HEART DISEASE PREDICTION BY USING CASE BASED REASONING (CBR)

CHAN HUE WAH

Bachelor of Computer Science (Computer System & Networking) with Honors

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : CHAN HUE WAH

Date of Birth

Title : HEART DISEASE PREDICTION BY USING CASED BASED REASONING (CBR)

Academic Session : SEMESTER I 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)

Date: 18/1/2023

(Supervisor's Signature)

NUR SHAZWANI BINTI KAMARUDIN
Date: 11/1/2023

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

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title HEART DISEASE PREDICTION BY USING CASED BASED REASONING (CBR)

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date: 3/6/2022

Stamp: DR. NUR SHAZWANI KAMARUDIN
PENSYARAH KANAN
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG.
TEL : 09-424 4736

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

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 (Computer System & Networking) with Hons.

(Supervisor's Signature)

Full Name : NUR SHAZWANI BINTI KAMARUDIN

Position : SENIOR LECTURER

Date : 11/1/2023

(Co-supervisor's Signature)

Full Name :

Position :

Date :

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 : CHAN HUE WAH

ID Number : CA19059

Date : 18 January 2023

HEART DISEASE PREDICTION BY USING CASE BASED REASONING (CBR)

CHAN HUE WAH

Thesis submitted in fulfillment of the requirements.
for the award of the degree of
Bachelor of Computer Science (Computer System & Networking) with Honors

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

I'd want to express my heartfelt gratitude to my supervisor, Dr. Nur Shazwani Binti Kamarudin, for accepting me as her supervisee and accepting the title of Heart Disease Prediction Using Intelligent system, which I offered. Dr. Nur Shazwani Binti Kamarudin also guide and help me a lot in doing this research and I came to know and learn may new things and knowledge. I consider myself really fortunate to have her mentorship throughout this project. Once again, I want to thanks to Dr. Nur Shazwani Binti Kamarudin.

Next, I'd want to thank my family members for always being there for me during this project and giving me advise when I'm stressed and having some difficulties when working on this project.

Finally, I'd want to show my heartfelt gratitude to my friends for constantly assisting me during this project and offering me advise or direction when I was experiencing difficulty with some aspect of it.

ABSTRAK

Kajian ini merupakan ramalan penyakit jantung dengan menggunakan pembelajaran mesin. Penyakit ramalan adalah penting dalam bidang perubatan. Ia sukar untuk mendapatkan keputusan yang tepat dengan menggunakan kaedah tradisional iaitu pengalaman doktor. Oleh itu, untuk mengatasi masalah ini, pembelajaran mesin akan digunakan untuk menggantikan pendekatan tradisional. Terdapat pelbagai pembelajaran mesin yang wujud tetapi, dalam penyelidikan ini, hanya tiga teknik pembelajaran mesin yang dipilih iaitu Fuzzy Logic, Neural Network dan Cased-Based Reasoning (CBR) akan dikaji. Perbandingan dari segi ketepatan akan dibuat antara tiga teknik pembelajaran mesin yang dipilih. Seterusnya, hanya Cased-Based Reasoning (CBR) yang akan dipilih untuk melakukan ramalan penyakit jantung. Dalam proses ramalan, set data penyakit jantung akan melalui pra-pemprosesan data untuk membersihkan data dan pemisahan data untuk memisahkan data kepada data latihan dan ujian. Kemudian, selepas data boleh digunakan, pembelajaran mesin yang dipilih akan digunakan untuk meramalkan hasil penyakit jantung.

ABSTRACT

This study provides an overview of heart disease prediction using intelligent system. Disease prediction is an important task in the medical industry. It is hard to get an accurate result by using the traditional method which is doctor's experience. Therefore, to overcome these issues, the intelligent system will be applied to replace the traditional approach. There are other intelligent system approaches available, but just three will be studied in this study such as Fuzzy Logic, Neural Network, and Cased-Based Reasoning (CBR). The comparison in term of accuracy will be made among the three chosen intelligent system techniques. Next, only the Cased-Based Reasoning (CBR) will be selected to perform the heart disease prediction. During the prediction phase, the heart disease dataset will go through data pre-processing to clean it and data splitting to divide it into training and testing data. Then, the selected intelligent system will then be used to identify the outcome of the heart disease after the data has been useful.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\sigma(y)$	Sigmoid Function
$\overrightarrow{W}$	Input
$\vec{x}$	Output
$\frac{1}{1+\ e^{-y}}$	Activation Function
δ_k	Error term calculated for each network output unit
δ_h	Error term calculated for each network hidden unit
W_{kh}	Weightage
$\eta\delta_j x_{ji}$	Updated Weight

LIST OF ABBREVIATIONS

CAD	Coronary Artery Disease
WHO	World Health Organization
ML	Intelligent system
AI	Artificial Intelligence
UCI	Intelligent system Repository
ECG	Electrocardiography
ANN	Artificial Neural Network
BA	Backpropagation Neural Network
CBR	Case-Based Reasoning
cp	Chest pain
trestbps	The person’s resting blood pressure(mm Hg)
chol	Cholesterol
fbs	Fasting blood sugar
restecg	Resting electrocardiographic results
thalach	The person’s maximum heart rate achieved.
exang	Exercise induced angina.
oldpeak	ST depression
slope	Slope of the peak exercise ST segment
cs	The number of major vessels
thal	Thalassemia
UMP	Universiti Malaysia Pahang

CHAPTER 1

INTRODUCTION

1.1 Introduction

Heart disease has caused a high level of concern among researchers since one of the most difficult aspects was to get an accurate and right prediction(Himanshu Sharma & M A Rizvi, 2017). The word "heart disease" describes a variety of heart problems(Chen et al., 2011). The four most frequent kinds of heart disease are coronary artery disease (CAD), arrhythmia, heart valve disease, and heart failure. According to a World Health Organization (WHO) study, a large number of people worldwide suffer from heart disease each year(Ramalingam et al., 2018). In United States (U.S.) also, heart disease has risen to become the top cause of mortality among people(Rahma Atallah & Amjed Al-Mousa, 2022).

In these days, there are various of techniques and tools that used to predict the heart disease, but it seems not efficiency to the medical field(Jabbar et al., 2012). This is because most methods are inefficient in calculating or predicting the outcome of heart disease in individuals, or the equipment are too expensive. Therefore, it was too challenging for them to predict the heart disease. Due to these challenges, it gives the motivation for us to do research on the prediction system which can predict heart disease accurately.

Hence, to resolve this concern, intelligent system technique will be applied to accurately identify heart disease. Intelligent system (ML) is a part of artificial intelligence (AI) that helps researchers to improve their prediction accuracy without being explicitly taught to do so. Intelligent system algorithms predict output result or outcome by using previous data as input. These data's attributes or features will be utilised to identify the heart disease outcome, such as positive,1 or negative,0. Intelligent system algorithms are

effective in predicting outcomes because they can handle massive amounts of data(Rajdhan et al., 2020) and it capable of accurately predicting the outcome of heart disease(Shah et al., 2020). As a result, in this study, intelligent system approaches will be studied and presented in order to properly predict the heart disease result.

1.2 Problem Statement

Heart disease has become a disease that can cause a people death if not predict accurately as there are many factors of the heart disease. Coronary artery disease (CAD), arrhythmia, heart valve dysfunction, and heart failure are a few examples. Therefore, it was challenging to identify whether the patient has heart disease or not. In a traditional approach, the doctors in the hospital or clinic will base on their experience and knowledge to diagnosis the result of heart disease for every patient(Himanshu Sharma & M A Rizvi, 2017). Therefore, the traditional approach will lead to not accurate in prediction. This is because sometimes there will be human error like the doctors are wrongly predict the heart disease result based on their experience. The hospital and clinic also collect and kept a large number of their patient result record in a folder. Those result record will leave behind and become raw data. This method can lead to unfavourable biases, errors, and additional medical costs, all of which have an influence on the quality of care delivered to patients.

Currently, the heart disease can be predicted by using the modern ways which is by using intelligent system. By using intelligent system methods, the doctors are only requiring keying in the data according to each input variable to get the outcome. These approaches will aid the doctors in hospital or clinic to provide the outcome result to the patients, but it has its limitation. Nowadays, there are many varieties of intelligent system techniques that can be used for prediction but not every intelligent system techniques are suitable. This is because some of the intelligent system techniques will only produce the images or text outcome where the outcome for the heart disease prediction is in binary form.

Besides that, for those intelligent system techniques that suitable to predict the result of heart disease has a critical problem which was accuracy. Every intelligent system model is using different algorithm to predict the heart disease result. The data cleansing procedure is very significant in deciding the accuracy of the results. The practise of fixing or eliminating erroneous, corrupted, badly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. It is essential to identify and remove errors and duplicate data in order to establish a dependable dataset and increase the accuracy of the final result. Therefore, each intelligent system model will have different accuracy in predicting the result outcome.

1.3 Objectives

The objective of this paper is to predict heart disease. There are three objectives in this study:

1) To study the effectiveness of prediction if the patient suffers from heart disease.

2) To develop the prediction technique such as Fuzzy Logic, Neural Network and Case Base Reasoning (CBR) for heart disease.

3) To evaluate the outcome of heart disease using the selected intelligent system approaches.

1.4 Scopes

The scope of the project are:

1.4.1 Research Scope

1) Get the heart disease dataset, "heart" from the Kaggle website, https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset.

2) Divide the dataset into training (60%) and testing (40%).

3) This study will execute the Case Base Reasoning (CBR) algorithm to measure the accuracy.

4) The prediction will display the result of the heart disease either 0 for negative or 1 for positive.

1.4.2 Development Scope

1) The Case Base Reasoning (CBR) algorithm or Intelligent system technique will be selected to predict the heart disease result outcome.

1.5 Significant of this Project

1) Doctors

The doctors in the hospital able to predict the patient heart disease result in an automatic way rather than using the traditional way which was predicting the result of heart disease base on the doctor's knowledge and experience. The doctors also can obtain the result of heart disease whether positive or negative in a fastest way when compared to the traditional approach that require a few hours or a few days to acquire the result.

2) Patients

The patients able to receive the heart disease result quickly. Therefore, for those patients who has the positive result of heart disease may take further treatment immediately to save their life. Other than that, for those patients who have early sign of heart disease symptom, they can receive the treatment early before the disease are spreading.

1.6 Thesis Organization

In this thesis organization, it consists of five chapter. The first chapter contains the introduction of the research title. It also includes the problem statement, objectives, scopes, significance of the project, and thesis organisation.

The second chapter is a review of the literature. It have the comparison of others technique or algorithm that able to predict the heart disease result. It also contains the advantages and disadvantages of each comparison.

The third chapter is methodology. This is the section where the methodology used in modelling the research project is illustrated.

The chapter four is review on the expected result and discussion. In this chapter presented the actual implementation of the research project.

The chapter five is review on the conclusion. It involves in reviewing the findings & results and discusses the outcome of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This Chapter 2 focuses on a review of the current system or techniques that have been used previously. This section will also compare and discuss past research on the prediction of heart disease using intelligent system.

2.2 Previous Research Works

2.2.1 Design of Heart Disease Diagnosis System using Fuzzy Logic

The authors of this research paper(Tanmay Kasbe & Ravi Singh Pippal, 2017), Tanmay Kasbe & Ravi Singh Pippal used Fuzzy Logic Intelligent system to diagnosis the heart disease. Fuzzy logic is a powerful reasoning method that performs best at dealing with uncertainty data. Heart disease diagnosis is an essential criterion in daily life, and yet due to a lot of uncertainty and risk factors, heart disease diagnosis can be difficult for experts at times. When a heart attack is detected, the speed of detection is critical in order to save the patient's life and prevent heart damage.

The authors used the heart disease dataset that taken from the UCI(Intelligent system Repository) to do the research. The dataset includes ten input variables, including Systolic Blood Pressure, Serum Cholesterol, Maximum Heart Rate, Chest Pain, Fasting Blood Sugar, Old Peak, Electrocardiography (ECG), Thallium Scan, gender, and age. The output attribute will be the Result. Next, the tools that used by the authors to diagnosis the heart disease result was MATLAB. The accuracy of the study article, heart disease diagnostic system using Fuzzy Logic, is 93.33 %. Fuzzy Logic Intelligent system

is used to diagnose heart diseases through a fuzzy expert system with membership function, fuzzy input and output variables, and a fuzzy rule base.

In the first stage, fuzzy membership functions will be used for implementing in the MATLAB tool. The fuzzy membership functions is used to convert the crisp input from the heart disease dataset to provide the fuzzy inference system. The authors selected two membership function such as triangular function and trapezoidal function. The lower limit, ‘a’ an upper limit, ‘b’ and a value of' ‘m' will clarify the triangular function. The range of the ‘m’ value will be a < m < b. Figure 2.1 below shows the triangular function. Then in the trapezoidal function, ‘a’ will represented the lower limit, ‘d’ will be the upper limit. Other than that, it also has lower support limit that defined as ‘b’ and an upper Support limit ‘c’. Figure 2.2 below shows the triangular function.

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{m-a}, & a < x \le m \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x \ge b \end{cases}$$

Figure 2.1 shows Triangular Function

$$\mu_A(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases}$$

Figure 2.2 shows Trapezoidal Function

The fuzzy expert system design is important in the second step for determining the input and output variables of the heart disease dataset. The UCI (Intelligent system Repository) heart disease dataset contains 10 input variables and 1 output variable.

The fuzzy data rule base will be declared at the final stage. The fuzzy rule base, which is one of the most important aspects of the fuzzy interface system, is used to determine the quality of the heart disease outcome. Fuzzy rule is a conditional statement. The form of fuzzy rules is given by IF THEN statements. The fuzzy data rule base declared by using AND/OR logic operator with either single attribute or combination of attributes. The authors had declared 86 rules by using the right combination of attributes. Thus, the more the new rule base, the higher the accuracy.

The advantage of Fuzzy Logic is it is dynamic and allows for rule changes. It even accepts the imprecise input information. Then, the disadvantage of this approach is the accuracy of these systems is compromised since they rely on imprecise data and inputs. However, the limitation of this techniques is the rule of the fuzzy logic is based on the predefined rules and if the rules are flawed, the result that predicted.

2.2.2 Prediction of Heart Disease Using Neural Network

Another research on the prediction of heart disease was proposed by Tülay Karay1lan and Özkan K1l1ç. The heart disease was predicted by using the Neural Network Intelligent system technique or algorithm(Adali & Akdeniz Üniversitesi, 2022). According to the researchers, developing a medical diagnosis system based on machine learning for the prediction of heart disease provides a more accurate diagnosis and lowers treatment costs. For the prediction system, the Backpropagation Algorithm, a widely used Artificial Neural Network learning methodology, was used to satisfy this need.

The dataset of this research was taken from the Cleveland database. This dataset has 303 cases and 76 attributes, but the authors only used 14 of the attributes. Figure 2.3 below shows all the attributes of the heart disease dataset in a table form. In this study, 13 variables will be used as input and 1 attribute, Num will be used as output. The authors of this study applied MATLAB R2015a tools to identify heart disease using a neural network approach. Backpropagation Algorithm with Artificial Neural Network (ANN) learning approach will be utilised in this study to predict heart disease.

Clinical Features	Description
Age	Age
Ca	Number of major vessels (0-3) colored by flourosopy
Chol(mg/dl)	Serum cholesterol
Cp	Chest pain type
Exang	Exercise induced angina
Fbs	Fasting blood sugar
Num	Diagnosis of heart disease
Oldpeak	ST depression induced by exercise relative to rest
Restecg	Resting electrocardiographic results
Sex	Gender
Slope	The slope of the peak exercise ST segment
Thal	3=normal ; 6 = fixed defect; 7= reversible defect
Thalach	Maximum heart rate achieved
Trestbps(mmHg)	Resting Blood Pressure

Figure 2.3 Cleveland Heart Disease Dataset Attributes

A multilayer perceptron neural network was built using the Artificial Neural Network (ANN) learning approach for the heart disease prediction system. It contains three layers like an input layer, a concealed layer, and an output layer. The input layer will have 13 neurons because the dataset has 13 input variables. Next, the authors used 3 neurons for hidden layer. The number of neurons in the input layer will be increased by one at a time by measuring their performance and selecting the best one. It will be good if the number of hidden layer neuron equal or same with the neuron of the input and output layer. Then the output layer will have 2 neurons. This is because the value of the output in the dataset was either disease absence or disease presence that represented by 0 and 1 respectively.

Besides that, the Backpropagation Neural Network algorithm (BA) will be used to build the multilayer neural networks. It is also known as an error-back propagation algorithm since it uses an error-correction learning rule base. The heart disease dataset will be split into 3 parts which are training, testing and validation. The training data will be used in the process of Backpropagation. At first, small random number will be use as the weightage for all the networks. The training data is then used as input, and the output for each unit is computed using the Sigmoid Function, $o = \sigma(\vec{W} . \vec{x})$, $\sigma(y) = \frac{1}{1+ e^{-y}}$ where $\vec{W}$ is the vector unit of weightage values while $\vec{x}$ is the vector unit for network input values. After that, error calculation will begin with the error signal (δ) calculation for every network output that propagated as input to all neurons in the network. The first error term, δ_k with the equation of $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$ will be used to compute for every network output unit where o_k reflect the network output for output unit k and define the target output for output unit k. The second error term, δ_h with the equation of $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} W_{kh}\delta_k$ will be used to compute for every hidden unit h where W_{kh} show the network weightage from the hidden unit h to the output unit k. Therefore, every network weightage will be updated by using the equation of $W_{ji} \leftarrow W_{ji} + \Delta W_{ji}$ where $\Delta W_{ji} = \eta\delta_j x_{ji}$. The η indicate the learning rate while x_{ji} refers to the input from unit i and unit j. The accuracy result from this study was 95%.

The advantage of Neural Network is at the ANN side. ANNs able to solve issues with the target function, and the output can be discrete, real, or a vector of many real or discrete-valued attributes. The presence of mistakes in the training data has no effect on the outcome. Then, the disadvantage of Neural Network is it is data-dependency as it

requires huge amount of data for the training phase, if not then the predicted result will be not accurate. However, the limitation of this techniques is the training phases require longer time and not every problem can be solved by using neural network. The data that contains a lot of attributes and many is not suitable in this technique.

2.2.3 Decision Support System in Heart Disease Diagnosis by Case Based Recommendation

Decision support system in heart disease diagnosis by case-based recommendation author is proposed by Prinsha Prakash(Prakash, 2015). According to the authors, most medical decisions must be made quickly, simply relying on the Doctor's unaided memory. Continuous training and recertification procedures push the Doctor to retain more important information in mind at all times, but limitations of human memory and recall, along with the expansion of knowledge, ensure that most of what is known cannot be known by most people. In order to identify the cardiac condition that was the result of this research, the author used a method known as Case Based Reasoning (CBR), which is an intelligent system. These methods are able to assist in organising, storing, and retrieving the information that is necessary when dealing with each difficult case, as well as propose appropriate diagnostic, prognostic, and therapeutic judgements and decision making processes.

Case-based reasoning differs from other AI strategies like knowledge-based systems in a number of ways (KBS). CBR uses the particular knowledge of previously encountered, concrete issue situations rather than relying only on generic knowledge of a problem area or drawing links along generalised relationships between problem descriptors and conclusions. Additionally, CBR provides incremental, continuous learning since each time a problem is resolved, a new learning is maintained and may be used to tackle similar issues in the future.

These authors use the heart disease dataset that can be found at Cleveland Clinic Foundation. Figure 2.4 provides a visual representation of the Cleveland Clinic Foundation's statistics on heart disease. There are fourteen properties in the dataset, thirteen of which are input variables and one of which is an output or result variable. In order to diagnose the cardiac illness that was the result of this research, the author used the Case Based Reasoning (CBR) approach. This was done with the help of the MATLAB programme. The CBR method or algorithm is broken down into four stages, which are retrieve, reuse, revise, and retain.

Clinical Features	Description
Age	Age
Ca	Number of major vessels (0-3) colored by flourosopy
Chol(mg/dl)	Serum cholesterol
Cp	Chest pain type
Exang	Exercise induced angina
Fbs	Fasting blood sugar
Num	Diagnosis of heart disease
Oldpeak	ST depression induced by exercise relative to rest
Restecg	Resting electrocardiographic results
Sex	Gender
Slope	The slope of the peak exercise ST segment
Thal	3=normal ; 6 = fixed defect; 7= reversible defect
Thalach	Maximum heart rate achieved
Trestbps(mmHg)	Resting Blood Pressure

Figure 2.4 Cleveland Clinic Foundation Heart Disease Dataset Attributes

Since the revise and retain process may be done manually, the author just used the retrieve and reuse technique in this research. The author uses scanned images of 2D echo cardio graphic, EEG, ECG, and heart images as input, and image processing techniques are used to validate the normal or abnormal condition of the heart. Then the result data will be saved in the case base. This result data will be reused by the doctor during the patient consultation by retrieving comparable instances from the case base and providing the heart disease result instantly. Aside from that, if a special case does not exist in the case base, the revise and retain approach will be utilised to get the result and preserve it in the case base for future usage.

The advantage of Case Based Recommendation (CBR) is that it is not complex to implement. CBR using case-based knowledge, therefore the output or result can be proposed rapidly. Then, the disadvantage of this approach is the CBR required more previous data cases to predict the result accurately. So, it needs huge amount of storage to store the previous cases. However, the limitation of this techniques is It takes longer time to compute the similarity between the training and testing data during the revise phases.

Table 2.1 Comparison Between Previous Research Works

Contents	Research 1	Research 2	Research 3
Research and Author	Design of Heart Disease Diagnosis System using Fuzzy Logic (Tanmay Kasbe et al., 2017)	Prediction of Heart Disease Using Neural Network (Tülay Karay1lan et al., 2017)	Decision Support System in Heart Disease Diagnosis by Case Based Recommendation (*Prinsha Prakash et al., 2015*)
Objective	Using Fuzzy Logic, measure the accuracy of heart disease diagnosis.	Analyze the efficiently of heart disease prediction accurate rate by using Neural Network	Design an intelligent clinical decision support system to aid in the diagnosis of heart disease.

Technique	Fuzzy Logic System (Fuzzy Membership Function, Fuzzy Expert System, Fuzzy Input and Output Variables and Fuzzy Rule Base)	Neural Network (Artificial Neural Network (ANN) and Backpropagation)	Case Based Reasoning (CBR) (Retrieve, Reuse, Revise and Retain)
Architecture	- Fuzzy Membership Function - Fuzzy Expert System, Fuzzy Input and Output Variables - Fuzzy Rule Base	- Artificial Neural Network (ANN) - Backpropagation	- Retrieve - Reuse - Revise - Retain
Data	UCI (Intelligent system Repository)	Cleveland Clinic Foundation	Cleveland Clinic Foundation

Features	There are 10 attributes and 1 output: ▪ Systolic Blood Pressure (BP) ▪ Serum Cholesterol (SCHL) ▪ Maximum Heart (MHR) ▪ Chest Pain (CP) ▪ Fasting Blood Sugar (FBS) ▪ Old Peak (OP) ▪ Electrocardiography (ECG) ▪ Thallium Scan (TScan) ▪ Gender ▪ Age ▪ Result (Output)	There are 13 attributes and 1 outputs: ▪ Age ▪ (Ca) Number of major vessels (0-3) coloured by fluoroscopy. ▪ (Chol) Serum cholesterol ▪ (Cp) Chest pain type ▪ (Exang) Exercise induced angina ▪ (Fbs) Fasting blood sugar ▪ (Oldpeak) ST depression induced by exercise relative to rest ▪ (Restecg) Resting electrocardiographic results ▪ Gender ▪ (Slope) The slope of the peak exercise ST segment ▪ Thal (3=normal ; 6 = fixed defect; 7= reversible defect)	There are 13 attributes and 1 outputs: ▪ Age ▪ Gender ▪ (Cp) Chest pain type ▪ (Trestbps) Resting Blood Pressure (in mmHg) ▪ (Chol) Serum cholesterol ▪ (Fbs) Fasting blood sugar ▪ (Restecg) Resting electrocardiographic results ▪ (Thalach) Maximum heart rate achieved ▪ (Exang) Exercise induced angina

		▪ (Thalach) Maximum heart rate achieved ▪ (Trestbps) Resting Blood Pressure (in mmHg) ▪ (Num) Diagnosis of heart disease	▪ (Oldpeak) ST depression induced by exercise relative to rest ▪ (Slope) The slope of the peak exercise ST segment ▪ (Ca) Number of major vessels (0-3) coloured by fluoroscopy. ▪ Thal (3=normal ; 6 = fixed defect; 7= reversible defect) ▪ (Result) Diagnosis of heart disease

2.3 Summary

According to Table 2.4, each research study on a different intelligent system technique to predict or diagnose the outcome of a heart disease. The three researchers used different algorithms, methodologies, or approaches to apply intelligent system. Therefore, it has different outcome.

Every intelligent system technique or algorithm have its own advantages and disadvantages. So, the suitable intelligent system techniques or algorithms will be select in this research or the heart result disease prediction.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter 3 will discuss about the methodology that used in the heart disease prediction by using selected intelligent system. This chapter will cover the Research Framework, Research Requirement, Proposed Design (in Flowchart), and data design.

3.2 Research Framework

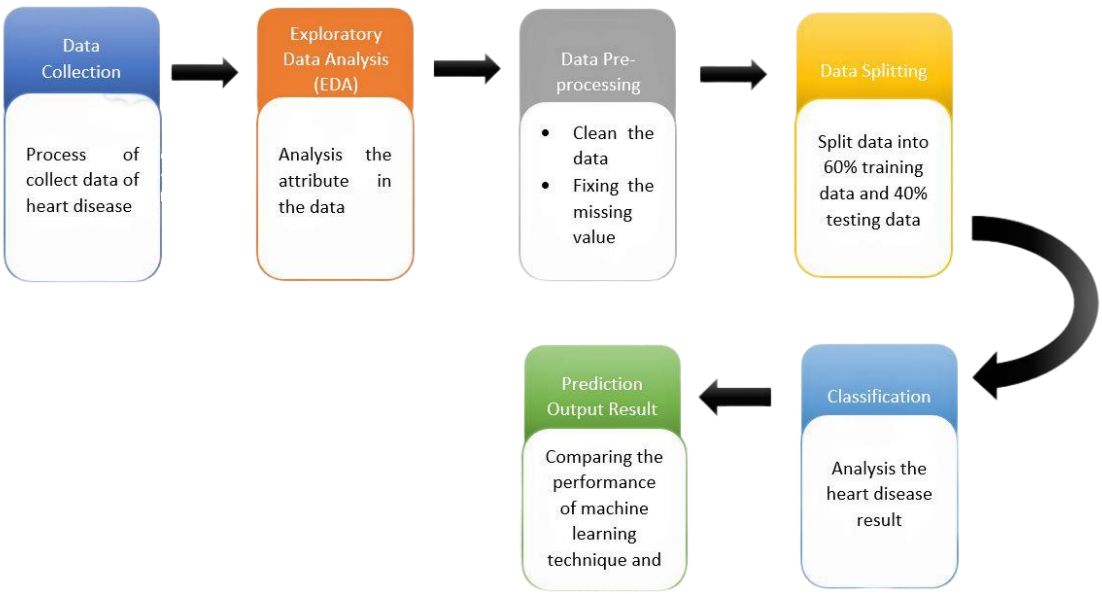


Figure 3.1 Research Framework

Table 3.1 Research Framework Description

Research Framework	Description
Data Collection	The heart disease dataset is a long-term study. This dataset was provided, and it was available in public. The heart disease dataset can be searched at the Kaggle website(David Lapp, 2019) and available to download if it is suitable. This heart disease dataset was created by David Lapp at the year of 2019. This dataset contains 14 column of attribute where 13 attributes are the input and 1 attribute is the output. This dataset also contains about 1025 data of cases.
Exploratory Data Analysis (EDA)	For the non-technical user will hard understand the data and its attribute. Therefore, EDA will be applied to show the details type of every attribute in the dataset.
Data Pre-processing	The heart disease dataset that downloaded from the Kaggle website was a raw data and it still cannot be use for prediction. It is crucial to ensure the dataset clean so that it can predict the heart disease result accurately and effectively. Therefore, the data will go through the data pre-processing stage to ensure the data in the dataset are usable. Data pre-processing have many techniques like Data Normalization, Data Cleaning, Data Transformation, and Data Reduction. By using the technique of data pre-processing, the heart disease data will be converted from raw data to usable and clean data.
Data Splitting	The heart disease dataset that had been went through the data pre-processing will be used to perform data splitting. The data splitting is aimed to separate the

	dataset into training data and testing data. A suitable ratio is required to split the data into training and testing to get the high accuracy in prediction. The splitting data ratio can be 80:20, 70:30 or 60:30 base on the dataset. The training data will be use as the input to predict the heart disease result while testing data use to evaluate its performance.
Classification	The chosen intelligent system algorithm or technique will be used to predict the heart disease result. The outcome result of the prediction will be either 0 or 1. The value 0 indicates that the patients has no heart disease while the value 1 shows that the patient has the heart disease.
Prediction Output Result	The prediction of heart disease will be conducted by using the training and testing data in the intelligent system technique or algorithm. Data will be trained to determine which intelligent system algorithm or technique will produce the best prediction result. The Case Based Reasoning (CBR) intelligent system technique will be used. These techniques will be used to test their performance. Then after the result is predicted, it will be used for visualized.

3.3 Research Requirement

3.3.1 Input

The input of the heart disease is in the csv file. The heart disease dataset contains 1025 set of cases. Apart from that, this dataset also consists of 14 columns of attributes where 13 columns of the attributes are the antecedent or input and 1 column will be the consequent or output. As a result, this study will only use the 13 variables to predict the heart disease outcome.

3.3.2 Output

The output of the heart disease dataset will be in the binary number. The output refers to the patient heart disease result. If the output is value 0, then the patient has no heart disease and if the value is 1 then the patient has heart disease. Lastly, the predicted output will compare with the actual output as shown in the Table 3.2 below.

Table 3.2 Predicted and Actual Output

Cases	**Output**	
	Predicted	**Actual**
…	…	…
141	1	1
142	1	1
143	1	0
…	…	…

3.3.3 Process Description

First, the heart disease dataset from the Kaggle website will be downloaded as .csv file. Then, the Jupyter Notebook software will be used to implement the intelligent system Python code. The heart disease dataset .csv file will be import to the software and perform the Exploratory Data Analysis (EDA) process. After that, the data in the dataset will be go through data pre-processing process to convert the raw data to high quality data by using data normalization technique. After data pre-processing, the data will be split into the ratio of 60:40 where 60% will be the training data and 40% will be the testing data. Next, the training and testing data will be apply to Case Based Reasoning (CBR). The next stage will be using the Case Based Reasoning (CBR) intelligent system algorithm or technique to predict the heart disease result. The more details explanation will provide in the flowchart section.

3.3.4 Constraints and Limitation

In this research, there are some constraints. In this study, it only focus on one intelligent system algorithm. Because there are many other techniques are not considered or offered in this study, claiming which is the best technique from others intelligent system techniques becomes the constraint. Next, since intelligent system approaches differ, there is only a limited amount of time to assess and research them all. Therefore, it is difficult to investigate, analyse, and assess all the others intelligent system approaches.

The limitation for this heart disease prediction research is static. This heart prediction is not a real time prediction. This is because the dataset that taken from the website do not contain the real latest heart disease data. Therefore, different heart disease dataset will have different result and the accuracy.

3.3.5 Flowchart

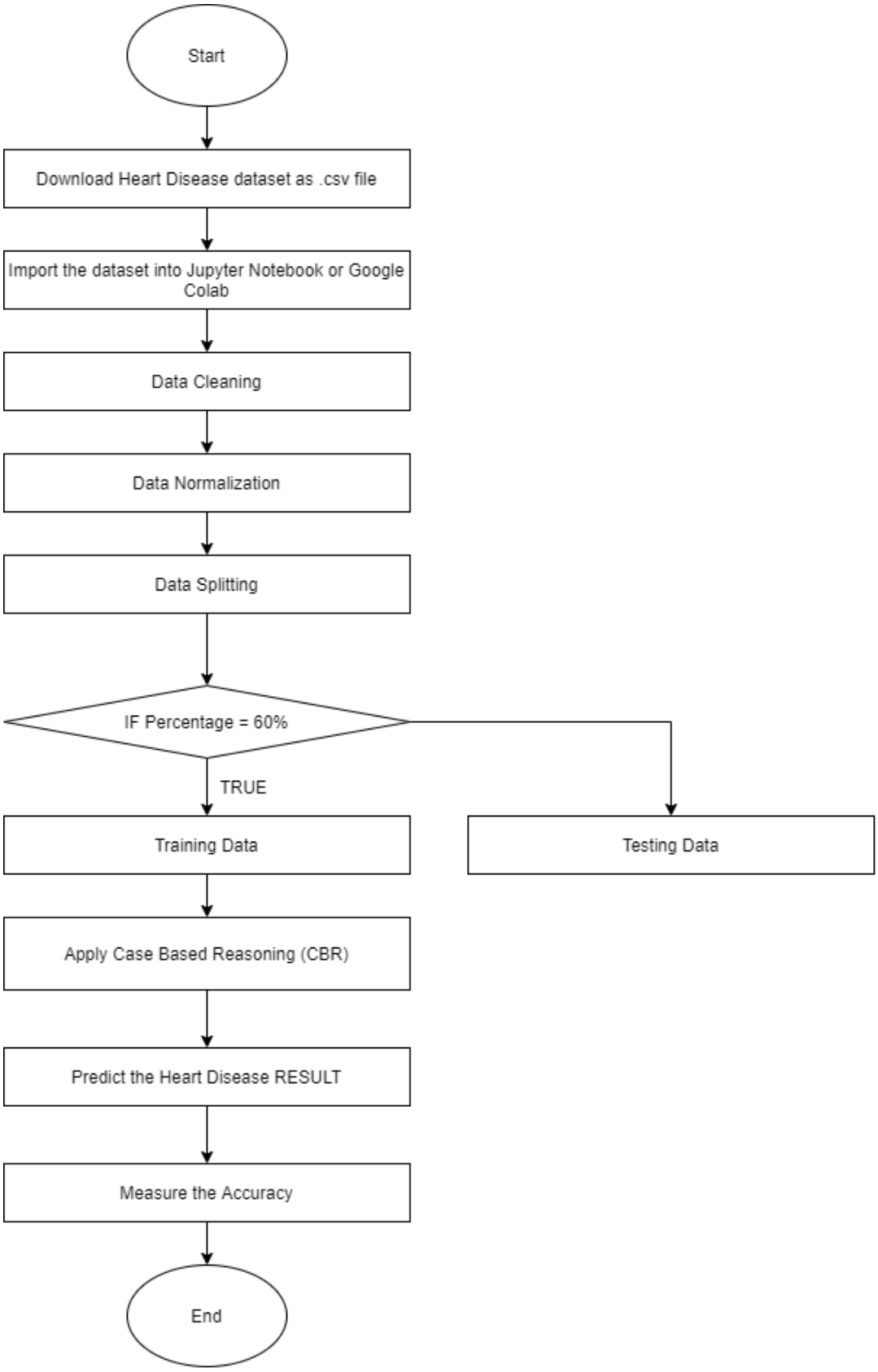


Figure 3.2 Flowchart

The Figure 3.2 above shows the process of the heart disease prediction in term of flowchart. In the first process, we need to download the heart disease dataset csv file from the Kaggle website. Then, use the dataset to go through the data pre-processing process to transform the raw data to become high quality data. In this process, data normalization technique will be applied to the heart disease data. It will normalize the data to ensure all the data are in the same range. Next, split the normalized data into the ratio of 60:40. 60% will be the training data and 40% will be the testing data.

After that, the 60% training data and 40% of testing data will be applied to conduct the prediction and the accuracy measurement. Then, the Case Based Reasoning (CBR) intelligent system algorithm will be used to predict the heart disease.

Finally, the outcome of the heart disease prediction will be predicted by using the selected intelligent system algorithm and then the accuracy will be measure by comparing the original dataset and the predicted dataset.

3.3.6 Software Requirement

Table 3.3 Software Requirement

Software	Version	Description
Visual Studio Code (Jupyter Notebook)	Version 1.74.3	Jupyter Notebook used as a Plugin to import heart disease dataset .csv, data pre-processing, data splitting and intelligent system algorithms.
Google Colab	Python 3.6. 9	Google Colab use for import heart disease dataset .csv, data pre-processing, data splitting and intelligent system algorithms if the data are too many.
Google Chrome	Version 100.0.4896.75	Used to do research or study of the heart disease prediction by using intelligent system.
Draw i.o	Version 16.0.0	Used to draw the research framework and flowchart.
Microsoft Words 365	Version 2109	Used for the research report
Microsoft Excel 365	Version 2109	Used for the heart disease dataset storage
Microsoft Power Point 365	Version 2109	Used for research work presentation
Microsoft Power BI	Version 2.111.590.0	Used to Visualize the predicted result.

3.3.7 Hardware Requirement

Table 3.4 Hardware Requirement

Hardware and Model	Features	Description
Laptop (MSI Thin GF63)	Processor: Intel® Core™ i5-10500H CPU @ 2.5GHz Graphic Card: RTX3050 RAM: 12GB (11.8 usable) System: 64-bit Operating System Processor Cores: 4 Edition: Window 10	Used for research documentation, presentation, and development.

3.3.8 Case Study

Since by using the traditional approaches by the doctor's experience and knowledge to predict the heart disease result was difficult and no accurate, the intelligent system algorithms or techniques will be proposed to overcome the problem. Intelligent system algorithms or techniques able to assist doctors to predict the heart disease result in an automatic way by applying the formula into the intelligent system algorithm. There are many different types of intelligent system algorithms available today, and not all of them are suitable to predict the heart disease outcome since the outcome is a binary number that can be either 0 or 1. This is because there are many types of intelligent system algorithms that only able to provide the outcome in images or text form. As a consequence, three best intelligent system algorithms that are suited for predicting the heart disease result are chosen, and only the best and most accurate algorithm is selected to predict the outcome.

3.4 Data Design

The heart disease dataset was provided by the author, David Lapp. This dataset can be found in the Kaggle website. The heart disease dataset characteristics, description, and datatypes are shown in Figure 3.3 below.

No	Attributes	Description	Datatypes
1	age	This indicates the person's age in years.	Integer
2	gender	This indicates the person's sex where: Value 0 = Female Value 1 = Male	Integer
3	cp	This indicates chest pain type: Value 0 = Typical Angina Value 1 = Atypical Angina Value 2 = Non-anginal Pain Value 3 = Asymptomatic	Integer
4	trestbps	This indicates the person's resting blood pressure (mmHg).	Integer
5	chol	This indicates the person's cholesterol measurements in mg/dl	Integer
6	fbs	This indicates the person's fasting blood sugar in measurement of less than 120 mg/dl where: Value 0 = not less than 120 mg/dl Value 1 = less than 120 mg/dl	Integer
7	restecg	This indicates resting electrocardiographic results. There are 3 values: Value 0 = Showing definite left ventricular hypertrophy by Estes' criteria Value 1 = Normal Value 2 = Having ST-T wave abnormality	Integer
8	thalach	This indicates the person's maximum heart rate achieved.	Integer
9	exang	This indicates the exercise induced angina where: Value 0 = no Value 1 = Yes	Integer
10	oldpeak	This indicates the ST depression induced by exercise relative to rest.	Float
11	slope	This indicates the slope of the peak exercise ST segment. Value 0 = downsloping Value 1 = flat Value 2 = upsloping	Integer
12	ca	The number of major vessels (0-3).	Integer
13	thal	This indicates where: Value 1 = normal Value 2 = fixed defect Value 3 = reversable defect	Integer
14	target	It indicates the presence of the heart disease in the patients. Value 0 = no heart disease Value 1 = has heart disease	Integer

Figure 3.3 Attributes and Description of the Heart Disease Dataset

3.5 Proof of Initial Concept

At first, the heart disease dataset will be taken from the Kaggle website. Figure 3.4 below shows the heart disease dataset from Kaggle website. Next, the dataset will be download from the Kaggle as shown in the Figure 3.5 below. Then, the downloaded dataset will be used to process by using the Case Based Reasoning intelligent system algorithms.

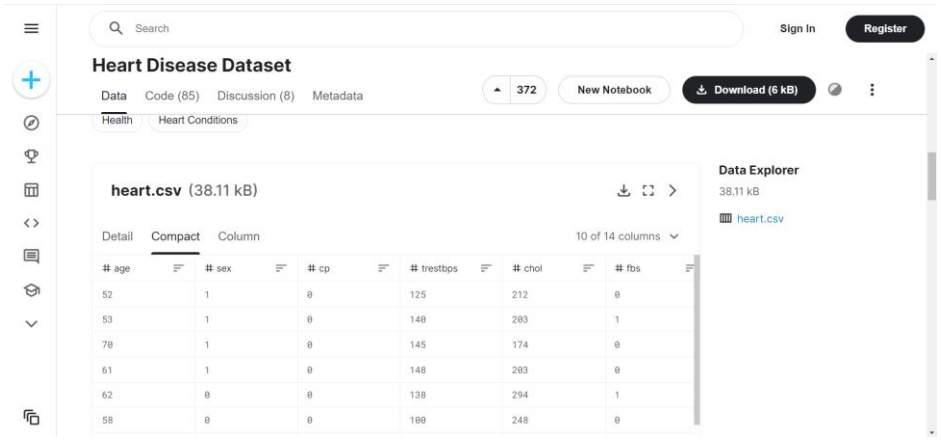
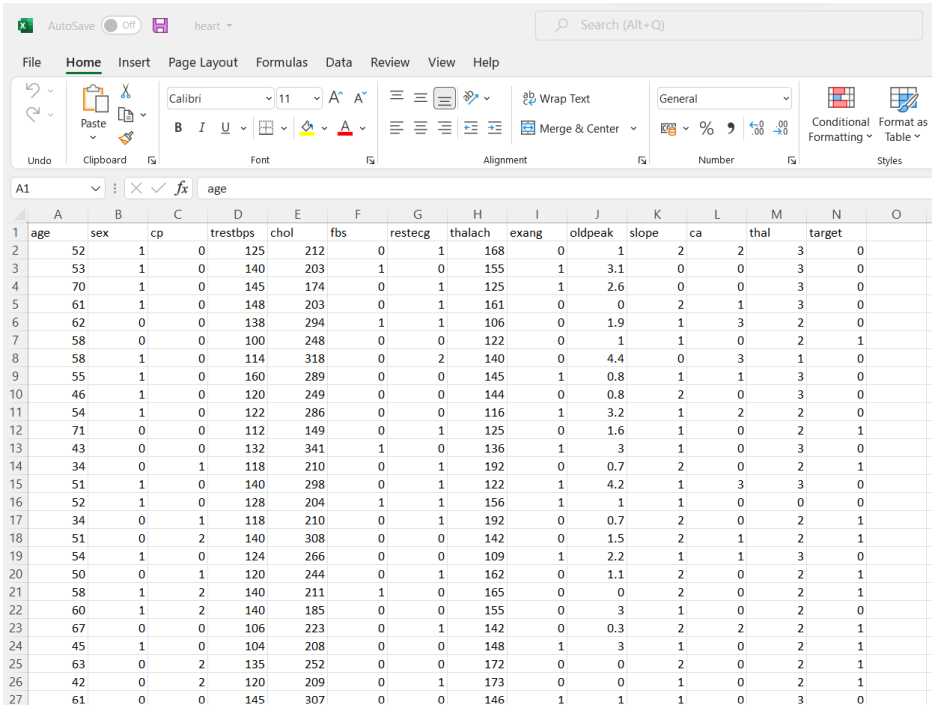


Figure 3.4 Heart Disease Dataset from Kaggle



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	52	1	0	125	212	0	1	168	0	1	2	2	3	0	
3	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	
4	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	
5	61	1	0	148	203	0	1	161	0	0	2	1	3	0	
6	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	
7	58	0	0	100	248	0	0	122	0	1	1	0	2	1	
8	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0	
9	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0	
10	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0	
11	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0	
12	71	0	0	112	149	0	1	125	0	1.6	1	0	2	1	
13	43	0	0	132	341	1	0	136	1	3	1	0	3	0	
14	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1	
15	51	1	0	140	298	0	1	122	1	4.2	1	3	3	0	
16	52	1	0	128	204	1	1	156	1	1	1	0	0	0	
17	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1	
18	51	0	2	140	308	0	0	142	0	1.5	2	1	2	1	
19	54	1	0	124	266	0	0	109	1	2.2	1	1	3	0	
20	50	0	1	120	244	0	1	162	0	1.1	2	0	2	1	
21	58	1	2	140	211	1	0	165	0	0	2	0	2	1	
22	60	1	2	140	185	0	0	155	0	3	1	0	2	0	
23	67	0	0	106	223	0	1	142	0	0.3	2	2	2	1	
24	45	1	0	104	208	0	0	148	1	3	1	0	2	1	
25	63	0	2	135	252	0	0	172	0	0	2	0	2	1	
26	42	0	2	120	209	0	1	173	0	0	1	0	2	1	
27	61	0	0	145	307	0	0	146	1	1	1	0	3	0	

Figure 3.5 Heart Disease Dataset Downloaded

3.5.1 Case Base Reasoning

Case-Based Reasoning (CBR) is another algorithm or approach that may be used to identify heart disease. The CBR algorithm comprises four phases: retrieve, reuse, modify, and retain.

In retrieve phase, the training and testing data that had been spitted will be processed to measure the similarity. There will be two types of similarity measured: local similarity and global similarity. The local and global similarities are depicted in Figures 3.15 and 3.16, respectively.

Numerical/Continuous value

$$sim(a,b) = 1 - \frac{|a-b|}{range}$$

Where,

a	is new feature,
b	is previous features, and
$range$	is the value of difference between the upper and lower boundary of the set.

Discrete value

$$sim\ (a,b) = \begin{cases} 1 & if\ a = b \\ 0 & if\ a \neq b \end{cases}$$

Where,

a	is new feature, and
b	is previous features.

Figure 3.6 Local Similarity

$$sim(A,B) = \frac{1}{\sum w_i} . \sum_{i=1}^{p} w_i . sim_i(a,b)$$

Where,

A	is new case,
B	is previous cases,
a	is new feature from local similarity,
b	is previous features from local similarity,
p	is the number of attributes,
i	is the iteration
w_i	is weight of attributes i $\sum_{i=1}^{p} w_i = 1$, and
sim_i	is local similarity calculate for attribute i.

Figure 3.7 Global Similarity

Then, the reuse phase will used back the previous cases solution that contain the highest global similarity to become the solution for the new cases.

Next, revise phase is used for verification and correct solution. It will verify the solution in the real world.

Finally, is the retain phase. In this phase, the new cases with the solution will be store into the case-base. This is because it can be used in the future when predicting the new patient heart disease problem as previous cases.

3.6 Testing Plan

To conduct the heart disease prediction, the Jupyter Notebook or Google Colab will be use. First of all, the downloaded heart disease dataset .csv file from the Kaggle website will be import to the Jupyter Notebook or Google Colab. Second, the data in the dataset will be go through data pre-processing process to clean the data. Third, once the data has been cleaned, the data normalisation method will be used to the heart disease dataset to normalise the data so that it lies within the same range. Fourth, split the data into 60:40 where 60% of the heart disease data will be the training data and 40% will be the testing data.

Fifth, execute the Case-Based Reasoning to determine each algorithm's accuracy. Sixth, after the accuracy had been measured, the Case Based Reasoning (CBR) intelligent system algorithms will be used to the prediction process. Seventh, the intelligent system method will be utilised to forecast the outcome of the heart disease data. Lastly, the heart disease result will be predicted.

3.7 Potential Used of Proposed Solution

The potential use of proposed solution can be used by patient. Patient was the early stage user. The selected intelligent system approach for predicting heart disease can be implemented as open source software or a system for public usage. Patient able can use it to perform a self-testing on their own to identify their heart disease result. It is significant for the patient to identify their heart disease result in the early stage so that the patient able to take the further action to cure it and stop it from spreading.

Secondly, proposed solution for the heart disease prediction can be also used by the clinic and hospital. The proposed solution system can be implemented in the medical system as a software or function to aid the doctor in the clinic and hospital to diagnosis the patient heart disease result rapidly. It is important to have the proposed solution system inside the clinic computer because if the patients were identified to affected with the heart disease, the patients can be directly transfer to the hospital for the further action to be taken to cure the disease.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter will explain in detail the implementation throughout this research. All the process and workspace has been done two type of software which is Jupyter Notebook, Visual Studio Code and Google Colab. Others than that, there will be a visualization process to visualize the predicted output. The result will be accurate for the right prediction. These tasks will explain in the implementation part meanwhile all results will be in details at result and discussion part.

4.2 Data Collection

The first step is to collecting data from the Kaggle website. The data was published by the author, David Lapp. The heart disease dataset that I take from Kaggle consist of 1025 of data. There are 13 attributes and 1 output in this dataset. The attributes are age, gender, (Cp) Chest Pain, (trestbps) Resting Blood Pressure, (Chol) Serum cholesterol, (fbs) Fasting blood sugar, (Restecg) Resting electrocardiographic results, (thalach) Maximum heart rate achieved, (exang) Exercise induced angina, (Oldpeak) ST depression induced by exercise relative to rest, (Slope) The slope of the peak exercise ST segment, (Ca) Number of major vessels and (thal) A blood disorder called thalassemia. Figure 4.1 below shows the dataset being import and read by using machine learning.

```
# Import and Read the Heart Disease Dataset under the variable of 'full_data'
full_data = pd.read_csv('heart.csv')

full_data
```

[2]

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Figure 4.1 Import Dataset

Figure 4.2 below shows the machine learning code to identify or to learn the dataset. The figure 4.2 below also shows the output of the attributes, the datatype and the missing value. As a result, the data in the 13 attributes did not contains any missing value. The datatype for the 13 attributes is also integer except attribute number 9 which is using float. For the experiment, I will use this dataset to split into training and testing data for prediction.

```
[3]  # prints information about the DataFrame
     full_data.info()

...  <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 1025 entries, 0 to 1024
     Data columns (total 14 columns):
      #   Column    Non-Null Count  Dtype
     ---  ------    --------------  -----
      0   age       1025 non-null   int64
      1   sex       1025 non-null   int64
      2   cp        1025 non-null   int64
      3   trestbps  1025 non-null   int64
      4   chol      1025 non-null   int64
      5   fbs       1025 non-null   int64
      6   restecg   1025 non-null   int64
      7   thalach   1025 non-null   int64
      8   exang     1025 non-null   int64
      9   oldpeak   1025 non-null   float64
      10  slope     1025 non-null   int64
      11  ca        1025 non-null   int64
      12  thal      1025 non-null   int64
      13  target    1025 non-null   int64
     dtypes: float64(1), int64(13)
     memory usage: 112.2 KB
```

Figure 4.2 Identifying Dataset

Aside from that, Figure 4.3 below is also another method of identifying the missing value. The result shows that there is no missing value in this dataset.

```
[4]  # Print the number of missing values
     print(full_data.isnull().sum())

...  age         0
     sex         0
     cp          0
     trestbps    0
     chol        0
     fbs         0
     restecg     0
     thalach     0
     exang       0
     oldpeak     0
     slope       0
     ca          0
     thal        0
     target      0
     dtype: int64
```

Figure 4.3 Identify Missing Value

4.3 Tools Require

To conduct this research, a few tools will be used. The tools that used to execute the algorithm are Jupyter Notebook in Figure 4.4 below and Google Colab. Then, Microsoft Power BI software from Figure 4.5 below, will be used to visualize the predicted output. Lastly, Microsoft Word software will be used for paperwork or report writing purpose. In this experiment, it is using Python Language in Machine Learning. A few libraries will require to install into the Jupyter Notebook and Google Colab such as pandas, NumPy, matplotlib and seaborn in the Figure 4.6 below. The first is pandas library. Pandas is an open source library. Pandas is a Python library for data analysis. Pandas are a set of data structures and data analysis tools that provide great performance, speed, and ease of use. It used to manipulate numeric data and time series. Next, is the NumPy. NumPy is also an open source library in Python that allows users to perform operations on arrays. In addition to that, it has tools for carrying out operations related to linear algebra, the Fourier transform, and matrix.

Figure 4.4 Jupyter Notebook & Python

Figure 4.5 Microsoft Power BI

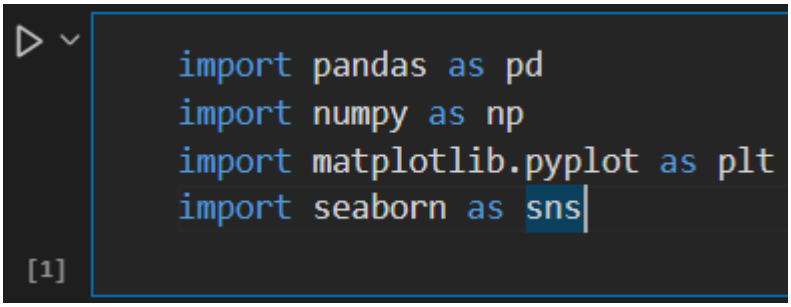


Figure 4.6 Python Libraries

Besides that, Matplotlib library also used in this research. Matplotlib is a Python library that allows us to create static, animated, and interactive visualisations. Matplotlib makes simple things simple and difficult things possible. Lastly, seaborn library package. Seaborn is a matplotlib-based Python data visualisation library. It offers a high-level interface for creating visually appealing and useful statistics visuals. The difference between Seaborn and Matplotlib is that Seaborn makes use of exciting themes, whilst matplotlib is used to create simple graphs. Seaborn has a few plots and patterns for data visualisation, while matplotlib visualises datasets using lines, scatter plots, pie charts, histograms, bar graphs, and so on.

4.4 Data Splitting

The process of data splitting involves dividing the data into two or more parts. When there is a split into two parts, one of those parts is often utilised to analyse or test the data, while the other part is used to train the model. The process of data splitting is a crucial component of data science, especially for the development of models that are based on data. Accuracy in the development of data models and the processes that apply data models, such as machine learning, may be improved with the aid of this approach. There are 3 types of data splitting. First is random sampling. The data modelling process is protected from bias using this data sampling strategy, which prevents against bias toward various potential data features. However, there may be problems with the random splitting method due to the unequal distribution of the data. Secondly, stratified random sampling. This method picks random samples of data from a set of parameters. It makes sure that the data in the training set and the test set are split up in the right way. Thirdly, non-random sampling. This method is usually used by data modellers in situations in which they want to use the most current data as the test set.

In this research, the data will be split into the ratio 60:40, where 60% will be the training data and 40% will be the testing data by using the non-random sampling method. This is because the result of this research requires the predicted dataset to compare with the previous dataset to identify the accuracy.

In the heart.csv dataset, there are 1025 data. Therefore, 615 data will be used for training and 410 data will be used for testing. To split the data into the ratio 60:40, I've used the sklearn.model_selection.train_test_split module. The code of the data splitting is shown in the Figure 4.7 below. The result of the training and testing result are shown in the Figure 4.8 and Figure 4.9 below respectively. Then, the testing data output column will be eliminated for prediction purpose.

```python
# Split Data into ratio 60:40 (60% Training Data & 40% Testing Data)
from sklearn.model_selection import train_test_split

attribute_column = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target']
data = full_data[attribute_column]  # attribute

training_data,testing_data = train_test_split(data, test_size=0.4, shuffle=False)

print(training_data) # 60% Random Training Dataset
print()

print(testing_data) # 40% Random Testing Dataset
print()

#Eliminate the outcome column(target) for testing_data
testing_data_no_output = testing_data.iloc[:,range(testing_data.shape[1]-1)]
```

Figure 4.7 Data Splitting

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     52    1   0       125   212    0        1      168      0      1.0
1     53    1   0       140   203    1        0      155      1      3.1
2     70    1   0       145   174    0        1      125      1      2.6
3     61    1   0       148   203    0        1      161      0      0.0
4     62    0   0       138   294    1        1      106      0      1.9
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
610   43    0   0       132   341    1        0      136      1      3.0
611   55    0   0       128   205    0        2      130      1      2.0
612   58    0   0       170   225    1        0      146      1      2.8
613   55    1   0       140   217    0        1      111      1      5.6
614   51    0   0       130   305    0        1      142      1      1.2

     slope  ca  thal  target
0        2   2     3       0
1        0   0     3       0
2        0   0     3       0
3        2   1     3       0
4        1   3     2       0
..     ...  ..   ...     ...
610      1   0     3       0
611      1   1     3       0
612      1   2     1       0
613      0   0     3       0
614      1   0     3       0

[615 rows x 14 columns]
```

Figure 4.8 Training Data Result

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
615    50    0   2       120   219    0        1      158      0      1.6
616    43    1   0       115   303    0        1      181      0      1.2
617    41    0   1       126   306    0        1      163      0      0.0
618    49    1   1       130   266    0        1      171      0      0.6
619    65    1   0       110   248    0        0      158      0      0.6
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
615       1   0     2       1
616       1   0     2       1
617       2   0     2       1
618       2   0     2       1
619       2   2     1       0
...     ...  ..   ...     ...
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0

[410 rows x 14 columns]
```

Figure 4.9 Testing Data Result

After the data had been split into training and testing, the training data will be saved as the csv file named, “heart_Training_data_Before_Normalization.csv”, while the testing data will be saved as “heart_Testing_data_Before_Normalization.csv” as shown in the Figure 4.10 below.

```
# Store 60% Random data into heart_Training_data_Before_Normalization.csv under variable of 'train'
training_data.to_csv('heart_Training_data_Before_Normalization.csv', index=False)
train = pd.read_csv('heart_Training_data_Before_Normalization.csv')

# Store 40% Random data without output into heart_Testing_data_Before_Normalization.csv under variable o
testing_data_no_output.to_csv('heart_Testing_data_Before_Normalization.csv', index=False)
test = pd.read_csv('heart_Testing_data_Before_Normalization.csv')
```

Figure 4.10 Store Training and Testing Data

4.5 Data Pre-Processing

Data pre-processing is the process of cleaning or dropping the data. It is an important step in data mining process. The purpose of data pre-processing is to convert or transform the raw data into high quality data or understandable format. Data normalization is also part of data pre-processing. Normalization is a procedure that is often used in the process of data preparation or data pre-processing in machine learning. The process of transforming the different columns of a dataset to the same scale is referred to as normalisation. The data in the dataset will be convert to the scale of 0 to 1 by using normalization method. It is not necessary to normalise each and every dataset when using machine learning. It is only required in situations in which the ranges of the characteristics differ. The purpose of normalizing the data is to enhance the accuracy of the prediction. In this research, the "from sklearn.preprocessing import Normalizer" module will be used to normalize the training and testing data.

4.5.1 Training Data

The training data will be normalized by using the code as shown in the Figure 4.11 below. The result of the training data that after normalized will be in the same range which is between 0 to 1 as shown in Figure 4.12 below.

```
# Normalization Process for 60% random train data
from sklearn.preprocessing import Normalizer

normal = Normalizer()

train_normalization.iloc[:,0:-1] = normal.fit_transform(train_normalization.iloc[:,0:-1])

train_normalization
```

Figure 4.11 Training Data Normalization

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	0.171891	0.003306	0.0	0.413199	0.700785	0.000000	0.003306	0.555339	0.000000	0.003306	0.006611	0.006611	0.009917	0
1	0.179004	0.003377	0.0	0.472842	0.685621	0.003377	0.000000	0.523503	0.003377	0.010470	0.000000	0.000000	0.010132	0
2	0.261156	0.003731	0.0	0.540967	0.649160	0.000000	0.003731	0.466351	0.003731	0.009700	0.000000	0.000000	0.011192	0
3	0.200274	0.003283	0.0	0.485910	0.666484	0.000000	0.003283	0.528591	0.000000	0.000000	0.006566	0.003283	0.009850	0
4	0.178548	0.000000	0.0	0.397413	0.846663	0.002880	0.002880	0.305259	0.000000	0.005472	0.002880	0.008639	0.005760	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	0.109549	0.000000	0.0	0.336290	0.868748	0.002548	0.000000	0.346480	0.002548	0.007643	0.002548	0.000000	0.007643	0
611	0.196486	0.000000	0.0	0.457277	0.732358	0.000000	0.007145	0.464422	0.003572	0.007145	0.003572	0.003572	0.010717	0
612	0.179660	0.000000	0.0	0.526589	0.696956	0.003098	0.000000	0.452247	0.003098	0.008673	0.003098	0.006195	0.003098	0
613	0.191977	0.003490	0.0	0.488668	0.757435	0.000000	0.003490	0.387444	0.003490	0.019547	0.000000	0.000000	0.010471	0
614	0.140000	0.000000	0.0	0.356864	0.837257	0.000000	0.002745	0.389805	0.002745	0.003294	0.002745	0.000000	0.008235	0

615 rows × 14 columns

Figure 4.12 Normalized Training Data

Thus, the training data after normalized will be save as “heart_Training_data_After_Normalization.csv” file in the code of Figure 4.13 below.

```
[12]
# Store train 60% Random Normalized data into 'heart_Training_data_After_Normalization.csv'
train_normalization.to_csv('heart_Training_data_After_Normalization.csv', index=False)
```

Figure 4.13 Store Normalized Training Data

4.5.2 Testing Data

The testing data will be normalized by using the code as shown in the Figure 4.14 below. The result of the testing data that after normalized will be in the same range which is between 0 to 1 as shown in Figure 4.15 below.

```
[17]
# Normalization Process for 40% random test data
from sklearn.preprocessing import Normalizer

norm = Normalizer()

test_normalization.iloc[:,0:-1] = norm.fit_transform(test_normalization.iloc[:,0:-1])

test_normalization
```

Figure 4.14 Testing Data Normalization

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
615	0.166817	0.000000	0.006673	0.400361	0.730660	0.0	0.003336	0.527143	0.000000	0.005338	0.003336	0.000000	0.006673	1
616	0.115065	0.002676	0.000000	0.307733	0.810809	0.0	0.002676	0.484344	0.000000	0.003211	0.002676	0.000000	0.005352	1
617	0.110460	0.000000	0.002694	0.339461	0.824406	0.0	0.002694	0.439144	0.000000	0.000000	0.005388	0.000000	0.005388	1
618	0.141860	0.002895	0.002895	0.376362	0.770095	0.0	0.002895	0.495061	0.000000	0.001737	0.005790	0.000000	0.005790	1
619	0.202726	0.003119	0.000000	0.343075	0.773478	0.0	0.000000	0.492780	0.000000	0.001871	0.006238	0.006238	0.003119	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	0.187676	0.003181	0.003181	0.445332	0.702988	0.0	0.003181	0.521674	0.003181	0.000000	0.006362	0.000000	0.006362	1
1021	0.184558	0.003076	0.000000	0.384496	0.793599	0.0	0.000000	0.433711	0.003076	0.008613	0.003076	0.003076	0.009228	0
1022	0.145834	0.003103	0.000000	0.341314	0.853285	0.0	0.000000	0.366137	0.003103	0.003103	0.003103	0.003103	0.006206	0
1023	0.154742	0.000000	0.000000	0.340433	0.786091	0.0	0.000000	0.492081	0.000000	0.000000	0.006190	0.000000	0.006190	1
1024	0.211086	0.003909	0.000000	0.469080	0.734891	0.0	0.003909	0.441717	0.000000	0.005473	0.003909	0.003909	0.011727	0

410 rows × 14 columns

Figure 4.15 Normalized Testing Data

Thus, the testing data after normalized will be save as "heart_Testing_data_After_Normalization.csv" file in the code of Figure 4.16 below.

```
# Store test 40% Random Normalized data into 'heart_Testing_data_After_Normalization.cs
test_normalization.to_csv('heart_Testing_data_After_Normalization.csv', index=False)
```

Figure 4.16 Store Normalized Testing Data

4.6 Algorithm

The data that has been split into the ratio of 70:30 and been normalized will be used in the Case Based Reasoning (CBR) algorithm for prediction purpose. Case-based reasoning, often known as CBR, is an approach for resolving novel issues by modifying approaches that have already shown successful in solving problems based on the past. It is an Artificial Intelligence (AI) technique that imitates how human make a decision. Memory, learning, as well as planning and problem-solving, are all investigated during CBR. This sets the way for a new technology that will include intelligent computer systems that are able to solve issues and adapt to new circumstances. In case-based reasoning (CBR), "intelligent" reuse of information from previously solved problems, commonly known as "cases," is predicated on the assumption that if two problems are similar, their solutions will also be similar. Cases may be thought of as examples of previously resolved issues. In CBR algorithm, there are 4 stages that will pass through which are retrieve, reuse, revise and retain shows in Figure 4.17 below.

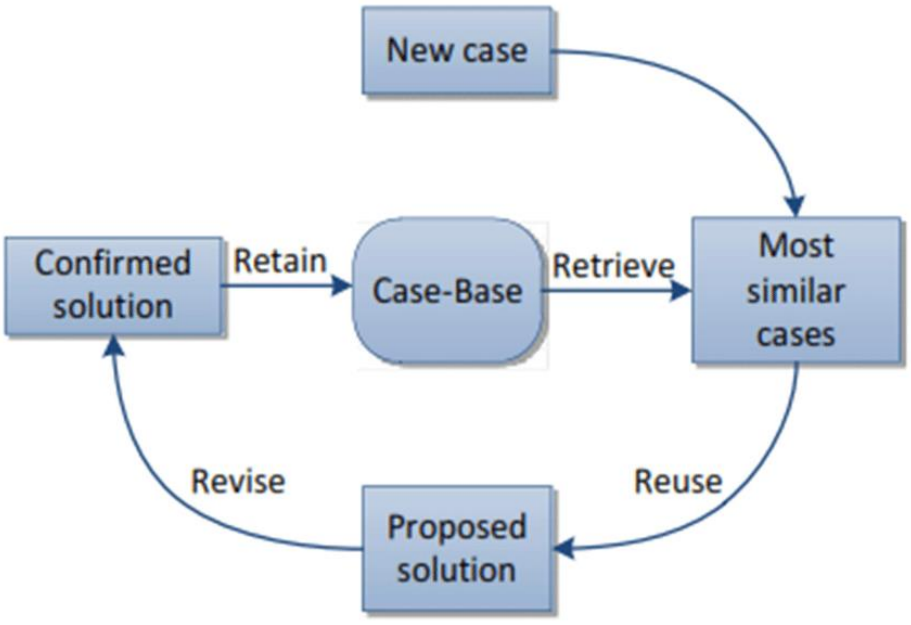


Figure 4.17 CBR Cycle

4.6.1 Retrieve

The first stage is retrieve. The similarities will be calculated between the testing data and training data. It used in problem solving and reasoning to match a previous experience/case (case base) with the new unseen problem to find solution. It also can be call as case matching in CBR match new case with the previous cases from the case base to find solution. The purpose of the similarities is to select cases that can be adapted easily to the current problem and select cases that have (nearly) the same solution than the current problem. The basic assumption for the retrieve phase is "similar problems have similar solutions" and the goal of similarity modelling is to provide a good approximation. There are two types of similarities that need to be calculated such as local similarity and Global Similarity. In Local Similarity, it used to compute the similarity between query (new problem) and case attributes values while Global Similarity is a build up from number of local similarities function It is a weight sum of the local similarity. The formula of the Local Similarity and Global Similarity are shown in the Figure 4.18 and Figure 4.19 below.

Numerical/Continuous value

$$sim(a,b) = 1 - \frac{|a-b|}{range}$$

Where,

a	is new feature,
b	is previous features, and
$range$	is the value of difference between the upper and lower boundary of the set.

Discrete value

$$sim\,(a,b) = \begin{cases} 1 & if\ a = b \\ 0 & if\ a \neq b \end{cases}$$

Where,

a	is new feature, and
b	is previous features.

Figure 4.18 Local Similarity Formula

$$sim(A,B) = \frac{1}{\sum w_i} \cdot \sum_{i=1}^{p} w_i \cdot sim_i(a,b)$$

Where,

A	is new case,
B	is previous cases,
a	is new feature from local similarity,
b	is previous features from local similarity,
p	is the number of attributes,
i	is the iteration
w_i	is weight of attributes i $\sum_{i=1}^{p} w_i = 1$, and
sim_i	is local similarity calculate for attribute i.

Figure 4.19 Global Similarity Formula

Thus, similarity measurement for local similarity is calculate between each attribute values, while Global Similarities is calculated between each case. The step in retrieve phase is:

Step 1:

In the first step, the data that had been split into training and testing will be import. It will import the original data and the data that had been normalized. Next, a variable will be used to declare for each data that being imported and the. The code is shown in the Figure 4.20 below. Then the output column of the normalized training data will be eliminated by using the code in the Figure 4.21 below for prediction purpose.

```
[19]
# Import and Read the 60% Random Normalized Train data by using variable 'train_normalize'
train_normalize = pd.read_csv('heart_Training_data_After_Normalization.csv')

# Import and Read the 40% Random Normalized Test data by using variable 'test_normalize'
test_normalize = pd.read_csv('heart_Testing_data_After_Normalization.csv')

# Import and Read the 70% Random Train data before Normalized  by using variable 'train_ori'
train_ori = pd.read_csv('heart_Training_data_Before_Normalization.csv')

# Import and Read the 30% Random Test data before Normalized by using variable 'test_ori'
test_ori = pd.read_csv('heart_Testing_data_Before_Normalization.csv')
```

Figure 4.20 Import Original and Normalized Dataset

```
[24]
# Eliminate the Output column of the 'train_normalize' variable
case_base = train_normalize.iloc[:,range(train_normalize.shape[1]-1)]
case_base
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	0.171891	0.003306	0.0	0.413199	0.700785	0.000000	0.003306	0.555339	0.000000	0.003306	0.006611	0.006611	0.009917
1	0.179004	0.003377	0.0	0.472842	0.685621	0.003377	0.000000	0.523503	0.003377	0.010470	0.000000	0.000000	0.010132
2	0.261156	0.003731	0.0	0.540967	0.649160	0.000000	0.003731	0.466351	0.003731	0.009700	0.000000	0.000000	0.011192
3	0.200274	0.003283	0.0	0.485910	0.666484	0.000000	0.003283	0.528591	0.000000	0.000000	0.006566	0.003283	0.009850
4	0.178548	0.000000	0.0	0.397413	0.846663	0.002880	0.002880	0.305259	0.000000	0.005472	0.002880	0.008639	0.005760
...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	0.109549	0.000000	0.0	0.336290	0.868748	0.002548	0.000000	0.346480	0.002548	0.007643	0.002548	0.000000	0.007643
611	0.196486	0.000000	0.0	0.457277	0.732358	0.000000	0.007145	0.464422	0.003572	0.007145	0.003572	0.003572	0.010717
612	0.179660	0.000000	0.0	0.526589	0.696956	0.003098	0.000000	0.452247	0.003098	0.008673	0.003098	0.006195	0.003098
613	0.191977	0.003490	0.0	0.488668	0.757435	0.000000	0.003490	0.387444	0.003490	0.019547	0.000000	0.000000	0.010471
614	0.140000	0.000000	0.0	0.356864	0.837257	0.000000	0.002745	0.389805	0.002745	0.003294	0.002745	0.000000	0.008235

615 rows × 13 columns

Figure 4.21 Eliminate Normalized Training Data Output Column

Step 2:

In this step, to calculate the local similarity for each case, the minimum and maximum data will be identified from the normalized training data from the Figure 4.22 and the range will be calculate by using the maximum value subtract the minimum value for each attribute as shown in the code below in Figure 4.23. Besides, for global similarity calculation purpose, the weightage will be declared to 1 for each attribute as shown in the Figure 4.24 below.

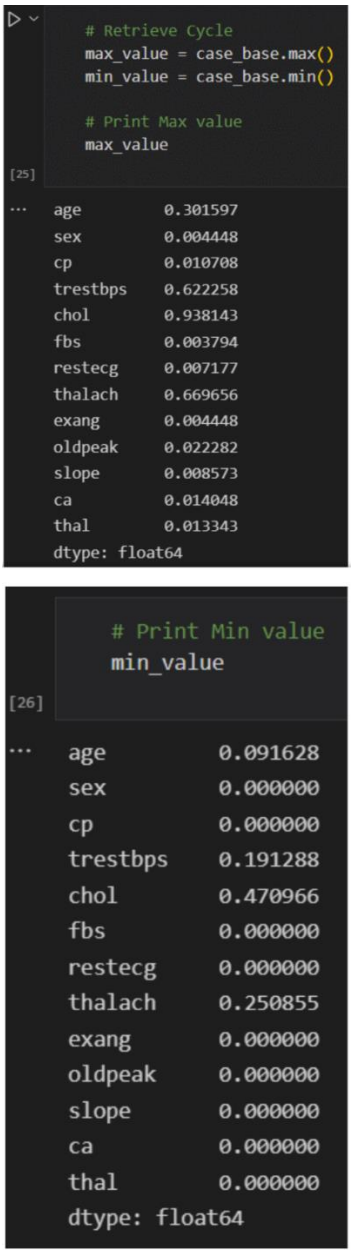


Figure 4.22 Identify Minimum and Maximum Value

```
[27]
    # Calculate Range Value
    range_value = max_value-min_value
    range_value

...
    age         0.209969
    sex         0.004448
    cp          0.010708
    trestbps    0.430970
    chol        0.467177
    fbs         0.003794
    restecg     0.007177
    thalach     0.418801
    exang       0.004448
    oldpeak     0.022282
    slope       0.008573
    ca          0.014048
    thal        0.013343
    dtype: float64
```

Figure 4.23 Calculate Range

```
[29]
    # Declaring weightage to all the attributes
    weightage=np.ones(test.shape[1])
    weightage

...
    array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])

[30]
    len(weightage)
    total_weightage=0
    # Calculate total weightage
    for i in range(len(weightage)):
        total_weightage+=weightage[i]

    total_weightage

...
    13.0
```

Figure 4.24 Declaring Weightage Value

Step 3:

In this step the local similarity and global similarity algorithm will be applied to predict the outcome result of the heart disease data. The code for both algorithm and some of the examples of the result will be shown in the Figure 4.25 and Figure 4.26 below respectively.

```
# Retrieve Cycle  -- Find similiar problems

for k in range(test_normalize.shape[0]):
    print('================================================================================')
    print(f'\t\t\t TEST DATA Case: {k+1}')
    print('================================================================================')
    for i in range(case_base.shape[0]):
        total_similarity = 0
        print('================================================================================')
        print(f'\t\t LOCAL SIMILARITY for TEST_DATA Case: {k+1}')
        print('================================================================================')
        for j in range(case_base.shape[1]):
            # Calculate Local Similarity
            local_similarity[i,j]=1-(abs(test_normalize.iloc[k,j]-case_base.iloc[i,j])/range_value[j])

            # Calculate Global Similarity
            total_similarity += (local_similarity[i,j] * weightage[j])
            global_similarity[i] = (1/total_weightage) * total_similarity
            print(f'Local Similarity TRAIN DATA Case {i+1}, Attribute: {train_normalize.columns[j]} \t = {local_similarity[i,j]}')

        print('--------------------------------------------------------------------------------')
        print(f'Global Similarity TRAIN DATA Case {i+1} = {global_similarity[i]}')
        print('--------------------------------------------------------------------------------')
```

Figure 4.25 Applying the Local and Global Similarity Algorithm

```
================================================================================
                        TEST DATA Case: 1
================================================================================
================================================================================
                LOCAL SIMILARITY for TEST_DATA Case: 1
================================================================================
Local Similarity TRAIN DATA Case 1, Attribute: age       = 0.9758371507239155
Local Similarity TRAIN DATA Case 1, Attribute: sex       = 0.2567899350444518
Local Similarity TRAIN DATA Case 1, Attribute: cp        = 0.3768263767791524
Local Similarity TRAIN DATA Case 1, Attribute: trestbps          = 0.9702127381697782
Local Similarity TRAIN DATA Case 1, Attribute: chol      = 0.936053353016284
Local Similarity TRAIN DATA Case 1, Attribute: fbs       = 1.0
Local Similarity TRAIN DATA Case 1, Attribute: restecg   = 0.9957148566721516
Local Similarity TRAIN DATA Case 1, Attribute: thalach   = 0.9326728717447913
Local Similarity TRAIN DATA Case 1, Attribute: exang     = 1.0
Local Similarity TRAIN DATA Case 1, Attribute: oldpeak   = 0.9087815432399864
Local Similarity TRAIN DATA Case 1, Attribute: slope     = 0.6180052527112954
Local Similarity TRAIN DATA Case 1, Attribute: ca        = 0.5293971421275363
Local Similarity TRAIN DATA Case 1, Attribute: thal      = 0.7568731029401983
--------------------------------------------------------------------------------
Global Similarity TRAIN DATA Case 1 = 0.7890126402438108
--------------------------------------------------------------------------------
```

Figure 4.26 Example Output Result from Local and Global Similarity Algorithm

4.6.2 Reuse

Next, after the local and global similarity had been calculated for each testing data, it will move to the reuse phase. The reuse process in the CBR cycle is responsible for proposing a solution for a new problem from the solutions in the retrieved cases where it reuse of previous experiences as the solution in a new problem or situation. Reusing a retrieved case can be as easy as returning the retrieved solution, unchanged, as the proposed solution for the new problem. It did not require any modification where it just copy back the solution in previous problem that achieve the highest similarity as the new case or problem solution. In Reuse phase, the highest similarity among the global similarity result will be identified by using the code in the Figure 4.27 below.

```
# Identify the Highest Value of the Global Similarity
highest_similarity = global_similarity
highest_index = np.argmax(highest_similarity)
print('================================================================================')
print(f'Highest Similarity for TEST DATA Case {k+1} = {highest_similarity.max()}')
print('================================================================================')
print('\n')
```

Figure 4.27 Identify Highest Similarity among the Global Similarity

After the highest similarity had been identified, the result of training case that achieve the highest global similarity will be append as the result for the testing case data. The code is shown in the Figure 4.28 below.

```
# Reuse Cycle -- Reuse a previous solution in a new situation
# Propose solution for new cases from the solutions in the retrieved cases
test_data = np.append(test_ori.iloc[k,:],train_normalize.iloc[highest_index,
```

Figure 4.28 Reuse

4.6.3 Revise

Besides, the revise phase is to figure out how the solution can be used in the new setting. It aims to verification and to ensure the correctness of the solution. Since the solution of the Breast Cancer Data has only 2 values which is 0 and 1, then the revise cycle is not require as if Result = 0 : Benign (non-cancerous) and if Result = 1 : Malignant (cancerous).

4.6.4 Retain

The last cycle is retain phase. The result for the new solution or test data will be merge and store to the train data before normalizing. So, it will store and merge the test data with the new result to the train data before normalizing and save into the case based. Therefore, the new solution will be saved to the file name "Breast_Cancer_Training_data_Before_Normalization.csv" by using the code in the Figure 4.29 below.

```
# Retain Cycle -- Integrated into case-based system
# Store the result for the new solution to the train_ori
test_data = pd.Series(test_data,index = train_ori.columns)


# Update / Merge the new solution with the previous dataset in 'heart_Training_data_Before_Normalizati
train_ori = train_ori.append(test_data,ignore_index=True)
train_ori.to_csv('heart_Training_data_Before_Normalization.csv', index=False)
```

Figure 4.29 Store the Predicted data to the Training Data before Normalized

4.7 Result

After the heart disease data had been predicted and save into the file name "Breast_Cancer_Training_data_Before_Normalization.csv", then the comparison will be made between the result in the original dataset with the result in the "Breast_Cancer_Training_data_Before_Normalization.csv" file. So, both csv file will be import and will be compare the output of both file by using the code in Figure 4.30 below. Then, the result of the comparison will be assigned with a DataFrame and being save as csv file named, "Results.csv".

```
[33]
df1 = pd.read_csv('heart.csv')
df2 = pd.read_csv('heart_Training_data_Before_Normalization.csv')
```

```
[34]
result = df1.apply(tuple, 1).isin(df2.apply(tuple, 1))
result = pd.DataFrame(result, columns=['Results'])

result
```

	Results
0	True
1	True
2	True
3	True
4	True
...	...
1020	True
1021	True
1022	True
1023	True
1024	True

1025 rows × 1 columns

Figure 4.30 Import Original and Predicted Data for Comparison

4.7.1 Visualization

4.7.1.1 Accuracy

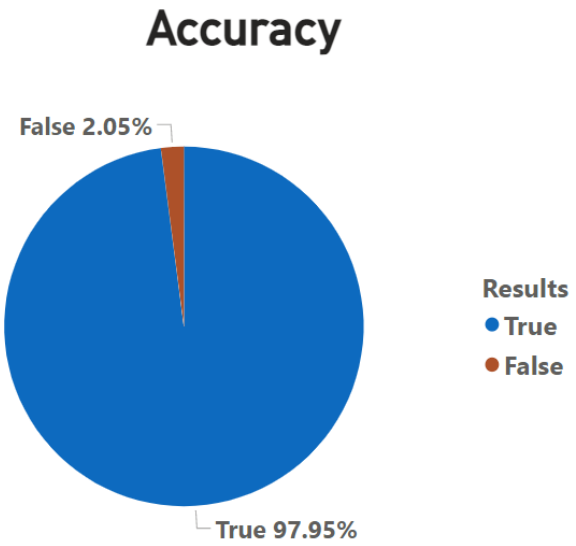


Figure 4.31 Accuracy in Pie Chart

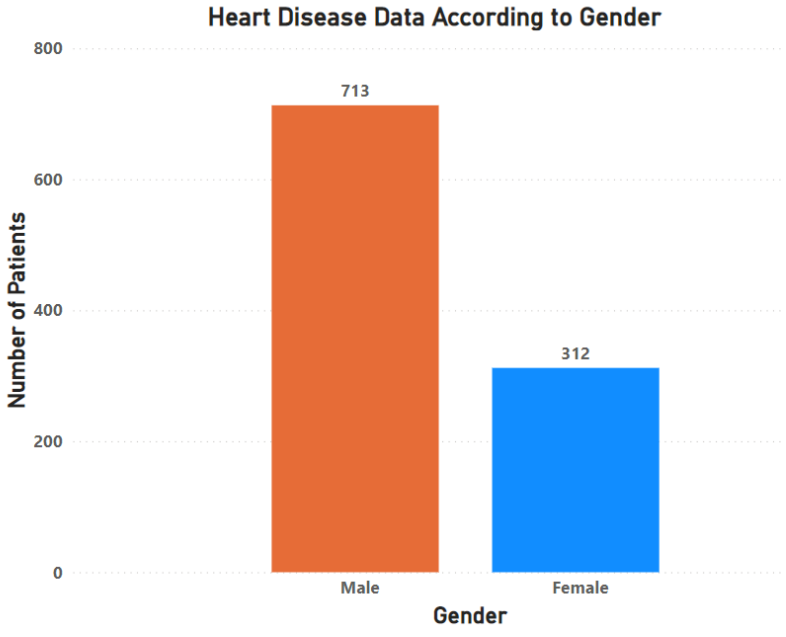


Figure 4.32 Number of Data According to Gender

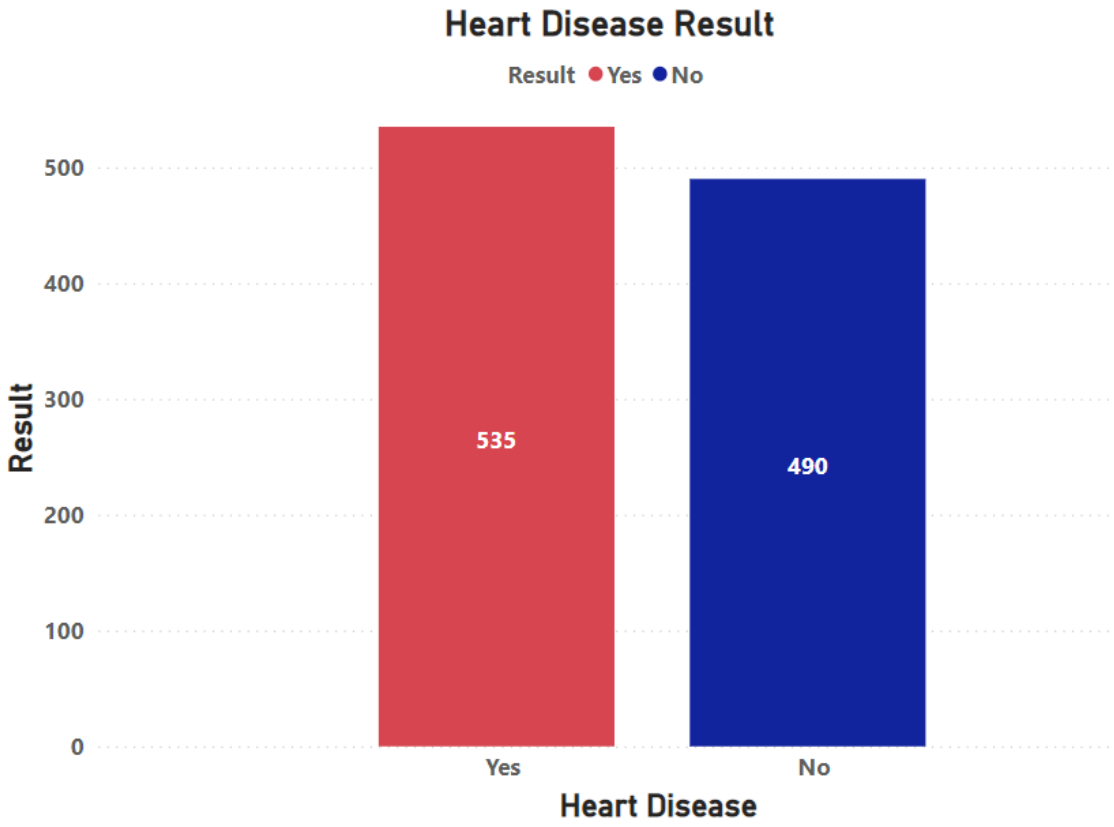


Figure 4.33 Number of Data According to the heart disease

From the result, a few visualizations technique will be apply to the predicted heart disease dataset and the result data by using the Microsoft Power BI. The Figure 4.31 above shows the graph of the accuracy results in Pie Chart. The overall prediction accuracy is 97.95%. Then, Figure 4.32 above shows the number of heart disease data according to the gender. The results show there are 1025 data altogether and 713 data come from Male and 312 data comes from Female. From the Figure 4.33 above, the number of patients that getting heart disease is 535 out of 1025.

4.7.1.2 Heart Disease According to Gender

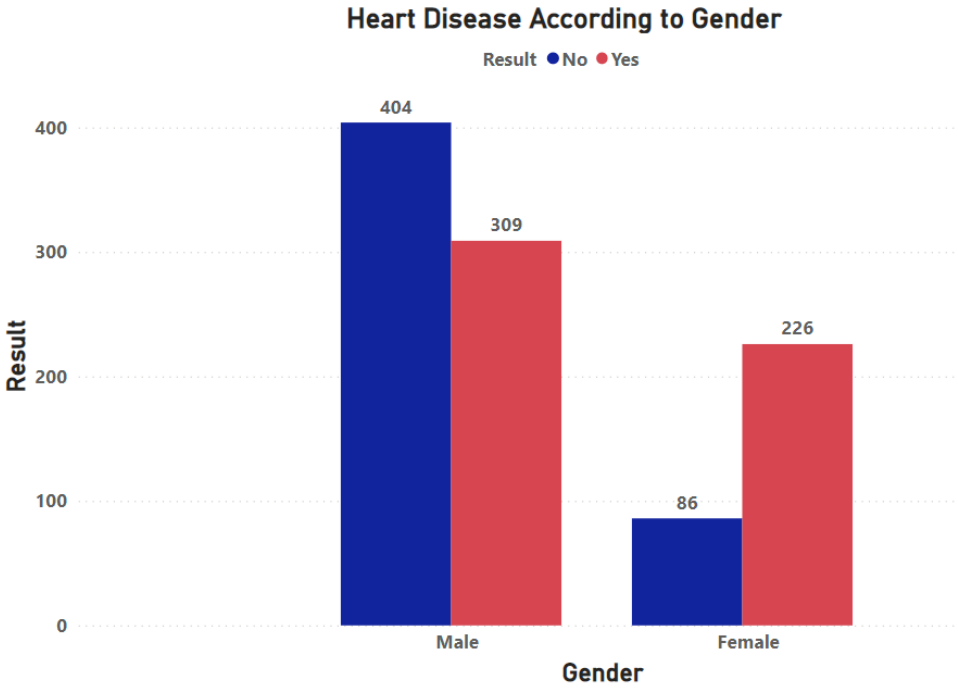


Figure 4.34 Heart Disease According to Gender (Overall)

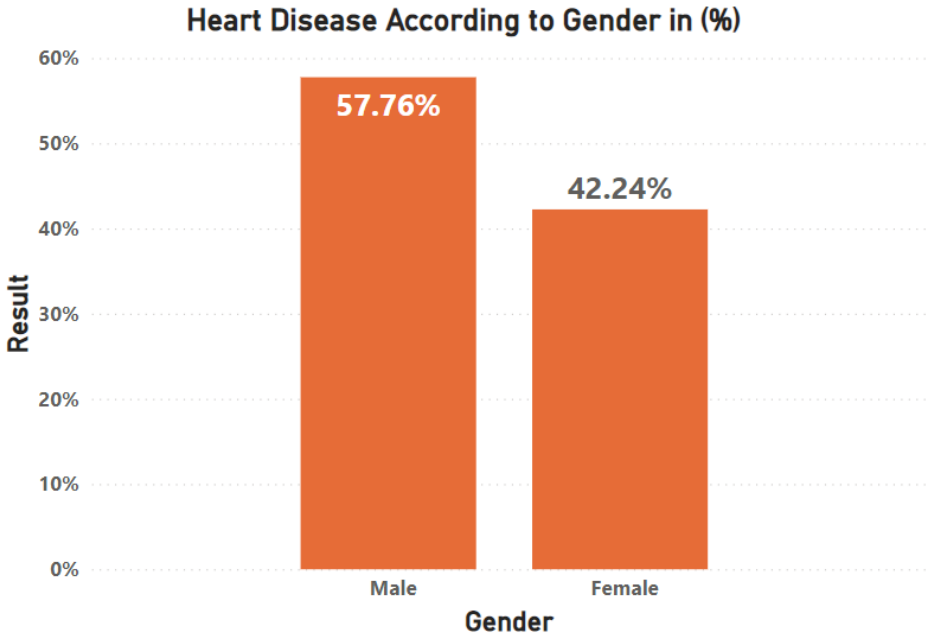


Figure 4.35 Heart Disease According to Gender (in %)

Next, the Figure 4.34 and 4.35 above shows the bar chart on the heart disease according to gender that visualized from the predicted result. The Figure 4.34 graph shows the overall number of patients who have and don't have heart disease according to the gender. The, the Figure 4.35 graph result shows that the gender, the percentage of gender male to get heart disease is higher than the female. The percentage for getting heart disease for male is 57.76% while the female is 42.24%. Based on the justification from (Jha et al., 2010) and (Donald M Lloyd-Jones et al., 1999) the factor activity that can cause the heart disease especially to male such as smoking and alcohol. Both activities are always do by male more than compared to female.

4.7.1.3 Heart Disease According to Age

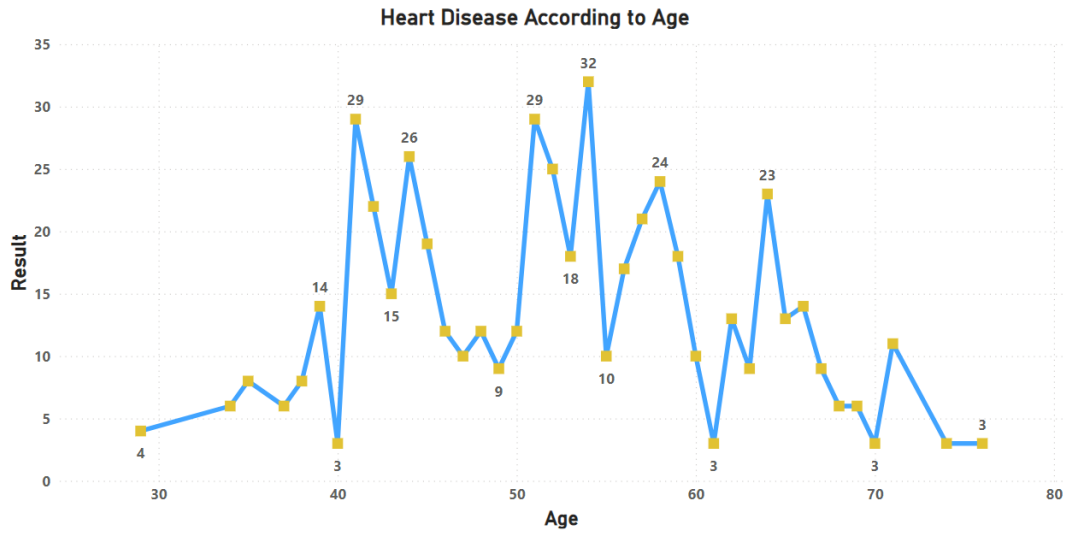


Figure 4.36 Heart Disease According to Age

Then, the Figure 4.36 above shows the line graph that visualized from the predicted result. The graph shows the number of heart disease according to the age. From the result, it shows that the people who in the age of 54 get the highest of heart disease hance. There are 32 numbers of patients from the sample of 535 patients who get the highest heart disease in the age 54. According to (Alexander et al., 2003), most of the male patient used to always focus on their work and didn't care on their health. So, it will lead to an unhealthy diet and slowly when the age grows older, they are getting the higher chance to have heart disease.

4.7.1.4 Heart Disease According to “thalach”

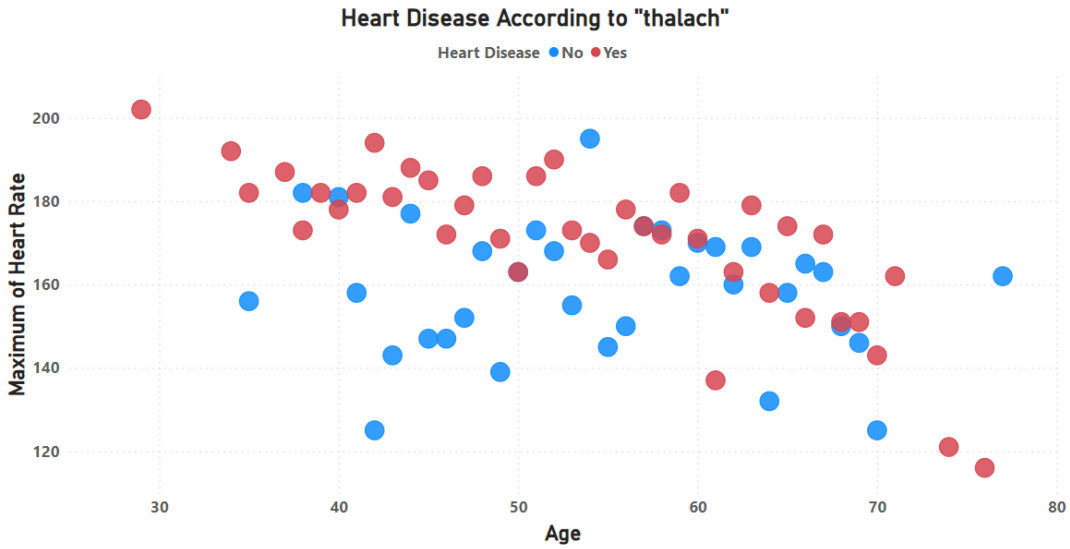


Figure 4.37 Maximum of Heart Rate

Next, the Figure 4.37 above shows the dotted graph that visualized from the predicted result. The graph shows the Maximum of Heart Rate according to the age. The maximum heart rate is calculate form the attribute “thalach”. It is maximum heart rate that a person's achieved. From the result, it shows that the patient who in the age 29 have the highest number of heart rate. According to (Erikssen & Rodahi, 1979), middle-age people used to have the higher heart rate due to their daily activities. This is because in that range of age, many people are stress with their work and so on. Therefore, they are high in risk to get heart disease if they not having a healthy diet regulary.

4.7.1.5 Heart Disease According to (cp) Chest Pain

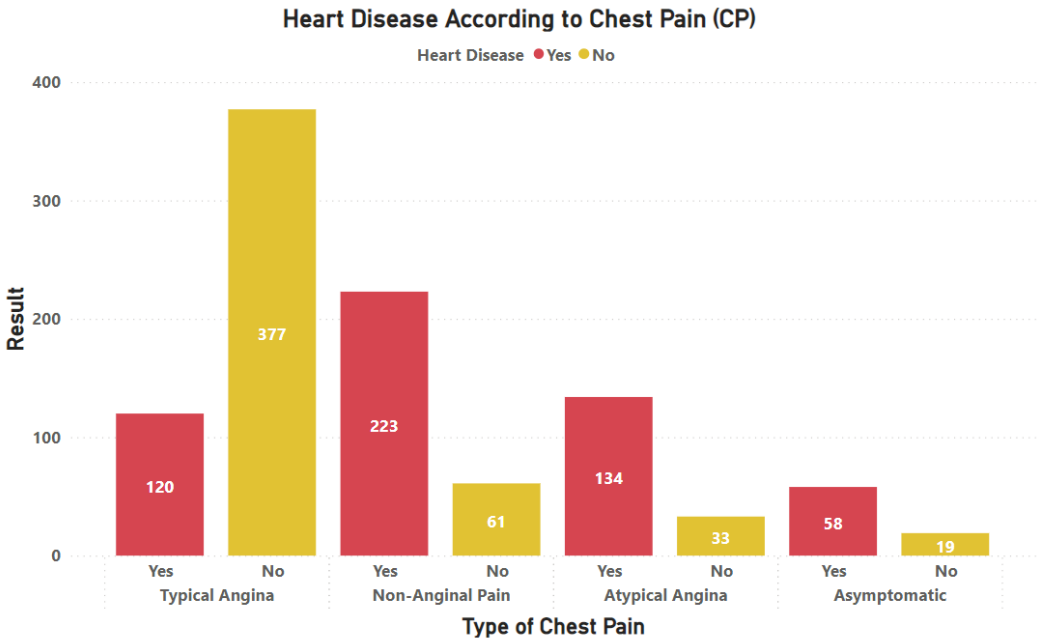


Figure 4.38 Heart Disease According to Chest Pain

Besides that, the Figure 4.38 shows the bar graph that visualized from the predicted result. The graph shows the Chest Pain Type according to the gender. The of chest pain is calculate according to the gender form the attribute “cp”. “cp” is a pain in any area of your chest. If you don't treat it right away, it could spread to other parts of your body, like your arms, neck, or jaw. Pain in the chest can feel sharp or dull. You might feel tight, achy, or like your chest is being squeezed or crushed. Pain in the chest can last a few minutes or a few hours. It has four types of Chests Pain such as Typical Angina, Atypical Angina, Non-anginal Pain, and Asymptomatic. Atypical pain is often defined as pain in the chest and abdomen or back, or as pain that is burning, stabbing, or like stomach aches(Kaski et al., 1991). Pain in the chest, arm, or jaw that is dull, heavy, tight, or crushing is a common sign. Non-Anginal Pain is one of the chest pains caused by heart disease(Kite et al., 2020). It feels like your chest is being squeezed or tightened, or like there is pressure or weight on it, especially behind your sternum. We might feel it on the right, left, or right in the middle. The Asymptomatic left ventricular systolic dysfunction (ALVSD), also called stage B heart failure, is low left ventricular pulse rate function that doesn't cause any symptoms(Gibbons et al., n.d.). From the graph above, we can observe that the patients that getting typical angina only 120 patients that get

disease out of 497 patients. Next, for the non-anginal pain patients, there are 223 patients that get disease out of 284. Next, the Atypical Angina patients that get heart disease have 134 patients out of . Besides that, for Asymptomatic patients that get heart disease have 58 patients from 77 patients.

4.7.1.6 Correlation between features

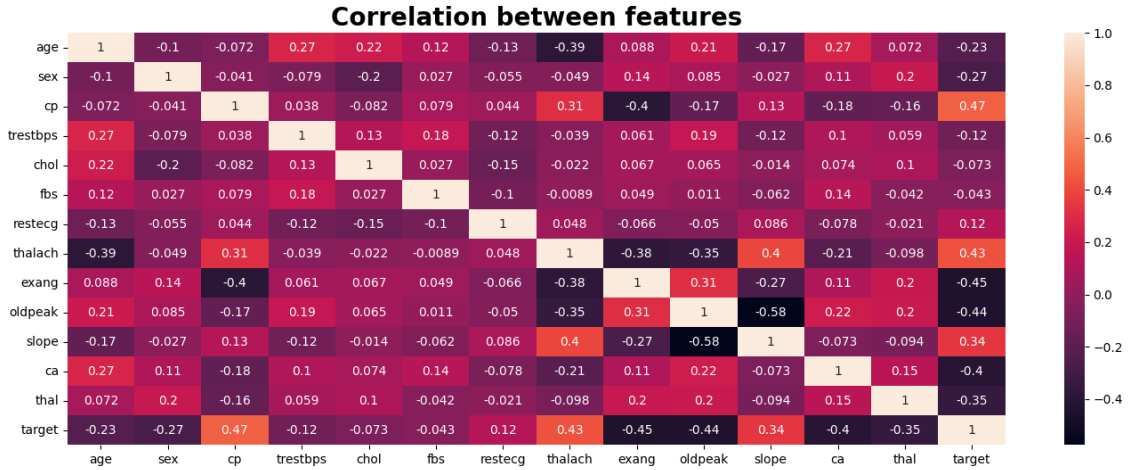


Figure 4.39 Correlation Coefficient

After that, the product moment correlation coefficient technique may be used to determine whether there is a correlation between the attributes, and the heatmap can then be constructed. The feature with the highest degree of correlation will be chosen for inclusion in the result of the model.

The technique that has been used to construct a predictive model is called feature selection. It is the procedure that has been used to limit the number of input variables by picking a relevant feature.

In the field of machine learning, the reason we suggested using feature selection was so that we could improve the overall performance of the algorithm and get highly effective results when it came to training the machine.

The technique of selecting the features that will most significantly contribute to the desired outcome of a prediction is known as "feature selection." This may be done either automatically or manually.

Having unnecessary characteristics in the dataset will lead to a decline in the accuracy of the models; nevertheless, the machine will function more effectively if the unnecessary dataset is distinguished.

CHAPTER 5

CONCLUSION

5.1 Objective Revisit

1) To study the effectiveness of prediction if the patient suffers from heart disease.

   This research was successfully to identify the result of the heart disease prediction by using the intelligent system, Case Base Reasoning (CBR).

2) To develop the prediction technique with and Case Base Reasoning (CBR) for heart disease.

   The Case Base Reasoning (CBR) algorithm with the data splitting and data pre-processing was used in this heart disease prediction. In the CBR algorithm, there are 4 stages which are Retrieve, Reuse, Revise and Retain where the Local and Global Similarity algorithm will be applied in the Retrieve stage. The data was split in to the specific ratio of training and testing data and the data will be going through the data pre-processing which is data normalization.

3) To evaluate the outcome of heart disease using the selected intelligent system approaches.

   In this research paper, the Case Base Reasoning (CBR) algorithm was used to calculate the local and global similarity of the data that has been split into training and testing. Then the outcome data that contain the highest similarity will be selected and used as the result for the new cases or data.

5.2 Limitation

Throughout this research work, I was suffered from a few limitations. Firstly, is regarding to the size of the dataset. Since Case Based Reasoning (CBR) is an Intelligent System and the algorithm that used in this system is to compare the similarity between each training data and gain the highest similarity data's result as the new cases result. Therefore, it requires a large or huge amount of data to ensure the accuracy of the prediction but in this research, there are only contain 1025 set of data where it is not enough to get a higher accuracy of prediction. The more data used in CBR, the more accurate its predictions are likely to be.

Next, is the Case Based Reasoning (CBR) algorithm. This algorithm consumes a lot of time during the prediction process begin. The reason is because this algorithm will compare the local and global similarity between each training data in order to identify the training data that achieve the highest similarity. It will compare one by one of the training data. So, if the number of the dataset is large, it will require a longer time to predict the result of heart disease.

Finally, is the software or tools that used for the prediction. There are 2 open source software or tools that I used in this research for prediction purpose which are Jupyter Notebook and Google Colab. The limitation in the Jupyter Notebook is that it consumes a lot of memory usage. Jupyter Notebook is a plugin that installed in the IDE tools for executing the Python machine learning. Since, the CBR is a time consuming algorithm, then during the process of the prediction, it used a lot of local host memory. Sometimes, it will show Page Unresponsive error during the middle of the prediction process. Then the Google Colab is a cloud-based tools that allow the user to write and execute arbitrary python code through the browser without downloading and installing anything to our laptop or PC. The constraints that I faced when using the Google Colab to execute the CBR algorithm for the heart disease prediction is the limitation of the line output. By using Google Colab, it only able to view or display up to 5000 lines only. If the output is more than 5000 lines, the Google Colab will pop out the "streaming output truncated to the last 5000 lines" warning.

5.3 Future Works

In future, this research can be improve by applying different dataset from different website. Currently, the dataset that used for this Heart Disease Prediction by Using Case Based Reasoning contain of 1025 number of data and 13 number of attributes The accuracy that can get from this dataset is 97.95%. In future, the different dataset can have a greater number of data and attributes so that the dataset that achieve the highest accuracy can be identify.

Another future works that could be done is to use another algorithm for this heart prediction. Currently, this research is using the Case Base Reasoning (CBR). Although this CBR able to perform the heart disease prediction, but it takes a long time to execute, and it consume a lot of memory in computer. Therefore, I would like to suggest the machine learning or deep learning such as artificial neural networks (ANN), naive Bayes, decision tree, or random forest to predict the heart disease.

REFERENCES

Donald M Lloyd-Jones, Martin G Larson, Alexa Beiser, & Daniel Levy. (1999). *Lifetime risk of developing coronary heart disease*.

Adali, E., & Akdeniz Üniversitesi. (2022). *2. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı = 2nd International Conference on Computer Science and Engineering : Antalya - Türkiye 5-8 Ekim (October) 2017*.

Alexander, C. M., Landsman, P. B., Teutsch, S. M., & Haffner, S. M. (2003). NCEP-Defined Metabolic Syndrome, Diabetes, and Prevalence of Coronary Heart Disease Among NHANES III Participants Age 50 Years and Older. In *1210 DIABETES* (Vol. 52). http://diabetesjournals.org/diabetes/article-pdf/52/5/1210/655112/db0503001210.pdf

Chen, A., Huang, S., Hong, P., Cheng, C., & Lin, E. (2011). HDPS: Heart Disease Prediction System. In *2011 Computing in Cardiology*.

David Lapp. (2019). *Heart Disease Dataset | Kaggle*. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Erikssen, J., & Rodahi, K. (1979). Resting Heart Rate in Apparently Healthy Middle-aged Men. In *European Journal of Applied Physiology and Occupational Physiology* (Vol. 42). Springer.

Gibbons, L. W., Mitchell, T. L., Wei, M., Blair, S. N., & Cooper, K. H. (n.d.). *Maximal Exercise Test as a Predictor of Risk for Mortality from Coronary Heart Disease in Asymptomatic Men*.

Himanshu Sharma & M A Rizvi. (2017). Prediction of Heart Disease using Machine Learning Algorithms: A Survey. *International Journal IJRITCC, 5*(8).

Jabbar, M. A., Chandra, P., & Deekshatulu, B. L. (2012). Prediction of risk score for heart disease using associative classification and hybrid feature subset selection. *International Conference on Intelligent Systems Design and Applications, ISDA*, 628–634. https://doi.org/10.1109/ISDA.2012.6416610

Jha, H. C., Divya, A., Prasad, J., & Mittal, A. (2010). Plasma circulatory markers in male and female patients with coronary artery disease. *Heart and Lung: Journal of Acute and Critical Care*, *39*(4), 296–303. https://doi.org/10.1016/j.hrtlng.2009.10.005

Kaski, J. C., Tousoulis, D., Gavrielides, S., Mcfadden, E., Galassi, A. R., Crea, F., & Maseri, A. (1991). *Comparison of Epicardial Coronary Artery Tone and Reactivity in Prinzmetal's Variant Angina and Chronic Stable Angina Pectoris* (Vol. 17, Issue 5).

Kite, T. A., Gaunt, H., Banning, A. S., Roberts, E., Kovac, J., Hudson, I., & Gershlick, A. H. (2020). Clinical outcomes of patients discharged from the Rapid Access Chest Pain Clinic with non-anginal chest pain: A retrospective cohort study. *International Journal of Cardiology*, *302*, 1–4. https://doi.org/10.1016/j.ijcard.2019.12.008

Prakash, P. (2015). Decision Support System In Heart Disease Diagnosis By Case Based Recommendation. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, *4*, 2. www.ijstr.org

Rahma Atallah, & Amjed Al-Mousa. (2022). *2019 2nd International Conference on New Trends in Computing Sciences (ICTCS) : proceedings : Amman, Jordan, 9-11 October 2019.*

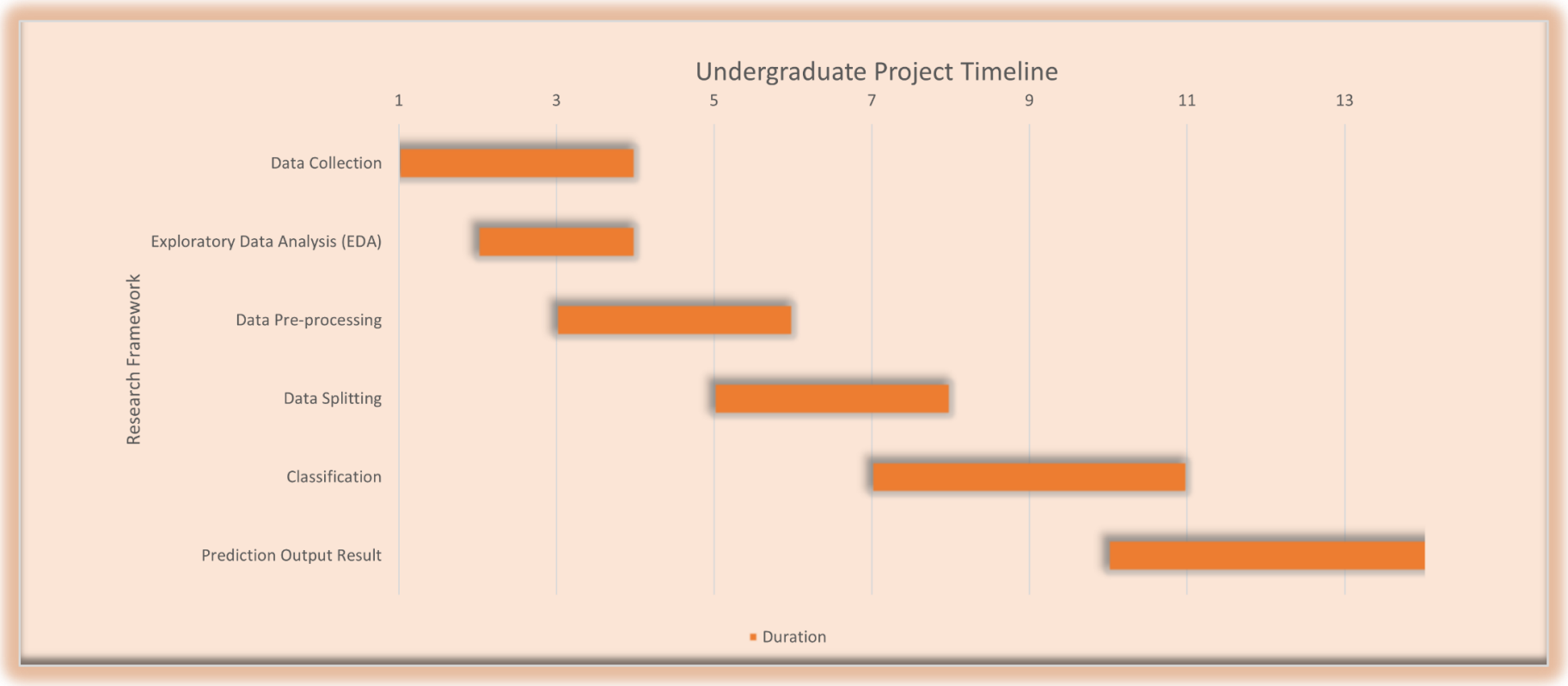
Rajdhan, A., Agarwal, A., & Sai, M. (2020). Heart Disease Prediction using Machine Learning. In *IJERT Journal International Journal of Engineering Research & Technology*. www.ijert.org

Ramalingam, V. v., Dandapath, A., & Karthik Raja, M. (2018). Heart disease prediction using machine learning techniques: A survey. *International Journal of Engineering and Technology(UAE)*, *7*(2.8 Special Issue 8), 684–687. https://doi.org/10.14419/ijet.v7i2.8.10557

Shah, D., Patel, S., & Bharti, S. K. (2020). Heart Disease Prediction using Machine Learning Techniques. *SN Computer Science*, *1*(6), 345. https://doi.org/10.1007/s42979-020-00365-y

Tanmay Kasbe, & Ravi Singh Pippal. (2017). *International Conference on Energy, Communication, Data Analytics & Soft Computing (ICECDS) - 2017 : 1st & 2nd August 2017*.

APPENDIX A
GANTT CHART



APPENDIX B
SOURCE CODE IN PYTHON

CBR Code (Retrieve, Reuse, Revise and Retain

```
   # Retrieve Cycle  -- Find similiar problems

   for k in range(test_normalize.shape[0]):
       print('==========================================================
====================')
       print(f'\t\t\t TEST DATA Case: {k+1}')
       print('==========================================================
====================')
       for i in range(case_base.shape[0]):
           total_similarity = 0
           print('==========================================================
========================')
           print(f'\t\t LOCAL SIMILARITY for TEST_DATA Case: {k+1}')
           print('==========================================================
========================')
           for j in range(case_base.shape[1]):
               # Calculate Local Similarity
               local_similarity[i,j]=1-(abs(test_normalize.iloc[k,j]-
case_base.iloc[i,j])/range_value[j])

               # Calculate Global Similarity
               total_similarity += (local_similarity[i,j] *
weightage[j])
               global_similarity[i] = (1/total_weightage) *
total_similarity
               print(f'Local Similarity TRAIN DATA Case {i+1},
Attribute: {train_normalize.columns[j]} \t = {local_similarity[i,j]}')

           print('----------------------------------------------------------
------------------------')
           print(f'Global Similarity TRAIN DATA Case {i+1} =
{global_similarity[i]}')
           print('----------------------------------------------------------
------------------------')

       # Identify the Highest Value of the Global Similarity
       highest_similarity = global_similarity
       highest_index = np.argmax(highest_similarity)
       print('==========================================================
====================')
```

```
        print(f'Highest Similarity for TEST DATA Case {k+1} =
{highest_similarity.max()}')
        print('==========================================================
====================')
        print('\n')

        # Reuse Cycle -- Reuse a previous solution in a new situation
        # Propose solution for new cases from the solutions in the
retrieved cases
        test_data =
np.append(test_ori.iloc[k,:],train_normalize.iloc[highest_index,-1])

        # Revise Cycle
        # There is no Revise Cycle implemented because the 'target'
output column only have 2 values(either 0 or 1)

        # Retain Cycle -- Integrated into case-based system
        # Store the result for the new solution to the train_ori
        test_data = pd.Series(test_data,index = train_ori.columns)


        # Update / Merge the new solution with the previous dataset in
'heart_Training_data_Before_Normalization.csv'
        train_ori = train_ori.append(test_data,ignore_index=True)
        train_ori.to_csv('heart_Training_data_Before_Normalization.csv',
index=False)
```

Data Splitting Code (60:40)

```
   # Split Data into ratio 60:40 (60% Training Data & 40% Testing Data)
   from sklearn.model_selection import train_test_split

   attribute_column = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs',
'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal',
'target']  # Attribute in the Dataset
   data = full_data[attribute_column]  # attribute

   training_data,testing_data = train_test_split(data, test_size=0.4,
shuffle=False)

   print(training_data) # 60% Random Training Dataset
   print()

   print(testing_data) # 40% Random Testing Dataset
   print()

   #Eliminate the outcome column(target) for testing_data
   testing_data_no_output =
testing_data.iloc[:,range(testing_data.shape[1]-1)]
```

Data Normalization Code

```
   # Normalization Process for 60% random train data
   from sklearn.preprocessing import Normalizer

   normal = Normalizer()

   train_normalization.iloc[:,0:-1] =
normal.fit_transform(train_normalization.iloc[:,0:-1])

   train_normalization
```