

A

Summer Internship Report

On

**Loan Prediction Using Machine Learning**

Submitted for partial fulfilment of the requirements for the award of the  
degree of

**BACHELOR OF ENGINEERING**

In

**INFORMATION TECHNOLOGY**

By

**P. Sri Charan Reddy**  
(2451-20-737- 002)

**Under Supervision of**  
**Pradip Narayankar, Director**

**PHN Technology Pvt. Ltd.**  
(Duration: 6<sup>th</sup> April, 2023 to 6<sup>th</sup> June, 2023)



**MATURI VENKATA SUBBA RAO (M.V.S.R) ENGINEERING COLLEGE**

**(An Autonomous Institution)**

Department of Information Technology

(Affiliated to Osmania University & Recognized by AICTE)

Nadargul, Saroor Nagar Mandal, Hyderabad – 501510.

**AY: 2023-24**



## CERTIFICATE

*This is to certify that the summer internship report entitled “**Loan Prediction Using Machine Learning**” is a bonafide work carried out by*

**P. Sri Charan Reddy (2451-20-737- 002)** *in partial fulfilment of the requirements for the award of degree of Bachelor of Engineering in Information Technology from Maturi Venkata Subba Rao (M.V.S.R.) Engineering College, an Autonomous Institution, affiliated to Osmania University Hyderabad, during the Academic Year 2023- 24 at **PHN Technology Pvt. Ltd., Pune.***

*The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.*

### **Summer Internship Co-ordinator**

**Dr.A.V.Krishna Prasad**

Associate Professor

Department of IT

MVSREC.

### **HOD**

**Dr.K.Venugopala Rao**

Dean – Academics,

Professor & Head

Department of IT

MVSREC

**External Examiner**



Date: 2023-06-16

**TO WHOMSOEVER IT MAY CONCERN**

This is to certify that **PABBATHI REDDY SRI CHARAN REDDY (Intern ID- 227825)** has successfully completed the Internship in **Machine Learning & Data Science with Python** presented by **PHN Technology Pvt Ltd, Pune.**

This program was designed to provide valuable hands-on experience and practical knowledge in **Machine Learning & Data Science with Python**. Through active participation and dedication, he/she has demonstrated exceptional skills, commitment, and a strong work ethic.

During the internship, he/she actively contributed to **Machine Learning & Data Science with Python** and exhibited professionalism, adaptability, and a willingness to learn. Their enthusiasm and positive attitude have made a significant impact on the overall success of the program.

We extend our sincere appreciation to **PABBATHI REDDY SRI CHARAN REDDY** for their valuable contributions and dedication throughout the duration of the virtual internship program.

Internship tenure was from **06/04/2023** to **06/06/2023**.

**Pradip Narayankar**  
**Director**  
**PHN Technology Pvt. Ltd.**



Document ID- PHN\_20230616349

 Solitaire Business Hub E wing 5010,  
Near Pheonix Mall, Viman Nagar,  
Pune, Maharashtra, 411014.

 1800 209 2288

 [www.phntechnology.com](http://www.phntechnology.com)

## DECLARATION

This is to certify that the work reported in the present Summer Internship entitled **“Loan Prediction Using Machine Learning”** is a record of bonafide work done by us in the department of Information Technology, Maturi Venkata Subba Rao (M.V.S.R.) Engineering College, an Autonomous Institution, affiliated to Osmania University. The reports are based on the project work done entirely by us and not copied from any other source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

**P. Sri Charan Reddy**  
(2451-20-737-002)

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Pradip Narayankar, Director PHN Technology Pvt. Ltd.** for giving me the opportunity and to do an internship within the organization and giving valuable suggestions and interest throughout the course of this work.

We would like to express our sincere gratitude and indebtedness to our Summer Internship Co-ordinator **Dr.A.V.Krishna Prasad, Associate Professor**, for their valued recommendations and attention throughout the course of this project.

We are also thankful to our Principal **Dr. G. Kanaka Durga, Professor**

and **Dr.K. Venugopala Rao**, Professor and Head - Department of Information Technology, Dean- Academics, Maturi Venkata Subba Rao Engineering College, for providing excellent infrastructure for completing this project successfully as a part of our B.E. (IT).

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**P. Sri Charan Reddy**  
(2451-20-737-002)

## **VISION & MISSION, PROGRAM EDUCATIONAL OUTCOMES**

### **Vision of the Department:**

To impart technical education producing competent and socially responsible engineering professionals in the field of Information Technology.

### **Mission of the Department:**

- M1. To make teaching learning process effective and stimulating.
- M2. To provide adequate fundamental knowledge of sciences and Information Technology with positive attitude.
- M3. To create an environment that enhances skills and technologies required for industry.
- M4. To encourage creativity and innovation for solving real world problems.
- M5. To cultivate professional ethics in students and inculcate a sense of responsibility towards society.

### **Program Educational Objectives:**

After 3 to 4 years of graduation, graduates of the Information Technology program will:

- I. Apply knowledge of mathematics and Information Technology to analyze, design and implement solutions for real world problems in core or in multidisciplinary areas.
- II. Communicate effectively, work in a team, practice professional ethics and apply knowledge of computing technologies for societal development.
- III. Engage in Professional development or postgraduate education to be a life-long learner.

## PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

### PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and

design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. 0`

## **PROGRAM SPECIFIC OUTCOMES (PSOs):**

PSO1: **Hardware design:** An ability to analyze, design, simulates and implements computer hardware / software and use basic analog /digital circuits, VLSI design for various computing and communication system applications.

PSO2: **Software design:** An ability to analyze a problem, design algorithm, identify and define the computing requirements appropriate to its solution and implement the same.



## **COURSE OBJECTIVES & COURSE OUTCOMES**

### **COURSE OBJECTIVES**

- To give an experience to the students in solving real life practical problems with all its constraints.
- To give an opportunity to integrate different aspects of learning with reference to real life problems.
- To enhance the confidence of the students while communicating with industry engineers and give an opportunity for useful interaction with them and familiarize with the work culture and ethics of the industry.

### **COURSE OUTCOMES**

After completion of the course student will be able to:

CO1 - Formulate a problem to map the requirements of real-world scenario

CO2 - Design/develop a small and suitable product in hardware or software

CO3 – Exhibit the skills to use contemporary technologies used by the industry

CO4 - Evaluate the solution against pre-existing alternatives with reference to pre specified criteria

.

CO5 - Demonstrate an understanding of work culture and ethics of the industry

**WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES**

1 <sup>st</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	06/04/23	Thursday	Introduction Session
	07/04/23	Friday	Session1 – Corporate Readiness Essentials
	10/04/23	Monday	Task 1- Analytics Task
	11/04/23	Tuesday	Time to complete the task
	12/04/23	Wednesday	Time to complete the task and submission

2 <sup>nd</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	13/04/23	Thursday	Discussion on Task
	14/04/23	Friday	Session 2 :- Corporate readiness essentials
	17/04/23	Monday	Task 2 :- Resume Building
	18/04/23	Tuesday	Time to complete the Task
	19/04/23	Wednesday	Submission of Task 2

3 <sup>rd</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	20/04/23	Thursday	Introduction to Excel
	21/04/23	Friday	Working with Excel
	24/04/23	Monday	Functions in Excel
	25/04/23	Tuesday	Visualization in Excel
	26/04/23	Wednesday	Filters in Excel

X

4 <sup>th</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	27/04/23	Thursday	Introduction to Jupiter Notebook
	28/04/23	Friday	Working with Jupiter notebook and Google Colab
	01/05/23	Monday	Introduction to various libraries in python.
	02/05/23	Tuesday	working with datasets
	03/05/23	Wednesday	Few data preprocessing steps

5 <sup>th</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	04/05/23	Thursday	Session :- Machine Learning – 1
	05/05/23	Friday	Task 3:- Machine Learning -1
	08/05/23	Monday	Reading data from an excel sheet and performing calculations on data using open pyxl library
	09/05/23	Tuesday	Writing the new calculated data to new excel sheet using open pyxl library and task submission
	10/05/23	Wednesday	Task discussion

6 <sup>th</sup> week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	11/05/23	Thursday	Session :- Machine learning – 2
	12/05/23	Friday	Task4 :- Machine learning-2
	15/05/23	Monday	Working with Numpy, shape attribute and concatenation of arrays.
	16/05/23	Tuesday	Performing calculations using mean, Median and SD and task submission
	17/05/23	Wednesday	Discussion of Task

7 <sup>th</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	18/05/23	Thursday	Session :- Machine learning -3
	19/05/23	Friday	Task machine learning - 3
	22/05/23	Monday	Working on visualization in python using matplotlib
	23/05/23	Tuesday	Analyzing different types of plots in python and task submission
	24/05/23	Wednesday	Task machine learning - 4

8 <sup>TH</sup> Week	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	25/05/23	Thursday	Working on loan dataset, performing data preprocessing, cleaning and finding correlations.
	26/05/23	Friday	Data visualization to find insights and splitting the data into train and test and feature scaling
	29/05/23	Monday	Working with different classification models to find out best accuracy for accurate predictions
	30/05/23	Tuesday	Creating user interface using Flask and choosing best accuracy model for prediction
	01/06/23	Wednesday	Submission of the task and discussion

xii

**Abstract**

When any financial institution lends the money to the person, it is always been a high risk. Today data is increasing with the rapid pace in the banks, therefore the bankers need to evaluate the person's data before giving the loan. It can be a big headache to evaluate the data. This problem is solved by analysing and training the data by using one of the Machine Learning algorithms. For this, we have generated a model for the prediction that the person will get the loan or not. The primary objective of this paper is to check whether the person can get the loan or not by evaluating the data with the help of decision tree classifiers which can gives the accurate result for the prediction.

xiii

**TABLE OF CONTENTS****CONTENTS****PAGE NO**

Certificate	i
Certificate (Organization)	ii
Declaration	iii
Acknowledgement	iv
Vision & Mission, PEOs	v
POs and PSOs	vi-vii
Course Objectives & Outcomes	viii
Weekly Overview of Internship Activities	ix-xi
Abstract	xii
Table of Contents	xiii
List of Figures	xiv
List of Tables	xv
List of Abbreviations	xvi
 Chapters	
<b>1. Introduction</b>	<b>1 - 2</b>
1.1 Problem statement	
1.2 Existing system	
1.3 Proposed system	
1.4 Scope	
<b>2. System Requirement Specifications</b>	<b>3</b>
2.1 Software Requirements	
2.2 Hardware Requirements	
 <b>3. Design &amp; Implementation</b>	 <b>4 – 16</b>
3.1 Features	
3.2 Environmental Setup	
3.3 System Architecture / Block Diagram	
3.4 UML Diagrams	
3.5 Algorithms	
<b>4. Test &amp; Results</b>	<b>17- 18</b>
4.1 Test cases	
4.2 Results	
<b>5. Conclusion &amp; Future Enhancements</b>	<b>19</b>
<b>References</b>	<b>20</b>
<b>Appendix</b>	<b>21-29</b>
<b>A: source/pseudo code</b>	

xiv

## LIST OF FIGURES

Figure No.	Figure Name	Page No.
------------	-------------	----------

3.2.1	Downloading Python software	5
3.2.2	Versions of Python	5
3.2.3	Installing Python	6
3.2.4	Setup successful	6
3.2.5	Verifying python	7
3.2.6	Downloading Visual Studio Code	8
3.2.7	Agreement to the conditions of Visual Studio Code	8
3.2.8	Image displaying completion of Visual Studio Code	9
3.3.1	System Architecture / Block Diagram	9
3.4.1	Use Case Diagram	11
3.4.2	Class Diagram	12
3.4.3	Activity Diagram	13
3.4.4	Sequence Diagram	14
3.5.1	KNN Algorithm	14
3.5.2	Logistic Regression	15
3.5.3	Decision Tree Diagram	16
3.5.4	Random Forest Classifier Diagram	16
4.2.1	Positive Result	18
4.2.2	Negative Result	18

**LIST OF TABLES**

<b>Table No.</b>	<b>Table Name</b>	<b>Page No.</b>
4.1.1	Test cases	17



## CHAPTER 1

### INTRODUCTION

A loan is the core business part of banks. The main portion the bank's profit is directly come from the profit earned from the loans. Though bank approves loan after a regress process of verification and testimonial but still there's no surety whether the chosen hopeful is the right hopeful or not. This process takes fresh time while doing it manually. We can prophesy whether that particular hopeful is safe or not and the whole process of testimonial is automated by machine literacy style. Loan Prognostic is really helpful for retainer of banks as well as for the hopeful also.

#### 1.1 Problem Statement

Understanding the problem statement is the first and foremost step. This would help you give an intuition of what you will face ahead of time. Let us see the problem statement. Any of the Finance company deals which deals with loans. They have a presence across all urban, semi-urban and rural areas. Customers first apply for a home loan after that company validates the customer's eligibility for a loan. The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling out the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. To automate this process, they have given a problem to identify the customer segments, that are eligible for loan amounts so that they can specifically target these customers.

#### 1.2 Existing System

Banks need to analyse for the person who applies for the loan will repay the loan or not. Sometime it happens that customer has provided partial data to the bank, in this case person may get the loan without proper verification and bank may end up with loss. Bankers cannot analyse the huge amounts of data manually; it may become a big headache to check whether a person will repay its loan or not. It is very much

necessary to know the person getting loan is going in safe hand or not. So, it is pretty much important to have an automated model which should predict the customer getting the loan will repay the loan or not.

### **1.3 Proposed System**

I have developed a prediction model for Loan sanctioning which will predict whether the person applying for loan will get loan or not. The major objective of this project is to derive patterns from the datasets which are used for the loan sanctioning process and create a model based on the patterns derived in the previous step. This model is developed by using the one of the machine learning algorithms in the proposed model for loan prediction, Dataset is split into training and testing data. After then training datasets are trained using the decision tree algorithm and a prediction model is developed using the algorithm. Testing datasets are then given to model for the prediction of loan. The motive of this paper is to predict the defaults who will repay the loan or not. Various libraries like pandas, numpy have been used. After the loading of datasets, Data Preprocessing like missing value treatment of numerical and categorical is done by checking the values. Numerical and categorical values are segregated. Outliers and frequency analysis are done, outliers are checked by getting the boxplot diagram of attributes.

### **1.4 Scope**

The Loan Prediction System project aims to develop a machine learning model for assessing the creditworthiness of loan applicants, enhancing the efficiency of loan approval processes. The scope includes collecting historical loan data, preprocessing to handle missing values and outliers, selecting and training an appropriate machine learning model, and deploying it in a production environment with a user-friendly interface. Ethical considerations, such as fairness and privacy, will be addressed, and comprehensive documentation will be provided. The project will deliver a trained model, a deployed system, and thorough documentation, with a defined timeline, risk mitigation strategies, and identification of key stakeholders and budget estimates

## **CHAPTER 2**

### **SYSTEM REQUIREMENT SPECIFICATIONS**

#### **2.1 Software Requirement**

- Operating System : Windows 7

#### **2.2 Hardware Requirement**

- Hard Disk : 160GB
- Ram : 8GB
- Processor : I3/Intel Processor

## CHAPTER 3

### DESIGN & IMPLEMENTATION

#### 3.1 Features

Users input relevant parameters into the form, and the data undergoes analysis by the most accurate model. After the prediction, users can view the result.

1. **Data Preprocessing:** Handles inconsistent data to ensure more accurate and results.
2. **Missing Value Identification:** Identifies and replaces missing values in the dataset with suitable values.
3. **Feature Selection:** Uses a correlation matrix to identify features that are least and most correlated with the dependent variable.
4. **Scaling and Normalization:** Performs feature scaling by normalizing the data to a range of 0 to 1, enhancing the algorithm's calculation speed.
5. **Splitting of Data:** Splits the dataset into training (70-80%) and testing (20-30%) sets.
6. **Implementation of Classification Model:** Utilizes various classification models such as Random Forest (RF), Decision Trees (DT), Logistic Regression (LR), k-Nearest Neighbors (KNN), and Naive Bayes (NB). Selects the model with the highest efficiency.
7. **Integration of ML Model with Frontend Using Flask:** Integrates the machine learning model seamlessly with the frontend using Flask.

#### 3.2 Environmental Setup

##### Installing Python:

**Step 1:** • Download python from <https://www.python.org/downloads/> as shown in the figure.

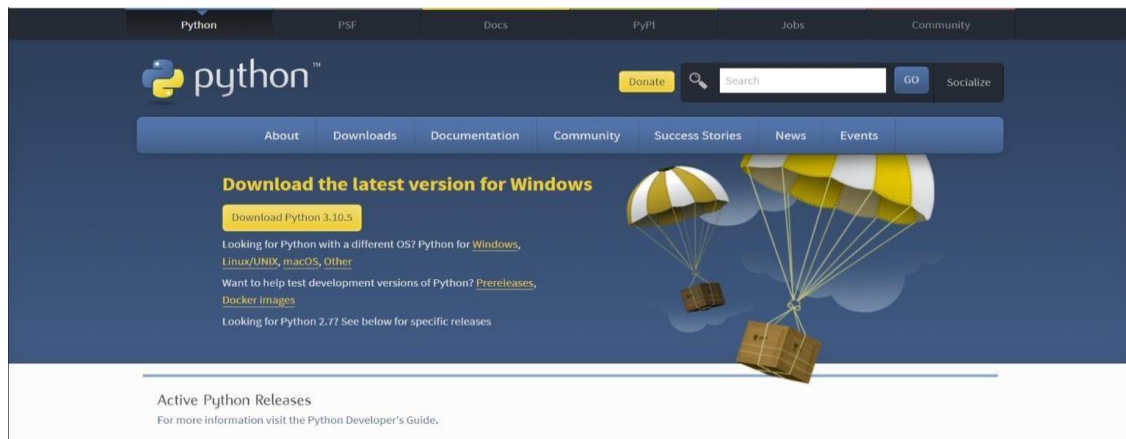


Fig 3.2.1 Downloading Python Software

- Download any of the latest versions of python as shown below.

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
<a href="#">Python 3.8.13</a>	March 16, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.13</a>	March 16, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.10</a>	Jan. 14, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.10.2</a>	Jan. 14, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.10.1</a>	Dec. 6, 2021	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.9</a>	Nov. 15, 2021	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.8</a>	Nov. 5, 2021	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)

Fig 3.2.2 Versions of Python

- Now select the required version and click on Windows installer 32 bit or Windows installer 64 bit to download it depending on your system type.

## Step 2:

- Once it is downloaded, run the setup for python.

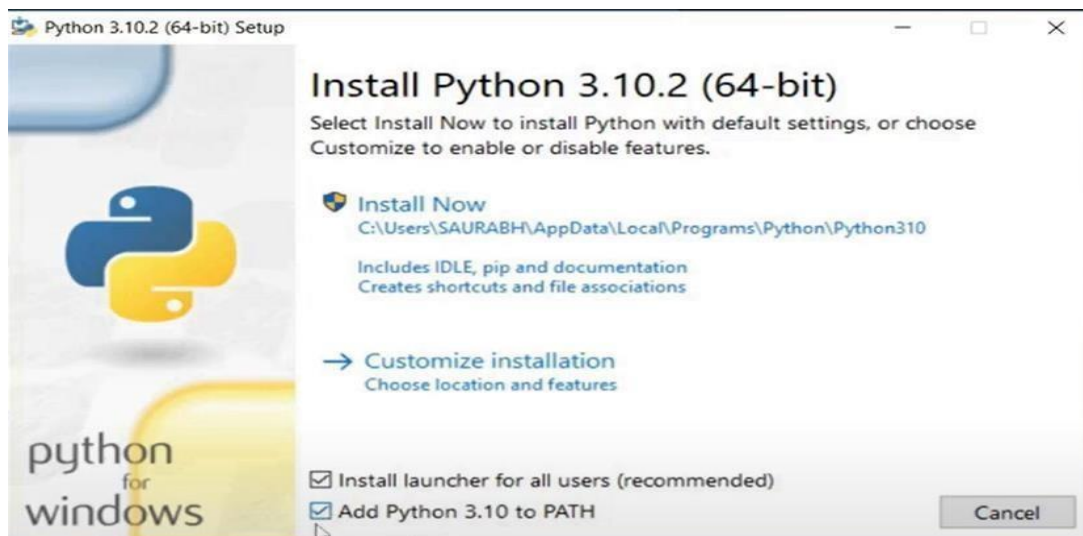


Fig 3.2.3 Installing Python

- Tick the checkbox 'Add Python 3.10 to PATH' to directly add the path to PATH in environmental variables.
- You can now start the installation either by clicking 'Install Now' or 'Customize installation'.
- After a while it shows the setup was successful as shown.

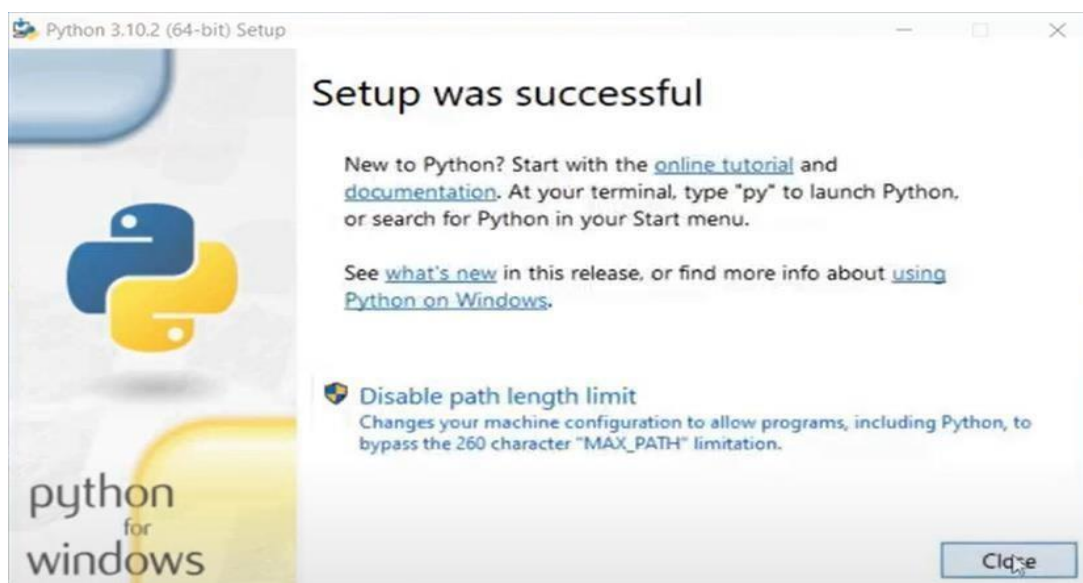
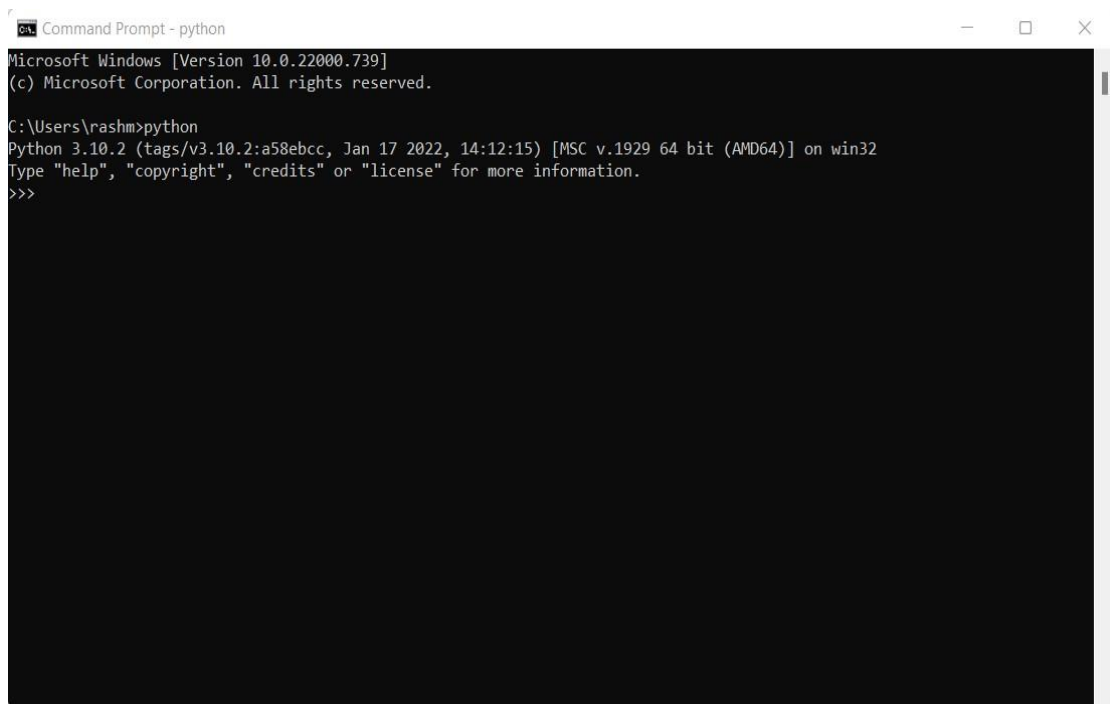


Fig 3.2.4 Setup successful

- Now you can verify whether python is successfully installed by entering python in the command prompt.
- It shows the details of the installed python.



```
Command Prompt - python
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Fig 3.2.5 Verifying python

## Installing Visual Studio Code:

### Step 1:

- Download VS code from <https://code.visualstudio.com/download>

### Step 2:

- Download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer (VSCode User Setup-{version}.exe). Then, run the file – it will only take a minute



Fig 3.2.6 Downloading Visual Studio Code

- Accept the agreement and click “next”

Secondly, accept the agreement and click on next.

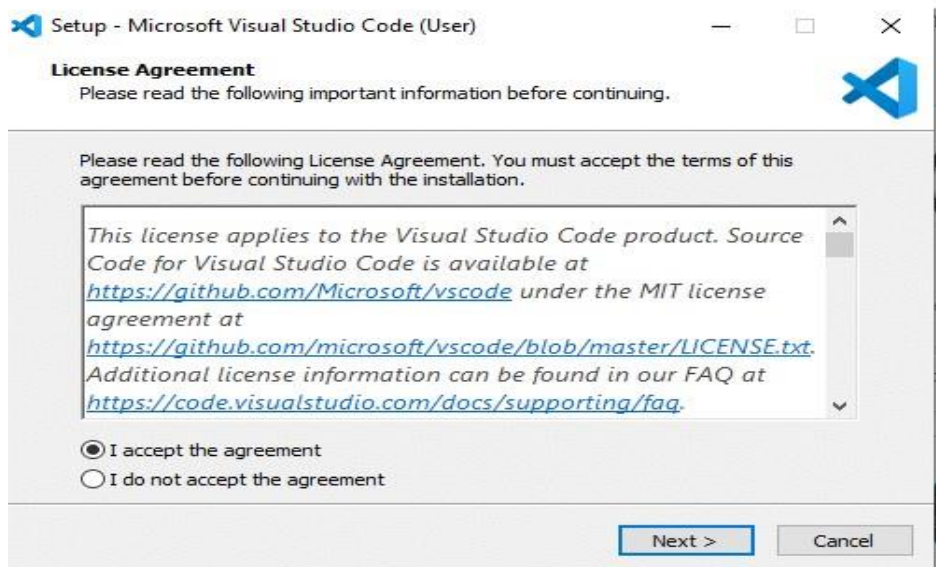


FIG 3.2.7 Agreement to the conditions of Visual Studio Code



- After accepting all the requests press finish button. By default, VS Code installs under:  
“C:\users{username}\AppData\Local\Programs\Microsoft VS Code.”



FIG 3.2.8 Image displaying completion of Visual Studio Code

- Finally, installation is successful.

### 3.3 System Architecture / Block Diagram

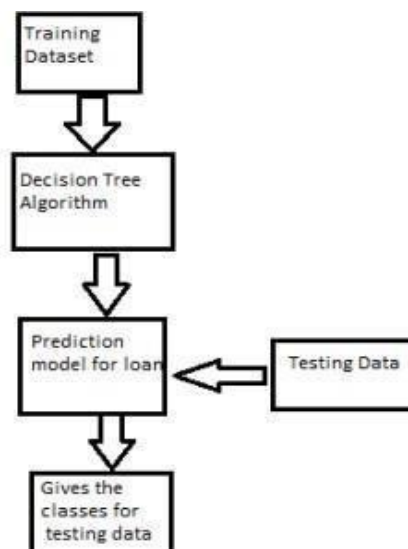


Fig 9: Architecture of propped system

FIG 3.3.1 System Architecture / Block Diagram

### 3.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Metamodel and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### 3.4.1 Use Case Diagram

- A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

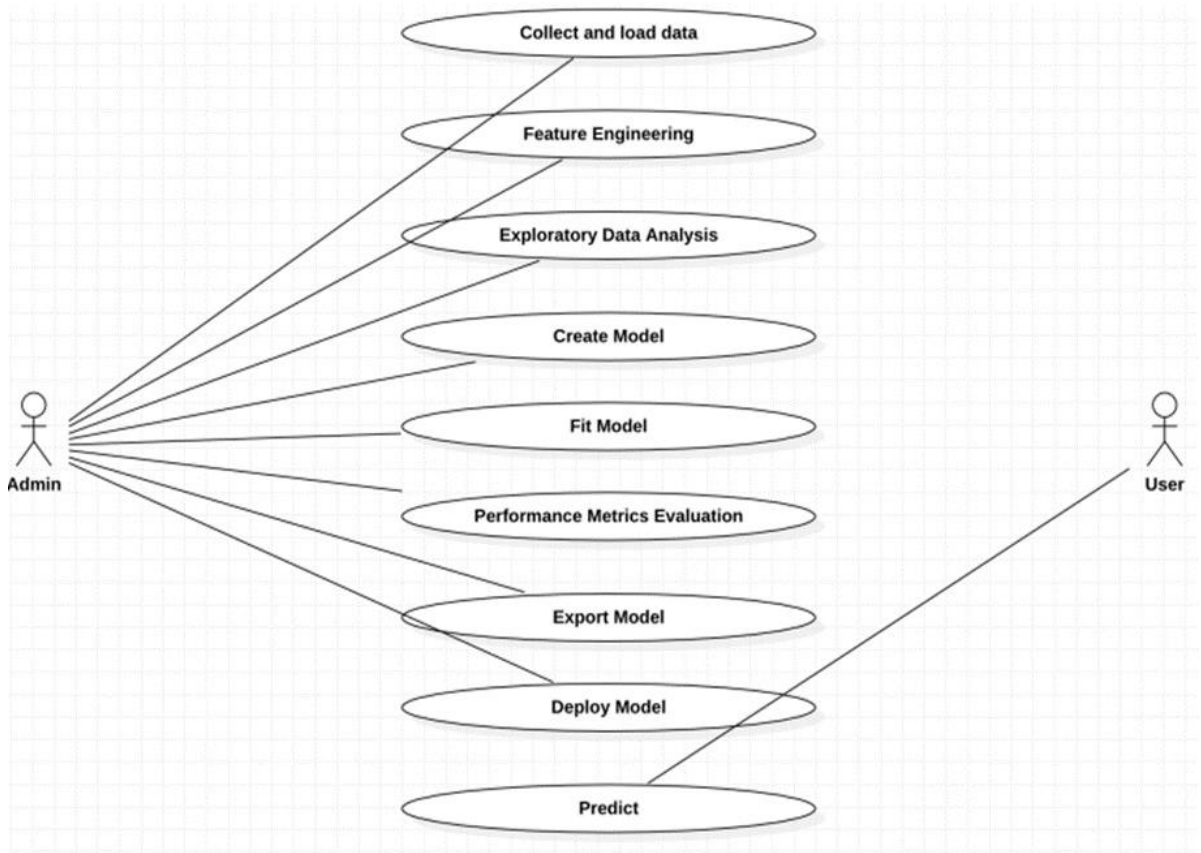


FIG 3.4.1 Use Case Diagram

### 3.4.2 Class Diagram

- In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains which information.

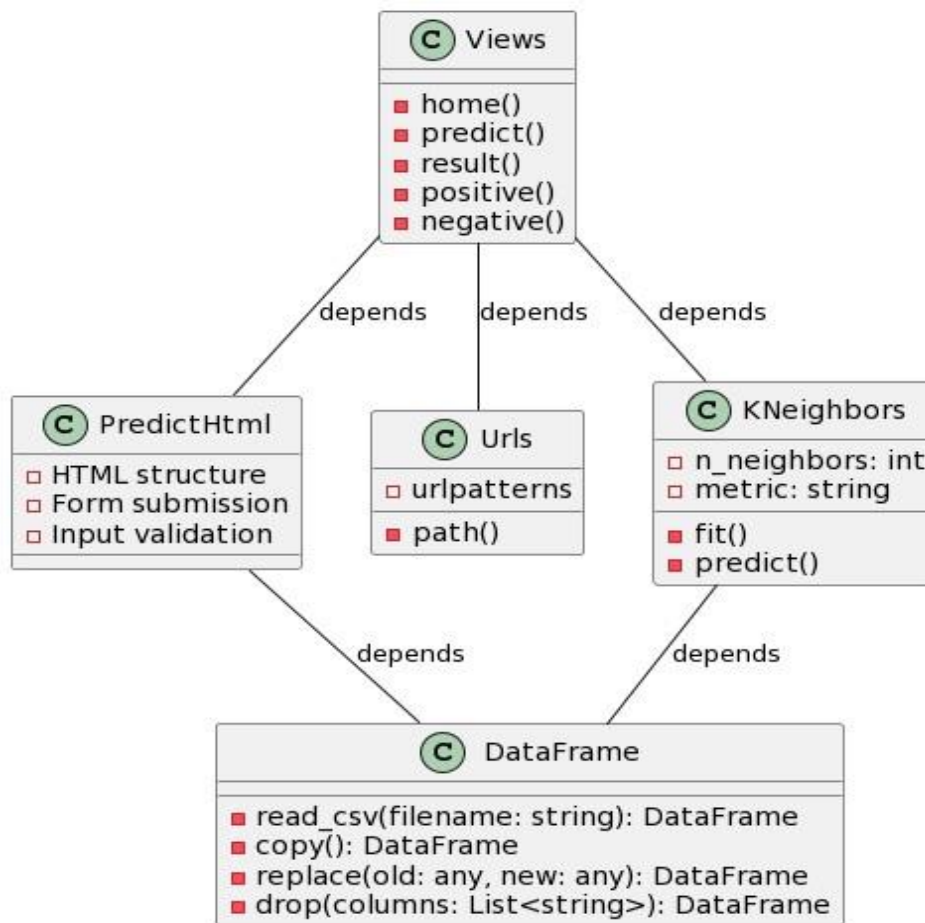


FIG 3.4.2 Class Diagram

### 3.4.3 Activity Diagram

- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step by-step workflows of components in a system. An activity diagram shows the overall flow of control.

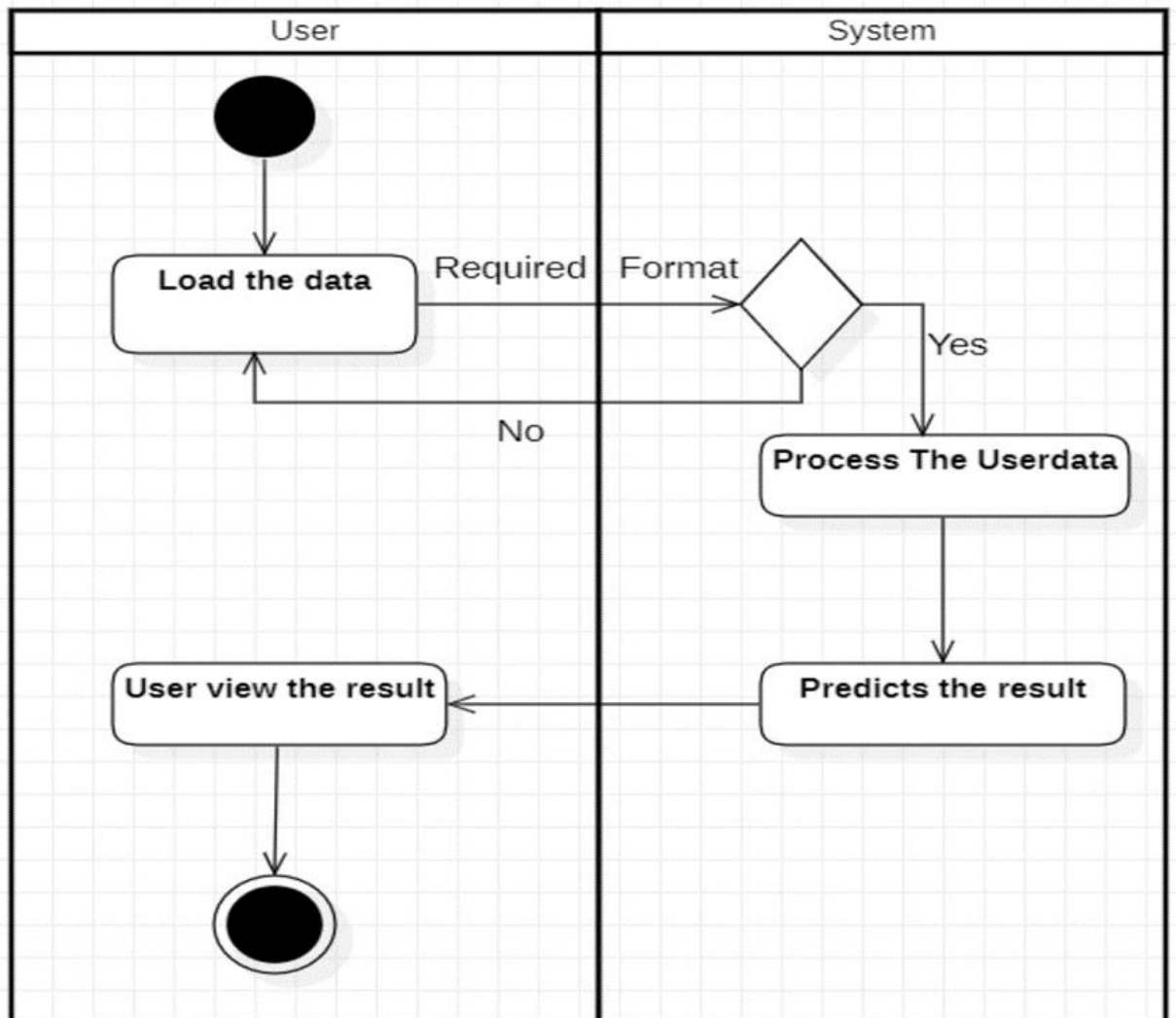


FIG 3.4.3 Activity Diagram

### 3.4.4 Sequence Diagram

- A sequence diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between different objects or components in a system over time. It represents the flow of messages, events, or actions between these entities, showing the order in which they occur.

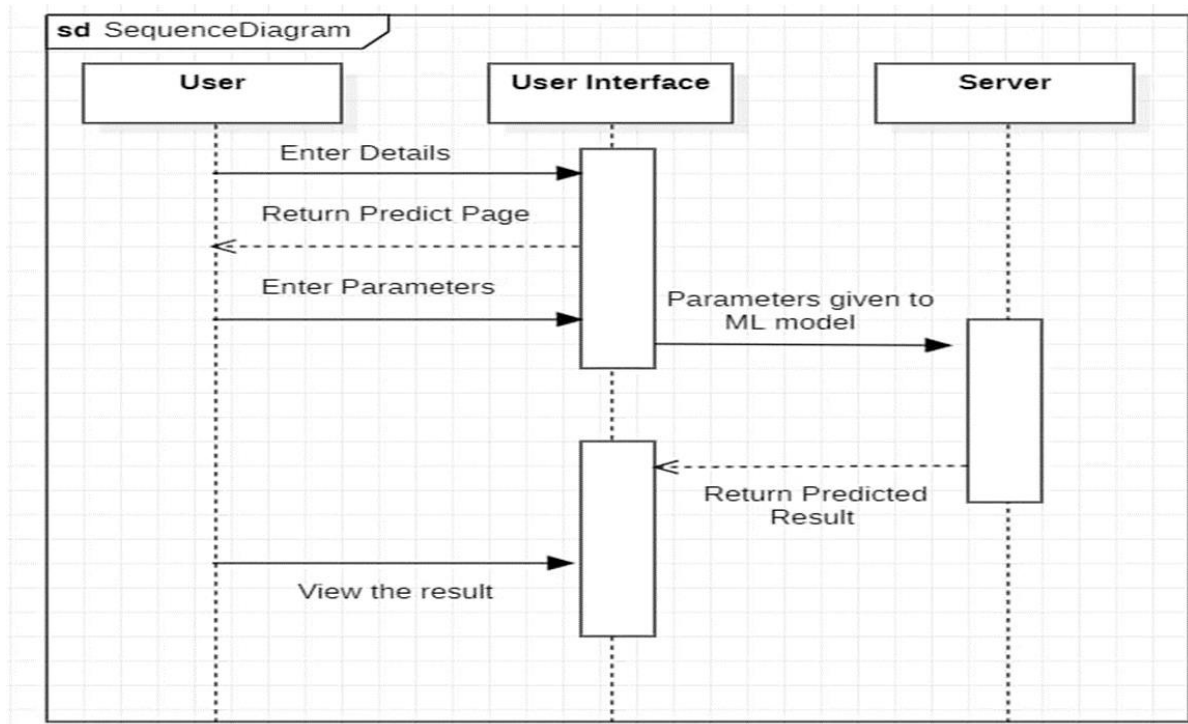


FIG 3.4.4 Sequence Diagram

### 3.5 Algorithm

- KNN Algorithm:**

KNN, a simple supervised learning method, classifies new data based on similarity to existing cases. It's versatile, handling both Regression and Classification, but often shines in classification tasks. Being non-parametric, it doesn't assume data patterns. Known as a lazy learner, it stores the dataset during training and classifies new data by finding the closest match.

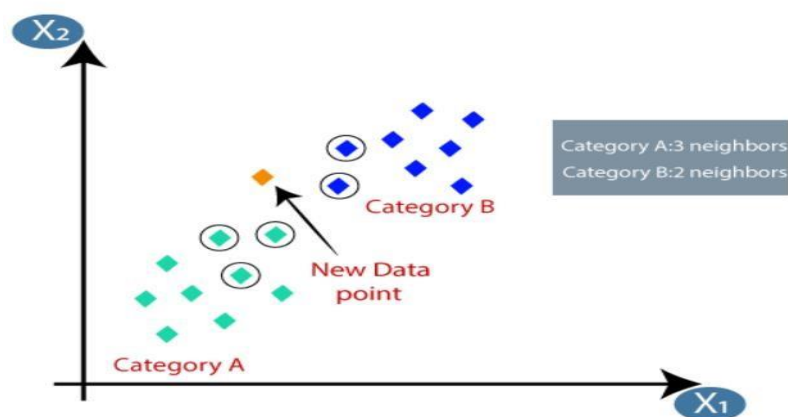


FIG 3.5.1 KNN Algorithm

- **Logistic Regression:**

Logistic regression, a popular supervised learning algorithm, predicts categorical outcomes (Yes/No, 0/1) using independent variables. Unlike linear regression, it employs an "S"-shaped logistic function, offering probabilistic values between 0 and 1. This algorithm is vital for classification tasks, determining probabilities, and effectively classifying new data based on continuous and discrete datasets. The logistic function's curve indicates the likelihood of outcomes.

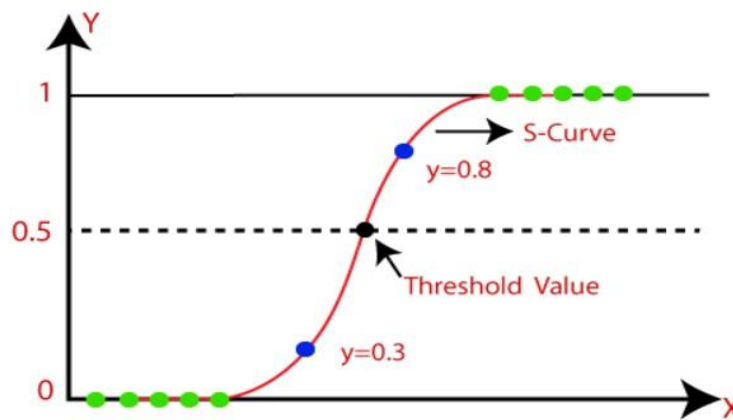


FIG 3.5.2 Logistic Regression

- **Decision Tree Algorithm:**

Decision Tree, a versatile supervised learning tool, excels in classification but handles regression too. Its tree structure comprises decision nodes for choices and leaf nodes for outcomes. Decisions are based on dataset features, forming a graphical representation for problem-solving. Employing the CART algorithm (Classification and Regression Tree), it constructs a tree-like structure, starting from the root and branching out based on answers to simple questions. The tree efficiently captures all possible solutions by iteratively asking questions and refining subtrees.

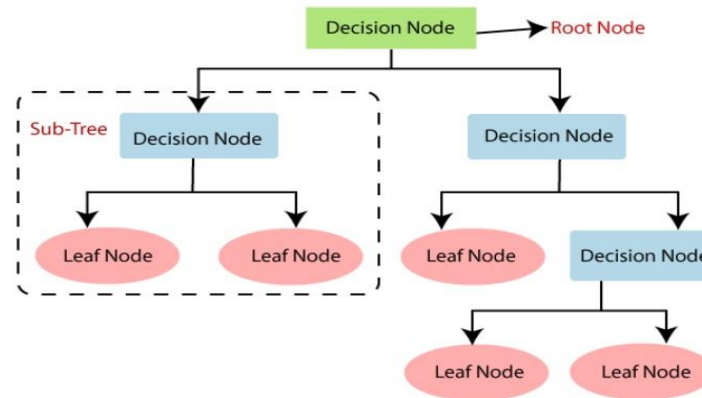


FIG 3.5.3 Decision Tree Diagram

- **Random Forest Classifier:**

Random Forest, a popular supervised learning algorithm, tackles both Classification and Regression in ML. It employs ensemble learning, combining multiple classifiers to boost model performance. By averaging predictions from numerous decision trees working on different data subsets, it enhances predictive accuracy. The majority-vote system prevents overfitting, with more trees in the forest ensuring higher accuracy.

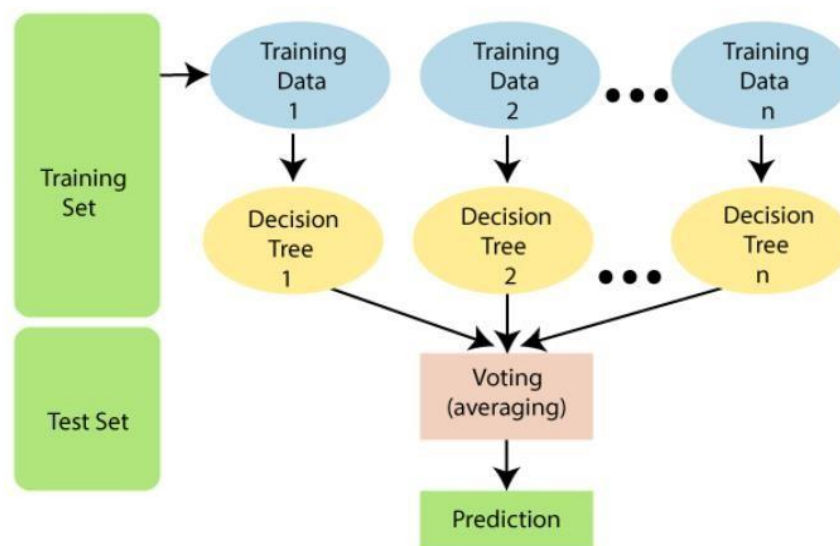


FIG 3.5.4 Random Forest Classifier Diagram



## CHAPTER 4

### TEST & RESULTS

#### 4.1 Test Cases

Table 4.1.1: Test Cases

Test Case No.	Testcase Name.	Test Case Description	Test Case Input	Test case output	Testcase status
1	Loan Prediction System	Analyzing the original and predicted values	Gender: Male Married: No Dependents:0 Education: Graduate Self-employed Income:5849 Loan amount:4.8520302639 Loanamount term:360.0 Credit history:1.0 Property/area: urban	Positive	Pass
			Gender: Male Married: yes Dependents:1 Education: Graduate Self-employed: No Income:4583 Loan amount:4.8520302639 Loan amount term:360.0 Credit history:1.0 Property/area: rural	Negative	Pass

## 4.2 Results

A screenshot of a web browser displaying a loan prediction interface. The browser's address bar shows '127.0.0.1:5000/predict'. The interface has a dark blue background. In the center, there is a white rectangular box with a light gray border. At the top of this box, it says 'Welcome!!!'. Below this, there are several input fields and buttons arranged in two columns. The first column contains: a text input with 'Male', a text input with '0', a text input with 'No', a text input with '0.0', a text input with '360.0', and a text input with 'Urban'. The second column contains: a text input with 'No', a text input with 'Graduate', a text input with '5849', a text input with '4.852030263919617', a text input with '1.0', and a button labeled 'Predict'. Below these inputs, the text 'Prediction result is Yes' and 'Model : Logistic Regression' is displayed.

Input	Value
Male	No
	Graduate
	5849
	4.852030263919617
	1.0
	Predict

Prediction result is Yes  
Model : Logistic Regression

FIG 4.2.1 Positive Result

A screenshot of a web browser displaying a loan prediction interface, similar to the one in FIG 4.2.1. The browser's address bar shows '127.0.0.1:5000/predict'. The interface has a dark blue background. In the center, there is a white rectangular box with a light gray border. At the top of this box, it says 'Welcome!!!'. Below this, there are several input fields and buttons arranged in two columns. The first column contains: a text input with 'Male', a text input with '1', a text input with 'No', a text input with '1508.0', a text input with '360.0', and a text input with 'Rural'. The second column contains: a text input with 'Yes', a text input with 'Graduate', a text input with '4583', a text input with '4.852030263919617', a text input with '1.0', and a button labeled 'Predict'. Below these inputs, the text 'Prediction result is No' and 'Model : Logistic Regression' is displayed.

Input	Value
Male	Yes
	Graduate
	4583
	4.852030263919617
	1.0
	Predict

Prediction result is No  
Model : Logistic Regression

FIG 4.2.2 Negative Result

## **CHAPTER 5**

### **CONCLUSION AND FUTURE ENHANCEMENT**

In conclusion, the implementation of a loan prediction system using machine learning has proven beneficial in evaluating creditworthiness and minimizing risk for financial institutions. The accuracy and adaptability of the model depend on continuous updates and retraining with fresh data. Future enhancements should include advanced algorithms, feature engineering, and selection techniques, as well as a focus on interpretability and compliance. Integrating non-traditional data sources, continuous monitoring, and incorporating user feedback for iterative improvement will contribute to the system's long-term effectiveness, providing a more comprehensive and reliable tool for informed lending decisions.

## REFERENCES

1. Kumar Arun, Garg Ishan, Kaur Sanmeet, —Loan Approval Prediction based on Machine Learning Approach, IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 18, Issue 3, pp. 79-81, Ver. I (May-Jun. 2016). Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., Tang, H.: Predicting diabetes mellitus with machine learning techniques. Bioinform. Comput. Biol. Sect. J. Front. Genet., published: 06 2018
2. Aboobyda Jafar Hamid and Tarig Mohammed Ahmed, —Developing Prediction Model of Loan Risk in Banks using Data Mining, Machine Learning and Applications: An International Journal (MLAIJ), Vol.3, No.1, pp. 1-9, March 2016.
3. S. Vimala, K.C. Sharmili, —Prediction of Loan Risk using NB and Support Vector Machine, International Conference on Advancements in Computing Technologies (ICACT 2018), vol. 4, no. 2, pp. 110-113, 2018.
4. Nikhil Madane, Siddharth Nanda, "Loan Prediction using Decision tree", Journal of the Gujrat Research History, Volume 21 Issue 14s, December 2019.

## APPENDIX

### SOURCE CODE

#### app.py code:

```

from flask import Flask, render_template, request, url_for
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
import joblib

app = Flask(__name__)

# Load the trained model
model = joblib.load('logistic_regression_model.pkl')

# Define a function to preprocess form data
def preprocess_form_data(data):
    labelencoder_X = LabelEncoder()
    data['Gender'] = labelencoder_X.fit_transform(data['Gender'])
    data['Married'] = labelencoder_X.fit_transform(data['Married'])
    data['Education'] = labelencoder_X.fit_transform(data['Education'])
    data['Self_Employed'] =
labelencoder_X.fit_transform(data['Self_Employed'])
    data['Property_Area'] =
labelencoder_X.fit_transform(data['Property_Area'])
    return data

@app.route('/')
def index():
    css_url = url_for('static', filename='css/styles.css')
    return render_template('index.html', css_url=css_url)

@app.route('/predict', methods=['POST'])
def predict():
    # Get form data
    form_data = request.form
    # Convert form data to DataFrame
    input_data = pd.DataFrame(form_data, index=[0])
    # Preprocess form data
    input_data = preprocess_form_data(input_data)
    # Drop 'Loan_ID' column if it exists
    if 'Loan_ID' in input_data.columns:
        input_data = input_data.drop('Loan_ID', axis=1)

```

```

# Make prediction
prediction = model.predict(input_data)
result=''
if prediction[0] == 'Y':
    result = result + 'Yes'
else:
    result = result + 'No'

# Render result template with prediction
return render_template('index.html', prediction_text=f"Prediction
result is {result} Model : Logistic Regression")

if __name__ == '__main__':
    app.run(debug=True)

```

### model.py:

```

# In[1]:
# Importing Libraries
import pandas as pd
import numpy as np                # For mathematical calculations
import seaborn as sns            # For data visualization
import matplotlib.pyplot as plt  # For plotting graphs
import pickle

# In[2]:
# Importing dataset
train = pd.read_csv('train_dataset.csv')
test = pd.read_csv('test_dataset.csv')

# Converting the values to number
train['Dependents'].replace('3+', 3,inplace=True)
test['Dependents'].replace('3+', 3,inplace=True)

# In[3]
# take a look at the top 5 rows of the train set, notice the column
"Loan_Status"
train.head()

# In[4]:

# take a look at the top 5 rows of the test set, notice the absense of
"Loan_Status" that we will predict
test.head()

# In[5]:
# Handling Missing Values

```

```
# Check How many Null Values in each columns
train.isnull().sum()

# Train Categorical Variables Missing values
train['Gender'].fillna(train['Gender'].mode()[0], inplace=True)
train['Married'].fillna(train['Married'].mode()[0], inplace=True)
train['Dependents'].fillna(train['Dependents'].mode()[0], inplace=True)
train['Self_Employed'].fillna(train['Self_Employed'].mode()[0],
inplace=True)
train['Credit_History'].fillna(train['Credit_History'].mode()[0],
inplace=True)

# Train Numerical Variables Missing Values
train['Loan_Amount_Term'].fillna(train['Loan_Amount_Term'].mode()[0],
inplace=True)
train['LoanAmount'].fillna(train['LoanAmount'].median(), inplace=True)

# Train Check if any Null Values Exits
train.isnull().sum()

# Test Check How many Null Values in each columns
test.isnull().sum()

# test Categorical Variables Missing values
test['Gender'].fillna(test['Gender'].mode()[0], inplace=True)
test['Married'].fillna(test['Married'].mode()[0], inplace=True)
test['Dependents'].fillna(test['Dependents'].mode()[0], inplace=True)
test['Self_Employed'].fillna(test['Self_Employed'].mode()[0],
inplace=True)
test['Credit_History'].fillna(test['Credit_History'].mode()[0],
inplace=True)

# test Numerical Variables Missing Values
test['Loan_Amount_Term'].fillna(test['Loan_Amount_Term'].mode()[0],
inplace=True)
test['LoanAmount'].fillna(test['LoanAmount'].median(), inplace=True)

# test Check if any Null Values Exits
test.isnull().sum()

# In[6]:

# Outlier treatment
train['LoanAmount'] = np.log(train['LoanAmount'])
test['LoanAmount'] = np.log(test['LoanAmount'])
```

```

# In[7]:

# Separating the Variable into Independent and Dependent
X = train.iloc[:, 1:-1].values
y = train.iloc[:, -1].values

# In[8]:

# Converting Categorical variables into dummy
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X = LabelEncoder()

# Gender
X[:,0] = labelencoder_X.fit_transform(X[:,0])

# Marriage
X[:,1] = labelencoder_X.fit_transform(X[:,1])

# Education
X[:,3] = labelencoder_X.fit_transform(X[:,3])

# Self Employed
X[:,4] = labelencoder_X.fit_transform(X[:,4])

# Property Area
X[:, -1] = labelencoder_X.fit_transform(X[:, -1])

# In[9]:

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.20, random_state = 0)

# In[10]:

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

```



```
# In[11]:

# Fitting Logistic Regression to our training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)

# In[12]:

# Predicting the results
y_pred = classifier.predict(X_test)

# In[13]:

# Printing values of whether loan is accepted or rejected
y_pred[:100]

# In[14]:

# import classification_report
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

# In[15]:

# implementing the confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
# f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(cm, annot=True, fmt="d")
plt.title('Confusion matrix of the classifier')
plt.xlabel('Predicted')
plt.ylabel('True')
# In[16]:
# Check Accuracy
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
# In[17]:
# Applying k-Fold Cross Validation
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y =
y_train, cv = 10)
accuracy = accuracies.mean()
pickle.dump(classifier, open('logistic_regression_model.pkl' , 'wb'))
# accuracies.std()
# In[ ]:
```

## Styles.css

```
body {
    margin: 0px;
    padding: 0px;
    display: flex;
    /* background: linear-gradient(to right, #a8dad9, #457b9d); */
    /* background: linear-gradient(131deg, rgb(253, 124, 140) 0.48%,
rgb(253, 112, 136) 51.67%, rgb(255, 211, 165) 100.00%); */
    /* background: linear-gradient(270deg, rgb(65, 80, 95) 0.48%,
rgb(36, 37, 38) 67.74%); */
    /* background: linear-gradient(132deg, rgb(251, 165, 116) 0.00%,
rgb(216, 245, 251) 100.00%); */
    /* background: linear-gradient(132deg, rgb(59, 55, 106) 0.00%,
rgb(0, 143, 186) 50.00%, rgb(255, 149, 213) 100.00%); */
    background: linear-gradient(90deg, hsla(221, 45%, 73%, 1) 0%,
hsla(220, 78%, 29%, 1) 100%);
    justify-content: center;
    align-items: center;
}

#container {
    height: 600px;
    width: 500px;
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    align-items: center;
    /* margin-left: 50vh;
margin-right: 50vh; */
    margin-top: 10vh;
    border: grey solid ;
    background-color: whitesmoke;
    border-radius: 2.5%;
}
```

```

}
h1 {
    font-size: 30px;
    font-family: monospace;
    padding: 10px;
}
#form-container{
    display: flex;
    flex-wrap: wrap;
    justify-content: left;
    height: 450px;
    width: 400px;
    z-index: 1;
    box-shadow: 5px 5px 5px rgba(0, 0, 0, 0.3);
    border-radius: 2.5%;
    margin-bottom: 25px;
}
#form{
    background-color: whitesmoke;
}
#form-container input{
    background-color: white;
    color: black;
    font-size: 14px;
    font-weight: 600 solid;
    border-radius: 10px;
    border-style: hidden;
    height: 35px;
    width: 150px;
    margin: 10px;
    padding: 2px 2px 2px 2px;
}
button{
    background: linear-gradient(131deg, rgb(253, 124, 140) 0.48%,
rgb(253, 112, 136) 51.67%, rgb(255, 211, 165) 100.00%);
    background: #4b749f;
    width: 150px;
    border-radius: 5%;
    font-size: 14px;
    margin-bottom: 5px;
}
h2 {
    font-size: 20px;
    font-family: monospace;
    padding: 10px;
    width: 300px;

```

```
}
```

## index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <link rel="stylesheet" href="{{ url_for('static',
filename='css/styles.css')}}">
  <title>Loan Predictor</title>
</head>

<body>
  <div id="container">
    <h1>Welcome!!!</h1>
    <div id="form-container">
      <form id="form" action="{{url_for('predict')}}"
method="post">
        <input type="text" placeholder="Gender" name="Gender"
required/>
        <input type="text" placeholder="Married" name="Married"
required/>
        <input type="text" placeholder="Dependents" name =
"Dependents" required/>
        <input type="text" placeholder="Education"
name="Education" required/>
        <input type="text" placeholder="Self Employed"
name="Self_Employed" required/>
        <input type="text" placeholder="Applicant Income"
name="ApplicantIncome" required/>
        <input type="text" placeholder="Co-Applicant Income"
name="CoapplicantIncome" required/>
        <input type="text" placeholder="Loan Amount"
name="LoanAmount" required/>
        <input type="text" placeholder="Loan Amount Term" name=
"Loan_Amount_Term" required/>
        <input type="text" placeholder="Credit History"
name="Credit_History" required/>
        <input type="text" placeholder="Property Area" name
="Property_Area" required/>
        <button type="submit">Predict</button>
      </form>
    <br>
```

```
        <h2>{{prediction_text}}</h2>
    </div>
</div>
</body>
</html>
```