*, 8(3), 58-66. doi: 10.5815/ijitcs.2016.03.07
- [11]. Blessie, C. (2019). Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 2714-2719. doi: 10.35940/ijitee.a4881.119119
- [12]. Uddin, S., Khan, A., Hossain, M., & Moni, M. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19(1). doi: 10.1186/s12911-019-1004
- [13]. Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. *IOP Conference Series: Materials Science and Engineering*, 1022, 012042. doi: 10.1088/1757-899x/1022/1/012042
- [14]. Cook, T., & McKay, C. (2015). How M-Shwari works: The story so far. Forum 10, Washington, DC: CGAP and FSD Kenya. Retrieved from <http://www.cgap.org/sites/default/files/Forum-How-M-Shwari-Works-Apr-2015.pdf>
- [15]. Bhandari, M. (2020, October 19). Predict Loan Eligibility using Machine Learning Models. Medium. <https://towardsdatascience.com/predict-loan-eligibility-using-machine-learning-models-7a14ef904057>
- [16]. Xu, Z. (. (2021, March 10). Loan default prediction with Berka dataset. Medium. <https://towardsdatascience.com/loan-default-prediction-an-end-to-end-ml-project-with-real-bank-data-part-1-1405f7aecb9e>
- [17]. Massaoudi, T. (2020, October 23). ML basics : Loan prediction. Medium. <https://towardsdatascience.com/ml-basics-loan-prediction-d695ba7f31f6>

- [18]. Aleksandrova, Y. (2021). Comparing Performance of Machine Learning Algorithms for Default Risk Prediction in Peer-to-Peer Lending. TEM Journal, 133-143. <https://doi.org/10.18421/tem101-16>
- [19]. Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. IOP Conference Series: Materials Science And Engineering, 1022, 012042. doi: 10.1088/1757-899x/1022/1/012042
- [20]. Bondora.com. Bondora.com. (2021). Retrieved 4 July 2021, from <https://www.bondora.com/en/public-reports>.
- [21]. Jafar Hamid, A., & Ahmed, T. (2016). Developing Prediction Model of Loan Risk in Banks Using Data Mining. Machine Learning and Applications: An International Journal, 3(1), 1-9.

APPENDICES

Appendix A: Gantt Chart

PROJECT TITLE		LOAN DEFAULT PREDICTION USING MACHINE LEARNING				
PROJECT MANAGER		GOLDA KISUTSA				
WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	% OF TASK COMPLETE
1 Milestone 1						
1.1	Introduction	Golda Kisutsa	2-Mar-2021	7-Mar-2021	5	100%
1.2	Literature Review	Golda Kisutsa	8-Mar-2021	18-Mar-2021	10	100%
1.3	Methodology	Golda Kisutsa	19-Mar-2021	6-Apr-2021	17	100%
1.4	Proposal Presentation	Golda Kisutsa	8-Apr-2021	8-Apr-2021	1	100%
2 Milestone 2						
2.1	Research Design	Golda Kisutsa	15-Apr-2021	22-Apr-2021	7	100%
2.2	Data Analysis	Golda Kisutsa	26-Apr-2021	17-May-2021	21	100%
2.3	Modelling	Golda Kisutsa	18-May-2021	23-Jun-2021	35	100%
2.4	Results	Golda Kisutsa	24-Jun-2021	4-Jul-2021	10	100%
2.5	Milestone 2 Presentation	Golda Kisutsa	6-Jul-2021	6-Jul-2021	1	100%
3 Milestone 3						
3.1	Model optimization	Golda Kisutsa	8-Jul-2021	18-Jul-2021	10	100%
3.2	Conclusion	Golda Kisutsa	19-Jul-2021	24-Jul-2021	5	100%
3.3	Recommendation	Golda Kisutsa	25-Jul-2021	29-Jul-2021	4	100%
3.4	Abstract	Golda Kisutsa	29-Jul-2021	31-Jul-2021	2	100%
3.5	Final Presentation	Golda Kisutsa	3-Aug-2021	3-Aug-2021	1	100%