



SCHOOL OF COMPUTER SCIENCE

ASSESSMENT TASK 4: Group Assignment (Weightage 30%)

AUGUST 2024 SEMESTER

MODULE NAME	:	DATA MINING
MODULE CODE	:	ITS61504
DUE DATE	:	TUESDAY, 13th FEBRUARY, 2024
PLATFORM	:	MyTIMES


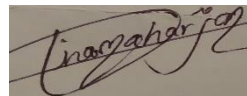
This paper consists of SEVEN (7) pages, inclusive of this page.

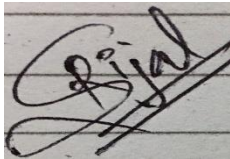
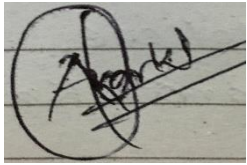

GROUP NO: “A”

ASSIGNMENT TOPIC: “PREDICTIVE MODELING for HEART DISEASE”

STUDENT DECLARATION

- I confirm that I am aware of the University's Regulation Governing Cheating in a University Test and Assignment and of the guidance issued by the School of Computing and IT concerning plagiarism and proper academic practice, and that the assessed work now submitted is in accordance with this regulation and guidance.*
- I understand that, unless already agreed with the School of Computing and IT, assessed work may not be submitted that has previously been submitted, either in whole or in part, at this or any other institution.*
- I recognise that should evidence emerge that my work fails to comply with either of the above declarations, then I may be liable to proceedings under Regulation*

No.	Student Name	ID	Date	Signature	Score
1	Prabin Joshi	0358667	13 th Feb, 2024		
2	Lina Maharjan	0358308	13 th Feb, 2024		

3	Srijal Rijal	0358227	13 th Feb, 2024		
4	Aayush Karki	0358660	13 th Feb, 2024		
5	Pooja Thapa	0358808	13 th Feb, 2024		

MARKING RUBRICS

Group Project Marking Rubrics (30 Marks)					
Criteria	Weight (Percentage)	Excellent (90-100)	Good (75-89)	Average (40-74)	Poor(0-39)
Abstract	5	Abstract is written in a clear and concise manner with information to understand the background of the case study.	Abstract is written in a clear manner with some information to understand the background of the case study.	Abstract is written with limited information to understand the background of the case study.	Abstract is poorly written with limited / no information to understand the background of the case study.
Introduction	10	Background is described in a clear and concise manner with information to understand the business nature and example.	Background is described in a clear and concise manner with some information about the business nature.	Background is described in clear manner with little / limited information about the business nature.	Background is described in brief manner with limited / no information about the business manner.
EDA	10	Dataset characteristics are elaborated in very precise and concise manner, with sample dataset given.	Dataset characteristics are elaborated in clear manner, with sample dataset given.	Dataset characteristics are briefly elaborated, with sample dataset given.	Dataset characteristic are listed, with sample dataset given.
Dataset Issue	15	3 dataset issues are described in precise manner with examples given to support the issues mentioned.	2-3 dataset issues are described in precise manner with brief / no examples to support the issues mentioned.	2-3 dataset issues are mentioned with brief / no examples to support the issues mentioned.	1-2 dataset issues are mentioned with brief / no examples to support the issues mentioned.

Data Preprocessing Techniques	15	3 appropriate pre-processing techniques used with well explanation on why the techniques are chosen	2 appropriate pre-processing techniques used with well explanation on why the techniques are chosen	1 appropriate pre-processing technique used with no explanation on why the technique is chosen	Incorrect preprocessing methods applied with no explanation on why the technique is chosen
Data Mining Technique(s) use	20	3 data mining techniques used with excellent parameters justification and explanation on why the techniques is chosen	2 data mining techniques used with good parameters justification and explanation on why the techniques is chosen	1 data mining technique used with well explanation on why the technique is chosen	1 data mining technique used with no explanation on why the technique is chosen
Results (Performance evaluation)	10	Performance evaluation for the 3 data mining techniques with comprehensive explanation	Performance evaluation for the 3 data mining techniques with sufficient explanation	Performance evaluation for the 3 data mining techniques without proper explanation	Performance evaluation for the 3 data mining techniques without explanation.
Conclusion	5	Appropriate conclusion with well explanation	Appropriate conclusion with no explanation	Incorrect conclusion with explanation	Incorrect conclusion with explanation
Submission requirements & team work	10	Fulfill all submission requirements with excellent content organization and team work among members.	Fulfill all submission requirements with consistent content organization and team work among members.	Some effort in compliance with submission requirement and team work among members.	Less obvious effort in compliance with submission requirement and team work among members

ACKNOWLEDGEMENTS

I would like to acknowledge the “IIMS College” for providing us an opportunity to carry out an group assignment of data mining entitled “Predictive Modeling of Heart Diseases”. We are profoundly grateful and highly indebted to express my deepest gratitude to my lecturer “Dipson Pokhrel” whose continuous support, inspiration, critical supervision, valuable suggestions, and constructive comments have been of invaluable importance for the fulfillment of this independent study. He has truly been the greatest lecturer for us and his mentorship was paramount in providing a well-rounded experience consistent with our long-term career goal. Respectfully, we would like to thank all the teachers and senior students of the “IIMS College” for their inspiration and encouragement during my study. In addition, we are grateful to our friends as well as family for all their continuous support throughout our study.

GROUP CONTRIBUTION

Group Members	Roles
Prabin Joshi	<ul style="list-style-type: none">- Chapter 5: Data Mining Techniques- Abstract- Report Editing and Formatting
Lina Maharjan	<ul style="list-style-type: none">- Chapter 3: Overview of the Datasets- Reference Formatting
Srijal Rijal	<ul style="list-style-type: none">- Chapter 6: Data Mining Results- Conclusion
Aayush Karki	<ul style="list-style-type: none">- Chapter 4: Preprocessing Techniques- Application of Predictive Model Part
Pooja Thapa	<ul style="list-style-type: none">- Chapter 1: Introduction- Chapter 2: Organizational Overview

TABLE OF CONTENTS

MARKING RUBRICS	3
ACKNOWLEDGEMENTS	5
GROUP CONTRIBUTION	6
TABLE OF FIGURES	8
TABLE OF TABLES	9
TABLE OF ABBREVIATIONS	10
ABSTRACT.....	11
CHAPTER – 1 INTRODUCTION	12
CHAPTER – 2 ORGANIZATIONAL OVERVIEW	14
CHAPTER – 3 OVERVIEW OF THE DATASETS	15
CHAPTER – 4 PRE-PROCESSING TECHNIQUES	24
CHAPTER – 5 DATA MINING TECHNIQUES	28
CHAPTER – 6 DATA MINING RESULTS	31
APPLICATION OF PREDICTIVE MODEL:.....	33
CONCLUSION.....	35
BIBLIOGRAPHY	36
TURNITIN REPORT	37

TABLE OF FIGURES

Figure 1:- Number of Deaths due to CVD's Every Year	12
Figure 2:- Sample of the Actual Data	15
Figure 3:- No. of Outliers in Each Numerical Variable	16
Figure 4:- Checking Target Variable Balance	16
Figure 5:- Correlation Matrix.....	16
Figure 6:- Summary Statistics for Numerical Variables	18
Figure 7:- Summary for Categorical Variables	19
Figure 8:- Histograms to Visualize Continuous Variables Distribution	20
Figure 9:- Bar Plots to Illustrate the Frequency of Categorical Variables	21
Figure 10:- Age Distribution for Presence and Absence of Diseases	22
Figure 11:- Gender Distribution for Presence and Absence of Diseases	23
Figure 12:- Check for Missing Values	24
Figure 13:- Target Variable Balance.....	24
Figure 14:- Identify and Remove Outliers	25
Figure 15:- Define Categorical Variables for One-Hot Encoding	27
Figure 16:-Perform One-Hot Encoding	27
Figure 17:- Feature Scaling.....	27
Figure 18:- Build the KNN Model on the Scaled Data.....	28
Figure 19:- Evaluation Results of Used Algorithms	31
Figure 20:- Confusion Matrix for Scaled KNN	31
Figure 21:- Confusion Matrix for Decision Tree	31
Figure 22:- Confusion Matrix for SVM.....	31
Figure 23:- Barplot of Performance Evaluation Metrics.....	32
Figure 24:- Turnitin Report.....	37

TABLE OF TABLES

Table 1:- Table of Abbreviations 10

TABLE OF ABBREVIATIONS

HD	Heart Disease
KNN	k-Nearest Neighbors
SVM	Support Vector Machine
DT	Decision Tree
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CVD's	Cardiovascular diseases

Table 1:- Table of Abbreviations

ABSTRACT

This case study revolves around implementing a predictive model for early detection of heart disease, addressing the pressing global issue of cardiovascular diseases. With a focus on enhancing cardiac care, the project utilizes machine learning and a dataset of patient information from various heart tests (heart.csv). The main challenge identified is the current healthcare industry's struggle with early detection and prevention of heart disease, emphasizing the need for a precise predictive model. The aim is to create a robust computer model that not only identifies individuals at higher risk but also provides personalized interventions. The project's significance lies in its response to the increasing prevalence of cardiovascular diseases, proposing a transformative tool to improve patient outcomes and alleviate strain on healthcare systems. The goals include developing an accurate predictive model, supporting medical practitioners in identifying high-risk individuals, identifying critical characteristics of heart disease, and enabling early treatment to enhance patient outcomes. Furthermore, the study employs K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree models, evaluating their performance using metrics such as accuracy, precision, recall, and F1-Score. Results indicate the superiority of SVM, with an 86% accuracy, 89% precision, 87% F1-Score, and 84% recall, showcasing its effectiveness in predicting heart disease. Confusion matrices provide detailed insights into true positive and true negative predictions, contributing to performance measurement and error analysis. The comprehensive approach to model development and evaluation ensures a nuanced understanding of each algorithm's predictive capabilities. This initiative reflects a commitment to advancing cardiac care through collaboration, advanced technology, and a comprehensive dataset.

Keywords: Predictive Modeling, Decision Tree, Support Vector Machine, k-Nearest Neighbors, Heart Disease, Model Development, Early Detection, Machine Learning, Evaluation Metrics, Confusion Matrix

CHAPTER – 1 INTRODUCTION

1. Project Background

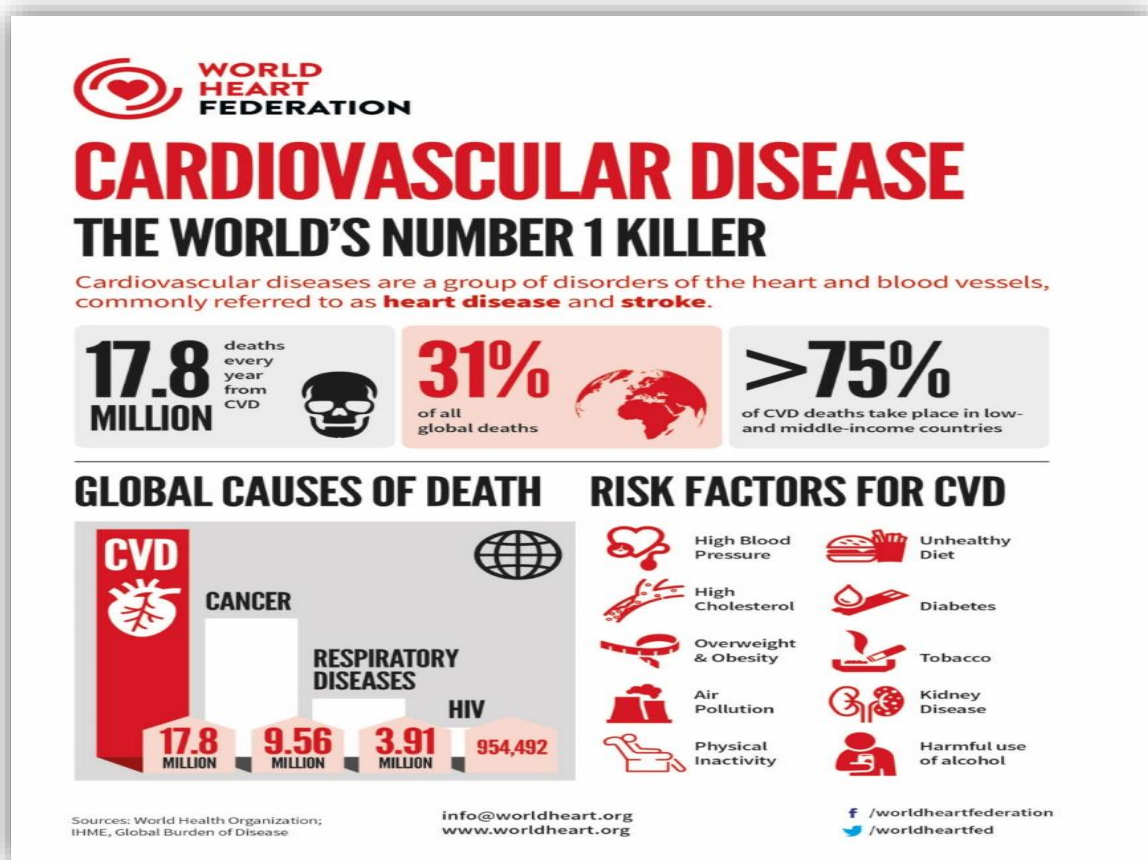


Figure 1:- Number of Deaths due to CVD's Every Year

Cardiovascular diseases (CVD's), particularly heart disease, stand as a prominent cause of illness and death globally. As shown in figure above or according to the WHO - around 17.8 million people die from cardiovascular diseases every year, which is about 31% of all global deaths. About 75% of these deaths occur in nations with lower to with lower to middle-income levels. The main reasons for 85% of cardiovascular disease-related deaths are heart attacks and strokes. Therefore, heart diseases become notable health issues, emphasizing the vital need for early detection to enhance patient outcomes.

To tackle the growing issue of heart disease, the healthcare sector, especially in cardiac care, is undergoing a transformative shift towards preventive measures. This change is happening because the traditional methods we use in healthcare often don't use all the information available. As heart disease becomes more common, there is a greater need for

advanced tools to help healthcare professionals or doctors find potential issues early on. In addition to that, this project comes from recognizing these challenges and a commitment to making heart care better. Using a predictive modeling through machine learning and a dataset of patient's information from various heart tests (heart.csv), the goal is to create a strong computer model. This model won't just find people at a higher risk of heart disease, it will also give ideas for personalized ways to help each person. The dataset, heart.csv, is really important for the project. It has a lot of different information that helps in creating a model to understand and deal with the complexities of heart disease. Furthermore, the project is not undertaken in isolation; it may involve collaboration with healthcare institutions, experts, or other stakeholders. Working together like this makes the project more trustworthy and shows that it could have a positive impact on patients and the healthcare system.

In summary, this project is a response to the urgent need for enhanced cardiac care in the face of rising cardiovascular diseases. By using advanced technology, working together, and a comprehensive dataset, the aim is to give doctors a powerful tool to find issues early and provide personalized help, ultimately making patients better and reducing the strain on healthcare systems.

2. Problem Statement

The healthcare industry faces a significant challenge in early detection and prevention of heart disease, a leading cause of mortality worldwide. Even with advanced medical tools, doctors struggle to predict who might develop heart issues because there are so many factors involved, like genes, lifestyle, and medical history. So, there is a critical need to develop an accurate predictive model that utilizes patient data to identify individuals who are at an elevated risk of heart disease.

In addition to that, we're dealing with a bunch of patient information collected during heart tests. By constructing a predictive model, healthcare providers aim to enhance cardiac care by enabling early detection of heart disease and facilitating targeted interventions. This means we can give people the right advice and treatment early on, hopefully preventing serious heart problems later. It's all about using technology to keep people healthier and save lives.

3. Objectives

- Develop a robust predictive model (using advanced techniques, such as machine learning algorithms) for early detection of heart disease.
- Providing medical practitioners a tool to help them identify those who are at high risk based on their medical characteristics.
- To determine the key characteristics of heart disease (HD) and research how to forecast HDs using different algorithms.
- Facilitate successful interventions and improve patient outcomes through early diagnosis.

CHAPTER – 2 ORGANIZATIONAL OVERVIEW

The organization undertaking this project is a healthcare provider committed to advancing cardiac care through innovative approaches and technology. Acknowledging the worldwide threat posed by cardiovascular diseases, especially heart disease, the company wants to revolutionize the healthcare industry by introducing an early detection prediction model. The project emphasizes a comprehensive and cooperative approach through engagement with stakeholders, specialists, and healthcare organizations. With the use of data mining techniques and a patient dataset from many cardiac tests, the organization aims to create a strong predictive model that can identify people at risk early on, for better treatment early and improve overall cardiac care. The project shows a dedication to improving patient outcomes, addressing the growing number of cardiovascular diseases, and developing health care in general. The organization's goal is to improve individual health and the entire healthcare system by utilizing innovative technology, teamwork, and a sophisticated comprehension of patient information.

CHAPTER – 3 OVERVIEW OF THE DATASETS

1. Dataset Issues

Data characteristics and Source:

- The dataset (heart.csv) contains information about patients who have undergone various tests regarding cardiovascular disease, and it is extracted from kaggle. It includes a mix of values, including categorical, binary, and continuous.
- **Characters:**
 - *Number of instances:* 1025
 - *Number of features:* 14
 - *Data Types:* Integer (represents whole numbers without fractional component), Numeric (represents both integers and decimal)
 - *Target Variable:* Presence or absence of heart disease - If the patient has heart disease, the target value is set to be 1, and if the patient doesn't have heart disease, the target value is set to be 0
- This Binary Classification helps machine learning models to analyze if the individual has risk of having heart disease or not based on their medical history record (i.e; Age, Sex, trestbps, chol, fbs, restecg, thalach,exang, slope, oldpear, ca, thal, target)
- Below screenshot is the basic information about dataset:

```
> # Display basic information about the dataset
> str(heart_data)
'data.frame': 1025 obs. of 14 variables:
 $ age      : int  52 53 70 61 62 58 58 55 46 54 ...
 $ sex      : int  1 1 1 1 0 0 1 1 1 1 ...
 $ cp       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ trestbps : int  125 140 145 148 138 100 114 160 120 122 ...
 $ chol     : int  212 203 174 203 294 248 318 289 249 286 ...
 $ fbs      : int  0 1 0 0 1 0 0 0 0 0 ...
 $ restecg  : int  1 0 1 1 1 0 2 0 0 0 ...
 $ thalach  : int  168 155 125 161 106 122 140 145 144 116 ...
 $ exang    : int  0 1 1 0 0 0 0 1 0 1 ...
 $ oldpeak  : num  1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...
 $ slope    : int  2 0 0 2 1 1 0 1 2 1 ...
 $ ca       : int  2 0 0 1 3 0 3 1 0 2 ...
 $ thal     : int  3 3 3 3 2 2 1 3 3 2 ...
 $ target   : int  0 0 0 0 0 1 0 0 0 0 ...
```

Figure 2:- Sample of the Actual Data

Dataset Issues:

- **Outliers:** These are the data points that differ drastically from the rest of the data points which may distort the outcome of analysis.

```
> print("Number of Outliers in Each Numerical Variable:")
[1] "Number of Outliers in Each Numerical Variable:"
> print(colSums(outliers))
      age trestbps      chol  thalach  oldpeak
      0       30       16       4       7
```

Figure 3:- No. of Outliers in Each Numerical Variable

For example:- in our dataset, outliers in numerical variables (age, trestbps, chol, thalach, oldpeak) were identified using boxplots and a quantile-based approach.

Also, later on, outliers were subsequently removed from the dataset too.

→ **Class Imbalance:** Class imbalance occurs when a dataset, particularly in binary classification, has a significant disparity in the number of instances between classes, with one having significantly fewer examples than the other.

```
> # Check the balance of the target variable
> target_balance <- table(heart_data$target)
> if (length(unique(heart_data$target)) > 1) {
+   print("Target Variable is Balanced:")
+   print(target_balance)
+ } else {
+   print("Target Variable is Not Balanced.")
+ }
[1] "Target Variable is Balanced:"
  0    1
499 526
```

Figure 4:- Checking Target Variable Balance

For example:- as shown in the figure no.4, or in our case, the output shows that the target variable is slightly imbalance (i.e. there are 499 observations in class 0 and 526 observations in class 1, means that the class imbalance is slightly in favour of class 1).

→ **Correlation Matrix:** The correlation matrix provides insights into the relationships between different numerical variables in the heart disease dataset. The values in the matrix range from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation.

```
> print("Correlation Matrix:")
[1] "Correlation Matrix:"
> print(correlation_matrix)
      age      trestbps      chol      thalach      oldpeak
age      1.0000000  0.27112141  0.21982253 -0.39022708  0.20813668
trestbps 0.2711214  1.00000000  0.12797743 -0.03926407  0.18743411
chol      0.2198225  0.12797743  1.00000000 -0.02177209  0.06488031
thalach  -0.3902271 -0.03926407 -0.02177209  1.00000000 -0.34979616
oldpeak   0.2081367  0.18743411  0.06488031 -0.34979616  1.00000000
```

Figure 5:- Correlation Matrix

Let's analyze the correlation matrix:

❖ **Age and Other Variables:**

Positive Correlation with Trestbps (Blood Pressure): There is a moderate positive correlation (0.27) between age and blood pressure (trestbps). This suggests that, on average, older individuals tend to have slightly higher blood pressure.

Positive Correlation with Cholesterol (Chol): Age also shows a mild positive correlation (0.22) with cholesterol levels. This implies that, as age increases, cholesterol levels may also show a slight increase.

Negative Correlation with Max Heart Rate (Thalach): There is a notable negative correlation (-0.39) between age and maximum heart rate (thalach). This suggests that younger individuals tend to have a higher maximum heart rate.

Positive Correlation with Oldpeak: Age has a positive correlation (0.21) with oldpeak, indicating a mild association between age and the extent of exercise-induced ST depression.

❖ **Blood Pressure (Trestbps) and Other Variables:**

Positive Correlation with Cholesterol (Chol): Blood pressure (trestbps) shows a mild positive correlation (0.13) with cholesterol levels. Individuals with higher blood pressure may also exhibit slightly elevated cholesterol levels.

Negative Correlation with Max Heart Rate (Thalach): There is a minimal negative correlation (-0.04) between blood pressure and maximum heart rate (thalach).

Positive Correlation with Oldpeak: Blood pressure exhibits a positive correlation (0.19) with oldpeak, suggesting a mild relationship between blood pressure and the extent of exercise-induced ST depression.

❖ **Cholesterol (Chol) and Other Variables:**

Weak Positive Correlation with Oldpeak: Cholesterol levels have a weak positive correlation (0.06) with oldpeak, indicating a subtle association

between cholesterol levels and the extent of exercise-induced ST depression.

❖ **Max Heart Rate (Thalach) and Oldpeak:**

Negative Correlation: There is a significant negative correlation (-0.35) between maximum heart rate (thalach) and the extent of exercise-induced ST depression (oldpeak). This implies that individuals with a higher maximum heart rate may experience less ST depression during exercise.

In summary, the correlation matrix provides valuable insights into the relationships between variables in the dataset. Understanding these correlations is crucial for feature selection, identifying potential multicollinearity, and gaining insights into the factors that may contribute to heart disease.

→ **Inconsistency Data Entry:** Uneven encoding of categorical variables, types, and formatting may bring errors and inconsistencies into the dataset.

For Example:- typeface mistakes in the variable “Sex”, with Male represented as “Male” and sometimes “M”.

→ **Data Quality:** For building accurate prediction models involves high data quality, encompassing relevance, completeness.

For Example:- addressing these quality issues such as outlier, inconsistency, class imbalance as well as scaling is crucial for maintaining accuracy of prediction model.

2. EDA

EDA is an essential step of the data mining process which includes data visualization and exploration of the dataset to gain insights about the data. In our case, “heart.csv”, dataset is used and the objective is to understand the characteristics of the dataset and also to find the potential relationships among variables.

➤ **Summary Statistics for Numerical Variables:**

```
> # Summary Statistics for Numerical Variables
> summary_stats_numerical <- summary(heart_data[, c("age", "trestbps", "chol", "thalach", "oldpeak")])
> print(summary_stats_numerical)
```

age	trestbps	chol	thalach	oldpeak
Min. :29.00	Min. : 94.0	Min. :126	Min. : 71.0	Min. :0.000
1st Qu.:48.00	1st Qu.:120.0	1st Qu.:211	1st Qu.:132.0	1st Qu.:0.000
Median :56.00	Median :130.0	Median :240	Median :152.0	Median :0.800
Mean :54.43	Mean :131.6	Mean :246	Mean :149.1	Mean :1.072
3rd Qu.:61.00	3rd Qu.:140.0	3rd Qu.:275	3rd Qu.:166.0	3rd Qu.:1.800
Max. :77.00	Max. :200.0	Max. :564	Max. :202.0	Max. :6.200

Figure 6:- Summary Statistics for Numerical Variables

The dataset consists of five numerical variables which includes “age”, “trestbps”, “chol”, “thalach” and “old peak”. The summary statics of these variables are presented which provides insights about central tendency including mean, median and spread of these variables from min to max.

➤ Summary for Categorical Variables:

```
[1] "Summary for sex :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	312	0		30.43902	
1	713	1		69.56098	

```
[1] "Summary for cp :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	497	0		48.487805	
1	167	1		16.292683	
2	284	2		27.707317	
3	77	3		7.512195	

```
[1] "Summary for fbs :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	872	0		85.07317	
1	153	1		14.92683	

```
[1] "Summary for restecg :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	497	0		48.487805	
1	513	1		50.048780	
2	15	2		1.463415	

```
[1] "Summary for exang :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	680	0		66.34146	
1	345	1		33.65854	

```
[1] "Count summary:"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	74	0		7.219512	
1	482	1		47.024390	
2	469	2		45.756098	

```
[1] "Summary for ca :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	578	0		56.390244	
1	226	1		22.048780	
2	134	2		13.073171	
3	69	3		6.731707	
4	18	4		1.756098	

```
[1] "Summary for thal :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	7	0		0.6829268	
1	64	1		6.2439024	
2	544	2		53.0731707	
3	410	3		40.0000000	

```
[1] "Summary for target :"
```

Category	Count	Percentage	Var1	Percentage	Freq
0	499	0		48.68293	
1	526	1		51.31707	

Figure 7:- Summary for Categorical Variables

The dataset consists of nine categorical variables such as sex, chest pain, fasting blood sugar etc. The count and unique values are displayed as a summary for a categorical variable including frequency percentage.

For our Exploratory Data Analysis (EDA), we'll conduct a comprehensive analysis in two main steps:

a. Univariate Analysis:

- ✚ We begin by examining each feature independently to understand its distribution and range.
- ✚ For continuous variables (age, trestbps, chol, thalach, oldpeak), we use histograms to visualize their distribution, which is shown figure number 7.

```
> # Univariate Analysis
> # Continuous Variables - Histograms
> continuous_columns <- c("age", "trestbps", "chol", "thalach", "oldpeak")
> par(mfrow=c(2, 3)) # Set up a 2x3 grid for subplots
> for (column_name in continuous_columns) {
+   hist(heart_data[[column_name]], main=column_name, xlab=column_name, col="skyblue", border="black")
+ }
```

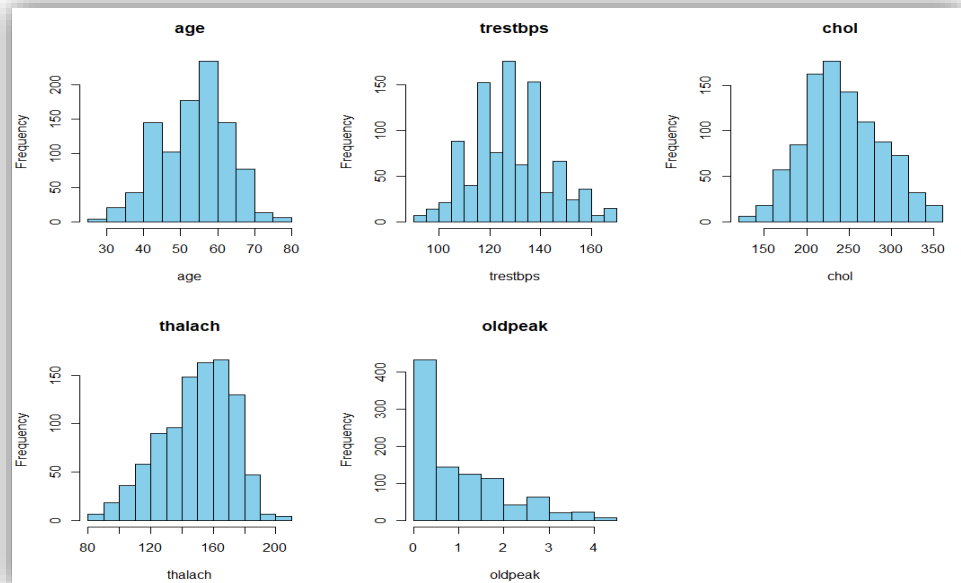


Figure 8:- Histograms to Visualize Continuous Variables Distribution

From the above histogram, it provides insights about the distribution frequency of various variables including age, thalach and so on. It can be clearly seen that the age group between 50-60 has high range of frequency or count in the observation and low of 30-40 and 70-80 age group people, likewise trestbps is high in 120 to 140 and low frequency in near to 100 and 160. Also the frequency is high in level 200 to 250 of cholesterol and less in below 200 and above 300. The frequency of observation is high of thalach in between 140 to 160 and is quite low in below 120 and in above 180. Also the frequency is high in oldpeak of group 0 to 1 and very less from above 1 to 4 comparatively.

✚ For categorical variables (sex, cp, fbs, restecg, exang, slope, ca, thal, target), we create bar plots to illustrate the frequency of each category, which is shown in figure number 8.

```
> # Categorical Variables - Bar Plots
> categorical_columns <- c("sex", "cp", "fbs", "restecg", "exang", "slope", "ca", "thal", "target")
> # Categorical Variables - Bar Plots
> par(mfrow=c(3, 3)) # Set up a 3x3 grid for subplots
> for (column_name in categorical_columns) {
+   counts <- table(heart_data[[column_name]])
+   # Mapping numeric codes to descriptive names
+   labels <- switch(
+     column_name,
+     cp = c("Typical angina", "Atypical angina", "Non-anginal pain", "Asymptomatic"),
+     sex = c("Male", "Female"),
+     restecg = c("Normal", "ST-T wave abnormality", "Probable/definite LV hypertrophy"),
+     slope = c("Upsloping", "Flat", "Downsloping"),
+     thal = c("Normal", "Fixed defect", "Reversible defect", "Not described"),
+     target = c("No disease", "Presence of disease"),
+     fbs = c("False", "True"),
+     exang = c("No", "Yes")
+   )
+   barplot(counts, main=column_name, col="lightblue", border="black", names.arg = labels)
+ }
```

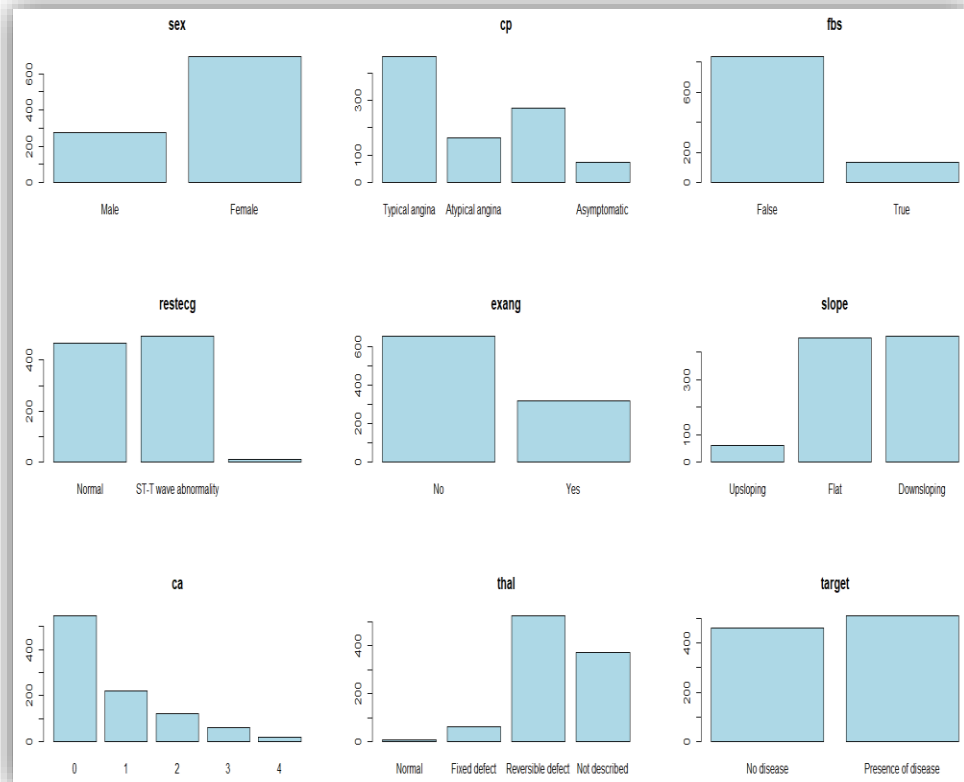


Figure 9:- Bar Plots to Illustrate the Frequency of Categorical Variables

When comparing the frequency of observation in between male and female or on the gender basis, it can be seen that frequency is very high in female compared to male it seems like double the frequency of male compared to female.

In comparison of restecg, it is divided into three groups Normal, ST-T wave abnormality and probable/definite LV hypertrophy. From the figure no. 8, it can be observed that the number compared to normal and ST-T wave abnormality is not very different, it's kind of similar but probable /definite LV hypertrophy group of restecg is very low in compared to other group.

In ca variable it is divided into five categories 0 to 4 and from the bar graph, it is visible that the frequency is high in group 0 and is decreasing frequency from 0 to 4. The frequency is in decreasing manner from groups 0 to 4.

In fps, the comparison is between fbs True observation and fps False observation. The bar graph shows that the frequency is high in fbs False observation than in True. The data shows that True observations is very low compared to False observation.

In comparison of slope, it is divided into three groups: upsloping, flat and downsloping. From the graph, it can be observed that the frequency of flat and downsloping group is quite similar where downsloping is little high than flat, but the group upsloping frequency is very low in comparison with flat and downsloping.

In comparison of the target, since it is binary either presence and absence of disease. The bar graph (figure no. 8) shows the frequency of presence of disease is little more higher than no disease.

b. Bivariate Analysis:

- ✚ In this step, we explore the relationship between two main features and the target variable (presence or absence of heart disease).
- ✚ We analyze the age distribution for both classes (presence and absence of disease) using side-by-side histograms to identify patterns, which is shown in figure number 9.

```
> # Bivariate Analysis
> # Age vs. Target
> # Separate age data for each target class
> age_no_disease <- heart_data$age[heart_data$target == 0]
> age_disease <- heart_data$age[heart_data$target == 1]
> # histograms for age by target class
> par(mfrow=c(1, 2)) # Set up a 1x2 grid for side-by-side plots
> hist(age_no_disease, main="Age Distribution (No Disease)", col="skyblue", border="black", xlab="Age")
> hist(age_disease, main="Age Distribution (Presence of Disease)", col="lightcoral", border="black", xlab="Age")
```

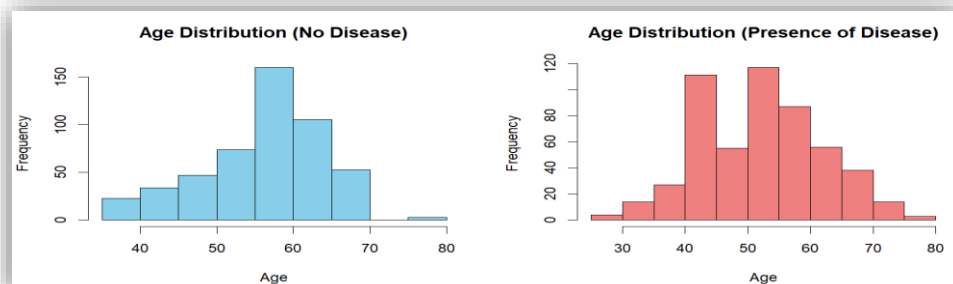


Figure 10:- Age Distribution for Presence and Absence of Diseases

For the purpose of getting insights and patterns of what age group individuals falls in presence of heart disease and absence of heart disease, two bar plot are made, and from it can be observed that the frequency of presence of heart disease is higher in age group of 40 to 45 and 50 to 55 also the age group of 55 to 60 has quite higher frequency of presence of heart disease and absence of heart disease is higher in age group of 55 to 60 and little higher in age group of 60 to 65. The low heart disease individual falls in group of 70 to 80 and 0 to 30 whereas the age group between 45 – 50 seems like 50:50 ratio of presence and absence of heart disease.

✚ We investigate the gender distribution for each class using bar plots to understand the distribution of heart disease among males and females, which is shown in figure number 10.

```
> # Sex vs. Target
> # Separate sex data for each target class
> sex_no_disease <- heart_data$sex[heart_data$target == 0]
> sex_disease <- heart_data$sex[heart_data$target == 1]
> # Create a 1x2 grid for side-by-side bar plots
> par(mfrow=c(1, 2))
> # Bar plot for the distribution of heart disease by gender
> barplot(table(sex_no_disease), main="No Disease by Gender",
+         col="skyblue", border="black", ylim=c(0, max(table(heart_data$sex))),
+         xlab="Gender", ylab="Count")
> barplot(table(sex_disease), main="Presence of Disease by Gender",
+         col="lightcoral", border="black", ylim=c(0, max(table(heart_data$sex))),
+         xlab="Gender", ylab="Count")
```

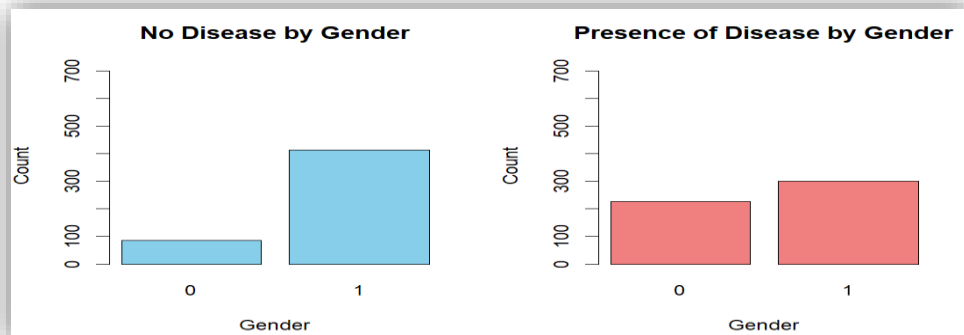


Figure 11:- Gender Distribution for Presence and Absence of Diseases

The figure no. 10 is a comparison of presence and absence of heart disease on the basis of gender where 0 is male group and 1 is female group. The first graph shows that the frequency of absence of heart disease is very low in male while comparing females. Also the second graph which is a bar graph of presence of heart disease it shows the number of presence of disease is also higher but not more than of female, the frequency of presence of heart disease is high in female.

These above mentioned analyses help us gain insights into the individual characteristics of the data and provide a deeper understanding of how each feature may influence our main goal: predicting the presence or absence of heart disease.

CHAPTER – 4 PRE-PROCESSING TECHNIQUES

a) Missing Values:

```
> # Check for missing values in the dataset
> missing_values <- colSums(is.na(heart_data))
> print("Missing Values Summary:")
[1] "Missing Values Summary:"
> print(missing_values)
  age    sex    cp    trestbps    chol    fbs    restecg    thalach    exang    oldpeak
  0      0      0      0          0      0      0          0          0          0
  slope    ca    thal    target
  0      0      0      0
```

Figure 12:- Check for Missing Values

The dataset was checked for missing values using the `colSums(is.na())` function. Fortunately, no missing values were found. This is crucial as missing values can disrupt model training. Imputation techniques or data removal are commonly employed to address missing values. In our dataset, the lack of missing values ensures the dataset's integrity by removing the need for imputation.

However, methods like mean, median or KNN imputations might have been used if there had been missing values. These techniques ensure that the integrity of the datasets is maintained by replacing missing values with plausible estimates derived from the available data. Imputation keeps important data from being lost and ensures that models can be trained on whole datasets, which improves prediction accuracy.

b) Target Variable Balance:

```
> # Check the balance of the target variable
> target_balance <- table(heart_data$target)
> if (length(unique(heart_data$target)) > 1) {
+   print("Target Variable is Balanced:")
+   print(target_balance)
+ } else {
+   print("Target Variable is Not Balanced.")
+ }
[1] "Target Variable is Balanced:"
  0      1
499 526
```

Figure 13:- Target Variable Balance

The balance of the target variable was assessed using the `table` function. A balanced target variable is crucial for robust model training.

It appears that the target variable is relatively balanced. The values represent the counts of each class in the target variable. In this case:

→ **Class 0 (No Disease):** 499 instances

→ **Class 1 (Presence of Disease):** 526 instances

The difference in counts between the two classes is not substantial, indicating a reasonably balanced distribution. So, therefore, a balanced target variable is beneficial for building robust machine learning models, as it helps prevent bias towards the majority class. In this context, the dataset seems well-suited for training classification models to predict the presence or absence of heart disease.

c) **Outlier Handling:**

```
> # Boxplots to identify outliers in numerical variables
> par(mfrow=c(2, 3)) # Set up a 2x3 grid for subplots
> for (column_name in continuous_columns) {
+   boxplot(heart_data[[column_name]], main=paste("Boxplot of", column_name), col="skyblue", border="black")
+ }
> # Alternatively, one can use quantile-based outlier detection
> outlier_detection <- function(column) {
+   Q1 <- quantile(heart_data[[column]], 0.25)
+   Q3 <- quantile(heart_data[[column]], 0.75)
+   IQR <- Q3 - Q1
+   outliers <- heart_data[[column]] < (Q1 - 1.5 * IQR) | heart_data[[column]] > (Q3 + 1.5 * IQR)
+   return(outliers)
+ }
> # Identify outliers in numerical variables
> outliers <- sapply(continuous_columns, outlier_detection)
> print("Number of Outliers in Each Numerical Variable:")
[[1] "Number of Outliers in Each Numerical Variable:"
> print(colSums(outliers))
   age trestbps   chol  thalach  oldpeak
    0       30     16       4        7
> # Identify outliers using quantile-based approach
> outliers <- sapply(continuous_columns, outlier_detection)
> # Remove outliers from the dataset
> heart_data_no_outliers <- heart_data[!apply(outliers, 1, any), ]
> # Display information before and after removing outliers
> cat("Original dataset size:", nrow(heart_data), "rows\n")
Original dataset size: 1025 rows
> cat("Dataset size after removing outliers:", nrow(heart_data_no_outliers), "rows\n")
Dataset size after removing outliers: 968 rows
```

Figure 14:- Identify and Remove Outliers

Outliers can significantly affect the performance of machine learning models by skewing the distribution and introducing noise. In this code, outliers are detected using both boxplots and a quantile-based approach. Outliers are removed because they can distort statistical analyses and model training.

The chosen technique, which involves removing data points lying outside a certain range from the first and third quartiles, helps in creating a more robust and accurate model by ensuring that extreme values do not unduly influence the results. Eliminating outliers guarantees that extreme values do not unreasonably affect the outcomes, leading to more precise and dependable forecasts. Moreover, outlier reduction contributes to the development of a more robust model by lessening the influence of noise and enhancing the results' interpretability.

d) **One-Hot Encoding:**

Categorical variables, including cp (chest pain type), restecg (resting electrocardiographic results), and thal (thalassemia), were subjected to one-hot encoding to facilitate their

incorporation into machine learning models. One-hot encoding transforms categorical variables into binary vectors, allowing models to interpret and utilize these variables effectively.

- **Process:**

- Decision for one-hot encoding:

- **Nominal Variables:** These lack inherent order and should be one-hot encoded to prevent unintentional ordinal relationships.

- **Ordinal Variables:** These possess a natural order and might not require one-hot encoding due to the meaningful information conveyed by their order.

- Based on the criteria:

"sex": Being binary, with categories "male" and "female," it does not require one-hot encoding.

"cp" (Chest pain type): Considered nominal due to the absence of a clear ordinal relationship among chest pain types; hence, it should be one-hot encoded.

"fbs": Being binary (true or false), no one-hot encoding is needed.

"restecg" (Resting electrocardiographic results): Since the results lack an apparent ordinal relationship, it should be one-hot encoded.

"exang": As a binary variable (yes or no), it does not need one-hot encoding.

"slope": Describing the slope with categories (Upsloping, Flat, Downsloping), it seems ordinal and thus doesn't require one-hot encoding.

"ca" (Number of major vessels): Having an inherent ordinal relationship as it represents a count, it doesn't need one-hot encoding.

"thal" (Result of thallium stress test): With different states like "Normal," "Fixed defect," and "Reversible defect," it suggests a nominal nature and should be one-hot encoded.

- In summary:

- Need One-Hot Encoding: "cp," "restecg," "thal"

- Don't Need One-Hot Encoding: "sex," "fbs," "exang," "slope," "ca"

- **Define Categorical Variables for One-Hot Encoding:**

The categorical variables selected for one-hot encoding were cp, restecg, and thal.

```
# Define the categorical variables for one-hot encoding
nominal_variables <- c("cp", "restecg", "thal")
```

Figure 15:- Define Categorical Variables for One-Hot Encoding

- **Perform One-Hot Encoding:**

```
# Perform one-hot encoding for nominal variables
heart_data <- heart_data %>%
  mutate(across(all_of(nominal_variables), as.factor)) %>%
  mutate(across(all_of(nominal_variables), ~factor(.)))
```

Figure 16:-Perform One-Hot Encoding

The ‘mutate’ function from the ‘dplyr’ package was used to convert the specified categorical variables into factors and subsequently apply one-hot encoding.

- **Result:**

The output of “str(heart_data)” after one-hot encoding reflects the creation of binary columns for each category within the selected variables. This transformation prepares the dataset for model training, ensuring that categorical information is appropriately represented and utilized by machine learning algorithms.

- **Importance:**

One-hot encoding is a crucial preprocessing step when dealing with categorical variables in machine learning. It allows models to interpret categorical information without imposing ordinal relationships between categories. The resulting binary vectors enhance the ability of models to capture the impact of different categories on the target variable.

The application of one-hot encoding ensures that the categorical variables are appropriately prepared for subsequent analyses and model training, contributing to the overall reliability and effectiveness of the predictive models.

e) Feature Scaling:

```
# Define predictor variables and target variable
predictors <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal")
target <- "target"

# Feature scaling using the preProcess function from caret
scaling_model <- preProcess(train_data[, predictors], method = c("center", "scale"))

# Apply the scaling transformation to both the training and test sets
train_data_scaled <- predict(scaling_model, train_data[, predictors])
test_data_scaled <- predict(scaling_model, test_data[, predictors])
```

Figure 17:- Feature Scaling

Feature scaling, a crucial preprocessing step for algorithms sensitive to feature magnitudes (e.g., SVM, KNN, and certain linear models relying on distances or gradients), ensures equitable contribution from all features, preventing bias towards those with larger magnitudes.

Despite its importance, not all algorithms necessitate scaled data. Decision Tree-based models, notably scale-invariant, highlight the need for a nuanced approach. Acknowledging the varied requirements of different models, a strategic decision has been made to postpone feature scaling until later stages, integrating it into pipelines where necessary.

This approach prioritizes flexibility and efficiency in the modeling process. Feature scaling is now selectively applied to models benefiting from it, avoiding unnecessary transformations for algorithms naturally accommodating varying feature scales. This tailored strategy ensures optimized preprocessing for each model, contributing to the overall performance enhancement of the machine learning pipeline.

CHAPTER – 5 DATA MINING TECHNIQUES

a) Develop 3 Predictive Models:

To develop a predictive model for early detection of heart diseases, we select three different machine learning algorithms – KNN, DT, and SVM.

KNN, Decision Tree Model, and SVM:

```
# Feature scaling using the preProcess function from caret
scaling_model <- preProcess(train_data[, predictors], method = c("center", "scale"))

# Apply the scaling transformation to both the training and test sets
train_data_scaled <- predict(scaling_model, train_data[, predictors])
test_data_scaled <- predict(scaling_model, test_data[, predictors])

# Build the KNN model on the scaled data
knn_model_scaled <- knn(
  train = train_data_scaled,
  test = test_data_scaled,
  cl = train_data[, target], # Use the original target variable
  k = 5
)

# Build the decision tree model
tree_model <- rpart(target ~ ., data = train_data, method = "class")

# Build the SVM model for classification
svm_model <- svm(
  formula = as.formula(paste(target, "~", paste(predictors, collapse = "+"))),
  data = train_data,
  type = "C-classification", # Change to C-classification
  kernel = "linear",
  cost = 1
)
```

Figure 18:- Build the KNN Model on the Scaled Data

Here, we develop three distinct machine learning models: k-Nearest Neighbors (KNN), Decision Tree, and Support Vector Machine (SVM) for a classification task on a heart

disease dataset. Each model aims to predict the presence or absence of heart disease based on several features.

In the KNN model building process, the training data is scaled using `preProcess` function, and the KNN model is built on the scaled data using `knn` function. This algorithm classifies a data point by considering the class labels of its k -nearest neighbors. The k parameter is set to 5, indicating that the model will consider the majority class among the five closest neighbors for classification.

Likewise, the Decision Tree model, implemented using the `rpart` function, is a tree-like structure where each internal node represents a decision based on a particular feature, leading to a final classification at the leaf nodes.

And lastly, the SVM model, specified with the `svm` function, employs a linear kernel and is configured for C-classification. The C parameter controls the trade-off between achieving a smooth decision boundary and classifying training points correctly.

The advantage of using this code lies in its comprehensive approach to building diverse classification models. KNN, Decision Trees, and SVM are distinct algorithms with different strengths and weaknesses. By employing all three, the code allows for model comparison and selection based on the specific characteristics of the dataset. Additionally, the inclusion of feature scaling enhances model performance, especially for KNN and SVM, contributing to more robust and accurate predictions.

b) Justification on the Predictive Model Chosen:

- ✓ **k - Nearest Neighbors (KNN):-** We chose the KNN model as it offers a simple yet effective approach to classification, well-suited for our heart disease prediction task. KNN operates on the principle that instances with similar feature values tend to belong to the same class. In the context of heart disease, where patterns may be complicated and non-linear, KNN's local learning nature allows it to adapt to the underlying complexity of the dataset. The model is especially beneficial when relationships between features are not easily characterized by a global model. Moreover, KNN is relatively easy to implement and understand, providing a good starting point for exploratory analysis and initial insights into the predictive patterns within the heart disease dataset.

- ✓ **Decision Tree (DT):-** The Decision Tree model was a natural choice for heart disease prediction due to its interpretability and ability to uncover hierarchical decision rules. DT's recursively split the feature space based on the most informative attributes, creating a tree structure that mirrors the decision-making process. In the field of heart disease prediction, where various risk factors may interact in complex ways, DT's excel at identifying critical combinations of features leading to disease outcomes. The interpretability of DT's is particularly crucial in healthcare settings, allowing practitioners to understand and communicate the factors contributing to predictions.

Additionally, DT's can handle non-linear relationships, providing a more nuanced understanding of the predictors influencing heart disease outcomes. They are capable of handling both numerical and categorical data and are interpretable, making them useful for understanding the decision-making process. Overall, the DT model aligns well with the need for transparent and interpretable predictive modeling in the context of heart disease prediction.

- ✓ **Support Vector Machine (SVM):-** We employed Support Vector Machine (SVM) as it is a powerful model known for its effectiveness in handling both linear and non-linear classification problems. SVM aims to find the optimal hyperplane that best separates data points of different classes while maximizing the margin.

In the context of heart disease prediction, where the decision boundary might be complex, SVM's ability to capture intricate relationships between features is valuable. The model is particularly useful when dealing with high-dimensional data, as is often the case in healthcare datasets. Furthermore, SVM allows for customization through different kernel functions, enabling the model to adapt to the specific characteristics of the data. While SVM may be less interpretable than some other models, its robustness and ability to handle complex relationships make it a compelling choice for heart disease prediction, especially when seeking a balance between accuracy and generalization.

CHAPTER – 6 DATA MINING RESULTS

The dataset (heat.csv) used in this study consists of 1025 data with 14 attributes in which various data preprocessing techniques are used for cleaning and dataset is splitted in 80:20 ratio of training and testing. The algorithms used in this study were KNN (K-Nearest Neighbors), SVM (Support Vector Method), Decision Tree and for evaluation of these algorithm, various evaluation metrics such as Accuracy, F1-Score, Recall and Precision are used.

```
> print("Model Comparison:")
[1] "Model Comparison:"
> print(model_metrics)
      Model Accuracy Precision Recall F1_Score
1   KNN Scaled 0.8247423 0.8061224 0.8404255 0.8229167
2 Decision Tree 0.8505155 0.7755102 0.9156627 0.8397790
3      SVM 0.8659794 0.8979592 0.8461538 0.8712871
```

Figure 19:- Evaluation Results of Used Algorithms

As shown in the figure no.17, where the evaluation result of predicting the presence and absence of heart disease with various used algorithm is displayed in which SVM (Support Vector Method)

```
> print("Confusion Matrix for Scaled KNN:")
[1] "Confusion Matrix for Scaled KNN:"
> print(conf_matrix_knn_scaled)

knn_model_scaled  0   1
                  0  81  19
                  1  15  79
```

Figure 20:- Confusion Matrix for Scaled KNN

```
> print("Confusion Matrix for Decision Tree:")
[1] "Confusion Matrix for Decision Tree:"
> print(conf_matrix_decision_tree)

predictions  0   1
              0  89  22
              1   7  76
```

Figure 21:- Confusion Matrix for Decision Tree

```
> print("Confusion Matrix for SVM:")
[1] "Confusion Matrix for SVM:"
> print(conf_matrix_svm)

svm_predictions  0   1
                 0  80  10
                 1  16  88
```

Figure 22:- Confusion Matrix for SVM

As shown in the figure no.s 18, 19, and 20, for the purpose of performance measurement and error analysis, confusion matrix which (i.e. performance measurement tool for classification model) is used. It shows True Negative (TN), False Positive (FP), False Negative (FN), True Positive (TP) values in a tabular form. Higher the number of True Positive (TP) and True Negative (TN), it shows the number of correct prediction by that specific model. Here also “Support Vector Machine” has the highest number of TP and TN which are 80 and 88 respectively.

After building three different models and evaluating them, we compare their performance using the metrics used in this study which includes Accuracy, precision, Recall and F1-Score comparison and is visualized by making a barplot for each metric as shown in figure no. 21.

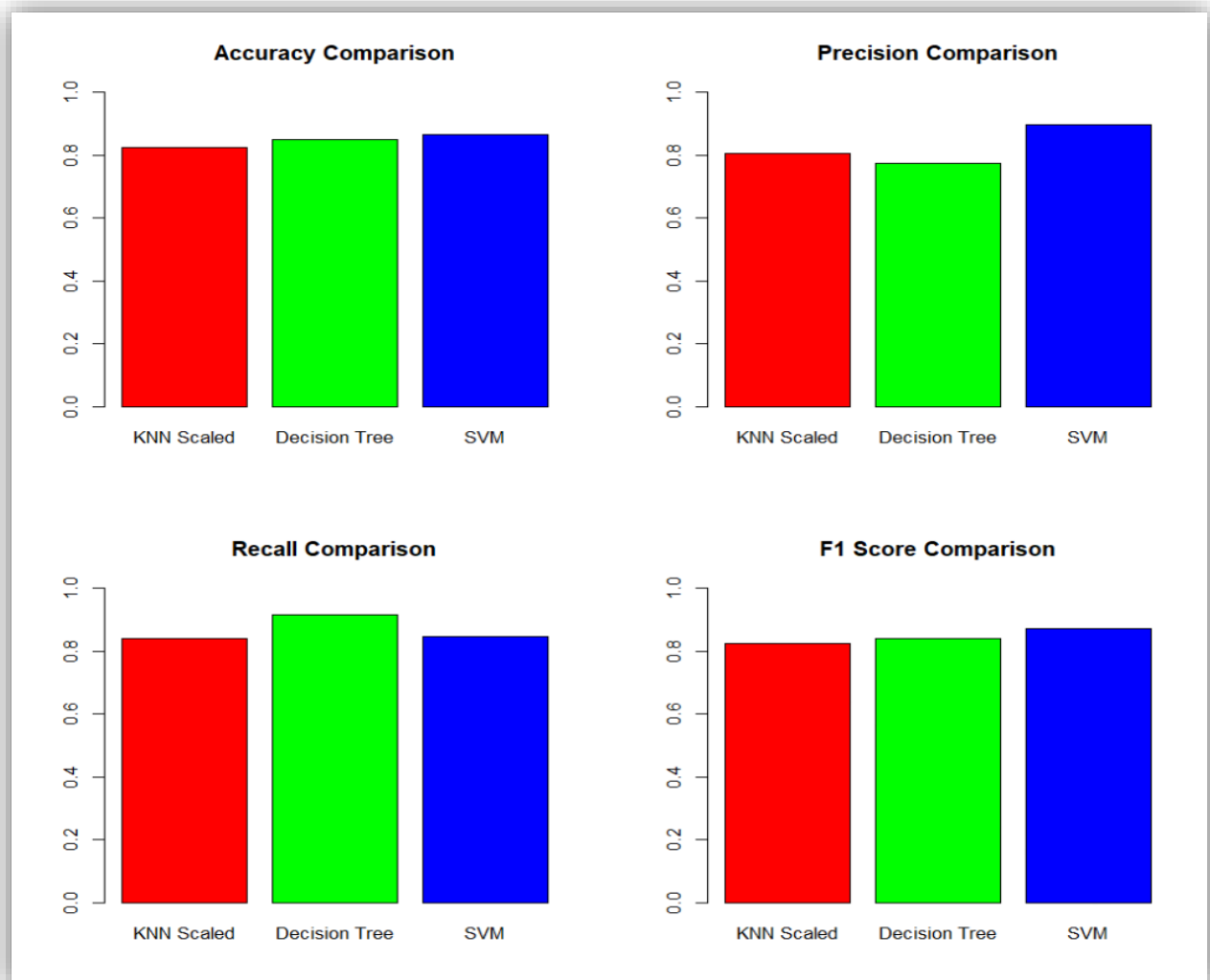


Figure 23:- Barplot of Performance Evaluation Metrics

APPLICATION OF PREDICTIVE MODEL:

→ Dummy Data:

```
> # Display the dummy data
> print(dummy_data)
  age sex cp trestbps chol fbs restecg thalach exang  oldpeak slope ca thal target
1  60  1  3    180  286  1      1    155    0 4.2891386    2  4    0      0
2  44  1  1    158  248  0      0     94    0 0.2291558    1  3    2      0
3  80  1  1    180  221  1      2    111    0 2.2110004    0  4    0      0
4  43  0  0    146  175  0      2    185    1 3.9946242    1  1    1      1
5  32  0  1    181  156  1      0    188    0 0.6094963    0  0    0      1
```

→ Result:

```
> # Display the results
> cat("Confusion Matrix for SVM on Dummy Data:")
Confusion Matrix for SVM on Dummy Data:
> print(conf_matrix_svm_dummy)

svm_predictions_dummy 0 1
                      0 2 0
                      1 1 2
> cat("Accuracy for SVM on Dummy Data:", round(accuracy_svm_dummy, 3))
Accuracy for SVM on Dummy Data: 0.8
> cat("Precision for SVM on Dummy Data:", round(precision_svm_dummy, 3))
Precision for SVM on Dummy Data: 1
> cat("Recall for SVM on Dummy Data:", round(recall_svm_dummy, 3))
Recall for SVM on Dummy Data: 0.667
> cat("F1 Score for SVM on Dummy Data:", round(f1_score_svm_dummy, 3))
F1 Score for SVM on Dummy Data: 0.8
```

→ Model Performance Metrics on New Data:

- i. **Accuracy:-** The SVM model achieved an accuracy of 80%, indicating the proportion of correctly classified instances.
- ii. **Precision:-** Precision is the ratio of correctly predicted positive observations to the total predicted positives. In this case, precision is 100%, suggesting that when the model predicts the positive class, it is correct.
- iii. **Recall:-** Recall measures the ratio of correctly predicted positive observations to the total actual positives. The recall for the SVM model on dummy data is 66.7%, indicating that it captures two-thirds of the actual positive instances.
- iv. **F1 Score:-** The F1 score is the harmonic mean of precision and recall. The SVM model achieved an F1 score of 80%, providing a balanced measure of precision and recall.

→ Conclusion:

Our decision to employ the SVM model for this classification task is validated by its performance on the provided dummy data. The model demonstrates high precision, showcasing its reliability in predicting the positive class. While the recall is not perfect, it captures a substantial portion of actual positive instances.

To further enhance model robustness and generalization, we recommend additional evaluation on more diverse and realistic datasets. Exploring hyper-parameter tuning and cross-validation can optimize the SVM model's performance, ensuring its effectiveness in real-world scenarios.

CONCLUSION

To sum up, the primary objective of this study was to provide a suitable model for the presence and absence of heart disease. Here three models: K-NN, Decision tree, SVM (Support Vector Method) was selected and the preprocessed dataset was provided to each models and is compared using various evaluation metrics including Accuracy, Recall, F1-Score, Precision and also confusion matrix is also displayed. After the result analysis SVM (Support Vector Methods) outcores with highest accuracy, recall, F1 score and also has higher number of True positive (TP), True Negative (TN) and from it can be concluded that “Support Vector Method” is suitable for the dataset and for the prediction of absence and presence of heart disease and also that it can surely enhance cardiac care and assist healthcare professional in identifying individual at higher risk of heart disease.

BIBLIOGRAPHY

Training, F. S. C. and F. A. (2021, May 11). *Cardiovascular Diseases - the number 1 cause of death globally*. First Support CPR and First Aid Training.

<https://firstsupportcpr.com/2021/05/11/cardiovascular-diseases-cause-most-death/>

TURNITIN REPORT

groupA_predictiveModelingOfHeartDiseases

ORIGINALITY REPORT

13%

SIMILARITY INDEX

10%

INTERNET SOURCES

3%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Taylor's Education Group

Student Paper

3%

2

Submitted to Sheffield Hallam University

Student Paper

1%

3

www.mdpi.com

Internet Source

1%

4

Submitted to University of Strathclyde

Student Paper

1%

5

Submitted to Southern New Hampshire
University - Continuing Education

Student Paper

1%

Figure 24:- Turnitin Report