



Texas A&M University - Commerce
Department of Computer Science

Comparative Analysis Of Heart Disease Detection Using Standard Machine Learning Models

Swetha Paspunuri

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

February 18, 2024

Declaration

I, Swetha Paspunuri, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Swetha Paspunuri
February 18, 2024

Abstract

Recently cardiovascular diseases has been on the rise, even affecting newborns. Detecting heart-related diseases early is vital because it helps doctors start treatment sooner, leading to better results for patients and less strain on healthcare resources. With more and more people facing heart problems, it's crucial to have advanced predictive tools. Using the abundant data available in cardiology, our project aims to integrate the technology into health care for predictive modelling. The primary goal of this project is to develop an efficient heart disease prediction system using various machine learning models to predict coronary artery disease(CAD) with utmost precision and effectiveness. We employed a dataset consisting of necessary patient information from online sources to train and validate our models. The first step is cleaning and preprocessing data that allow us to find key patterns for training the models. This research trains a Logistic Regression (LR), Random Forest (RF) and Naive Bayes (NB) model for classification on the heart disease dataset. We evaluate these models using standard measures like precision, which tells us how accurate positive predictions are; recall, which shows how well the models capture all actual positive cases; and the F1 score, which balances both precision and recall.

Keywords: Logistic Regression(LR), Random Forest(RF), Naive Bayes(NB) , F1 score, Precision.

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

1	Introduction	1
1.1	Background	1
1.2	Research Question	2
1.2.1	Aims and objectives	2
1.3	Solution approach	2
1.4	Summary of contributions and achievements	3
2	Literature Review	4
2.1	Introduction to Heart Disease	4
2.2	Background on Machine Learning Models	4
2.2.1	Logistic Regression	4
2.2.2	Naïve Bayes	4
2.2.3	Random Forest	4
2.3	Performance Measures for Evaluation	5
2.4	Description of the Dataset	5
2.5	Summary of Literature Reviewed	5
3	Methodology	6
3.1	Algorithms Descriptions	6
3.1.1	Logistic Regression	6
3.1.2	Naïve Bayes	6
3.1.3	Random Forest	6
3.2	Implementations	7
3.2.1	Logistic Regression	7
3.2.2	Naïve Bayes	7
3.2.3	Random Forest	7
3.3	Experiments Design	7
3.4	Algorithms	7
3.5	Examples of the sections of a methodology chapter	9
3.5.1	Example of a software/Web development main text structure	9
3.5.2	Example of an algorithm analysis main text structure	9
3.5.3	Example of an application type main text structure	10
3.5.4	Example of a science lab-type main text structure	11
3.6	Example of an Equation in \LaTeX	11
3.7	Example of a Figure in \LaTeX	11

3.8	Example of an algorithm in \LaTeX	13
3.9	Example of code snippet in \LaTeX	13
3.10	Example of in-text citation style	15
3.10.1	Example of the equations and illustrations placement and reference in the text	15
3.10.2	Example of the equations and illustrations style	15
3.11	Summary	16
4	Results	17
4.1	A section	17
4.2	Example of a Table in \LaTeX	18
4.3	Example of captions style	18
4.4	Summary	18
5	Discussion and Analysis	19
5.1	A section	19
5.2	Significance of the findings	19
5.3	Limitations	19
5.4	Summary	19
6	Conclusions and Future Work	20
6.1	Conclusions	20
6.2	Future work	20
7	Reflection	21
	Appendices	23
A	An Appendix Chapter (Optional)	23
B	An Appendix Chapter (Optional)	24

List of Figures

3.1	Example figure in \LaTeX	12
-----	---	----

List of Tables

3.1	Undergraduate report template structure	9
3.2	Example of a software engineering-type report structure	10
3.3	Example of an algorithm analysis type report structure	10
3.4	Example of an application type report structure	10
3.5	Example of a science lab experiment-type report structure	11
4.1	Example of a table in \LaTeX	18

List of Abbreviations

LR	Logistic Regression
RF	Random Forest
NB	Naive Bayes
CAD	Coronary Artery Disease

Chapter 1

Introduction

Today, heart problems are a major health concern affecting individuals worldwide. Many people are suffering from heart issues like heart disease, heart failure, and irregular heartbeats Webb et al. (2015). Heart problems can affect people, not just physically but also emotionally. Those with heart conditions often find it difficult to live normally and face many difficulties. Additionally, the financial side of managing heart problems adds an extra layer of challenges. Spotting heart-related issues early is crucial. It helps healthcare professionals to step in quickly, enhance patient outcomes, and ease the strain on healthcare resources. Early detection allows for timely intervention, potentially preventing the progression of heart conditions. To address the need for early detection, our project focuses on developing a machine learning model capable of accurately identifying the presence of heart diseases. In this endeavor, we utilize a heart-related issue dataset from (Janosi et al., 1988), sourced from the online repository UC Irvine. The data undergoes thorough cleaning and pre-processing to extract useful information essential for training the machine learning model. The machine learning algorithms employed, as highlighted by (Sharma et al., 2020), include LR, NB, and RF classification. These algorithms have demonstrated effectiveness in detecting coronary artery disease by evaluating outputs based on various factors such as resting blood pressure, serum cholesterol, maximum heart rate achieved, and more. Furthermore, our project aims not only to detect heart-related issues but also to contribute valuable insights to the broader field of cardiovascular health. By leveraging advanced algorithms, we seek to ensure the effective prediction of heart-related problems, potentially revolutionizing the early diagnosis and management of cardiovascular conditions.

1.1 Background

Our project focuses on addressing the issue of cardiovascular diseases in today's world, affecting everyone irrespective of their age. The primary motivation behind our work is to detect the heart-related diseases as early as possible. This identification helps doctors to start the treatment sooner, to improve patient results and effectively using the healthcare resources. For this, our project focuses on integrating technology into healthcare by using the abundant data in cardiology for predictive modeling. The primary goal is to develop an efficient heart disease prediction system by concentrating on predicting CAD with precision and effectiveness. So, we use different machine learning models such as LR, RF, NB. These models play a crucial role in predicting and understanding heart-related issues. The project commences with data preprocessing to extract

essential patterns required for training the models, aligning with the objectives and research approach outlined in subsequent sections. Our project becomes significant as it can give better resources to doctors for finding and handling heart problems early on. We want to help make hearts healthier by explaining some crucial ideas and ways to use them in a simple way.

1.2 Research Question

How can machine learning models, specifically Logistic Regression, Naïve Bayes, and Random Forest classification algorithms, be effectively utilized to develop a heart disease prediction system for early detection of Coronary Artery Disease, with a focus on improving patient outcomes and contributing to advancements in cardiovascular health?

1.2.1 Aims and objectives

Aims: To develop and implement an advanced heart disease prediction system, utilizing machine learning models for early detection of CAD, with the ultimate goal of enhancing patient outcomes and contributing to the ongoing global efforts in cardiovascular health. **Objectives:**

- Obtain and analyze the heart disease dataset, clean and preprocess the data for model training.
- Train LR classifier, optimizing meta-parameters for improved performance.
- Develop NB classifier, focusing on feature selection and parameter tuning.
- Utilize RF algorithm to construct decision tree ensembles, refining predictive capabilities.
- Integrate trained models into healthcare systems for real-time heart disease prediction.
- Provide healthcare professionals with valuable insights and resources for informed decision-making.

1.3 Solution approach

The solution approach involves a thorough step-by-step method designed to create an advanced system for predicting heart disease, specifically focusing on early detection of CAD, with the ultimate goal of achieving our defined aim and objectives. The project commences with the acquisition of a dataset related to cardiac issues obtained from the research conducted by Janosi et al. (1988), which is accessible through UC Irvine. Following that, we carefully clean and process the data to find important patterns needed for training our machine learning models. We incorporate technology into healthcare by using different smart algorithms, like LR, NB, and RF classification. These specific algorithms are chosen because they are proficient in effectively CAD, by taking into account key factors such as resting blood pressure, serum cholesterol, and maximum heart rate achieved. The models are evaluated using specific metrics like precision, recall, and the F1 score to provide us with the understanding of how well they are performing.

1.4 Summary of contributions and achievements

Describe clearly what you have done/created/achieved and what the major results and their implications are.

Chapter 2

Literature Review

2.1 Introduction to Heart Disease

Cardiovascular diseases pose a significant threat to global health, affecting individuals of all ages. These conditions, including heart disease, heart failure, and irregular heartbeats, have profound physical and emotional impacts on affected individuals. Early detection and effective management are critical in mitigating the adverse effects of heart-related issues and improving patient outcomes.

2.2 Background on Machine Learning Models

2.2.1 Logistic Regression

LR is a statistical method used for binary classification tasks. It models the probability of a binary outcome based on one or more predictor variables. In the context of heart disease prediction, LR can analyze patient parameters such as age, cholesterol levels, and blood pressure to estimate the likelihood of the presence of heart disease. LR is widely used in healthcare research due to its simplicity, interpretability, and ability to handle linear relationships between variables.

2.2.2 Naïve Bayes

NB is a probabilistic classifier based on Bayes' theorem with an assumption of independence between features. Despite its simplistic assumption, NB has been shown to perform well in various classification tasks, including text categorization and medical diagnosis. In heart disease prediction, NB can effectively analyze patient attributes and calculate the conditional probability of heart disease given the observed features.

2.2.3 Random Forest

RF is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. RF is known for its robustness and ability to handle high-dimensional data. In heart disease prediction, RF can analyze a large number of patient parameters and identify complex patterns associated with cardiovascular conditions.

2.3 Performance Measures for Evaluation

To evaluate the performance of our machine learning models, we will employ several performance measures, including accuracy, precision, recall, and F1-score. These metrics provide insights into the models' ability to correctly classify instances of heart disease and non-heart disease cases. By evaluating multiple performance measures, we can assess the overall effectiveness of our predictive models and identify areas for improvement.

2.4 Description of the Dataset

Our project utilizes the heart disease dataset sourced from UC Irvine, compiled by Janosi et al. (1988). This dataset contains various patient attributes, such as age, sex, cholesterol levels, and resting blood pressure, along with the presence or absence of heart disease. We preprocess the dataset to handle missing values and normalize the features to ensure optimal model performance.

2.5 Summary of Literature Reviewed

The literature review highlights the significance of early detection and effective management in combating cardiovascular diseases. Previous studies have demonstrated the utility of machine learning algorithms in predicting heart disease, with research highlighting the importance of feature selection, parameter tuning, and model evaluation. By building upon existing literature and leveraging advanced predictive tools, our project aims to contribute to the ongoing efforts in cardiovascular health and improve patient outcomes.

Chapter 3

Methodology

Recognizing the need for early detection and management of cardiovascular conditions, we utilize machine learning algorithms to analyze patient data and predict the likelihood of heart disease. Our methodology encompasses data collection, preprocessing, feature extraction, model development, and evaluation, aiming to deliver a robust and effective predictive tool for doctors. By integrating innovative learning algorithms and conducting experiments, we aspire to contribute meaningful solutions and enhance patient outcomes in cardiovascular health.

3.1 Algorithms Descriptions

3.1.1 Logistic Regression

LR is a statistical method used for binary classification tasks. It models the probability of a binary outcome based on one or more predictor variables. In the context of heart disease prediction, LR can analyze patient parameters such as age, cholesterol levels, and blood pressure to estimate the likelihood of the presence of heart disease.

3.1.2 Naïve Bayes

NB is a probabilistic classifier based on Bayes' theorem with an assumption of independence between features. In heart disease prediction, Naïve Bayes can effectively analyze patient attributes and calculate the conditional probability of heart disease given the observed features. Its simplicity and computational efficiency make Naïve Bayes a popular choice for healthcare applications.

3.1.3 Random Forest

RF is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. In heart disease prediction, Random Forest can analyze a large number of patient parameters and identify complex patterns associated with cardiovascular conditions.

3.2 Implementations

3.2.1 Logistic Regression

For LR implementation, we'll utilize the 'LogisticRegression' class from the 'sklearn.linear_model' module. We'll preprocess the dataset, including handling missing values and scaling features, before fitting the model to the training data.

3.2.2 Naïve Bayes

For NB implementation, we'll use the 'GaussianNB' class from the 'sklearn.naive_bayes' module. Similar to Logistic Regression, we'll preprocess the dataset and fit the model to the training data.

3.2.3 Random Forest

For RF implementation, we'll employ the 'RandomForestClassifier' class from the 'sklearn.ensemble' module. We'll preprocess the dataset and tune hyperparameters, such as the number of estimators and maximum depth, to optimize model performance.

3.3 Experiments Design

In our experimental approach to assess the predictive performance of each algorithm for heart disease, we'll begin by dividing our dataset into separate training and testing sets. This division ensures that the models are trained on a subset of the data and evaluated on an independent portion, enabling us to gauge their generalization capability. To further fortify the reliability of our findings, we'll employ cross-validation techniques. This involves iteratively partitioning the dataset into multiple subsets, training the models on different combinations, and validating them on the remaining data, thus providing a more comprehensive evaluation. Subsequently, we'll utilize a range of performance metrics, including accuracy, precision, recall, and F1-score, to quantify the algorithms' effectiveness. These metrics will allow us to discern not only the models' overall correctness but also their ability to precisely identify positive cases and recall them accurately. By meticulously analyzing these performance indicators, we aim to determine the most optimal approach for heart disease prediction, considering factors such as model interpretability and computational efficiency alongside predictive accuracy.

3.4 Algorithms

In our project, we implement three distinct machine learning algorithms—Logistic Regression, Random Forest, and Naive Bayes—to predict heart disease. Logistic Regression is a simple yet powerful algorithm used for binary classification tasks, where it models the probability of a binary outcome. Random Forest, on the other hand, is an ensemble learning method that constructs multiple decision trees and aggregates their predictions to improve accuracy. Naive Bayes is a probabilistic classifier based on Bayes' theorem, particularly effective for datasets with high dimensionality and strong feature independence assumptions. Each algorithm offers unique advantages and approaches in identifying patterns and making predictions, contributing to our comprehensive analysis of heart disease prediction.

Algorithm 1 Logistic Regression

Input: Training dataset (X_{train}, Y_{train}) , Test dataset X_{test} **Output:** Predicted class labels for X_{test}

```

1: function LOGISTICREGRESSION( $X_{train}, Y_{train}, X_{test}$ )
2:   Initialize logistic regression classifier
3:   Standardize features:  $X_{train} \leftarrow \text{StandardScaler.fit\_transform}(X_{train})$ 
4:   Fit classifier to training data:  $\text{classifier.fit}(X_{train}, Y_{train})$ 
5:   Standardize test features:  $X_{test} \leftarrow \text{StandardScaler.transform}(X_{test})$ 
6:   Predict probabilities for test data:  $y_{prob} \leftarrow \text{classifier.predict\_proba}(X_{test})$ 
7:   Convert probabilities to class labels:  $y_{pred} \leftarrow \text{threshold\_function}(y_{prob})$ 
8:   return  $y_{pred}$ 
9: end function

```

Algorithm 2 Random Forest

Input: Training dataset (X_{train}, Y_{train}) , Test dataset X_{test} **Output:** Predicted class labels for X_{test}

```

1: function RANDOMFOREST( $X_{train}, Y_{train}, X_{test}$ )
2:   Initialize random forest classifier with specified parameters
3:   Handle missing values:  $X_{train}, X_{test} \leftarrow \text{Imputer.fit\_transform}(X_{train}, X_{test})$ 
4:   Fit classifier to training data:  $\text{classifier.fit}(X_{train}, Y_{train})$ 
5:   Predict class labels for test data:  $y_{pred} \leftarrow \text{classifier.predict}(X_{test})$ 
6:   return  $y_{pred}$ 
7: end function

```

Algorithm 3 Naive Bayes

Input: Training dataset (X_{train}, Y_{train}) , Test dataset X_{test} **Output:** Predicted class labels for X_{test}

```

1: function NAIVEBAYES( $X_{train}, Y_{train}, X_{test}$ )
2:   Initialize naive Bayes classifier
3:   Discretize continuous features:  $X_{train}, X_{test} \leftarrow \text{Binarizer.fit\_transform}(X_{train}, X_{test})$ 
4:   Fit classifier to training data:  $\text{classifier.fit}(X_{train}, Y_{train})$ 
5:   Predict class labels for test data:  $y_{pred} \leftarrow \text{classifier.predict}(X_{test})$ 
6:   return  $y_{pred}$ 
7: end function

```

3.5 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project’s needs.

Table 3.1: Undergraduate report template structure

Frontmatter	Title Page
	Abstract
	Acknowledgements
	Table of Contents
	List of Figures
	List of Tables
	List of Abbreviations
Main text	Chapter 1 Introduction
	Chapter 2 Literature Review
	Chapter 3 Methodology
	Chapter 4 Results
	Chapter 5 Discussion and Analysis
	Chapter 6 Conclusions and Future Work
	Chapter 7 Refection
End matter	References
	Appendices (Optional)
	Index (Optional)

3.5.1 Example of a software/Web development main text structure

Notice that the “methodology” Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that “Chapter 5” in Table 3.2 is “Testing and Validation” which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

3.5.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

Table 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Requirements specifications Analysis Design Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

Table 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Algorithms descriptions Implementations Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.5.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Table 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Problems (tasks) descriptions Algorithms/tools/technologies/etc. descriptions Implementations Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.5.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the “Methodology” section as “Materials and Methods.”

Table 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	Problems (tasks) description Materials Procedures Implementations Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.6 Example of an Equation in \LaTeX

Eq. 3.1 [note that this is an example of an equation’s in-text citation] is an example of an equation in \LaTeX . In Eq. (3.1), s is the mean of elements $x_i \in \mathbf{x}$:

$$s = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3.1)$$

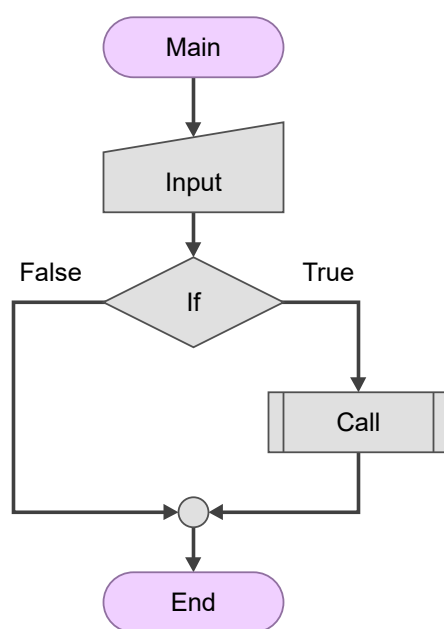
Have you noticed that all the variables of the equation are defined using the **in-text** maths command $\$$, and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

3.7 Example of a Figure in \LaTeX

Figure 3.1 is an example of a figure in \LaTeX . For more details, check the link:

wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.

Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the “Flow algorithm software”. Here is the link for that: flowgorithm.org.

Figure 3.1: Example figure in \LaTeX .

3.8 Example of an algorithm in \LaTeX

Algorithm 4 is a good example of an algorithm in \LaTeX .

Algorithm 4 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \dots, x_N$

Output: *EvenSum* (Sum of even numbers in \mathbf{x})

```

1: function EVENSUMMATION( $\mathbf{x}$ )
2:   EvenSum  $\leftarrow$  0
3:    $N \leftarrow \text{length}(\mathbf{x})$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:     if  $x_i \bmod 2 == 0$  then                                ▷ check if a number is even?
6:       EvenSum  $\leftarrow$  EvenSum +  $x_i$ 
7:     end if
8:   end for
9:   return EvenSum
10: end function

```

3.9 Example of code snippet in \LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 4 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```

1 import numpy as np
2
3  $\mathbf{x}$  = [0, 1, 2, 3, 4, 5] # assign values to an array
4 evenSum = evenSummation( $\mathbf{x}$ ) # call a function
5
6 def evenSummation( $\mathbf{x}$ ):
7     evenSum = 0
8      $n = \text{len}(\mathbf{x})$ 
9     for  $i$  in range( $n$ ):
10         if np.mod( $\mathbf{x}[i]$ ,2) == 0: # check if a number is even?
11             evenSum = evenSum +  $\mathbf{x}[i]$ 
12     return evenSum

```

Listing 3.1: Code snippet in \LaTeX and this is a Python code example

Here we used the “\clearpage” command and forced-out the second listing example onto the next page.

```

1 public class EvenSum{
2     public static int evenSummation(int[] x){
3         int evenSum = 0;
4         int n = x.length;
5         for(int i = 0; i < n; i++){
6             if(x[i]%2 == 0){ // check if a number is even?
7                 evenSum = evenSum + x[i];
8             }
9         }
10        return evenSum;
11    }
12    public static void main(String[] args){
13        int[] x = {0, 1, 2, 3, 4, 5}; // assign values to an array
14        int evenSum = evenSummation(x);
15        System.out.println(evenSum);
16    }
17 }

```

Listing 3.2: Code snippet in \LaTeX and this is a Java code example

```

1 int evenSummation(int x[]){
2     int evenSum = 0;
3     int n = sizeof(x);
4     for(int i = 0; i < n; i++){
5         if(x[i]%2 == 0){ // check if a number is even?
6             evenSum = evenSum + x[i];
7         }
8     }
9     return evenSum;
10 }
11
12 int main(){
13     int x[] = {0, 1, 2, 3, 4, 5}; // assign values to an array
14     int evenSum = evenSummation(x);
15     cout<<evenSum;
16     return 0;
17 }

```

Listing 3.3: Code snippet in \LaTeX and this is a C/C++ code example

3.10 Example of in-text citation style

3.10.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more than once.

3.10.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase “Eq” for an equation before using an equation number with (`\eqref{.}`). Use “Table” to refer to a table, “Figure” to refer to a figure, “Algorithm” to

refer to an algorithm and “Listing” to refer to listings (code snippets). Note that, we do not use the articles “a,” “an,” and “the” before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence “A report structure is shown in **the** Table 3.1” should be written as “A report structure is shown **in** Table 3.1.”

3.11 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter “**Testing and Validation**” should precede the “Results” chapter. See Section 3.5.1 for report organization of such project.

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in \LaTeX

Table 4.1 is an example of a table created using the package \LaTeX “booktabs.” do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of lines (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in \LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork)** goes **below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table** goes **above** the table. See the example in Table 4.1.
- The **caption of an Algorithm** goes **above** the algorithm. See the example in Algorithm 4.
- The **caption of a Listing** goes **below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with “Results” Chapter as “ Results and Discussion” as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader’s understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summarizes the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most significant contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in \LaTeX and use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planned aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is “**References**,” which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between “References” and “Bibliography.” The list of “References” strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of “References.”

References

- Janosi, A., Steinbrunn, W., Pfisterer, M. and Detrano, R. (1988), 'Heart disease', UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C52P4X>.
- Sharma, V., Yadav, S. and Gupta, M. (2020), Heart disease prediction using machine learning techniques, *in* '2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)', pp. 177–181.
- Webb, G., Mulder, B. J., Aboulhossn, J., Daniels, C. J., Elizari, M. A., Hong, G., Horlick, E., Landzberg, M. J., Marelli, A. J., O'Donnell, C. P. et al. (2015), 'The care of adults with congenital heart disease across the globe: current assessment and future perspective: a position statement from the international society for adult congenital heart disease (isachd)', *International journal of cardiology* **195**, 326–333.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...