

LOAN ELIGIBILITY CLASSIFICATION
USING MACHINE LEARNING APPROACH

PAUL LAW LIK PAO

Bachelor of Computer Science (Software
Engineering)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : PAUL LAW LIK PAO

Date of Birth

Title : LOAN ELIGIBILITY CLASSIFICATION USING
MACHINE LEARNING APPROACH

Academic Session : SEMESTER 2, SESSION 2022/2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

(Supervisor's Signature)

TS. DR MOHD ARFIAN BIN ISMAIL

New IC/Passport Number
Date: 18 MAY 2023

Name of Supervisor
Date: 18 MAY 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) with Honours.

(Supervisor's Signature)

Full Name : TS. DR MOHD ARFIAN BIN ISMAIL

Position : SENIOR LECTURER

Date : 18 MAY 2023



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : PAUL LAW IK PAO

ID Number : CB20025

Date : 18 MAY 2023

LOAN ELIGIBILITY CLASSIFICATION USING MACHINE LEARNING
APPROACH

PAUL LAW LIK PAO

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

MAY 2023

ACKNOWLEDGEMENTS

I would like to express my heartfelt thanks and immense appreciation to the divine providence for granting me the opportunity and favor to successfully accomplish my Final Year Project titled "Loan Eligibility Classification using Machine Learning Approach" within the designated timeframe.

First and foremost, I am immensely grateful to my supervisor, Ts. Dr Mohd Arfian Bin Ismail, for his guidance, support, and invaluable expertise throughout the entire duration of this project. His insightful feedback, encouragement, and continuous assistance have been instrumental in shaping and refining the direction of my research.

I would also like to extend my sincere appreciation to the lecturers of the Faculty of Computing at Universiti Malaysia Pahang for their exceptional teaching and providing me with a solid foundation in the field of machine learning. Their dedication to education and willingness to share their knowledge have been pivotal in developing my skills and understanding in this area.

I am indebted to the researchers and authors whose work and publications I extensively studied to gain a comprehensive understanding of different machine learning algorithms. Their contributions to the field have been instrumental in shaping the theoretical framework of this project.

Last but not least, I would like to thank my family and friends for their unwavering support, encouragement, and patience throughout this academic journey. Their belief in my abilities and constant motivation have been the driving force behind my perseverance and determination to excel.

ABSTRAK

Kertas penyelidikan ini membentangkan satu kajian mengenai pengelasan kelayakan pinjaman menggunakan pendekatan pembelajaran mesin dengan membandingkan prestasi tiga algoritma Pembelajaran Mesin iaitu Regresi Logistik, Hutan Rawak, dan Pohon Keputusan. Kajian ini dijalankan menggunakan Python dan Jupyter Notebook untuk analisis data dan pembangunan model. Model-model tersebut kemudiannya dinilai menggunakan set ujian dengan menggunakan metrik penilaian seperti Ketepatan, Presisi, Pemanggilan, dan Skor F1. Prestasi model-model tersebut dibandingkan untuk mengenal pasti algoritma yang paling berkesan dalam pengelasan kelayakan pinjaman. Antara ketiga-tiga pendekatan Pembelajaran Mesin, model RL kelihatan paling berkesan dalam mengelas kelayakan pinjaman, dengan skor ketepatan 82%, skor pemanggilan 82%, skor presisi 81%, dan skor F1 79%.

ABSTRACT

Machine learning is becoming increasingly vital in various domains, including loan eligibility classification, due to its ability to analyze large amounts of data, develop predictive models, adapt to new information, and automate processes. This research paper presents a study on loan eligibility classification using a machine learning approach by comparing the performance of three Machine Learning algorithms which were Logistic Regression, Random Forest, and Decision Tree. This research was conducted using Python and Jupyter Notebook for data analysis and model development. The models were then evaluated on the testing set using evaluation metrics such as Accuracy, Precision, Recall, And F1-Score. The performance of the models was compared to identify the most effective algorithm for loan eligibility classification. Among the three ML approach, the LR model appears to be the most effective at classify loan eligibility, with the 82% accuracy score, 82% recall score, 81% precision score and 79% F1 score.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS	xi
LIST OF ABBREVIATIONS	xii
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE	4
1.4 SCOPE	5
1.5 SIGNIFICANCE OF PROJECT	5
1.6 REPORT ORGANIZATION	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 INTRODUCTION	8
2.2 EXISTING METHODS	9
2.2.1 Logistic Regression	9

2.2.2	Decision Tree	11
2.2.3	Random Forest	14
2.3	COMPARATIVE ANALYSIS OF EXISTING APPROACHES	17
2.4	SUMMARY	19
CHAPTER 3 METHODOLOGY		20
3.1	INTRODUCTION	20
3.2	RESEARCH FRAMEWORK	20
3.2.1	Literature Review	21
3.2.2	Collection of Dataset	22
3.2.3	Data Preprocessing	26
3.2.4	Logistic Regression Algorithm	31
3.2.5	Result	32
3.3	PROJECT REQUIREMENT	33
3.3.1	Logistic Regression	33
3.3.2	Input	35
3.3.3	Output	36
3.4	PROOF OF INITIAL CONCEPT	37
3.5	POTENTIAL OF PROPOSED SOLUTION	39
3.6	HARDWARE AND SOFTWARE	40
3.7	SUMMARY	42
CHAPTER 4 IMPLEMENTATION, RESULTS AND DISCUSSION		43
4.1	INTRODUCTION	43
4.2	IMPLEMENTATION PROCESS	43
4.2.1	Exploratory Data Analysis (EDA)	43

4.2.2	Data Visualization	46
4.2.3	Data Preprocessing	55
4.3	TESTING	60
4.3.1	Training and Testing Data Ratio	60
4.3.2	Training Machine Learning Models	61
4.4	ML MODEL PROPOSED	64
4.5	DISCUSSION	65
4.6	RESULT	67
4.6.1	Comparison Performance Result of Proposed ML Models	67
4.6.2	Loan Eligibility Prediction System Proposed	70
CHAPTER 5 CONCLUSION		74
5.1	INTRODUCTION	74
5.2	RESEARCH CONSTRAINTS CHALLENGES	75
5.3	FUTIRE WORK	76
REFERENCES		78
APPENDIX A CORRELATION HEATMAP		84

LIST OF TABLES

Table 1. 1 Summary of Problem Statement	4
Table 2. 1 Results of Evaluation Metrics	16
Table 2. 2 Comparisons of the Existing Method	18
Table 3. 1 Attributes of the Loan Prediction Dataset	23
Table 3. 2 Training Dataset of Loan Applicants	24
Table 3. 3 Second Dataset of Loan Applicants	25
Table 3. 4 Dataset of the loan applicant	36
Table 3. 5 Hardware Descriptions and Specifications	40
Table 3. 6 Software Description and Specification	40
Table 4. 1 Comparison Result of Performance Metric	68

LIST OF FIGURES

Figure 2. 1	Steps of the DT	12
Figure 2. 2	Flowchart of DT Algorithm	14
Figure 2. 3	Steps of the RF algorithm	15
Figure 3. 1	Summary of phases in the research methodology	21
Figure 3. 2	Histogram of the Gender	27
Figure 3. 3	Histogram of the Education	27
Figure 3. 4	Histogram of the Dependents	27
Figure 3. 5	Histogram of the Self-Employed	28
Figure 3. 6	Histogram of the Loan Status	28
Figure 3. 7	Histogram of the Credit History	28
Figure 3. 8	Histogram of the Total Income	29
Figure 3. 9	Histogram of the Loan Amount	29
Figure 3. 10	Histogram of the Loan Amount Term	30
Figure 3. 11	Histogram of the Property Area	30
Figure 3. 12	Process of LR	31
Figure 3. 13	Outcome of the Heat Map	32
Figure 3. 14	Outcome of the Correlation Matric	33
Figure 3. 15	The Sigmoid Function	34
Figure 3. 16	Flowchart of LR Algorithm	37
Figure 4. 1	Flowchart of the Exploratory Data Analysis	44
Figure 4. 2	Check Data Type of Variables	45
Figure 4. 3	Check Data Type of Variables	45
Figure 4. 4	Identify the Loan_ID Contains Duplicate Values	46
Figure 4. 5	Calculate Total Number of Missing Values	46
Figure 4. 6	Visualization of the Gender Variable	47
Figure 4. 7	Visualization of the Dependents Variable	48
Figure 4. 8	Visualization of the Education Variable	49
Figure 4. 9	Visualization of the Loan Status Variable	49
Figure 4. 10	Visualization of the Self-Employed Variable	50

Figure 4. 11	Visualization of the Total Income Variable	51
Figure 4. 12	Visualization of the Total Income Variable	51
Figure 4. 13	Visualization of the Loan Amount Variable	52
Figure 4. 14	Visualization of the Loan Amount Term Variable	53
Figure 4. 15	Visualization of the Loan Amount Term Variable	54
Figure 4. 16	Visualization of the Credit History Variable	54
Figure 4. 17	Visualization of the Property Area Variable	55
Figure 4. 18	Flowchart of Data Preprocessing	56
Figure 4. 19	Feature Selection	57
Figure 4. 20	Handling Missing and Null Values	57
Figure 4. 21	Convert String Value into Numeric Values	58
Figure 4. 22	Modify and Reset the DataFrame Structure	58
Figure 4. 23	Create New model_data DataFrame	59
Figure 4. 24	Label Encoding	59
Figure 4. 25	Adding New Column	60
Figure 4. 26	Training and Testing Data Ratio	61
Figure 4. 27	Training ML Models	61
Figure 4. 28	Result of Performance Metrics	62
Figure 4. 29	Training RF Model	62
Figure 4. 30	Training DT Model	63
Figure 4. 31	Training LR Model	64
Figure 4. 32	model.py	65
Figure 4. 33	Interface of Loan Eligibility Prediction Web Application	67
Figure 4. 34	Radar chart of the performance of the ML methods	70
Figure 4. 35	Data Row of Successful Loan Status	72
Figure 4. 36	Input of the Successful Loan Status Data	72
Figure 4. 37	Data Row of Unsuccessful Loan Status	73
Figure 4. 38	Input of the Unsuccessful Loan Status Data	73

LIST OF SYMBOLS

/	Division
+	Division
^	Exponentiation
-	Subtraction
*	Multiplication
e	Euler's Constant

LIST OF ABBREVIATIONS

ML	Machine Learning
MCO	Movement Control Order
ATD	Association for Talent Development
RF	Random Forest
DT	Decision Tree
LR	Logistic Regression
SEER	Surveillance, Epidemiology, and End Results
CM	Confusion Matrix
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
THAID	THeta-Alpha-Input-Design
WEKA	Waikato Environment for Knowledge Analysis
KNN	K-Nearest Neighbors
AD	Alzheimer Disease
HC	Healthy Controls
MCI	Mild Cognitive Impairment
sMCI	stable Mild Cognitive Impairment
pMCI	progressive Mild Cognitive Impairment
SVM	Support Vector Machine
UCI ML	University of California Irvine Machine Learning

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

One of the oldest industries in this world is banking. This industry started when merchants made grain loans to farmers and traders transported products between towns or cities. Before the banking system appear, the people would trade product or service for another in exchange which called barter system. This barter system has been in existence for centuries around the world until the invention of money and start the banking activities by using the money. Banking activities at Greece more varied as the lenders at Greece starts made loans, accept the deposits, and provide money exchange service by changing from one currency to another currency. The first modern bank in the history was founded in Siena, where located at the Italian Renaissance in 15th century. The bank is called Banca dei Pashi di Siena (Specialist, 2022).

In modern era, a major challenging of buying a properties or things is having the sufficient money as the money is needed to pay for the basic needs and stuffs that needed in life. To solve these issues, people normally apply the loan from the bank to purchase the properties or stuffs needed as they can get the cash directly after the loan application approved by the bank. Thus, there has been an increasing in loan application yearly in Malaysia especially the inflation occurs these few years. The raising prices of properties and cars forced the people needed to apply the loan from local bank and pay the interest that charged by bank included principal amount of loans Sharen (Kaur, 2022). Loans are the primary revenue source for banks in Malaysia as the interest charged on loans is where their significant profit comes from.

As been stated, loans occasionally become necessary for those needed the money for business purpose. Thus, loan eligibility classification by using the ML method could be extremely helpful to every bank in Malaysia as the loan applications can be process faster and efficient. The model for classify the loan eligibility needs to be trained through a dataset that comprises data of the loan applicants. For example, the name, age, gender, marital state, income, credit card history, and loan amount (Massaoudi, 2021).

ML methodology using the LR will be apply in this study as it covers the classification. machine learning, which is becoming increasingly common as it is relevant in many fields and industry. For example, the ML method applied in medical field called medical diagnosis which user only need choose the symptoms of sick in the system and the system will discern patterns in symptoms. The user will receive the summary of report that contains the possible illness based on symptoms and recommends treatment options for user (Sinhasane, S, 2018).

By applying the ML ensure the better outcome of the loan application besides speed up the loan approval progress automatically. Through this study, the employees of the financial institution can classify the loan applications is accept or reject as the whole loan approval process is automated by ML method. The classification model which is LR will be applied in this study as the information of loan applicants will be processed to classify whether the loan application is sanctioned or not

1.2 PROBLEM STATEMENT

There are some significant issues that related to the loan application has been figure out and this study is based an observation and experiences from the loan applicants in Malaysia. Nowadays, inflation happened around worldwide include Malaysia causes the Malaysians face difficulties in maintain the lifestyle before inflation period. Mostly, the Malaysians faced the economic crisis during the Movement Control Order (MCO) began on 18th March 2020. Total of 10,317 Malaysians was declared bankrupt and total of 1,246 businesses also shut down during the MCO (Bernama, 2021).

Most of the businesses closed down during the MCO because the business loan that applied at the local bank took lengthy procedure to approve the loans. This can be threat for the company which needs immediate cash to fill the void as the poor cash flow can sink the business anytime. The local financial institution still using the traditional methods manually examine the applicant's income, credit history, and several other factors to establish the risk. Therefore, the progress of loan approval takes long duration by local financial institution.

The others issue was the local financial institutions senior employees need to train the new hired employees to get familiar with loan approval procedure. According to Association for Talent Development (ATD), the average organization spent 1,252 US dollar per employee on development initiatives and training (Markovic, I, 2020). The existing knowledges and skills of the new hired employees may not be able to suit the loan approval procedure as the different policies and requirements. Thus, the senior employees need to spend time in new hired employee training, and this will lower the organization productivity as the senior employees cannot devote the time to expand the organization business.

The new hired employees of the bank need to take time to understand and manage the loan applicant documents for reviewing and determining the risk of loan. This scenario significantly is time-consuming as some of the new employees are quick learners, while others need more supervision from the senior employee. Table 1.1 shows the summary of the problem statement.

Table 1. 1 Summary of Problem Statement

Problem	Description	Effect
Lengthy loan approval process	Examine and determine the loan applications manually.	The duration of loan approval is longer as the determination manually is time-consuming.
Significant employee training which increase the resourcing costs	The employees of the financial need to be training of loan approval procedures.	The senior employees need to train the new employees and it is time-consuming. Therefore, the productivity of the financial institution becomes low.

1.3 OBJECTIVE

The aim of this study is to develop a loan eligibility classification for the financial institutions in Malaysia. To achieve the aim, several objectives need to be meet. There are three objectives in this study and stated below:

- 1) To study on how to predict the loan eligibility by using the ML method.
- 2) To design and implement the ML method on a system that can automate the loan eligibility process.

- 3) To evaluate the functionality of using loan eligibility classification to shorten the loan approval process.

1.4 SCOPE

The scopes of this study are:

Employees of the financial institutions are the user of this system. Loan applicants submit their documents needed for loan application to the financial institution. In this study, the loan approval process applied the LR model to see the loan application can be sanctioned or not. Employees of the loan institution insert the data of the loan applicants like name, gender, age, marital state, income, credit card history, and loan amount to analysis the eligibility of the loan application.

This study of the ML method which LR model applied in loan eligibility classification allows the loan approval process to become more digital and automated as the employees of the financial institution able to save time in approve the loan application automatically but not manually. The LR model that performs classification task by organizing the loan application data while speed up the loan approval process.

1.5 SIGNIFICANCE OF PROJECT

- i) Employees

The financial institution employees able to process the loan approval in paperless and automated way as the employees only need to insert the dataset of loan applicants to test the eligibility of the loan.

- ii) Loan Applicants

The loan applicants able to get the result of the loan application submitted faster compared to the old fashion loan approval process that done by financial institution employees manually in determine the documents submitted by loan applicants whether eligible or not.

iii) Financial Institution

The financial institution able to speed up the loan approval process significantly from manually to automated in digital form. The loan approval documents able to be managed in efficient way as the loan applicants 'documents store in digital form. The financial institution able to get more business opportunities and optimize revenue.

1.6 REPORT ORGANIZATION

This thesis consists of five chapters.

Chapters 1 explained the project introduction that includes the problem statement, the objectives, the scope, and significance of the project as well as the thesis organization.

Chapter 2 illustrates the literature review of three existing ML method by examining the advantages, disadvantages of the three existing ML methodology, as well as the comparison of these methodology benefits.

Chapter 3 discusses the methodology applied in this study which covered research framework, project requirement like input, output, process description, constraints and limitations. Besides, the proposed design also covered in this chapter like flowchart and

dataset description used in this study. The testing approach or strategy will cover in this chapter besides potential use of the system proposed.

Chapter 4 explains the implementation process of the different ML models such as RF (RF), DT (DT), and LR (LR) in python and testing and result discussion of the loan status on the Streamlit web application.

Chapter 5 briefly summarize the whole study and figure out the research constraints and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter covers on the classification method that has been widely used. As a direct outcome of this, research on the ML approaches will be conducted. The ML classifies into three primary categories which are supervised ML, unsupervised ML, and semi-supervised learning (Machine Learning, 2021). This study focusses in classification categories of supervised ML and research will be conducted.

Classification in ML is the supervised learning method that allow the computer program to categorize the set of data provided in structured or unstructured into classes (Waseem, M, 2022). The process like identifying, interpreting, and organizing concepts and objects into specified classes will be conducted based on the training data given in the classification model (5 Types of Classification Algorithms in Machine Learning, 2020). In the binary classification, there only two distinct class will be divided from a set of data as well as it can be applied to both structured and unorganized data (What Is Binary Classification, 2022). In binary classification algorithms, there are many methodologies such as RF, Support Vector Machine (SVM), Naïve Bayes and K Nearest Neighbour (Gong, 2022).

LR, RF and DT are three existing classification method in ML that have been investigated. The reasons choose these three ML classification methods are ease to use in solving the binary classification issues. These classification ML methods able to predict the discrete value into classes and it is easy to interpret as these algorithms can

smoothly handle qualitative target variables (Duggal, 2022). In this study, a literature review was conducted from the fundamental aspects of all methodologies, including the advantages and disadvantages of each approach.

2.2 EXISTING METHODS

In this section, an overview of current will be observed and delivered. This section discusses on the methodology and algorithm that has been proposed. In addition, this section provides the information on why, what, and how the current work behaved.

2.2.1 Logistic Regression

Joseph Berkson, the Statistician who initially introduced the LR in 1944. Although the LR model may be applied to any type of data, it is most commonly utilized with cross-sectional data. The LR model can be implemented when only two mutually incompatible outcomes exist for a categorical variable (LR, 2022). To anticipate the likelihood of a target variable, a supervised learning classification method known as LR is utilized. There are only two feasible classes since the nature of the goal or dependent variable is dichotomous (ML- LR, n.d.). In other words, the dependent variable is binary, with data represented by the values 1 or 0, with 1 denoting success and 0 denoting failure. $P(Y=1)$ as a function of X is mathematically predicted by a LR model. It is one of the most fundamental ML algorithms, and it may be used to a variety of classification issues, such as spam identification, and illness diagnosis (ML- LR, n.d.).

According to Firda Anindita Latifah, Isnandar Slamet and Sugiyanto research proposed, they make a comparison of the LR technique versus the RF algorithm for heart disease categorization. The data used for the LR are attained from the Kaggle website where the residents of the Framingham, Massachusetts (Latifah et al., 2020). There is total 3656 data obtained which consists of the patient has a 10-year chance of developing coronary heart disease with only one dependent variable “YES” or “NO” and 15 independent variables. The data is separated into training and testing segments in a 7:3

ratio. Based on the research, the accuracy of the training data calculated by using the LR is 85.04% (Latifah et al., 2020).

In the research which conducted by Ping Lin, Milton Soto-Ferrari and Odette Chams-Anturi, they utilized the data gathered by the Surveillance, Epidemiology, and End Results (SEER) program that contains 8602 patients from 2007 to 2012 in Atlanta. In the research, the LR algorithm was applied to assess radiation clinical route concordance in patients with early-stage breast cancer (Lin et al., 2022). To calculate radiation after surgery concordance, the data gathered is separated into 70% training data and 30% test dataset from 16 SEER data characteristics. The multivariate LR approach may be used to assess the mathematical correlation between the attribute-treatment route represented by radiation after surgical and no radiotherapy after surgery. The resulting confusion matrix had been developed to calculate the accuracy, sensitivity, and specificity.

The mathematical correlations with the attribute-treatment course expressed as radiation after surgery represent positive class and no radiotherapy after surgery represent negative class. The acceptable confidence level was chosen at 95% (p-value =0.05), indicating a statistically significant outcome. They also used the generated confusion matrix (CM) to calculate the model's performance where TP symbolizes true positives, TN denotes true negatives, FP symbolizes false positives, and FN symbolizes false negatives. The equations are shown below. It scored 81.82% in accuracy, 97.39% in sensitivity, and 48.29% in specificity (Lin et al., 2022).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.1)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (2.2)$$

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (2.3)$$

Based on Bernard X. W. Liew, Francisco M. Kovacs, David Rugamer, and Ana Royuela works, they conducted an observation study where 47 medical facilities have been chosen by the Spanish Back Pain Research Network. This observation research comprised a clinical database of 3001 people with neck discomfort. In this study, the total 3001 data obtained was split into a ratio 8:2 which training data set has 2402 and testing data set has 599. The data of the participants suffering from neck pain calculated by using LR has best performance for predicting which is 0.777 compared to Rain Forest and Xgboost.

2.2.2 Decision Tree

DTs are extensively used methodology approaches that are at the foundation of statistics, data mining, and ML. This appeal stems from their readability, simplicity, and superior outcomes. For the classification and regression issues, both can be solved by applying the DTs algorithm as DT is a supervised ML algorithm (Chow, 2022). In 1963, Morgan and Sonquist who invented the DT regression in their AID project by splitting the data collected into two subsets recursively. The THeta-Alpha-Input-Design (THAID) project conducted by Messenger and Mandell in 1972 produced the first classification tree by separating data to maximize the amount of the cases in the modal category (The Complete Guide to DT Analysis, 2019). DTs begin at the base of the tree and contrast the values of the root attribute with the record's value to predict a classifier. Then proceed to the next node based on the comparison.

There are some primary terms develop a complete DT. The Root Nodes are the node that begin of the DT where divide the population based on the characteristics. The

nodes that result from separating the root nodes are called Decision Nodes. Terminal Nodes are the nodes where further splitting is not feasible. A sub-section and small components of this DT is referred to as a sub-tree or branch. The complete DT graph is as shown in the Figure 2.1.

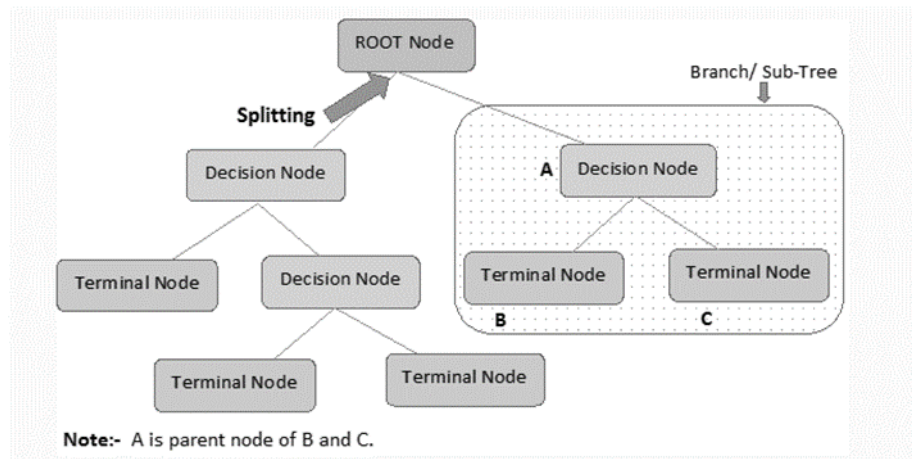


Figure 2.1 Steps of the DT

Vasiliki Matzavela and Efhimios Alepis conducted research titled DT learning using a prediction model for student academic accomplishment by utilizing four key processes: data collection, categorization, the development of a predictive model, and evaluation. There are 213 students' data were integrated during classification stage with the variables like sex, rating, parent education, parent salary, first child or not, and employed or not are the factors to consider. The DT-Quest technique was used in the development of the predictive model since it may use DT weights. Examination scores were used to evaluate each student's coursework performance, and grading is performed on a range of 0 to 100, with 50 scores required for passing. When assessments are customized to students' learning skills, they are more successful. The system generates a smart ML environment by allowing for individualization; it was widely evaluated by students, and the findings revealed a great rating while keeping a high degree of educational affordance. The comparison of the data obtained from the evaluation criteria and then from the algorithm's classification algorithm indicated that there is a link among student achievement and their personal characteristics.

DT was utilized in a study proposed by Casper Kaum, N.Z Jhanjh, Wei Wei Goh and Sanath Sukumaram. A pilot test was used to establish the needed training model and assess its performance using a sample size of more than 100. The DT algorithm applied to determine whether the knowledge is of high, medium, or low quality based on the knowledge quality attributes that are met. The flow chart of the DT algorithm as shown in Figure 2.2. The total of 8 data set which conducted in four different experimentations, they found out 3 data set which are dataset B, dataset C and dataset F with both accuracy index 1. As a result, the variable combination generated a 1.0 accuracy rate, proving that the discovered ML algorithms were acceptable for classifying knowledge quality (DTs for Decision Making, 2014). Based on the results of the pilot test, the DT method may be used to categorise knowledge inside a knowledge dense system and increase its performance.

To predict diabetes mellitus, three classification systems were used in the study which are RF, DT, and neural network in the study conducted by Quan Zou, Kaiyang Qu, Yamei Luo, Dehui, Ying Ju and Hua Tang. The total of 178,131 physical examination dataset collected from hospital Luzhou, China was split into two datasets. The 164,431 sample data selected as a training set while 13700 sample data selected as independent set. In this study, the J48 DT in Waikato Environment for Knowledge Analysis (WEKA) was chosen as the root node because it produces a branch for each probable attribute value, splits the instance into several subsets, each of which matches to a branch of the node n, and then continues the repetitive procedure on each branch. The algorithm stops when all cases have the same categorization. According to the research, the Luzhou dataset J48 DT has best result compared to RF and neural network as the accuracy is above 0.8084 (Zou et al., 2018).

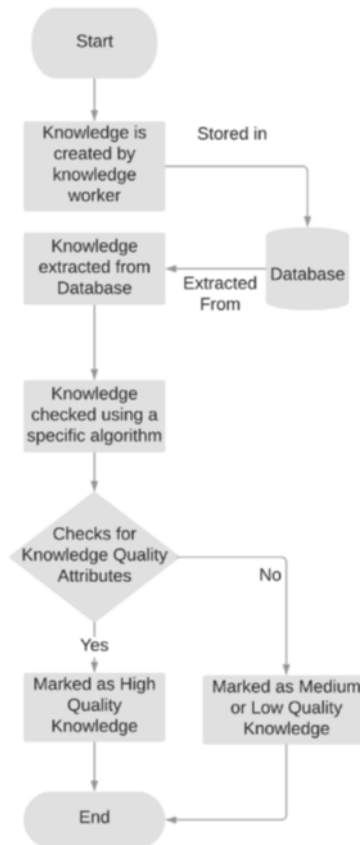


Figure 2. 2 Flowchart of DT Algorithm

2.2.3 Random Forest

Leo Breiman is the inventor of the RF algorithm. Leo Breiman is a statistician from University of California at Berkeley who invented the RF, ML approach in 2001. RF is a supervised ML strategy for dealing with complex classification and regression problems by combining multiple classifiers to improve model performance (Chaudhary, 2022). The RF Algorithm is simple to use. It is accomplished in 2 phases: the first includes integrating N classification trees with the construction of the RF, and the second includes making predictions for each tree formed in the first step. RF works on the Bagging principle of the ensemble methods by using replacements to establish a new training subset from sample training data, and the result is determined by majority of votes (R, 2022).

Numerous DTs are utilized in a RF approach. Decision nodes, leaf nodes, and a root node generate a complete DT. The leaf node of each tree represents the DT's final result. The final output is chosen using a majority-voting procedure. In this situation, the output determined by the majority of DTs becomes the RF 's Final Class. The classifiers of RF algorithm as shown below in Figure 2.3.

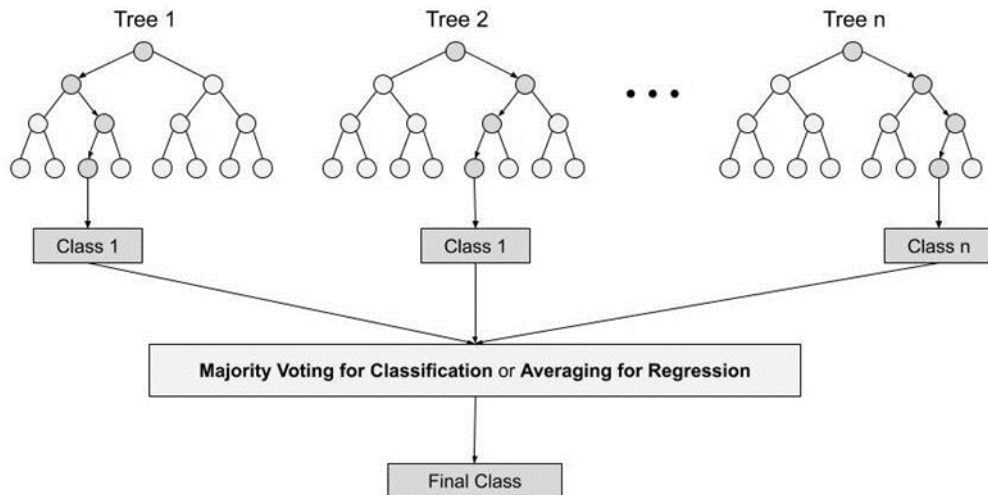


Figure 2.3 Steps of the RF algorithm

In Devika R, Sai Vaishnavi Avilala and V. Subramaniaswamy research, they utilized the public source dataset in their research. A total of 400 dataset samples were collected from various hospitals, community health clinics, and medical labs in order to perform research on classifier for chronic kidney disease prediction using three distinct approaches such as Nave Bayes, RF (RF), and K-Nearest Neighbors (KNN). The algorithms written in C sharp programming language are used to predict the chronic kidney disease as the accuracy that what number patients with chronic nephropathy square measure intervals at a specified time obtained. Based on the research, the presented of RF accuracy is better, and Naïve Bayes showed better precision (Ardekani et al., 2016). As a result, they discovered that the RF outperformed KNN and Nave Bayes in the prediction of chronic kidney disease in the study. Table 2.1 shows the result of Evaluation Metrics.

Table 2. 1 Results of Evaluation Metrics

Name of Classifier	Evaluation Parameter			
	Accuracy	Precision	Recall	F-measure
Naïve Bayes	99.635	1	0.996	0.998
RF	99.844	0.9985	0.99	0.99
K-Nearest Neighbors	87.78	0.879	0.877	0.8775

Source: (Ardekani et al., 2016).

The data collected for identifying neuroimaging data in Alzheimer's disease is obtained from four well-known web of science, according to Alessia Sarica, Antonio, and Aldo Quattrone were Pubmed, Scopus, Google Scholar, and Web of Science. For the RF classification applied in this research, three binary datasets were used: Alzheimer Disease (AD) vs. Healthy Controls (HC), Mild Cognitive Impairment (MCI) vs. Healthy Controls (HC), and stable Mild Cognitive Impairment (sMCI) vs. progressive Mild Cognitive Impairment (pMCI). A stratified repeated random sampling technique was used to evaluate the performance of each classifier by separating dataset into training set (75%) and testing set (25%) (Sarica et al., 2017). The mean of all 100 repetitions was used to calculate the accuracy on the testing set. Using 5,000 trees, The RF classifiers are trained individually on data instances of each of the four modalities and the feature significance ranking was derived. Based on the research the presented highest accuracy among three dataset is AD vs. HC as 87% (Sarica et al., 2017).

Based on Weiwei Lin, Ziming Wu, Longxin Lin, Angzhan Wen, and Jin Li research, The ensemble RF technique was utilized to forecast potential buyers using insurance business information from China Life Insurance Company. More than 500,000 clients purchase behaviors data were extracted and positive cases accounted for 20,787 of the total data, accounting for 4.1% (W. Lin et al., 2017). Analyzing the information gain and collecting the business information by identifying segmentation points for categorical variables or the threshold value for sequential values, as well as selecting one of the 16 possible user characteristics. Classification algorithms such as Support Vector

Machine (SVM) and LR are ineffective in the categorization of unbalanced distribution characteristic dataset. However, when the feature number approaches 16, the ensemble RF method performs better. Based on the investigation, the ensemble RF algorithm predicted 60831 potential clients, and 24% of them purchased the insurance, with a recall of up to 30.9% within a prediction interval of 60% to 90%. (W. Lin et al., 2017).

2.3 COMPARATIVE ANALYSIS OF EXISTING APPROACHES

The advantages of implementing a DT, which is relevant to both regression and classification issues DTs are effective for both regression and classification scenarios because they can forecast both discrete and continuous variables (Naik, 2021). DTs are simple to grasp, analyses, and depict since the data type can accommodate any form of data, whether numerical, category, or Boolean. Normally, data must be normalized before being fed into an algorithm for execution. However, in DT algorithm, continuous and categorical variables can co-exist in the algorithm as it does not rely on the inputs directly to predict an outcome (Duggal, 2022b). Instead, it is dependent on the relationship of the many inputs to anticipate what will happen next, which is the outcome.

In other words, decision-trees have a propensity to construct too complicated trees that may not generalized effectively. This is known as overfitting. Pruning, reducing the number of samples required at a leaf node, as well as the maximum depth of the tree, are all required to prevent these issues. Besides, outcome of predictions is neither smooth nor continuous, but rather piecewise constant approximations. As a result, DT is poor at extrapolation.

The LR algorithm is the ideal approaches when the connection between the independent and dependent variables is linear as it is less complexity compared to other algorithms (GeeksforGeeks, 2022). LR models are simpler to implement, analyze, and train.

In other words, the limitations of the LR method can only forecast discrete functions. As a result, in LR, the dependent variable is constrained to a discrete number set. Because it has a linear decision surface, LR cannot handle non-linear scenarios and real-world data is rarely linearly separable.

RF capable of doing implicit feature selection and producing uncorrelated DTs. It does this by populating each DT with a random set of attributes (Singh, 2021). Thus, it an ideal model for dealing with a variety of data properties. RF is capable of handling both linear and nonlinear relationships and provide high accuracy and balance the bias-variance trade-off well (Singh, 2021).

On the other hand, the RF has higher complexity compared to other classification algorithms. In comparison to DTs, RF creates a vast number of trees and then mixes their outputs. By default, the Python sklearn package generates 100 trees. This approach demands significantly need more processing power and resources to train the as compared to DTs and make decision (Kumar et al., n.d.). The comparisons of LR, DT, and RF are summarized in Table 2.2.

Table 2. 2 Comparisons of the Existing Method

Characteristics	LR	DT	RF
Problem Type	Regression and Classification	Regression and Classification	Regression and Classification
Classification Accuracy	Lower	Lower	Higher
Training Speed	Fast	Fast	Slow
Complexity	Simple	Simple	Moderate
Prediction Speed	Fast	Fast	Fast
Ease of Implementation	Easy	Middle	Hard

2.4 SUMMARY

This chapter has already covered the existing ML methods and their comparisons, which include the DT, LR, and RF. The LR method will be chosen among these three algorithms to classify the eligibility of the loan prediction who success to get the loan approval or not from the local financial institutions. Because of its ability to solve issues stochastically, LR can perform the tasks of classification and regression, constructs a tree-like structure by categorizing the cases. The results of loan eligibility are successful or unsuccessful. Data or information could include like loan applicants' name, gender, age, marital state, income, credit card history, and loan amount.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The classification of loan eligibility using LR was described in this chapter. This chapter starts with the research framework where it covers literature review, data collection, data pre-processing and applying algorithm and model. This chapter also include the project requirement which discuss the step of LR works, mathematical formulation used. The discussion of the proof of initial concept and hardware and software that used in this research also conducted in this chapter,

3.2 RESEARCH FRAMEWORK

This section describes the research framework of this study. Figure 3.1 shows the research framework. The research framework includes five sections which are Literature review, Collection of datasets, Data processing, Applying algorithm, and result. Next is the describe of each activity of each research framework in respective sub-chapter.

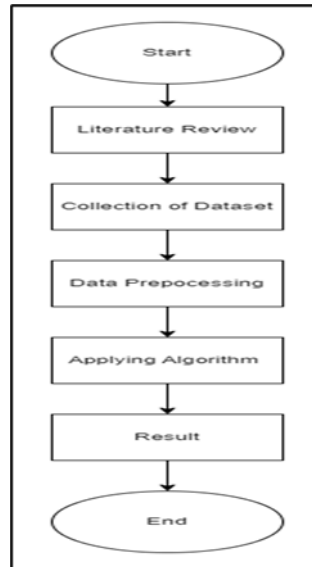


Figure 3. 1 Summary of phases in the research methodology

3.2.1 Literature Review

Credit line is the primary source of income of any financial institutions as they can earn interests of these approved loan. However, the prediction of loan eligibility is one of the complex and difficult tasks for any bank. This makes the research in loan eligibility become more important. This study provides a solution to automate loan approval process by implementation of ML algorithm. The data is collected from University of California Irvine ML (UCI ML) repository and Kaggle for prediction and studying.

The research on the classification methods has been conducted thoroughly and several methods have been reviewed such as LR, RF, and DT. Among of all the ML methods that have been mentioned, this research chooses LR in classify the loan eligibility. This is because the LR is the best method to achieve the objective study which is classify the loan eligibility (S., S., A., A., & M. Mohamed, R. 2021).

The classification of loan eligibility using LR was described in this chapter. This chapter starts with the methodology where it covers literature review, data collection, experimental design and testing and result. This chapter also include the discussion of hardware and software that used in this research,

3.2.2 Collection of Dataset

The dataset of the loan applicant information was taken from Kaggle, which is one of the most popular ML and data science hackathon platforms. Kaggle has over 50,000 public datasets that can used by researcher or learner to build the science projects. The datasets provided in the Kaggle website is suitable for the educational purpose. Thus, the two dataset which are related to the loan application was taken to classify the eligibility of loan application by using the python. Both of the datasets taken is for the model training purpose and model testing purpose.

When looking for the accessible Bank datasets, it was discovered that one of the datasets from the dream house finance company at Kaggle was regularly studied using ML methods. In this research, the focus is on the loan applicant's data that is believed to influence the loan approval. For example, the variables of loan applicant dataset are Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Applicant_Income, Coapplicant_Income, Loan_Amount, Loan_Amount_Term, Credit_History, Property_Area, and Loan_Status (Zaidan, 2022). The description of the variables of loan prediction dataset are show below in Table 3.1.

There were two bank-related data sets on Kaggle. All the dataset that being collected from the Kaggle that can be accessed online using the link of <https://www.kaggle.com/code/zaidandhman/prediciting-loan-default-probability>. The training dataset contains 614 data samples with 13 features, of which six were categorical and seven were numeric. The dataset of the training dataset is show below at Table 3.2. Additionally, there are approximately 367 data instances in the testing sample with 12 attributes, of which five were numeric features and seven were categorical variables

(Zaidan, 2022). The dataset of the testing dataset is show below at Table 3.3. Both of testing and training dataset was used for predicting whether the loan applicant applied by loan applicant will be successful or not.

Table 3. 1 Attributes of the Loan Prediction Dataset

Variables	Description
Loan_ID	A uniques loan ID
Gender	Male or Female
Married	Married (YES)/Not Married (NO)
Dependents	Number of persons depending on client
Education	Graduate/Undergraduate
Self_Employed	Self Employed (YES/NO)
Applicant_Income	Applicant Income
Coapplicant_Income	Coapplicant Income
Loan_Amount	Loan amount in Thousands (\$000)
Loan_Amount_Term	Term of loan in months
Credit_History	Credit history meets guidelines
Property_Area	Urban/Semi and Rural
Loan_Status	Loan Approval (YES/NO)

Source: (Zaidan, 2022).

Table 3. 2 Training Dataset of Loan Applicants

Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
LP001002	Male	No	0	Graduate	No	5849	0	154	360	1	Urban	Y
LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
LP001014	Male	Yes	3	Graduate	No	3036	2504	158	360	0	Semiurban	N
LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban	Y
LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y

Table 3.3 Second Dataset of Loan Applicants

Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
LP001015	Male	Yes	0	Graduate	No	5720	0	110	360	1	Urban
LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1	Urban
LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1	Urban
LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360	0	Urban
LP001051	Male	No	0	Not Graduate	No	3276	0	78	360	1	Urban
LP001054	Male	Yes	0	Not Graduate	Yes	2165	3422	152	360	1	Urban
LP001055	Female	No	1	Not Graduate	No	2226	0	59	360	1	Semiurban
LP001056	Male	Yes	2	Not Graduate	No	3881	0	147	360	0	Rural
LP001059	Male	Yes	2	Graduate	No	13633	0	280	240	1	Urban
LP001067	Male	No	0	Not Graduate	No	2400	2400	123	360	1	Semiurban
LP001078	Male	No	0	Not Graduate	No	3091	0	90	360	1	Urban
LP001082	Male	Yes	1	Graduate	No	2185	1516	162	360	1	Semiurban
LP001083	Male	No	3	Graduate	No	4166	0	40	180	0	Urban
LP001094	Male	Yes	2	Graduate	No	12173	0	166	360	0	Semiurban
LP001096	Female	No	0	Graduate	No	4666	0	124	360	1	Semiurban
LP001099	Male	No	1	Graduate	No	5667	0	131	360	1	Urban

3.2.3 Data Preprocessing

During the data preprocessing of the dataset taken from Kaggle, the data cleaning techniques was applied. The practice of correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or missing data from a dataset is known as data cleaning. For the missing data occurred in the loan applicant dataset, there are few ways to handle the missing data either ignore the tuples or fill in the missing values. For the best approaches to solve the missing data in the dataset, the approach by filling the missing values was chosen to deal with the issues.

The placement dataset for handling missing values using means was chosen instead of using mode and median. For example, the missing value of the attributes like loan amount, loan amount term, and credit history. The placement dataset for handling missing values using mode was chosen instead of using mean and median. For example, the missing value of the attributes like gender, married, dependents, and self-employed.

For the Exploratory Data Analysis, visualization of the distributions of the dataset was conducted by using the Seaborn. The graphical graph of the attributes of the datasets was generated by Seaborn by using Python. For example, attributes like gender, dependents, education, loan status, self-employed, total income, loan amount, loan amount term, and credit history. The details of the graph visualization of these attributes are shown in Figure 3.2 to Figure 3.11.

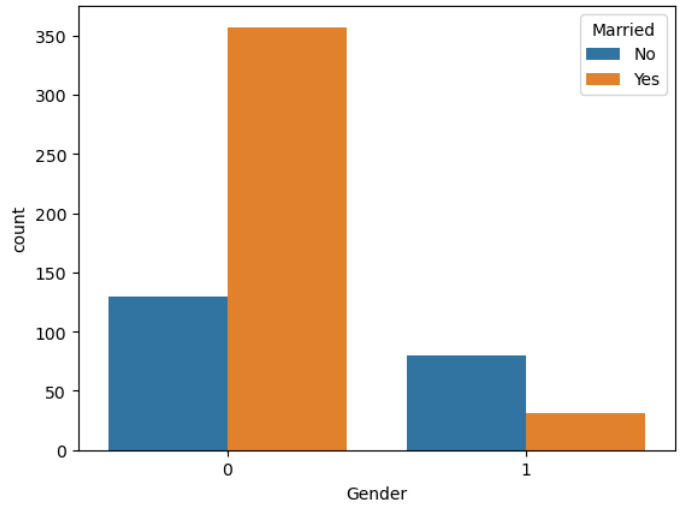


Figure 3.2 Histogram of the Gender

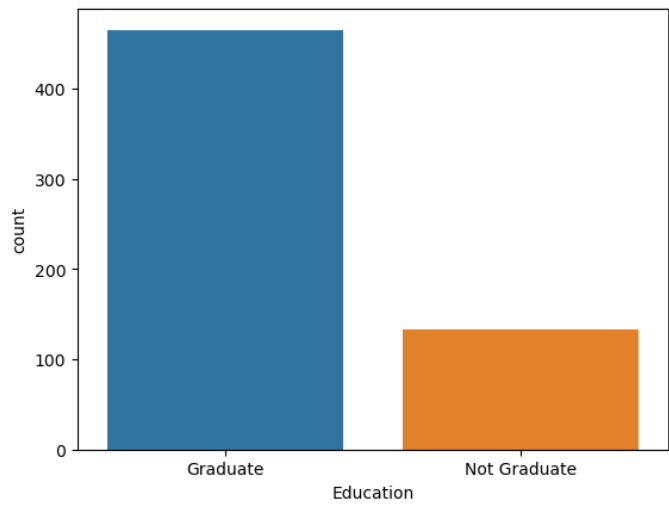


Figure 3.3 Histogram of the Education

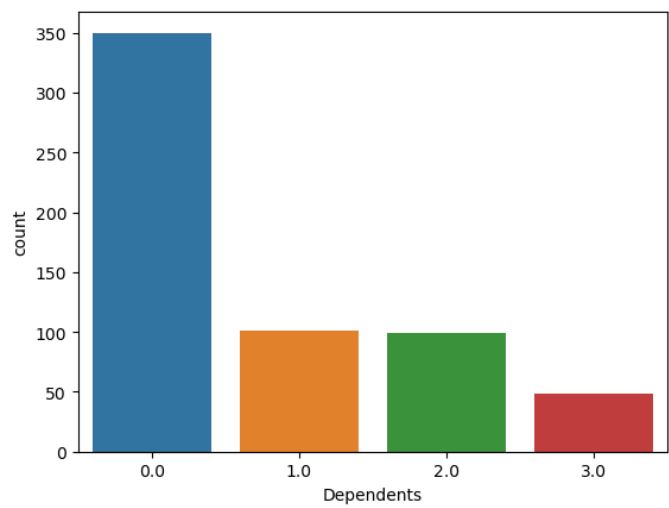


Figure 3.4 Histogram of the Dependents

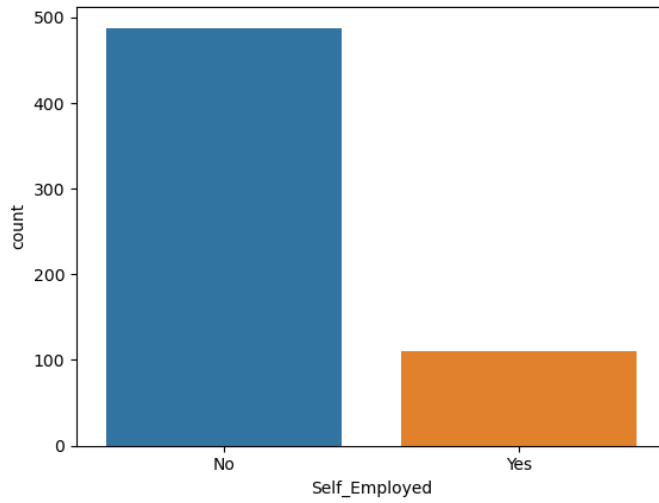


Figure 3.5 Histogram of the Self-Employed

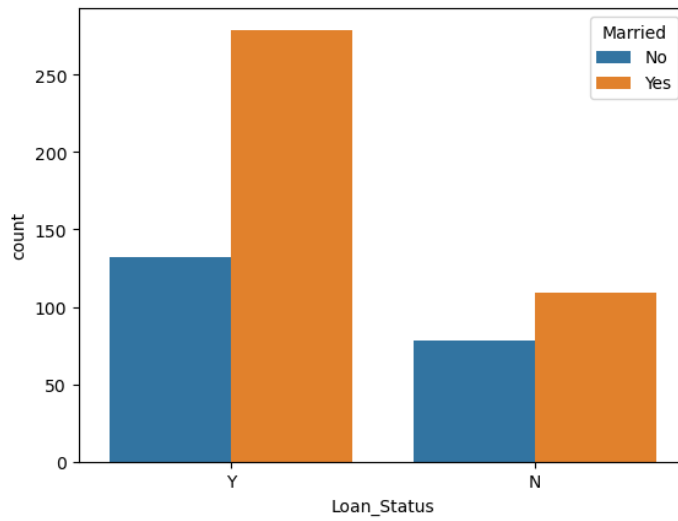


Figure 3.6 Histogram of the Loan Status

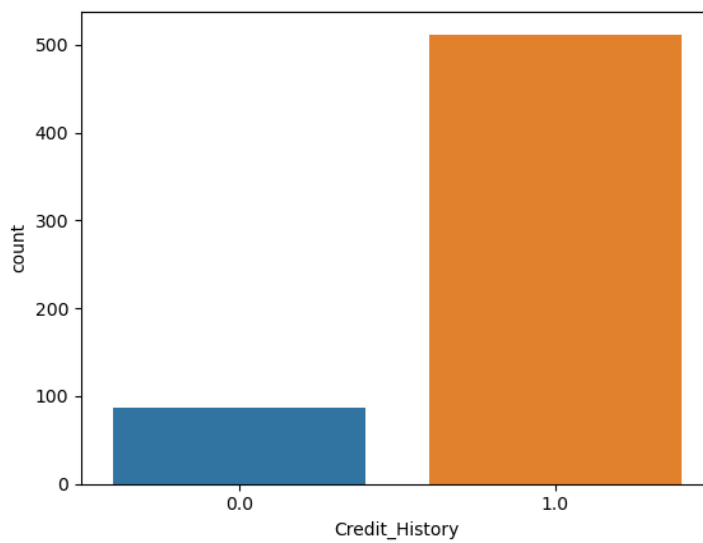


Figure 3.7 Histogram of the Credit History

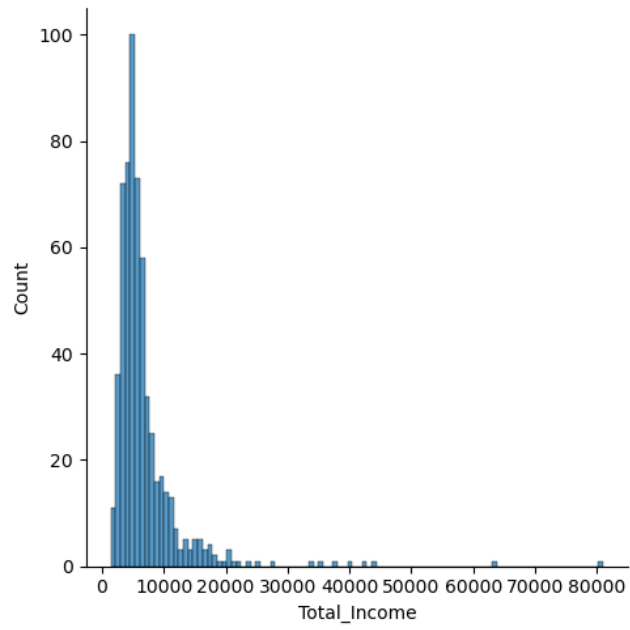


Figure 3. 8 Histogram of the Total Income

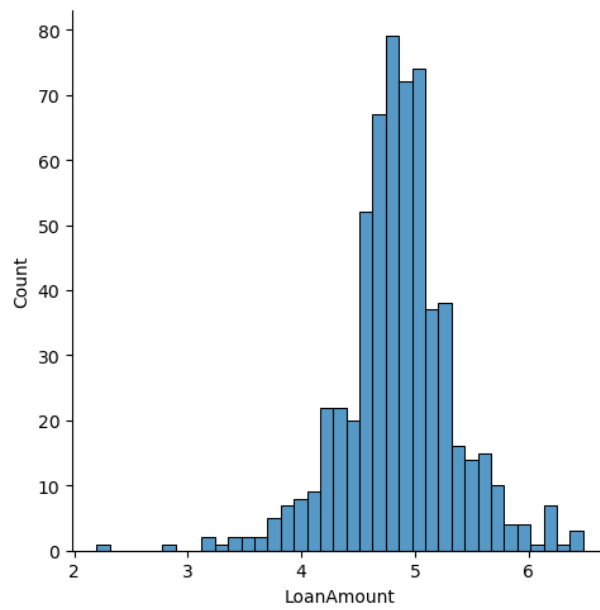


Figure 3. 9 Histogram of the Loan Amount

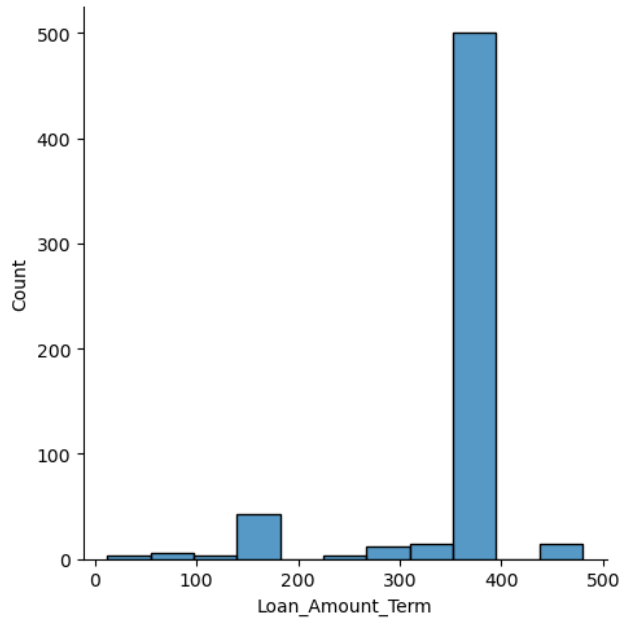


Figure 3.10 Histogram of the Loan Amount Term

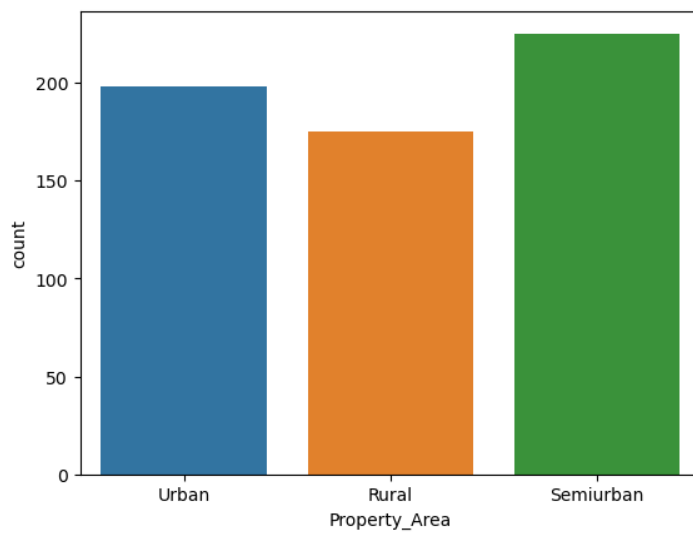


Figure 3.11 Histogram of the Property Area

3.2.4 Logistic Regression Algorithm

In this study, the ML algorithm which is LR was used to forecast the financial institution will issue the loan applicant a loan or not. To predict the loan eligibility, the LR using Python was implemented with the following steps shown in Figure 3.12[34]. The tools used for conducting this study is Kaggle and Jupyter Notebook. The following figures shows the flows of the process of LR by using the Python.

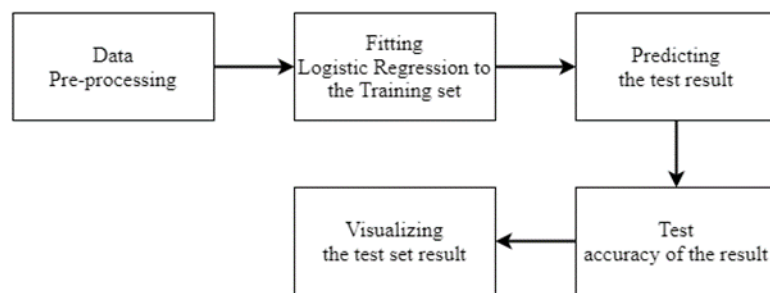


Figure 3. 12 Process of LR

The following shows the libraries imported and used in this study.

- i. pandas: for data manipulation and analysis
- ii. numpy: for numerical computation
- iii. matplotlib: for creating static, animated, and interactive visualizations.
- iv. seaborn: a data visualization library based on matplotlib, which provides a high-level interface for creating attractive and informative statistical graphics.
- v. plotly: a library for creating interactive, web-based visualizations.
- vi. Scikit-learn library: popular MLlibrary for Python that provides a wide range of tools for MLtasks
- vii. tabulate: provides a simple way to generate formatted tables from data in Python.

3.2.5 Result

With the comparison of the three ML algorithms which are RF, DT, and LR, the LR gave the best performance in term of accuracy, recall, precision and FI score among the RF and DT. The accuracy of the LR obtained was 0.83 which was the highest value compared to RF and DT. The figure of the outcome of the heatmap is shown in Figure 3.13 and Figure 3.14 shows the outcome of the correlation matrix.

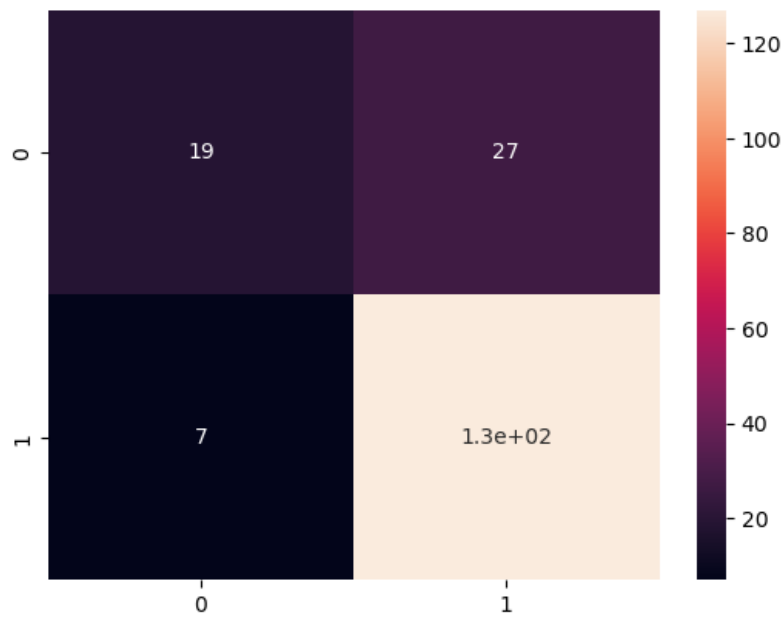


Figure 3. 13 Outcome of the Heat Map

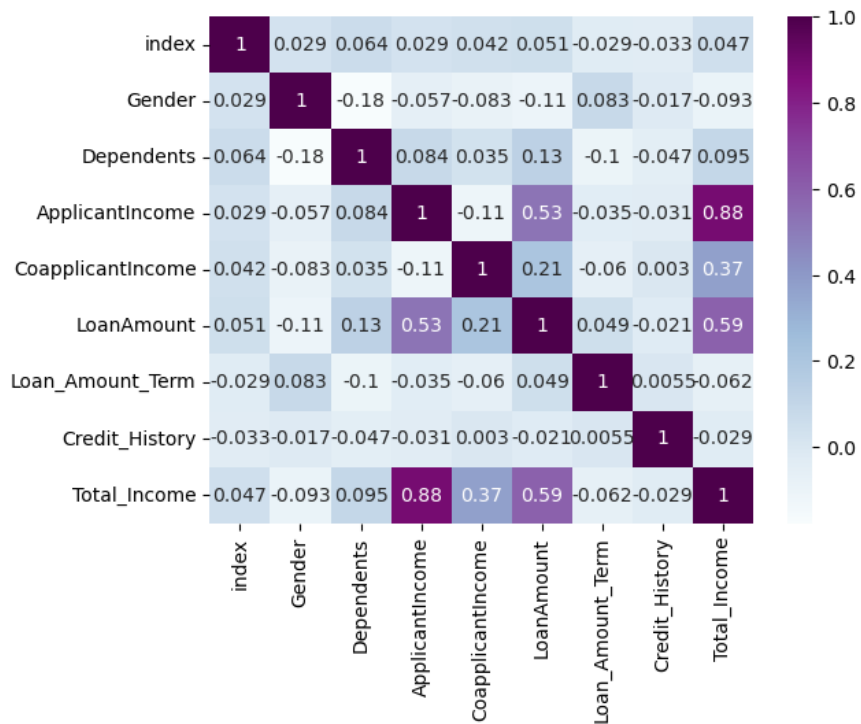


Figure 3. 14 Outcome of the Correlation Matrix

3.3 PROJECT REQUIREMENT

3.3.1 Logistic Regression

In this study, the loan eligibility classification is a classification problem which can be solved by using the ML algorithm, LR as it produces the probabilistic outcome likes ‘0’ or ‘1’ and ‘Success’ or ‘Fail’. The LR technique used to examine correlation between variables by using the ‘S’ shaped logistic function which called the Sigmoid function (Sonia Jessica, 2021). Sigmoid function transforms numerical data into an expression of probability between 0 and 1 as it assigns probabilities to discrete outcomes. Depending on whether the loan are successful issues by the bank or not, probability ranges from 0

and 1. In the binary classification, the population need to be divided into two groups with a cut-off of 0.5. Group A includes everything that is greater than 0.5, while another group includes everything that is less than 0.5. Figure 3.15 shows the sigmoid function of the LR algorithm.

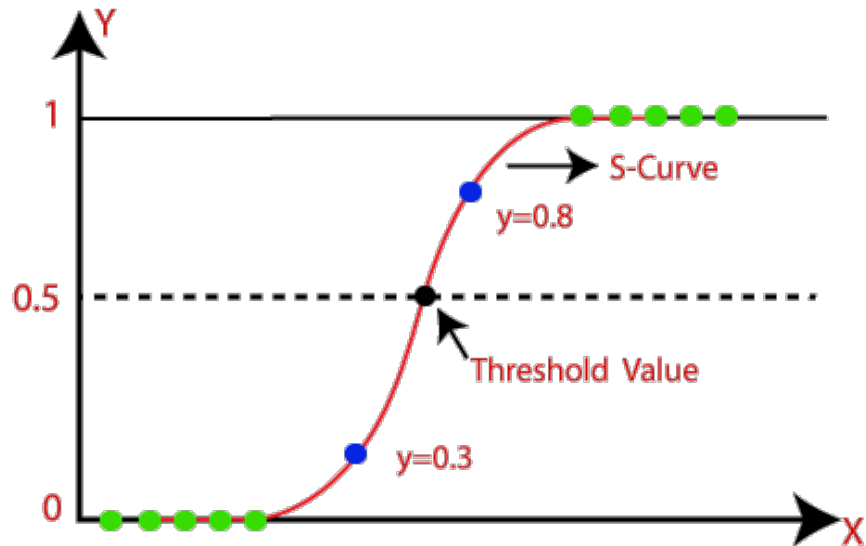


Figure 3. 15 The Sigmoid Function

The logistic function is the core of the LR method (Sonia Jessica, 2021). The logistic function can be expressed in mathematical form which is given by equation 3.3.

$$1/(1+e^{-value}) \quad (3.3)$$

when:

1. 'e' represents the base of the natural logarithms
2. *value* represents the resembles the actual numerical value that we want to transform using the sigmoid function

For the classification of the loan eligibility, the LR uses an equation as the representation. The coefficients or weights are used to mix the input numbers linearly for the equation to function. The output value modelled in linear regression as opposed to linear regression is a binary value '0' or '1'. The LR equation is shown by equation 3.4.

$$y = \frac{e^{(b_0 + b_1 * x)}}{1 + e^{(b_0 + b_1 * x)}} \quad (3.4)$$

where:

1. 'y' represents the predicted output
2. 'x' represents the input value
3. 'b0' represents the bias or intercept term
4. 'b1' represents the coefficient for input value

3.3.2 Input

The data of loan applicant needed to insert for the loan eligibility classification system which are Gender, Marital Status, Dependents, Education, Self Employed, Income, Coapplicant Income, Loan Amount, Loan Amount Term, Credit history, and Property Area. By using the data or information provided by loan applicant, the loan eligibility classification system able to ease the financial institution staff to speed up the loan approval process automatically instead of manually. The Table 3.4 shows the dataset of loan applicant needed in the loan eligibility classification system.

Table 3. 4 Dataset of the loan applicant

Variable	Description
Gender	Gender of the loan applicant.
Marital Status	Marital Status of the loan applicant. (YES/NO)
Dependents	Applicant has any dependents or not. (None/One/Two/More Than Two)
Education	Applicant is graduated or not. (Graduated/Not Graduated)
Self Employed	Applicant is self-employed or not. (YES/NO)
Applicant Income	Amount of the loan applicant's income. (In MYR)
Coapplicant Income	Amount of the loan coapplicant's income. (In MYR)
Loan Amount	Loan amount (In thousands, MYR)
Loan Amount Term	Duration of time a loan applicant expected to repay a loan (in Months)
Credit History	Record of a loan applicant. (0/1)
Property Area	Physical area or size of a piece of property or real estate. (Urban/Semi, Rural)

3.3.3 Output

The loan eligibility prediction will ease the old and traditional loan approval process that applied at local financial institution until today. The digital adoption applied in loan approval process by using ML brings great impacts to the financial industry. The financial institution able to speed up the loan approval by inserting the data or information needed that provided by loan applicant. Besides, the loan applicant can also use this loan eligibility system to predict the current loan information provided able to apply the loan successfully or not.

3.4 PROOF OF INITIAL CONCEPT

The flowchart in this study consists of four fundamental steps. The steps in the LR algorithm flowchart shown in Figure 3.16. The discussion of each step conducted thoroughly.

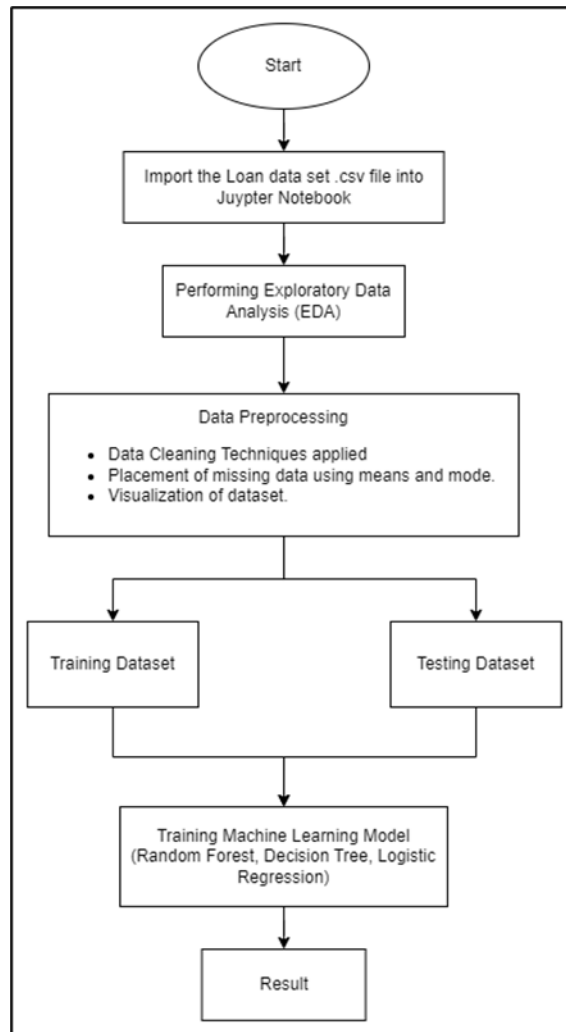


Figure 3. 16 Flowchart of LR Algorithm

The Figure 3.16 provides a visual representation of the steps involved in training a ML model. The first step is to download and obtain the training dataset in .csv file. After that, the Exploratory Data Analysis (EDA) performed to gain insights into the data and identify any patterns or relationships that may affect the performance of the model before the data pre-processing.

EDA involves visualizing and analysing the distribution of the variables, identifying any outliers or missing values, and examining the relationship between the variables and the outcome variable. Performing EDA helps to identify any issues with the data and make informed decisions on how to pre-process the data before fitting any ML models.

Additionally, during the data preprocessing stage, we also need to consider other factors such as feature scaling and feature selection. Feature scaling ensures that all features are on the same scale and prevents any one feature from dominating the others. Feature selection involves identifying and selecting the most important features for the model, which can help to reduce overfitting and improve the model's accuracy.

After performing Exploratory Data Analysis (EDA), the next step is to pre-process the data before fitting the ML model. The data preprocessing step involves cleaning the data, placement of missing data, and visualization the dataset to improve the accuracy and performance of the model. Data cleaning techniques applied in data preprocessing, missing values, anomalies, and outliers identified during EDA are handled by either dropping the affected rows, filling in the missing values, or using imputation methods to estimate the missing values.

Additionally, visualization of dataset is applied during this stage. It involves the use of graphs, charts, and other visual aids to help us better understand and analyse the data. Data visualization helps to identify patterns, trends, and relationships between variables that might not be immediately apparent from the raw data.

Next, the data is transformed into a suitable format for ML models by splitting it into two sets which are the 70% of training set and the 30% of testing set. The training

set is used to train the model, while the testing set is used to evaluate the performance of the model. After the model has been trained, it is tested on the testing set to evaluate its performance. The performance of the model is measured using various evaluation metrics such as accuracy, precision, recall, and F1-score, which are used to assess the effectiveness of the models in predicting the outcome variable.

At the end, a best ML which shows the best detection performance will be proposed.

3.5 POTENTIAL OF PROPOSED SOLUTION

Nowadays rapidly growing financial industry has high demands of the creative and new technologies to replace the old traditional method and technologies used in the daily business activities like loan approval. The creation of the new technology introduce in the financial industry will bring a lot of advantages to the user as it can eradicate the involvement of human resource and increase the job productivity compared to the old traditional method which is manually.

The employees of the financial institutions can boost the work efficiency by using the loan eligibility prediction system able to accelerate the loan application period and the loan applicant's waiting duration will be shortened. Besides, the committing fraud during the loan approval process can be prevented as the digital processing of the online documents submitted compared to the old method that required loan applicant submit the document needed in hard copy that may contains fake information.

In the future, this loan eligibility prediction module might be improved and incorporated as the current system is built on prior training data to predict the successful of loan approval or not, but in the future, it is possible to modify the program so that it may predict the actual loan amount that can be borrowed by the loan applicant of the local financial institution.

3.6 HARDWARE AND SOFTWARE

The hardware and software requirements are determined depending on what is required during the classification process for the loan applicant, whether he or she can get the loan approval or not. Tables 3.5 and 3.6 detail the software and hardware specifications, respectively.

Table 3.5 Hardware Descriptions and Specifications

Hardware Descriptions	Specifications
Laptop HP 15-ec1xxx Notebook	Importance:
<ul style="list-style-type: none"> Processor: AMD Ryzen 7 4800H with Radeon Graphics @ 2.90 GHz Memory: 16 GB RAM System Type: 64-bit operating system Dimension: 40 x 36 x 3 cm Weight: 2kg 	<ul style="list-style-type: none"> The notebook is essential to explore for research-related material. A laptop is essential to draught and revise a proper and official research report.
	Function:
	<ul style="list-style-type: none"> Act as a channel for internet search engine interaction.

Table 3.6 Software Description and Specification

Software	Specification
Microsoft Office Word 2019	<ul style="list-style-type: none"> It was used to write adequate report documentation. Tools for word processing operations such as composing, editing, formatting, and printing.
<ul style="list-style-type: none"> Graphical word processing program Developed by: Microsoft Version: 2019 	

<p>Microsoft Office Power Point 2019</p> <ul style="list-style-type: none"> • Presentation program • Developed by: Microsoft • Version: 2019 	<ul style="list-style-type: none"> • Used to create a presentation slide show. • Presentation slide design software.
<p>Microsoft Office Excel 2019</p> <ul style="list-style-type: none"> • Spreadsheet program • Developed by: Microsoft Office • Version: 2019 	<ul style="list-style-type: none"> • Used to import all the datasets of applicant data. • Contains tables of rows and columns that may preserve and organize all the datasets.
<p>Diagrams.net</p> <ul style="list-style-type: none"> • Cross-platform graph drawing software • Developed by: JGraph Ltd • Version: 20.6 	<ul style="list-style-type: none"> • Create diagrams such as flowcharts, and organizational charts.
<p>Scikit-learn</p> <ul style="list-style-type: none"> • Software MLLibrary • Developed by David Cournapeau • Version: 1.0 	<ul style="list-style-type: none"> • As a useful tool for data mining and data analysis of the loan eligibility classification.
<p>Jupyter Notebook</p> <ul style="list-style-type: none"> • Open-Source Web Application • Developed by Fernando Pérez and Brian Granger • Version: 5.0 	<ul style="list-style-type: none"> • Producing and distributing computational documents

3.7 SUMMARY

In conclusion, this chapter has covered the methodology of research implemented. First, it shows the flow how the research methodology conducted and followed by the project requirement. The project requirement covered the explain how the LR algorithm works as the ML algorithm used to classify the loan eligibility based on the information of loan applicant. The dataset used to conduct the LR algorithm in python was explained and how the dataset is preprocessing. The dataset's description is also stated ad the LR function in equation form also included to provide the better understanding of the LR algorithm works. The hardware and software required used in this project development covered in this chapter also.

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter is briefly discussing the Implementation Process, Testing, and Result Discussion of the research. This chapter also discuss the implementation of the basic step of the ML models such as RF, DT, and LR in designing the loan eligibility prediction system in Python.

4.2 IMPLEMENTATION PROCESS

The implementation is defined to meet the objectives that were stated to show the result is relevance to the research. ML method which is LR was used in this study where the dataset was divided into training and testing phase. About 70% from dataset was used for training and another 30% was used for testing purpose. Implementing LR, RF, and DT algorithms in Python by using the popular libraries such as scikit-learn or stats models. This chapter also discuss the implementation of the basic step of the LR in designing the loan eligibility prediction system.

4.2.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted in this study. There are several steps involved in EDA, such as check the data type of each variable, check the count of the missing values, identify the Loan_ID contains duplicate values, and calculate the total number of the missing values (D'Agostino, 2023). The Figure 4.1 shows the details of the steps of the EDA implemented in this study.

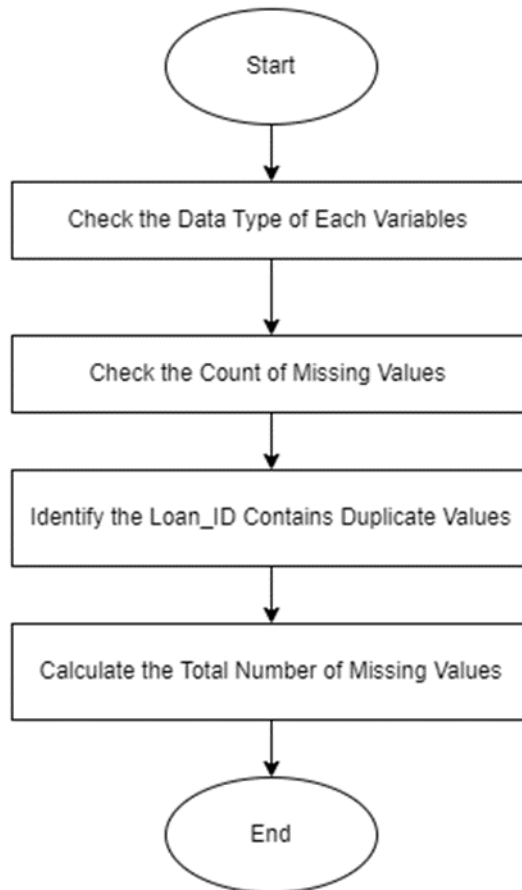


Figure 4. 1 Flowchart of the Exploratory Data Analysis

4.2.1.1 Check the Data Type of Each Variable

In this section, there are 13 variables in the dataset called 'data'. The 'data.info()' provides the details information about the dataset 'data' and this provides the better understanding of the data structure. It can use for the other purpose likes data cleaning, data preparation, and exploratory data analysis. The Figure 4.2 shows the data type of the variables in the dataset.

```
In [2]: # To get the information about the LoanApprovalPrediction.csv dataset.
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 598 entries, 0 to 597
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID                598 non-null    object
1   Gender                 598 non-null    object
2   Married                598 non-null    object
3   Dependents             586 non-null    float64
4   Education              598 non-null    object
5   Self_Employed          598 non-null    object
6   ApplicantIncome        598 non-null    int64
7   CoapplicantIncome      598 non-null    float64
8   LoanAmount             577 non-null    float64
9   Loan_Amount_Term       584 non-null    float64
10  Credit_History         549 non-null    float64
11  Property_Area          598 non-null    object
12  Loan_Status            598 non-null    object
dtypes: float64(5), int64(1), object(7)
memory usage: 60.9+ KB
```

Figure 4.2 Check Data Type of Variables

4.2.1.2 Check the Count of Missing Values

Null values are the blank values and will affect the performance to loan eligibility predication of ML models. In this section, there are four variables that have missing values which are Dependents, Loan Amount, Loan Amount Term, and Credit History. There are 12 null values in Dependents variables, 21 null values in Loan Amount, 14 null values in Loan Amount Term, and 49 null values in Credit History. Figure 4.3 illustrates the count of missing values in the dataset.

```
In [3]: # Data Preprocessing by checking the count of the missing values in the LoanApprovalPrediction.csv dataset.
data.isna().sum()

Out[3]: Loan_ID          0
Gender                0
Married               0
Dependents            12
Education             0
Self_Employed        0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            21
Loan_Amount_Term      14
Credit_History        49
Property_Area         0
Loan_Status           0
dtype: int64
```

Figure 4.3 Check Data Type of Variables

4.2.1.3 Identify the Loan_ID Contains Duplicate Values

Duplicate values in a dataset can lead to inaccurate results and bias in data analysis and modeling. It is crucial to identify and handle duplicates appropriately. The statement implies that there are 0 instances of duplicate records in the dataset 'data'. Figure 4.4 illustrates the identification of duplicate values in the Loan_ID variable.

```
In [5]: # Identify the Loan_ID contains Duplicate Values
data.duplicated().sum()
Out[5]: 0
```

Figure 4. 4 Identify the Loan_ID Contains Duplicate Values

4.2.1.4 Calculate the Total Number of Missing Values

In this section, a total of 96 missing values in dataset 'data' can affect the quality of the data analysis and loan eligibility prediction modeling results. Therefore, it is essential to identify and handle the missing values appropriately. Figure 4.5 shows the calculation of total number of missing values in the dataset.

```
In [6]: # Calculates the total number of missing values in a Pandas DataFrame named 'data'.
data.isna().sum().sum()
Out[6]: 96
```

Figure 4. 5 Calculate Total Number of Missing Values

4.2.2 Data Visualization

In this section, the data visualization technique applied in this study by representing DataFrame 'data' information and variables through visual means, such as charts, and graphs. It provides the better understanding of the complex data and able to identify patterns, trends, and relationships that may not be immediately apparent in raw data such as patterns and trends in loan approval and rejection.

Thus, data visualization plays a critical role in loan eligibility prediction, helping both lenders and borrowers to make more informed decisions and improve the efficiency and accuracy of the loan approval process.

4.2.2.1 Visualization of the Distribution of a Variable ‘Gender’

In this code, the data visualization of the distribution of a variable 'Gender' of the DataFrame ‘data’ was created by using the Seaborn library in Histogram form. The Figure 4.6 shows the count of individuals categorized by gender and marital status. From the dataset, there were 130 single males and 357 married males, as well as 80 single females and 31 married females.

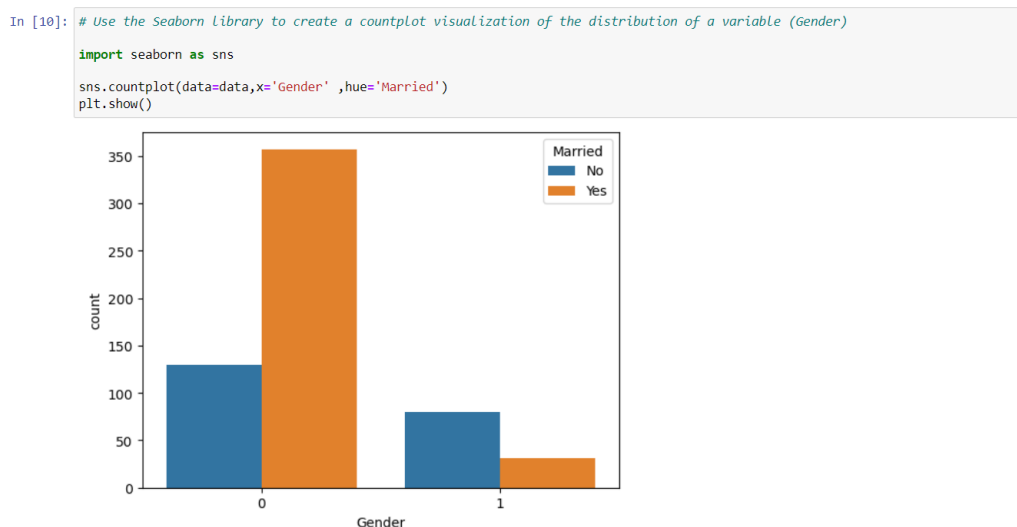


Figure 4. 6 Visualization of the Gender Variable

4.2.2.2 Visualization of the Distribution of Variable ‘Dependents’

In this code, the data visualization of the distribution of a variable 'Dependents' of the DataFrame ‘data’ was created by using the Seaborn library in Histogram form. Figure 4.7 illustrates the visualization of the Dependents variable. From the dataset, there

were 350 loan applicants have 0 dependents, 101 loan applicants have 1 dependent, 99 loan applicants have 2 dependents, and 48 loan applicants have 3 or more dependents.

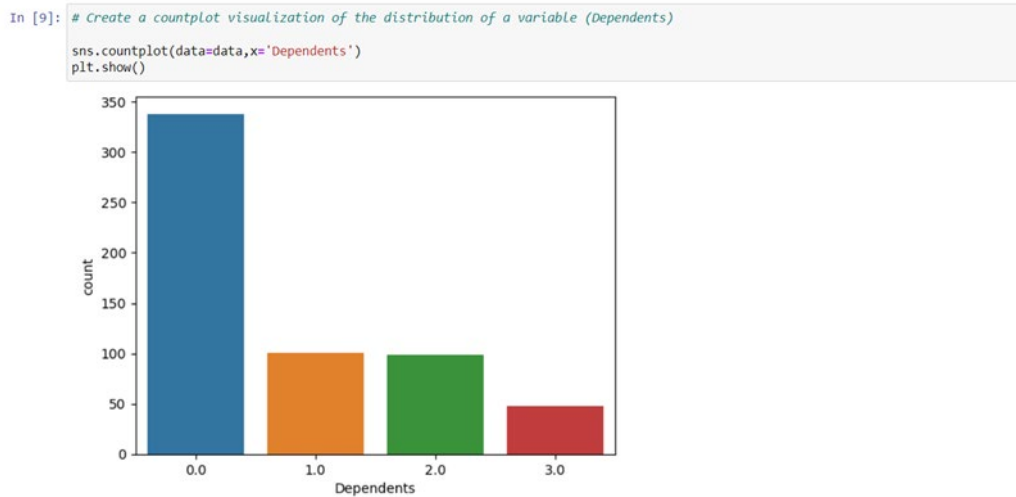


Figure 4. 7 Visualization of the Dependents Variable

4.2.2.3 Visualization of the Distribution of a Variable 'Education'

In this code, the data visualization of the distribution of a variable 'Education' of the DataFrame 'data' was created by using the Seaborn library in Histogram form. The Figure 4.8 shows that there are 465 individuals who are graduated and 133 individuals who are not graduated.

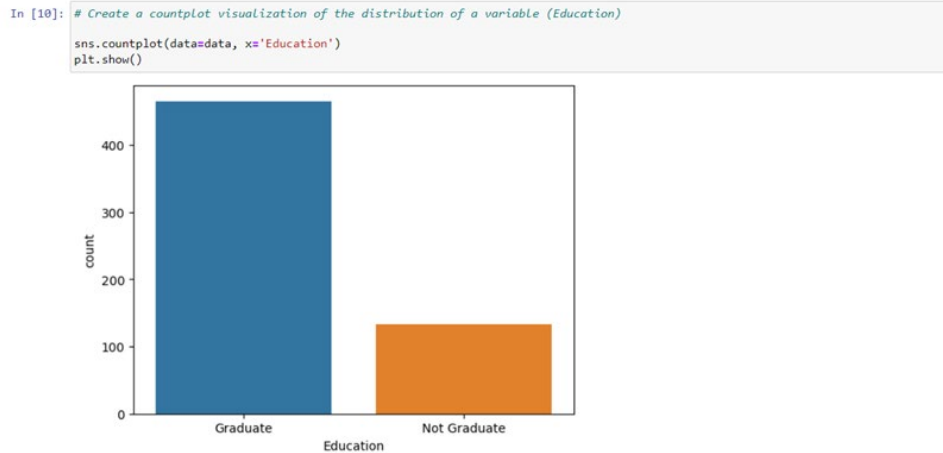


Figure 4. 8 Visualization of the Education Variable

4.2.2.4 Visualization of the Distribution of a Variable ‘Loan_Status’

In this code, the data visualization of the distribution of a variable 'Loan_Status' of the DataFrame ‘data’ was created by using the Seaborn library in Histogram form. Figure 4.9 shows that among single individuals, there are 78 instances where the loan status is not approved and 132 instances where the loan status is approved. Among married individuals, there are 109 instances of not approved loans and 279 instances of approved loans.

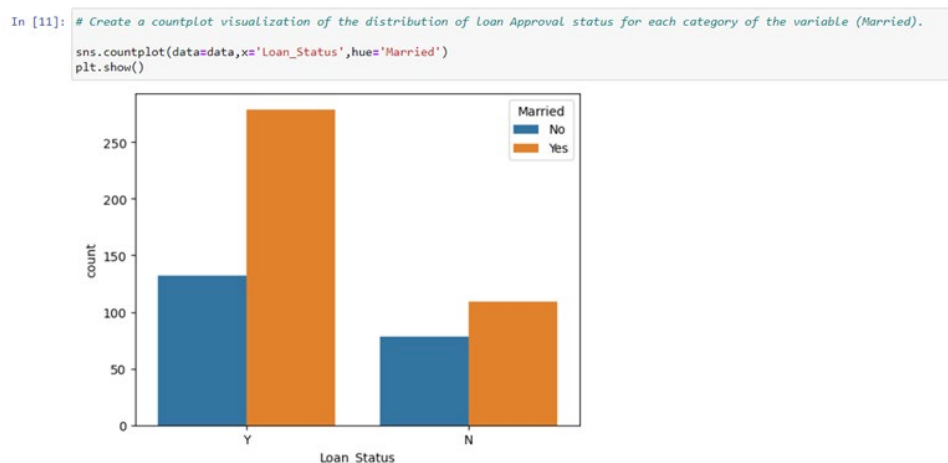


Figure 4. 9 Visualization of the Loan Status Variable

4.2.2.5 Visualization of the Distribution of a Variable ‘Self_Employed’

In this code, the data visualization of the distribution of a variable 'Self_Employed' of the DataFrame 'data' was created by using the Seaborn library in Histogram form. The Figure 4.10 shows that there are 488 loan applicants are not self-employed, and 110 instances where loan applicants are self-employed.

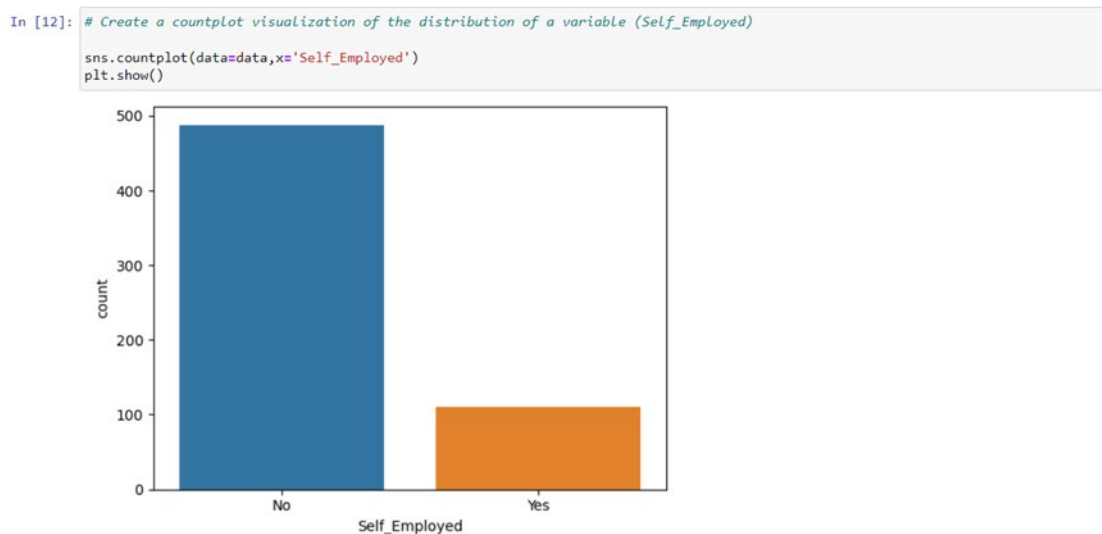


Figure 4. 10 Visualization of the Self-Employed Variable

4.2.2.6 Visualization of the Distribution of a Variable ‘Total_Income’

In this code, the data visualization of the distribution of a variable 'Total_Income' of the DataFrame 'data' was created by using the Seaborn library in Histogram form. Figure 4.11 illustrates a distribution of the variable 'Total_Income' that is skewed to the left. This indicates that the majority of loan applicants have lower incomes, with a potential presence of outliers on the higher income range.



Figure 4. 11 Visualization of the Total Income Variable

4.2.2.7 Visualization of the Distribution of a Variable ‘Total_Income’

In this code, the data visualization of the distribution of a variable 'Total_Income' of the DataFrame 'model_data' was created by using the Seaborn library in Histogram form. Figure 4. 12 shows the visualization of the Total Income Variable.

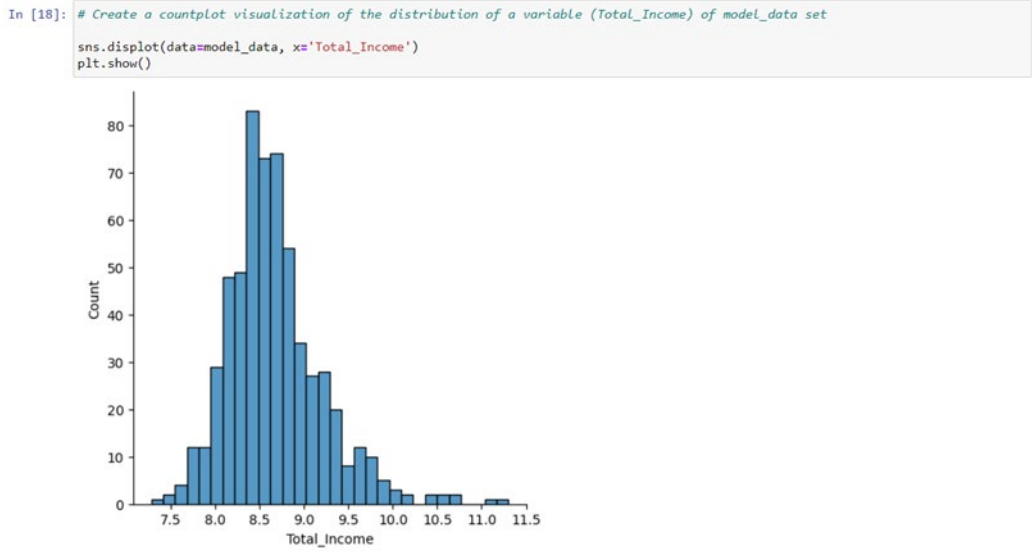


Figure 4. 12 Visualization of the Total Income Variable

4.2.2.8 Visualization of the Distribution of a Variable 'Loan_Amount'

In this code, the data visualization of the distribution of a variable 'LoanAmount' of the DataFrame 'model_data' was created by using the Seaborn library in Histogram form. Figure 4. 13 shows the visualization of the Loan Amount Variable.

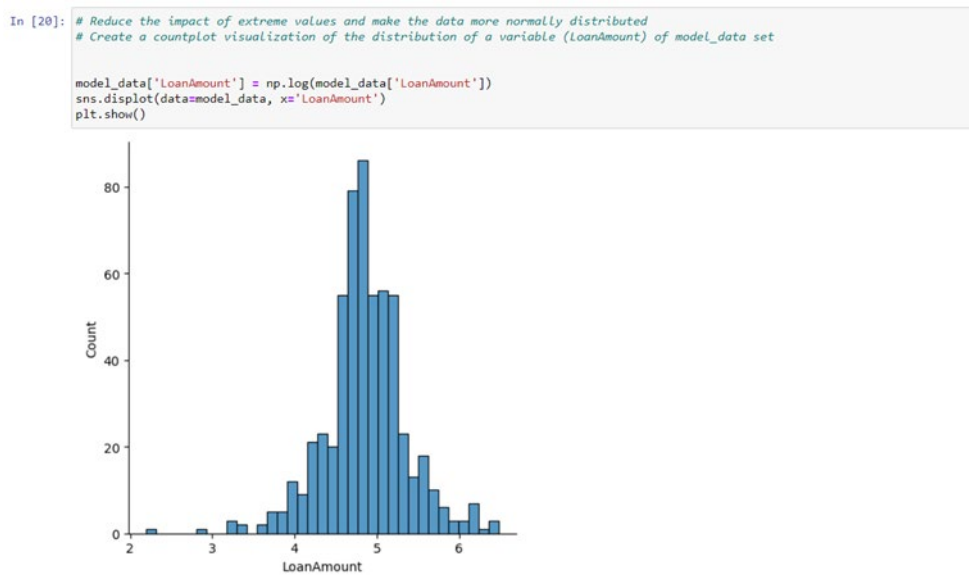


Figure 4. 13 Visualization of the Loan Amount Variable

4.2.2.9 Visualization of the Distribution of a Variable 'Loan_Amount_Term'

In this code, the data visualization of the distribution of a variable 'Loan_Amount_Term' of the DataFrame 'model_data' was created by using the Seaborn library in Histogram form. Figure 4.14 shows a visualization of the Loan Amount Term variable.

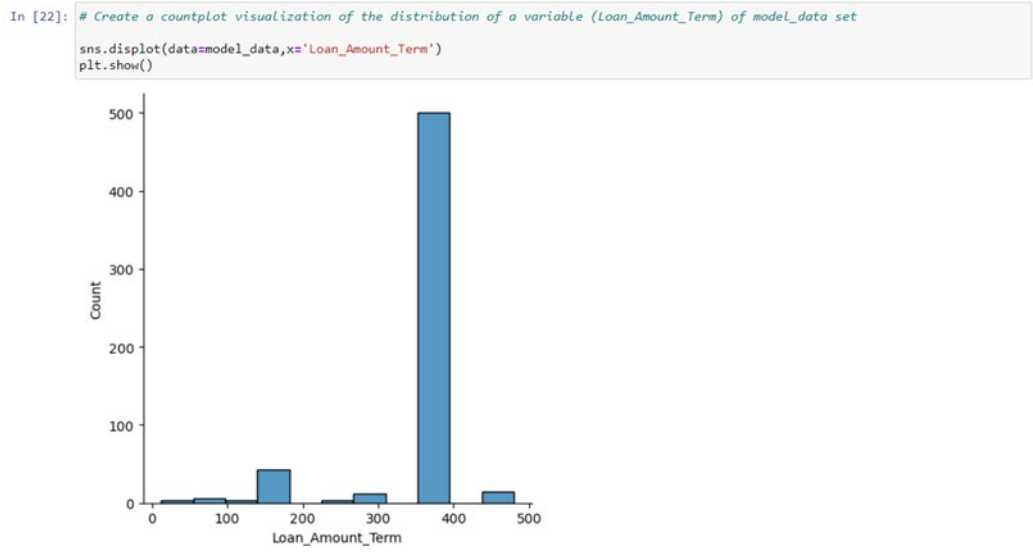


Figure 4. 14 Visualization of the Loan Amount Term Variable

4.2.2.10 Visualization of the Distribution of a Variable ‘Loan_Amount_Term’

In this code, the data visualization of the distribution of a variable 'Loan_Amount_Term' of the DataFrame 'model_data' was created by using the Seaborn library in Histogram form. Numpy's log function applied on the 'Loan_Amount_Term' column before creating a histogram is to reduce the effect of outliers in the data. The logarithmic transformation compresses large values into a smaller range, reducing the skewness of the data and making it more symmetric. Figure 4.15 visually represents the distribution of the 'Loan Amount Term' variable.

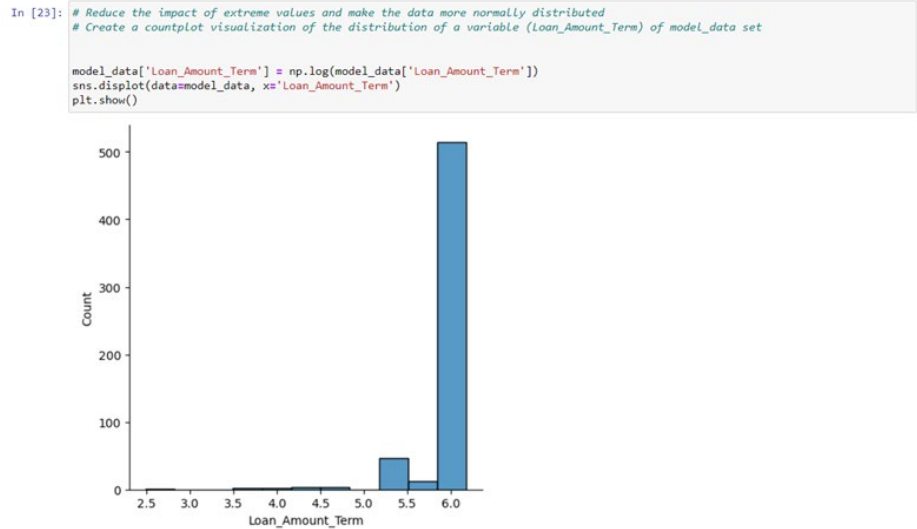


Figure 4. 15 Visualization of the Loan Amount Term Variable

4.2.2.11 Visualization of the Distribution of a Variable ‘Credit_History’

In this code, the data visualization of the distribution of a variable 'Credit_History' of the DataFrame ‘model_data’ was created by using the Seaborn library in Histogram form. Figure 4.16 illustrates that out of the 598 loan applicants, 86 individuals do not have credit history, whereas 512 applicants have credit history.

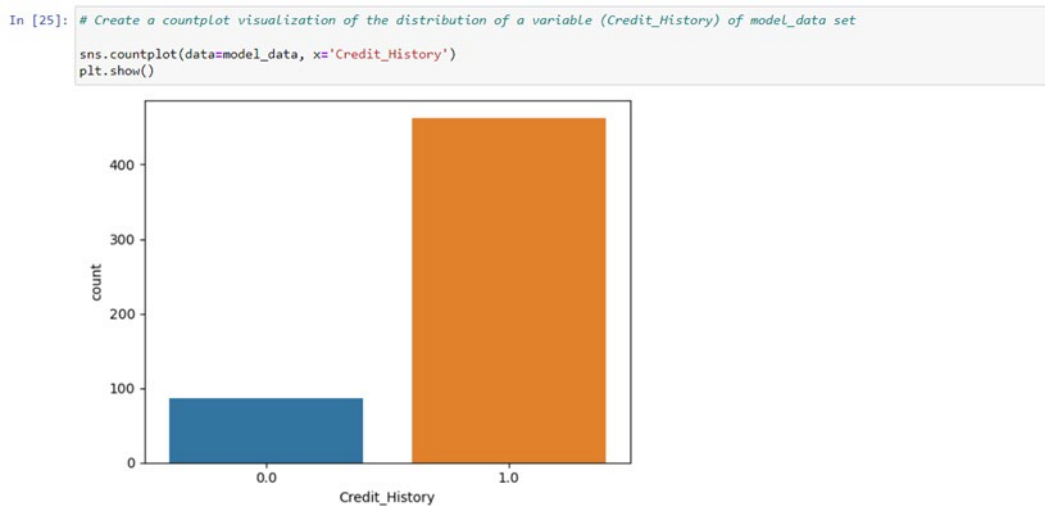


Figure 4. 16 Visualization of the Credit History Variable

4.2.2.12 Visualization of the Distribution of a Variable 'Property_Area'

In this code, the data visualization of the distribution of a variable 'Property_Area' of the DataFrame 'data' was created by using the Seaborn library in Histogram form. From the Figure 4.17 shows that there are 175 properties located in rural areas, 225 properties in semi-urban areas, and 198 properties in urban areas.

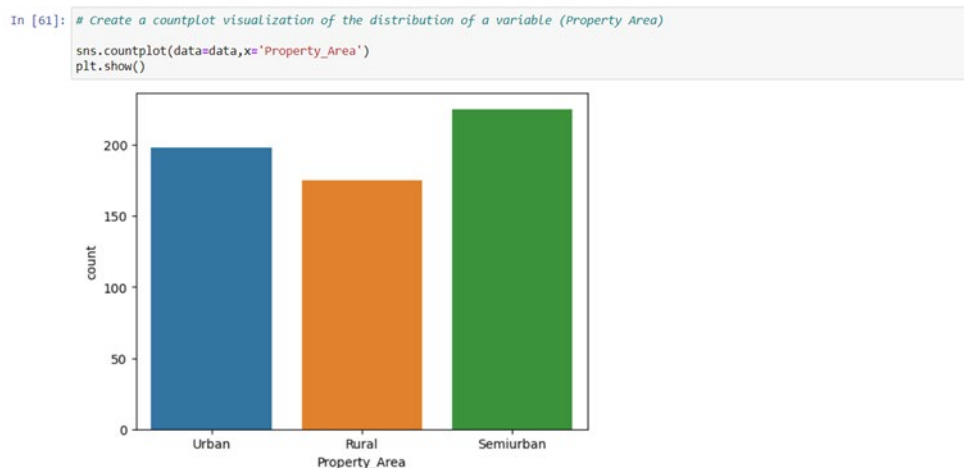


Figure 4. 17 Visualization of the Property Area Variable

4.2.3 Data Preprocessing

This section covers various techniques involved in data preprocessing, including feature selection, handling missing and nulls values, and duplicates, converting data to the appropriate type, label encoding, and others (GeeksforGeeks, 2023). A flowchart depicting the data preprocessing process is also provided. The aim of data preprocessing is to clean and transform data to make it suitable for analysis and ML tasks (Lawton, 2022). These techniques may vary depending on the type of data and analysis required. Figure 4.18 shows the steps of the data processing.

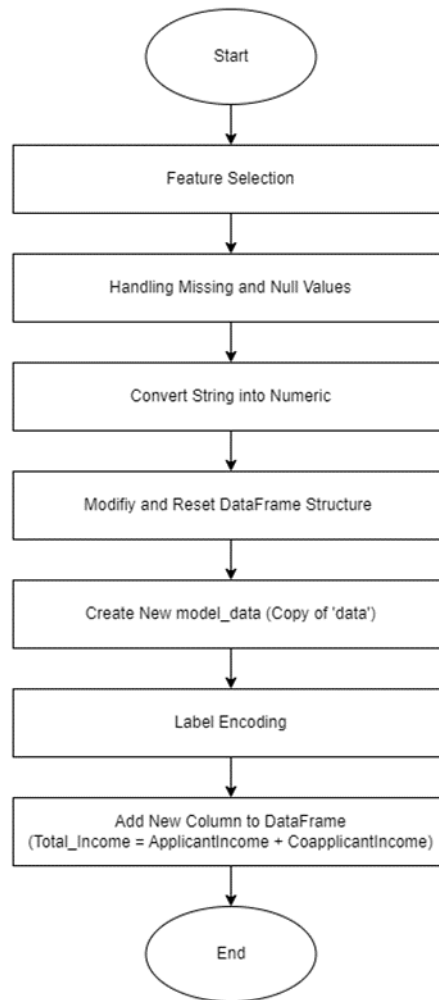


Figure 4. 18 Flowchart of Data Preprocessing

4.2.3.1 Feature Selection

Feature selection is a useful technique that can improve the detection performance of ML models by selecting only the most relevant and informative features of the dataset (Gupta, 2023). This can lead to better accuracy, reduced noise, and improved interpretability of the model. While dropping unnecessary columns is one way to perform feature selection. In this study, the Loan_ID column was dropped and removed from the 'data' DataFrame as it not be used in the model training. Figure 4.19 shows the feature selection of the dataset.


```
In [5]: # Drop and remove the Loan_ID column from the DataFrame
data.drop(['Loan_ID'], axis=1, inplace=True)
```

Figure 4. 19 Feature Selection

4.2.3.2 Handling Missing and Null Values

Missing or null values of the variables in a dataset can be problematic for ML algorithms, as they are unable to process or identify these values. To address missing and null values in the 'Dependents', 'LoanAmount', 'Loan_Amount_Term', and 'Credit_History' variables.

To deal with the missing and null values in this study, the method of imputation used to fill in the missing and null values with the mode, median and numeric. Imputation is one of the approaches to deal with missing or null values, and the choice of method will depend on the specific characteristics of the dataset and the modeling task (Kumar, 2021). For example, the missing values of categorical values filled with mode and missing values of numeric values filled with mean. Figure 4.20 shows the method to handle missing and null values in dataset.

```
In [8]: # Handling the Null Values of the variables

# Fill the categorical values with the mode
data['Dependents'] = data['Dependents'].fillna(data['Dependents'].mode()[0])

# Fill the missing values with the mean starting with Loan Amount
data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mean())

# Fill the missing values in Loan Amount Term
data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mean())

# Fill the missing values in Credit History
data['Credit_History'] = data.Credit_History.fillna(1)
```

Figure 4. 20 Handling Missing and Null Values

4.2.3.3 Convert String Value into Numeric Values

In this section, the 'data' has the Gender column which are Male and Female that both are categorical variables. For ML algorithms that require numerical inputs and create data visualization in this study, categorical variable needed to be converted into numerical variable. Thus, create a mapping dictionary with keys 'Male' and 'Female' that map to the values 0 and 1 respectively that executed in this study. Figure 4.21 demonstrates how to convert a string value into a numeric value for gender feature.

```
In [7]: #Snippet that maps the values of a column named "Gender" in a Pandas DataFrame named data from string values to numerical values.  
data.Gender = data.Gender.map({'Male': 0, 'Female':1})
```

Figure 4. 21 Convert String Value into Numeric Values

4.2.3.4 Modify and Reset the DataFrame Structure

In this section, the index of the DataFrame of 'data' was reset by assigning an index starting from 0 and incrementing by 1 for each row as the Loan_ID column had been removed and dropped. Figure 4.22 shows how to modify and reset the DataFrame structure of dataset.

```
In [13]: #Reset the index of the DataFrame 'Data'.  
data = data.reset_index()
```

Figure 4. 22 Modify and Reset the DataFrame Structure

4.2.3.5 Create New model_data dataframe

In this line of code, it creates a new DataFrame named model_data and assigns to it a copy of an existing variable data. Since the copy() method is used, any changes made to model_data will not affect the original data variable. This is useful when the

modifications happen on a dataset ‘model_data’ without altering the original dataset ‘data’. Figure 4.23 shows the method to create new model_data DataFrame in this study.

```
In [16]: # Create a new DataFrame that is a copy of the original one 'Data'.  
# In case the numeric values skewed - Log the data  
  
model_data = data.copy()
```

Figure 4. 23 Create New model_data DataFrame

4.2.3.6 Label Encoding

Label encoding is a technique that is commonly used to convert categorical variables into a numerical format that can be read by ML algorithms. This is because most of the ML algorithms are not able to process with the categorical data directly (Saxena, 2022).

Thus, by using label encoding which allow each category in a categorical variable is assigned a unique numerical code, allowing the algorithm to process the data more easily. It is important to note that label encoding does not increase the dimension of the dataset, as each category is represented by a single numerical value. Figure 4.24 shows the label encoding method applied in Python.

```
In [27]: # Categorical variables in the DataFrame will have been replaced with integer codes  
  
from sklearn.preprocessing import LabelEncoder  
label_encoder = LabelEncoder()  
  
obj = (data.dtypes == 'object')  
for col in list(obj[obj].index):  
    data[col] = label_encoder.fit_transform(data[col])
```

Figure 4. 24 Label Encoding

4.2.3.7 Adding New Column

In this section, the new column called Total_Income was added into DataFrame ‘data’. The values in ‘Total_Income’ column is calculated by adding the ‘ApplicantIncome’ column and the ‘CoapplicantIncome’ column together. Total_Income

may be a useful feature for predicting loan eligibility or other financial outcomes. This is because the total income of loan applicants and coapplicant can be a better predictor of loan eligibility outcomes than the individual incomes of each applicant. Thus, the MLmodels applied may be able to make more accurate predictions. Figure 4.25 shows the method that add the new column into the dataset.

```
In [14]: #Add a new column to DataFrame 'Data' called 'Total_Income', which is calculated by summing the 'ApplicantIncome' and 'CoapplicantIncome'
data['Total_Income'] = data['ApplicantIncome'] + data['CoapplicantIncome']
```

Figure 4. 25 Adding New Column

4.3 TESTING

4.3.1 Training and Testing Data Ratio

In order to train and test a model, a DataFrame ‘data’ will be split of 70:30, where 70% of the data being used for training purpose and 30% for testing purpose. This means that out of a total dataset of 598 data points, 418 are used for training and 180 are used for testing. Figure 4.26 shows the data splitting method by splitting the data into two set for training and testing purpose.

```
In [32]: # Splits the data into a training set and a testing set and 30% of the data will be used for testing and 70% will be used for training

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=7)

print("Number of Original Data:", model_data.shape[0])
print("Number of Original Variable Data:", model_data.shape[1])
print("\n")

print("Training Data:", x_train.shape[0])
print("Variables of Training Data:", x_train.shape[1])

print("\n")
print("Testing Data:", x_test.shape[0])
print("Variables of Testing Data:", x_test.shape[1])
```

Number of Original Data: 598
Number of Original Variable Data: 14

Training Data: 418
Variables of Training Data: 13

Testing Data: 180
Variables of Testing Data: 13

Figure 4. 26 Training and Testing Data Ratio

4.3.2 Training Machine Learning Models

Figure 4.27 shows the different ML model applied in this study such as RF algorithm model, DT algorithm model, and LR algorithm model. The result of performance metrics of ML models such as accuracy, F1 Score, Precision, and Recall is shown in Figure 4.28.

```
In [31]: #Trains and evaluates multiple machine Learning models including Logistic Regression, Decision Tree Classifier, and Random Forest Classifier

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

models = []
models.append(('Logistic Regression', LogisticRegression(max_iter=1000)))
models.append(('Decision Tree Classifier', DecisionTreeClassifier()))
models.append(('Random Forest Classifier', RandomForestClassifier()))

def modeling(model):
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    return accuracy_score(y_test, y_pred) * 100

for name, model in models:
    print(f'{name} model accuracy is {modeling(model)}')
```

Logistic Regression model accuracy is 81.66666666666667
Decision Tree Classifier model accuracy is 73.33333333333333
Random Forest Classifier model accuracy is 81.11111111111111

Figure 4. 27 Training ML Models

```

# Evaluate the machine learning models
results = []
for name, model in models:
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    acc = accuracy_score(y_test, y_pred) * 100
    f1 = f1_score(y_test, y_pred, average='weighted') * 100
    prec = precision_score(y_test, y_pred, average='weighted') * 100
    rec = recall_score(y_test, y_pred, average='weighted') * 100
    results.append([name, acc, f1, prec, rec])

# Print the results in a table
headers = ['Model', 'Accuracy', 'F1 Score', 'Precision', 'Recall']
table = tabulate(results, headers=headers, tablefmt='orgtbl')
print(table)

```

Model	Accuracy	F1 Score	Precision	Recall
Logistic Regression	81.6667	79.3356	81.1677	81.6667
Decision Tree Classifier	72.7778	73.6488	75.0502	72.7778
Random Forest Classifier	81.1111	79.1435	80.0677	81.1111

Figure 4. 28 Result of Performance Metrics

4.3.2.1 Random Forest (RF)

This model is imported from sklearn RandomForestClassifier_module. The coding is shown as below. The model and coding implemented are used to build and train the RF model. Figure 4.29 shows the RF model that selected for training model.

```

from sklearn.ensemble import RandomForestClassifier

# Divide into training and testing data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=7)

# Define the model
model = RandomForestClassifier()

# Fit the model on the training data
model.fit(x_train, y_train)

# Save the train model
with open('train_model.pkl', mode='wb') as pkl:
    pickle.dump(model, pkl)

```

Figure 4. 29 Training RF Model

4.3.2.2 Decision Tree (DT)

This model is imported from sklearn DecisionTreeClassifier_module. The coding is shown as below. The model and coding implemented are used to build and train the DT model. Figure 4.30 shows the DT model that selected for training model.

```
from sklearn.tree import DecisionTreeClassifier
```

```
# Divide into training and testing data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=7)

# Define the model
model = DecisionTreeClassifier()

# Fit the model on the training data
model.fit(x_train, y_train)

# Save the train model
with open('train_model.pkl', mode='wb') as pkl:
    pickle.dump(model, pkl)
```

Figure 4. 30 Training DT Model

4.3.2.3 Logistic Regression (LR)

This model is imported from sklearn LogisticRegression_module. The coding is shown as below. The model and coding implemented are used to build and train the LR model. Figure 4.31 shows the LR model that selected for training model.

```
from sklearn.linear_model import LogisticRegression
```

```
# Divide into training and testing data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=7)

# Define the model
model = LogisticRegression(max_iter=1000)

# Fit the model on the training data
model.fit(x_train, y_train)

# Save the train model
with open('train_model.pkl', mode='wb') as pkl:
    pickle.dump(model, pkl)
```

Figure 4. 31 Training LR Model

4.4 ML MODEL PROPOSED

This python script ‘model.py’ used for training the different ML models such as RF, DT, and LR on the LoanApprovalPrediction.csv dataset. This python preprocesses the data, splits it into training and testing sets, fits the model on the training data according to the ML model selected, and saves the trained model using pickle. The saved model in pickle file can be loaded easily later for making predictions on new data. The ‘model.py’ python script is shown below in Figure 4.32.


```
1 from sklearn.preprocessing import LabelEncoder
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.metrics import accuracy_score
5 import pickle
6 import pandas as pd
7
8
9 # Load the data
10 data = pd.read_csv('LoanApprovalPrediction.csv')
11
12
13 # Drop Loan_ID column
14 data.drop(['Loan_ID'], axis=1, inplace=True)
15
16
17 # Convert to int datatype
18 label_encoder = LabelEncoder()
19 obj = (data.dtypes == 'object')
20 for col in list(obj[obj].index):
21     data[col] = label_encoder.fit_transform(data[col])
22
23
24 # Fill in missing rows
25 for col in data.columns:
26     data[col] = data[col].fillna(data[col].mean())
27
28
29 # Drop the column 'Loan_Status' from an 'data' DataFrame
30 x = data.drop(['Loan_Status'], axis=1)
31 y = data.Loan_Status
32
33
34 # Divide into training and testing data
35 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=7)
36
37 # Define the model
38 model = LogisticRegression(max_iter=1000)
39
40 # Fit the model on the training data
41 model.fit(x_train, y_train)
42
43 # Save the train model
44 with open('train_model.pkl', mode='wb') as pkl:
45     pickle.dump(model, pkl)
46
47
```

Figure 4. 32 model.py

4.5 DISCUSSION

In this study, the LR model will be used in develop the interactive and highly customizable web applications in Python with Streamlit which is a wildely-used open-source framework. The objective of the web application is to leverage this framework to deploy a LR model that can predict loan eligibility.

For first step involves training the LR model on an appropriate dataset for loan eligibility prediction. This includes pre-processing and feature engineering to ensure that

the model is based on relevant and significant features. Subsequently, it will integrate the model with the Streamlit framework to establish a web application that users can engage with.

The Streamlit application will have an intuitive interface that allows users to enter their information, such as income, credit score, and other pertinent data. The model will then use the information provided by the user, in addition to the weights and biases of the trained LR model, to generate a prediction of their loan eligibility.

The main advantage of using Streamlit is that it provides readily available UI components to design the web application's elements, such as sliders, buttons, and dropdown menus. This simplifies the development process and enables us to create a more interactive and user-friendly web application.

To sum up, Streamlit used in Python to deploy a LR model for loan eligibility prediction. By integrating the model with Streamlit, it able to create an interactive and user-friendly web application that enables users to input their data and receive a prediction result of their loan eligibility whether approve or not. The following Figure 4.33 shows the interface of Loan Eligibility Prediction system in Streamlit.



The image shows a web application interface for loan eligibility prediction. At the top left is an illustration of a person jumping over a stack of gold coins. To the right of this is the title "Loan Eligibility Prediction" in a large, bold, black font. Below the title is a form with several input fields:

- Gender:** A dropdown menu with "Male" selected.
- Married:** A dropdown menu with "Yes" selected.
- Dependents:** A dropdown menu with "None" selected.
- Education:** A dropdown menu with "Graduate" selected.
- Self-Employed:** A dropdown menu with "Yes" selected.
- Applicant Income:** A numeric input field with "0" and minus/plus buttons.
- Coapplicant Income:** A numeric input field with "0" and minus/plus buttons.
- Loan Amount (Thousands):** A numeric input field with "0" and minus/plus buttons.
- Loan Tenor (Months):** A numeric input field with "0" and minus/plus buttons.
- Credit History:** A numeric input field with "0" and minus/plus buttons.
- Property Area:** A dropdown menu with "Semiurban" selected.

At the bottom of the form is a "Predict" button.

Figure 4. 33 Interface of Loan Eligibility Prediction Web Application

4.6 RESULT

4.6.1 Comparison Performance Result of Proposed ML Models

In this study, the performance of three different ML models for predicting loan eligibility was compared which were RF, DT, and LR. The models were evaluated using four different performance metrics which were accuracy, recall, precision, and F1-score. The accuracy, precision, recall and F1 score of the three ML method are shown below in Table 4.1.

Table 4. 1 Comparison Result of Performance Metric

Method	Accuracy	Recall	Precision	F1 Score
RF	0.80	0.81	0.79	0.78
DT	0.71	0.72	0.74	0.72
LR	0.82	0.82	0.81	0.79

For loan eligibility prediction, accuracy is an important metric to consider because it measures the overall effectiveness of the model in correctly predicting loan eligibility. In this study, the LR model has the highest accuracy score of 0.82, indicating that it correctly predicts loan eligibility for 82% of the applicants in the dataset. This means that out of all the loan applications, 82% of them are accurately classified as either eligible or not eligible by the model. The RF model has an accuracy score of 0.80, indicating that it correctly predicts loan eligibility for 80% of the applicants in the dataset. However, the DT model has the lowest accuracy score of 0.71, indicating that it correctly predicts loan eligibility for only 71% of the applicants in the dataset.

Besides, recall is also an important metric in loan eligibility prediction, as it measures the ability of the model to correctly identify all eligible individuals, without incorrectly excluding any eligible individuals. The higher the recall score, the better the model is at correctly identifying eligible individuals. Based on the comparison of the result of the ML methods, the LR model obtained the highest recall score of 0.82. This indicates that the LR model is less likely to miss any eligible individuals and is therefore more reliable in identifying all eligible individuals. In contrast, the RF model has a slightly lower recall score of 0.81, which suggests that it may have a higher rate of false negatives, meaning that it may incorrectly identify some eligible individuals as ineligible. However, the RF model still has a high recall score, indicating that it is also effective at identifying eligible individuals. The DT model has the lowest recall score of 0.72, suggesting that it may have a higher rate of false negatives compared to the other two models. This means that the DT model is more likely to miss eligible individuals, which could lead to potentially risky lending decisions.

Next, Precision is a key metric to consider when predicting loan eligibility, as it measures the model's ability to accurately identify applicants who are eligible for a loan without mistakenly identifying those who are not. This is particularly important because incorrectly identifying individuals as eligible can have significant repercussions, such as loan defaults that can bring a negative impact to the financial institutions. Based on the result obtained, the LR model has the highest precision score which is 0.8, indicating that it is the most effective at correctly identifying eligible individuals. The RF model has a slightly lower precision score which is 0.79, indicating a slightly higher rate of false positives. The DT model has the lowest precision score which is 0.74, indicating that it may have a higher rate of false positives.

F1-score is an important metric to consider when evaluating the performance of ML models for loan eligibility prediction. The LR model has an F1-score of 0.79, which is the highest score among the three models. A higher F1-score indicates that the model is more effective at identifying eligible loan applicants while minimizing false positives, which is important for financial institutions to minimize risk and ensure that eligible applicants are not denied a loan. The RF model has an F1-score of 0.78, which is second-highest score among the three ML models. The DT model has an F1-score of 0.72, indicating a relatively low balance between precision and recall. Therefore, the F1-score is a key metric to consider when evaluating the performance of ML models for loan eligibility prediction.

A radar chart can be used as a useful visualization tool to compare multiple variables across different models. In this study, the performance of the LR, RF, and DT models in terms of their accuracy, recall, precision, and F1-score compared using a radar chart. The radar chart of the performance of the three ML methods are shown in Figure 4.34. From the radar chart, it shows the performance of the three ML models in predicting loan eligibility.

The LR model performs the best among the three models, with the longest line that is closest to the outer edge of the chart for all four-evaluation metrics which were accuracy, recall, precision, and F1 score. The RF model also performs well, especially in correctly identifying eligible loan applicants. However, the DT model performs the worst, with the shortest line among the three models, indicating its poor performance across all evaluation metrics.

Overall, based on the performance metrics and radar chart in this study, the LR model appears to be the most effective at predicting loan eligibility, with the highest precision score which is 0.81 and a relatively high recall score of 0.82. However, the RF model is a close second, with a high precision score of 0.79 and the highest recall score which is 0.81. The DT model has the lowest precision and recall scores and is therefore the least effective at predicting loan eligibility.



Figure 4. 34 Radar chart of the performance of the ML methods

4.6.2 Loan Eligibility Prediction System Proposed

Python is a programming language that is commonly used in ML algorithms (Gülen, 2022). Jupyter Notebook is one of the popular tools for working with Python in ML which allows Python code executed in an interactive, browser-based environment.

In this study, a LR model is used for training to make predictions about loan eligibility by using the Streamlit to integrated with. After the model has been trained, it can be tested using a set of testing data (Nantasenamat, 2023).


This can help users to better understand the performance of their models and prediction outcome of the loan eligibility is shown. The process of the predict the loan eligibility executed step by step by inserting the input of data in the Streamlit web application interface such as Gender, Married, Dependents, Education, Self Employed, Applicant Income, Coapplicant Income, Loan Amount, Loan Amount Term, Credit History, and Property Area.

Loan Eligibility Prediction System is a web application created using Streamlit. This application allows users to predict their loan eligibility based on a set of input parameters. The app is hosted on Streamlit Cloud and can be accessed at <https://paul0426-psm-loan-eligibility-prediction-loan-lg13sb.streamlit.app/>. The GitHub link access to a Python project titled 'Loan Eligibility Classification Using Machine Learning Approach,' which showcases a machine learning-based solution for determining loan eligibility at https://github.com/Paul0426/PSM_Loan_Eligibility_Prediction.git.

The random data row was selected from the dataset for testing purpose. The data selected for testing the successful loan approval is show below at Figure 4.35 and the result of the loan status according to the input inserted is show below at Figure 4.36. Additionally, the data selected for testing the unsuccessful loan approval is show below at Figure 4.37 and the result of the loan status according to the input inserted is show below at Figure 4.38.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	applicantIncome	LoanAmount	Loan_Amount_Term	credit_History	property_Area	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
5	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
8	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
9	LP001014	Male	Yes	3	Graduate	No	3036	2504	158	360	0	Semiurban	N
10	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
11	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
12	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y

Figure 4. 35 Data Row of Successful Loan Status



Loan Eligibility Prediction

Gender

Male

Married

No

Dependents

None

Education

Graduate

Self-Employed

No

Applicant Income

6000 - +

Coapplicant Income

0 - +

Loan Amount (Thousands)

139 - +

Loan Tenor (Months)

360 - +

Credit History

1 - +

Property Area

Urban

Predict

You are ELIGIBLE for the loan

Figure 4. 36 Input of the Successful Loan Status Data

18	LP001034	Male	No	1	Not Graduate	No	3596	0	100	240		Urban	Y
19	LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
20	LP001038	Male	Yes	0	Not Graduate	No	4887	0	133	360	1	Rural	N
21	LP001041	Male	Yes	0	Graduate	Yes	2600	3500	115		1	Urban	Y
22	LP001043	Male	Yes	0	Not Graduate	No	7660	0	104	360	0	Urban	N
23	LP001046	Male	Yes	1	Graduate	No	5955	5625	315	360	1	Urban	Y
24	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116	360	0	Semiurban	N
25	LP001052	Male	Yes	1	Graduate	Yes	3717	2925	151	360		Semiurban	N
26	LP001066	Male	Yes	0	Graduate	Yes	9560	0	191	360	1	Semiurban	Y
27	LP001068	Male	Yes	0	Graduate	No	2799	2253	122	360	1	Semiurban	Y
28	LP001073	Male	Yes	2	Not Graduate	No	4226	1040	110	360	1	Urban	Y

Figure 4. 37 Data Row of Unsuccessful Loan Status



Loan Eligibility Prediction

Gender: Male | Married: Yes

Dependents: None | Education: Not Graduate

Self-Employed: No | Applicant Income: 7660 - +

Coapplicant Income: 0 - + | Loan Amount (Thousands): 104 - +

Loan Tenor (Months): 360 - + | Credit History: 0 - +

Property Area: Urban

Predict

You are NOT ELIGIBLE for the loan

Figure 4. 38 Input of the Unsuccessful Loan Status Data

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

ML is becoming increasingly important in loan eligibility prediction due to its ability to analyse large volumes of loan application data and identify complex patterns. Traditional methods of assessing loan eligibility often rely on manual evaluations and predefined rules, which can be subjective and limited in their accuracy.

With the ML algorithms involved, it able to leverage vast amounts of historical loan data to identify relevant features and create predictive models that assess creditworthiness more accurately. With the ever-increasing availability of data and advancements in ML techniques, financial institutions can make more informed and objective decisions, leading to improved risk assessment, reduced default rates, and enhanced overall efficiency in the loan approval process.

As a result, ML algorithm which are RF, LR, and DT selected to conduct this research. This research's contributions are evaluating the performance of the three ML algorithms in predicting the loan eligibility. To conduct this study, the Loan Dataset which consists of 614 data samples is used for performance evaluation in term of accuracy, precision, recall, and F1-measure. From the 13 features, there are six were categorical and seven were numeric. All the dataset are trained using three algorithms, which are RF, LR, and DT in the Python coding by Jupyter Notebook.

There are several processes conducted before the performance evaluation like Exploratory Data Analysis (EDA), data preprocessing to ensure a fair and accurate performance results are generated. Before a performance evaluation is conducted, several essential steps need to be taken to ensure accurate and unbiased results. These steps include Exploratory Data Analysis (EDA) and data preprocessing. EDA involves the checking the data type of each variable, checking the count of missing values, identify the duplicate values, and calculate the total number of missing values. Data preprocessing involves the feature selection, handle missing and null values, variables transformation, and encoding of categorical variables. These steps ensure that the data is ready for analysis and that the performance evaluation is conducted fairly and accurately.

Results have shown that LR presents better results than other ML models. LR stands out as the best-performing ML algorithm based on the performance metrics in this study. It achieved the highest accuracy score of 0.81, surpassing both DC with a score of 0.72 and RF, which also achieved 0.81. In terms of F1 Score, LR obtained 0.79, while DC scored 0.73, and RF achieved 0.79 as well. Furthermore, LR obtained a precision score of 0.81, surpassing DC with a score of 0.75, and RF with a score of 0.80. LR also achieved the highest recall score of 0.81, outperforming DC with a score of 0.72, and tying with RF at 0.81.

According to this research, LR has been identified as the superior ML algorithm for predicting loan eligibility compared to RF and DT.

5.2 RESEARCH CONSTRAINTS CHALLENGES

During this research conducted, there are various of constraints were identified and acknowledged as significant challenges that needed to be addressed and overcome in order to ensure the validity and reliability of the research findings.

This study encountered a significant constraint in the form of limited time available for conducting comprehensive evaluations of various ML algorithms. Due to the restricted timeframe, it was not feasible to thoroughly explore and include additional methods in this research. Regrettably, due to the limited timeframe, it was not feasible to extensively evaluate and incorporate additional ML methods that could have potentially offered valuable insights and allowed for comparative analysis. Consequently, only three ML algorithms, namely LR, RF, and DT, were selected for inclusion in this study. If more time had been available, additional methods such as Support Vector Machines (SVM) and Naive Bayes could have been chosen and incorporated to broaden the scope of the research.

Other limitation of this research is that the ML technique's accuracy may be compromised when the dataset is limited. The quantity and quality of the dataset can have an impact on the performance of the trained and tested ML model. The dataset related to bank or loan is limited and it is not easy to obtain as it is confidential information of the financial institution's customer. Thus, the result of loan eligibility prediction may be less accurate due to the limitations of the dataset used for prediction during the prototype in Streamlit.

Finally, all the constraints encountered during this research can serve as valuable recommendations for improvement in future work. These constraints highlight areas where enhancements can be made to enhance the quality and scope of future studies. By taking these recommendations into consideration, future research can overcome the limitations faced in this study and contribute to more robust and insightful findings in the field of loan eligibility prediction.

5.3 FUTURE WORK

Based on the obtained results, it was observed that the performance metrics such as accuracy, precision, recall, and F1 score were influenced by the different ML methods

used for training the models. This shows that the selection of the ML method has a great impact on the result of loan eligibility prediction.

Thus, it is recommended to explore and consider more ML methods in future research. By incorporating a broader range of ML algorithms, it becomes possible to compare their performance and determine the most effective approach for loan eligibility prediction. Each algorithm has its own unique strengths and weaknesses, and conducting a comparative analysis allows for the identification of the most accurate and reliable algorithm.

Including more ML methods in future studies enables researchers to gain a comprehensive understanding of the predictive capabilities of different algorithms. This will help in selecting the most suitable approach based on the specific requirements and characteristics of the loan eligibility prediction task. By considering a wider range of ML methods, future research can enhance the accuracy, robustness, and applicability of loan eligibility prediction models.

REFERENCES

Specialist, G. (2022, June 27). First bank in the world. Money Gate. <https://money-gate.com/first-bank-in-world/>

Sharen Kaur. (2022, April 7). Higher loan approvals in the first two months of 2022 indicate that the market is recovering, says MIDF Research <https://www.nst.com.my/property/2022/04/786681/higher-loan-approvals-first-two-months-2022-indicate-market-recovering-says>

Massaoudi, T. (2019, June 7). ML basics: Loan prediction. [towardsdatascience.com. https://towardsdatascience.com/ml-basics-loan-prediction-d695ba7f31f6#:~:text=These%20details%20are%20Gender%2C%20Marital,can%20specifically%20target%20these%20customers.](https://towardsdatascience.com/ml-basics-loan-prediction-d695ba7f31f6#:~:text=These%20details%20are%20Gender%2C%20Marital,can%20specifically%20target%20these%20customers.)

Sinhasane, S. (2018, July 1). Medical Diagnosis Apps - A Game Changer in the World of On-demand Healthcare. Mobisoft Infotech. <https://mobisoftinfotech.com/resources/blog/medical-diagnosis-apps/>

Over 10,000 declared bankrupt during MCO period. (2021, September 29). MalaysiaNow. <https://www.malaysianow.com/news/2021/09/28/over-10000-declared-bankrupt-during-mco-period>

Markovic, I. (2020, November 22). What is the average cost of training a new employee? eduMe. <https://www.edume.com/blog/cost-of-training-a-new-employee>

Machine Learning. (2021, July 2). <https://www.ibm.com/my-en/cloud/learn/machine-learning>

Waseem, M. (2022, March 28). How To Implement Classification In Machine Learning? Edureka. <https://www.edureka.co/blog/classification-in-machine-learning/>

5 Types of Classification Algorithms in Machine Learning. (2020, August 26). MonkeyLearn Blog. <https://monkeylearn.com/blog/classification-algorithms/>

What is Binary Classification. (2021, August 5). Deepchecks. <https://deepchecks.com/glossary/binary-classification/>

Gong, D. (2022, February 23). Top 6 ML Algorithms for Classification. Towards Data Science. <https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501>

"LR." International Encyclopedia of the Social Sciences. Encyclopedia.com. (October 28, 2022). <https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/logistic-regression>

ML- LR. (n.d.). Retrieved November 6, 2022, from https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm

Latifah, F. A., Slamet, I., & Sugiyanto. (2020). Comparison of heart disease classification with LR algorithm and RF algorithm. INTERNATIONAL CONFERENCE ON SCIENCE AND APPLIED SCIENCE (ICSAS2020). <https://doi.org/10.1063/5.0030579>

Lin, P., Soto-Ferrari, M., & Chams-Anturi, O. (2022). A LR Assessment to Measure Radiotherapy Clinical Pathway Concordance for Early Stages Breast Cancer Patients. *Procedia Computer Science*, 203, 559–564. <https://doi.org/10.1016/j.procs.2022.07.080>

Liew, B.X.W., Kovacs, F.M., Rügamer, D. et al. MLversus LR for prognostic modelling in individuals with non-specific neck pain. *Eur Spine J* 31, 2082–2091 (2022). <https://doi.org/10.1007/s00586-022-07188-w>

Chow, R. (2022, March 23). *DT and RF Algorithms: Decision Drivers*. History of Data Science. <https://www.historyofdatascience.com/decision-tree-and-random-forest-algorithms-decision-drivers/>

The Complete Guide to DT Analysis. (2019, December 10). Explorium. <https://www.explorium.ai/blog/the-complete-guide-to-decision-trees/>

DTs for Decision Making. (2014, August 1). Harvard Business Review. <https://hbr.org/1964/07/decision-trees-for-decision-making>

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus With ML Techniques. *Frontiers in Genetics*, 9. <https://doi.org/10.3389/fgene.2018.00515>

Chaudhary, M. (2022, June 28). RF Algorithm - How It Works & Why It's So Effective. <https://www.turing.com/kb/random-forest-algorithm>

R, S. E. (2022, June 21). Understanding RF. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

Ardekani, B. A., Bermudez, E., Mubeen, A. M., & Bachman, A. H. (2016). Prediction of Incipient Alzheimer's Disease Dementia in Patients with Mild Cognitive Impairment. *Journal of Alzheimer's Disease*, 55(1), 269–281. <https://doi.org/10.3233/jad-160594>

Sarica, A., Cerasa, A., & Quattrone, A. (2017). RF Algorithm for the Classification of Neuroimaging Data in Alzheimer's Disease: A Systematic Review. *Frontiers in Aging Neuroscience*, 9. <https://doi.org/10.3389/fnagi.2017.00329>

Lin, W., Wu, Z., Lin, L., Wen, A., & Li, J. (2017). An Ensemble RF Algorithm for Insurance Big Data Analysis. *IEEE Access*, 5, 16568–16575. <https://doi.org/10.1109/access.2017.2738069>

Naik, S. (2021, September 23). DT Advantages and Disadvantages. EDUCBA. <https://www.educba.com/decision-tree-advantages-and-disadvantages/>

Duggal, N. (2022, September 8). Advantages of DTs. Simplilearn.com. <https://www.simplilearn.com/advantages-of-decision-tree-article>

GeeksforGeeks. (2022, August 23). Advantages and Disadvantages of LR. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/>

Singh, J. (2022, December 18). RF: Pros and Cons. DataDrivenInvestor. <https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04>

Kumar, N., Kumar, N., & Profile, V. M. C. (n.d.). Advantages and Disadvantages of RF Algorithm in Machine Learning. <http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html>

Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using DTs and RF: A comparative study. *IOP Conference Series: Materials*

Science and Engineering, 1022(1), 012042. <https://doi.org/10.1088/1757-899x/1022/1/012042>

Banking Dataset Classification. (2020, September 6). Kaggle. <https://www.kaggle.com/datasets/rashmiranu/banking-dataset-classification>

Soham, S. (2018, September 26). Personal Loan Prediction: Using DT. Kaggle. <https://www.kaggle.com/code/sohamsave/personal-loan-prediction-using-decision-tree/notebook>

Zaidan, dhman. (2022, October 21). Prediciting loan default probability. Kaggle. Retrieved December 29, 2022, from https://www.kaggle.com/code/zaidandhman/prediciting-loan-default-probability/data?select=train_u6lajuX_CVtuZ9i.csv

S., S., A., A., & M. Mohamed, R. (2021). Loan Prediction Using LR in ML(p. 2794) [Review of Loan Prediction Using LR in Machine Learning

How Does LR Work? (n.d.). KDnuggets. Retrieved January 5, 2023, from <https://www.kdnuggets.com/2022/07/logistic-regression-work.html#:~:text=is%20Logistic%20Regression%3F->

Gülen, K. (2022, November 17). Best Language For MLIn 2022: Is It Python? Dataconomy. <https://dataconomy.com/2022/11/best-language-for-machine-learning/#:~:text=Because%20Python%20is%20one%20of,Python%20engineers%20are%20in%20demand.>

Nantasenamat, C. (2023c, January 4). How to Build your First MLModel in Python. *Medium*. <https://towardsdatascience.com/how-to-build-your-first-machine-learning-model-in-python-e70fd1907cdd>

D'Agostino, A. (2023, April 4). Exploratory Data Analysis in Python — A Step-by-Step Process. *Medium*. <https://towardsdatascience.com/exploratory-data-analysis-in-python-a-step-by-step-process-d0dfa6bf94ee>

GeeksforGeeks. (2023). Data Preprocessing in Data Mining. *GeeksforGeeks*. <https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/>

Lawton, G. (2022). data preprocessing. *Data Management*. <https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing>

Gupta, A. (2023). Feature Selection Techniques in ML(Updated 2023). *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/>

Kumar, S. (2021, December 15). 7 Ways to Handle Missing Values in Machine Learning. *Medium*. <https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e>

Saxena, S. (2022). Here's All you Need to Know About Encoding Categorical Data (with Python code). *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/>

APPENDIX A CORRELATION HEATMAP

