

HEART DISEASE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and
Engineering by

Nayana Vikraman
(38110358)
Srija Saha (38110553)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**
Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI - 600 119**

MARCH – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Nayana Vikraman (38110358)** and **Srija Saha (38110553)** who carried out the project entitled "**Heart disease prediction using machine learning**" under my supervision from November 2021 to March 2022.

Internal Guide

Ms.C.M.Suja

Head of the Department

Dr. S. Vigneshwari M.E., Ph.D., and Dr. L. Lakshmanan M.E., Ph.D.,

Submitted for Viva-voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Nayana Vikraman(Reg No:38110358)** and **Srija Saha(Reg No: 38110553)** hereby declare that the Project Report entitled “**Heart disease prediction using machine learning**” done by us under the guidance of **Ms.C.M.Suja.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Ms.C.M.Suja**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Over the years we have seen that the spread of disease is a global problem. Accurate and timely analysis of any health-related problems is essential to treat and treat them early but with this growing number it becomes more difficult. Machine learning strategies have been accelerating the field of health with more research and methodologies.

Machine learning strategies can contribute significantly to the process of predicting and diagnosing diseases early, becoming a field of research and proving to be a viable option in the health sector.

The research paper focuses on the prediction of diseases caused in Heart, predicting the likelihood of a person having a specific disease that can lead to heart attack based on various medical factors and related parameters.

We set up 4 different predictive systems using data set to predict whether a patient might be diagnosed or not. This diagnosis is a difficult task because it must be done accurately and effectively. We undertake various machine learning classifications such as SVM and all to predict and differentiate an infected patient and used a temperature map to analyze the disease.

The most useful method was used to control how the model could be used to improve the accuracy of the disease prediction in any individual. The power of the proposed model is relatively satisfactory and able to predict disease and show good accuracy compared to the previously used category such as non-abrasive bays etc. So, the silent amount of pressure has been removed using the given model in identifying opportunities for the divider to be able to diagnose diseases correctly and efficiently.

Keywords: Customer segmentation, Clustering, K-means clustering, Mini Batch Kmeans clustering, hierarchical clustering

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
		v
	ABSTRACT	
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1. OVERVIEW	2
	1.2. MACHINE LEARNING	3
	1.2.1 HISTORY & RELATIONSHIPS	
	1.3 MACHINE LEARNING STRATEGIES	4
	1.3.1. SUPERVISED LEARNING	5
	1.3.2. UNSUPERVISED LEARNING	6
2	LITERATURE SURVEY	11
	2.1. RELATED WORK	11
3	METHODOLOGY	14
	3.1. EXISTING SYSTEM	14
	3.2. PROPOSED SYSTEM	14
	3.3. OBJECTIVE	14
	3.4. SOFTWARE AND HARDWARE REQUIREMENTS	15
	3.4.1. SOFTWARE REQUIREMENTS	15
	3.4.2. HARDWARE REQUIREMENTS	15
	3.4.3. LIBRARIES	15
	3.5. PROGRAMMING LANGUAGES	16
	3.5.1 PYTHON	16
	3.5.2. DOMAIN	18

3.6. SYSTEM ARCHITECTURE	21
3.7. ALGORITHMS USED	23
3.7.1. LOGISTIC REGRESSION	23
3.7.2. K NEAREST NEIGHBOR	23
3.7.3. SUPPORT VECTOR MACHINE	23
3.7.4. RANDOM FOREST	24
3.8. MODULES	25
3.8.1. DATASET COLLECTION	25
3.8.2. TRAIN AND TEST THE MODELS	25
3.8.3. HYPERPARAMETER TUNNING	25
3.9. WEB APPLICATION	26
RESULT AND DISCUSSION	29
4.1. PERFORMANCE ANALYSIS	29
CONCLUSION	30
5.1. CONCLUSION	30
REFERENCES	31
APPENDICES	31
A. SOURCE CODE	32
B. SCREENSHOTS	38

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1.	MACHINE LEARNING CLASSIFICATION	7
3.1.	WORKFLOW DIAGRAM	21
4.1.	ACCURACIES OBTAINED	29
B.1.	DATASET	38
B.2.	CORRELATION MATRIX	38
B.3.	HISTOGRAM	39
B.4.	BOX PLOT	40
B.5.	SEABORN VISUALIZATION	41
B.6.	BAR PLOT	42
B.7.	GRAPH PLOT	44
B.8.	LR CLASSIFICATION REPORT	46
B.9.	KNN CLASSIFICATION REPORT	46
B.10.	SVM CLASSIFICATION REPORT	46
B.11.	RF CLASSIFICATION REPORT	47
B.12.	ACCURACY TABLE	47
B.13.	WEB APPLICATION OF PREDICTION	48
C.	RESEARCH PAPER PUBLICATION	49
D.	PLAGARISM REPORT	51

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
ML	Machine Learning
AI	Artificial Intelligence
RF	Random Forest
UI	User Interface
SVM	Support Vector Machine
KNN	K Nearest neighbor
LR	Logistic Regression

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION OF THE PROJECT

Health care data is generally large in volume and complexity in structure. Records of a large set of medical data created by medical professionals are available to analyze and extract important information from it. Machine learning strategies are able to manage large contents of data and helps to obtain needful information.

This type of Machine learning framework can encourage physicians to take immediate action so that more patients can receive medication in a shorter period of time, thus saving a significant number of lives.

This paper covers the predictions of different types of diseases caused in Heart that causes death using different ML strategies. Heart disease, also known as cardiovascular disease, encompasses a wide range of cardiovascular conditions and has been a major cause of death worldwide for the past few decades. It encompasses a wide range of risk factors as well as the need for time to find accurate, reliable, and logical ways to make an early diagnosis.

Heart is a vital organ of the human frame. It pumps blood to every part of our anatomy. If it fails to feature efficiently, then the brain and diverse different organs will stop operating, and inside few minutes, the man or woman will die. exchange in lifestyle, paintings related pressure and awful food behavior contribute to the growth in the price of numerous heart-associated sicknesses. coronary heart diseases have emerged as one of the most distinguished reasons of dying all around the global. in step with global fitness employer, heart related sicknesses are responsible for taking 17.7 million lives every year, 31% of all international deaths. In India too, coronary heart-associated diseases have emerged as the main reason of mortality coronary heart sicknesses have killed 1.7 million Indians in 2016, in accordance to the 2016 global Burden of ailment file, released on September 15,2017. Heart-associated diseases increase the spending on fitness care and also lessen the productiveness of a man or woman. Estimates made by the sector fitness employer (WHO), advocate that India has lost up to \$237 billion,

from 2005-2015, because of heart-related or cardiovascular diseases. accordingly, possible and accurate prediction of heart-related sicknesses could be very vital. medical organizations, all over the international, gather facts on various fitness-associated troubles. This information may be exploited using numerous machine mastering strategies to benefit beneficial insights. but the data accrued is very huge and, often, these statistics may be very noisy. These datasets, which might be too overwhelming for human minds to realize, may be easily explored using various device getting to know techniques. hence, those algorithms have end up very useful, in recent instances, to expect the presence or absence of coronary heart-related diseases as it should be the usage of records technology inside the fitness care enterprise is increasing day by day to resource docs in choice- making activities. It facilitates docs and physicians in ailment management, medicines, and discovery of styles and relationships amongst analysis information. cutting-edge procedures to predict cardiovascular hazard fail to discover many folks that would advantage from preventive treatment, while others acquire pointless intervention. system-gaining knowledge of offers an opportunity to improve accuracy by way of exploiting complicated interactions among risk elements. We assessed whether system-getting to know can enhance cardiovascular danger prediction. Using a variety of techniques, it was found that the SVM technique provided the best accuracy among others.

The main purpose of this work is to determine whether patients are diagnosed with the disease or not by using not only one algorithm in prior. Using database for finding the disease and using different algorithms to have a better understanding about the algorithms and analysing the data for the most effective result. It did provide excellent accuracies for each disease.

1.2 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide

variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Despite the fact that the reasons mentioned are valid, we have added a dimension in the last decade where data is being utilized for predicting what could potentially happen in the future. Then comes Machine Learning which play a significant role in doing so. Machine learning is a subset/subfield of Artificial Intelligence.

Generally, the main aim of Machine learning is to understand the structure of data and apply the best possible models that can be utilized or identify a hidden pattern. Developing a machine learning model is one of the key factors in predicting a future problem which again requires machine learning algorithms. There are numerous machine learning algorithms that have been developed and mature enough to solve various real-world business problems.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

Using Machine learning, information is being turned into knowledge. In the last 5-6 decades, enormous data has been recorded or collected which will be of no use if we don't utilize or analyze to find hidden patterns. In order to find useful and significant patterns with complex data, we have several Machine Learning techniques available to ease our struggle for discovery. Subsequently, those identified hidden patterns and knowledge of the problem can be helpful to perform complex decision making and predict future occurrence.

1.2.1 History and relationships to other fields

The term machine learning was coined in 1959 by Arthur Samuel, an

American IBMer and pioneer in the field of computer gaming and artificial intelligence. Also, the synonym self-teaching computers was used in this time period. A representative book of machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. Whereas, a machine learning algorithm for stock trading may inform the trader of future potential predictions.

1.3 MACHINE LEARNING APPROACHES

In machine learning, tasks square measure is typically classified into broad classes. These classes square measure supported however learning is received or however, feedback on the education is given to the system developed. Two of the foremost wide adopted machine learning strategies are square measure supervised learning that trains algorithms supported example input and output information that's tagged by humans, and unattended learning that provides the algorithmic program with no tagged information to permit it to search out structure at intervals its computer file.

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

Supervised learning: The computer is presented with example inputs and their

desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

1.3.1 Supervised Learning

In supervised learning, the pc is given example inputs that square measure labeled with their desired outputs. The aim of this technique is for the algorithmic program to be ready to "learn" by comparing its actual output with the "taught" outputs to search out errors, and modify the model consequently. Supervised learning thus uses patterns to predict label values on extra unlabeled information. For example, with supervised learning, an algorithm may be fed data with images of sharks labeled as fish and images of oceans labeled as water. By being trained on this data, the supervised learning algorithm should be able to later identify unlabeled shark images as fish and unlabeled ocean images as water.

A common use case of supervised learning is to use historical information to predict statistically probably future events. It's going to use historical stock exchange info to anticipate approaching fluctuations or be used to filter spam emails. In supervised learning, labeled photos of dogs are often used as input files to classify unlabeled photos of dogs.

Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to

a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

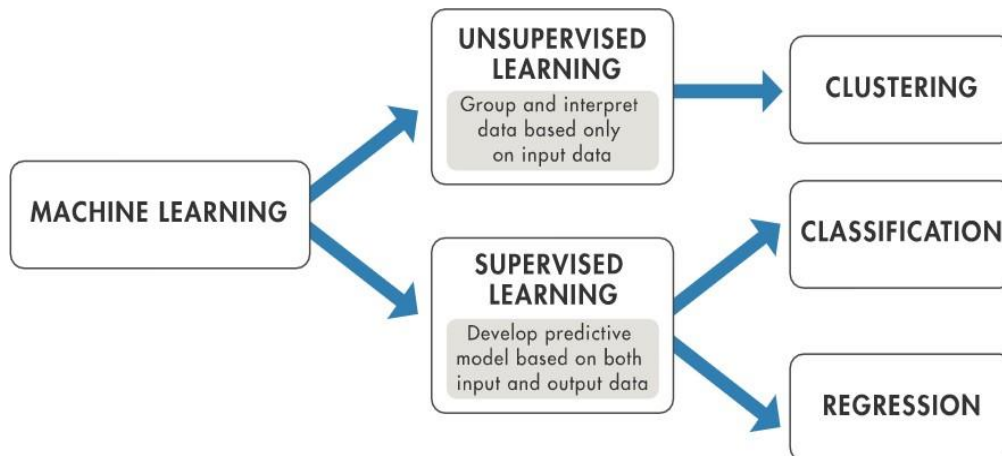
Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

1.3.2 Unsupervised Learning

In unsupervised learning, information is unlabeled, and the learning rule is left to seek out commonalities among its input file. The goal of unattended learning is also as easy as discovering hidden patterns at intervals in a dataset, however, it should even have a goal of feature learning, that permits the procedure machine to mechanically discover the representations that square measure required to classify data.

Unsupervised learning is usually used for transactional information. You will have an oversized dataset of consumers and their purchases, however, as a person, you'll probably not be able to add up what similar attributes will be drawn from client profiles and their styles of purchases.

With this information fed into the Associate in Nursing unattended learning rule, it should be determined that ladies of a definite age vary UN agency obtain unscented soaps square measure probably to be pregnant, and so a promoting campaign associated with physiological condition and baby will be merchandised



1.1 Machine Learning Classification

CLASSIFIERS IN MACHINE LEARNING:

Logistic Regression in Machine Learning

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

Advantages: Logistic regression is easier to implement, interpret, and very efficient to train. It makes no assumptions about distributors of classes in feature space. It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.

Disadvantages: If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to

overfitting. The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.

Support Vector Machine:

Definition: Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Advantages: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

Examples of hyperplanes

H1 is not a good hyperplane as it doesn't separate the classes

H2 does but only with small margin

H3 separates them with maximum margin (distance)

Parameters of SVM

There are three main parameters which we could play with when constructing a SVM classifier:

- Type of kernel
- Gamma value
- C value

K-NEAREST NEIGHBOUR (KNN):

KNN classified an object by a majority vote of the object's neighbors, in the space

of input parameter. The object is assigned to the class which is most common among its **k (an integer specified by human) nearest neighbor**.

It is a **non-parametric, lazy algorithm**. It's non-parametric since it does not make any assumption on data distribution (the data does not have to be normally distributed). It is lazy since it does not really learn any model and make generalization of the data (It does not train some parameters of some function where input X gives output y).

So strictly speaking, this is not really a learning algorithm. It simply classifies objects based on **feature similarity** (feature = input variables).

Classification is computed from a simple majority vote of the k nearest neighbors of each point.

Advantages: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

Disadvantages: Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

Random Forest

Random forest is an ensemble model that grows multiple trees and classify objects based on the "votes" of all the trees. i.e., An object is assigned to a class that has most votes from all the trees. By doing so, the problem with high bias (overfitting) could be alleviated.

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Pros of RF:

- It could handle large data set with high dimensionality, output Importance of Variable, useful to explore the data
- Could handle missing data while maintaining accuracy

Cons of RF:

- Could be a black box, users have little control on what the model does

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, & complex algorithm.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

- R. Perumal et al. developed a heart disease prediction model using the Cleveland dataset of 303 data instances through feature standardization and feature reduction using PCA, where they identified and utilized seven principal components to train the ML classifiers. They concluded that LR and SVM provided almost similar accuracy values (87% and 85%, respectively) compared to that of k-NN with 69%. C. B. C. Latha et al.
- C. B. C. Latha et al. performed a comparative analysis to improve the predictive accuracy of heart disease risk using ensemble techniques on the Cleveland dataset of 303 observations. They applied the brute force method to obtain all possible attribute set combinations and trained the classifiers. They achieved a maximum increase in the accuracy of a weak classifier of 7.26% based on ensemble algorithm, and produced an accuracy of 85.48% using majority vote with NB, BN, RF, and MLP classifiers using an attribute set of nine attributes.
- D. Ananey-Obiri developed three classification models, namely, LR, DT, and Gaussian naïve Bayes(GNB), for heart disease prediction based on the Cleveland dataset. Feature reduction was performed using single value decomposition, which reduced the features from 13 to 4. They concluded that both LR and GNB had predictive scores of 82.75% and AUC of 0.87. It was suggested that other models, such as SVM, k-NN, and random forest, be included.
- N. K. Kumar et al. trained five machine learning classifiers, namely, LR, SVM, DT, RF, and KNN, using a UCI dataset with 303 records and 10 attributes to predict cardiovascular disease. The RF classifier achieved the highest accuracy of 85.71% with an ROC AUC of 0.8675 compared to the other classifiers.
- Gupta et al. replaced the missing values based on the majority label and

derived 28 features using the Pearson correlation coefficient from the Cleveland dataset and trained LR, KNN, SVM, DT, and RF classifiers using the factor analysis of mixed data (FAMD) method; the results based on a weight matrix RF achieved the best accuracy of 93.44%.

- M. Sultana et al. explored KStar, J48, sequential minimal optimization (SMO), BN, and MLP classifiers using Weka on a standard heart disease dataset from the UCA repository with 270 records and 13 attributes; they achieved the highest accuracy of 84.07% with SMO.
- S. Mohan et al. developed an effective hybrid random forest with a linear model (HRFLM) to enhance the accuracy of heart disease prediction using the Cleveland dataset with 297 records and 13 features. They concluded that the RF and LM methods provided the best error rates.
- S. Kodati et al. developed a heart disease prediction system (HDPS) with the Cleveland dataset of 297 instances and 13 attributes using Orange and Weka data mining tools, where they evaluated the precision and recall metrics for the naïve Bayes, SMO, RF, and KNN classifiers.
- Ed-daoudy et al. researched the Cleveland dataset of 303 records and 14 attributes from UCI. They evaluated the performance of the four main classifiers, namely, SVM, DT, RF, and LR, using Apache Spark with its machine learning library MLlib.
- I. Tougui et al. compared the performances of LR, SVM, KNN, ANN, NB, and RF models to classify heart disease with the Cleveland dataset with 297 observations and 13 features using six data mining tools: Orange, Weka, RapidMiner, Knime, MATLAB, and Scikit-Learn.
- V. Pavithra et al. proposed a new hybrid feature selection technique with the combination of random forest, AdaBoost, and linear correlation (HRFLC) using the UCI dataset of 280 instances to predict heart disease. Eleven (11) features were selected using filter, wrapper, and embedded methods; an improvement of 2% was found for the accuracy of the hybrid model.
- C. Gazeloglu et al. projected 18 machine learning models and 3 feature selection techniques (correlation-based FS, chi-square, and fuzzy rough set)

to find the best prediction combination for heart disease diagnosis using the Cleveland dataset of 303 instances and 13 variables.

- N. Louridi et al. proposed a solution to identify the presence/absence of heart disease by replacing missing values with the mean values during pre-processing. They trained three machine learning algorithms, namely, NB, SVM (linear and radial basis function), and KNN, by splitting the Cleveland dataset of 303 instances and 13 attributes into training and testing ratios.
- M. Kavitha et al. implemented a novel hybrid model on the Cleveland heart dataset of 303 instances and 14 features with a 70:30 ratio for training and testing by applying DT, RF, and hybrid (DT + RF) algorithms.
- B. A. Tama et al. designed a stacked architecture to predict heart disease using RF, gradient boosting machine, and extreme gradient boosting with particle swarm optimization(PSO) feature selection using various heart disease datasets, including the Cleveland with 303 instances and 13 attributes

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

3.2 PROPOSED SYSTEM

In the proposed system, the customer segmentation model includes not only centroid-based but also hierarchical clustering. • The three clustering algorithms K means, Minibatch K means and the hierarchical algorithm has been selected from the literature survey. • By deploying the three different algorithms, the clusters are formed and analyzed respectively. • The most effective and efficient algorithm is determined by comparing and evaluating the precision rate among the three algorithms

3.3 OBJECTIVE OF PROJECT

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarities among customers in each group. The main objective of segmenting customers is to decide how to relate to customers in each segment to maximize the value of each customer to the business

The emergence of many competitors and entrepreneurs has caused a lot of tension among competing businesses to find new buyers and keep the old ones. As a result of the predecessor, the need for exceptional customer service becomes appropriate regardless of the size of the business. Furthermore, the ability of any business to understand the needs of each of its customers will provide greater customer support in providing targeted customer services and developing customized customer service plans. This understanding is possible through structured customer service.

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1 Software Requirements:

- ✓ Python

- ✓ Anaconda
- ✓ Jupyter Notebook

3.4.2 Hardware Requirements:

- ✓ Processor: Intel Core i5
- ✓ RAM: 8GB
- ✓ OS: Windows

3.4.3 Libraries:

- ✓ **Tkinter**- Tkinter is a library of python used often by everyone. It is a library that is used to create GUI-based applications easily. It contains so many widgets like radio buttons, text files and so on. We have used this for creating an account registration screen, log in or register screen, prediction interface which is a GUI based application
- ✓ **Sklearn**- Scikit Learn also known as sklearn is an open-source library for python programming used for implementing machine learning algorithms. It features various classification, clustering, regression machine learning algorithms. In this, it is used for importing machine learning models, getting accuracy, get a confusion matrix.
- ✓ **Pandas**- Library of python which can be used easily. It gives speed results and is also easily understandable. It is a library that can be used without any cost. We have used it for data analysis and to read the dataset.
- ✓ **Matplotlib**- Library of python used for visualizing the data using graphs, scatterplots, and so on. Here, we have used it for data visualization.
- ✓ **NumPy**- Library of python used for arrays computation. It has so many functions. We have used this module to change the 2-dimensional array into a contiguous flattened array by using the ravel function.

- ✓ **Pandas Profiling**-This is a library of python which can be used by anyone free of cost. It is used for data analysis. We have used this for getting the report of the dataset.

3.5. PROGRAMMING LANGUAGES

3.5.1 Python

Python is the best programming language fitted to Machine Learning. In step with studies and surveys, Python is the fifth most significant language yet because the preferred language for machine learning and information science. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Features of python

There are many features in Python, some of which are discussed below –

1. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Java script, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.
2. Free and Open Source: Python language is freely available at the official website and can download it easily.
3. Object-Oriented Language: One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
4. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
5. Extensible feature: Python is an Extensible language. We can write us some Python code into C or C++ language and also, we can compile that code in C/C++ language.
6. Python is Portable language: Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.
7. Python is Integrated language: Python is also an integrated language because we can easily integrated python with other languages like C, C++, etc.
8. Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode
9. Large Standard Library: Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

10. Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

Advantages of python

- ***Integration Feature:*** Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Python. Python also processes XML and other mark-up languages as it can run on all modern operating systems through same byte code.
- ***Improved Programmer's Productivity:*** The language has extensive support libraries and clean object-oriented designs that increase two to tenfold of programmer's productivity while using the languages like Java, VB, Perl, C, C++ and C#.
- ***Productivity:*** With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

Disadvantages of Python

Python has varied advantageous features, and programmers prefer this language to other programming languages because it is easy to learn and code too. However, this language has still not made its place in some computing arenas that includes Enterprise Development Shops. Therefore, this language may not solve some of the enterprise solutions, and limitations include-

- ***Difficulty in Using Other Languages:*** The Python lovers become so accustomed to its features and its extensive libraries, so they face problem

in learning or working on other programming languages. Python experts may see the declaring of cast “values” or variable “types”, syntactic requirements of adding curly braces or semi colons as an onerous task.

- **Weak in Mobile Computing:** Python has made its presence on many desktop and server platforms, but it is seen as a weak language for mobile computing. This is the reason very few mobile applications are built in it like Carbon NELLE.
- **Gets Slow in Speed:** Python executes with the help of an interpreter instead of the compiler, which causes it to slow down because compilation and execution help it to work normally. On the other hand, it can be seen that it is fast for many web applications too.
- **Run-time Errors:** The Python language is dynamically typed so it has many design restrictions that are reported by some Python developers. It is even seen that it requires more testing time, and the errors show up when the applications are finally run.
- **Underdeveloped Database Access Layers:** As compared to the popular technologies like JDBC and ODBC, the Python’s database access layer is found to be bit underdeveloped and primitive. However, it cannot be applied in the enterprises that need smooth interaction of complex legacy data.

3.5.2 Domain

Machine learning could be a subfield of computer science (AI). The goal of machine learning typically is to know the structure information of knowledge of information and match that data into models which will be understood and used by folks. Although machine learning could be a field inside technology, it differs from ancient process approaches. In ancient computing, algorithms are sets of expressly programmed directions employed by computers to calculate or downside solve. Machine learning algorithms instead give computers to coach on knowledge inputs and use applied math analysis to output values that fall inside a particular vary. Thanks to this, machine learning facilitates computers in building models from sample knowledge to modify decision-making processes supported knowledge inputs

3.6. SYSTEM ARCHITECTURE

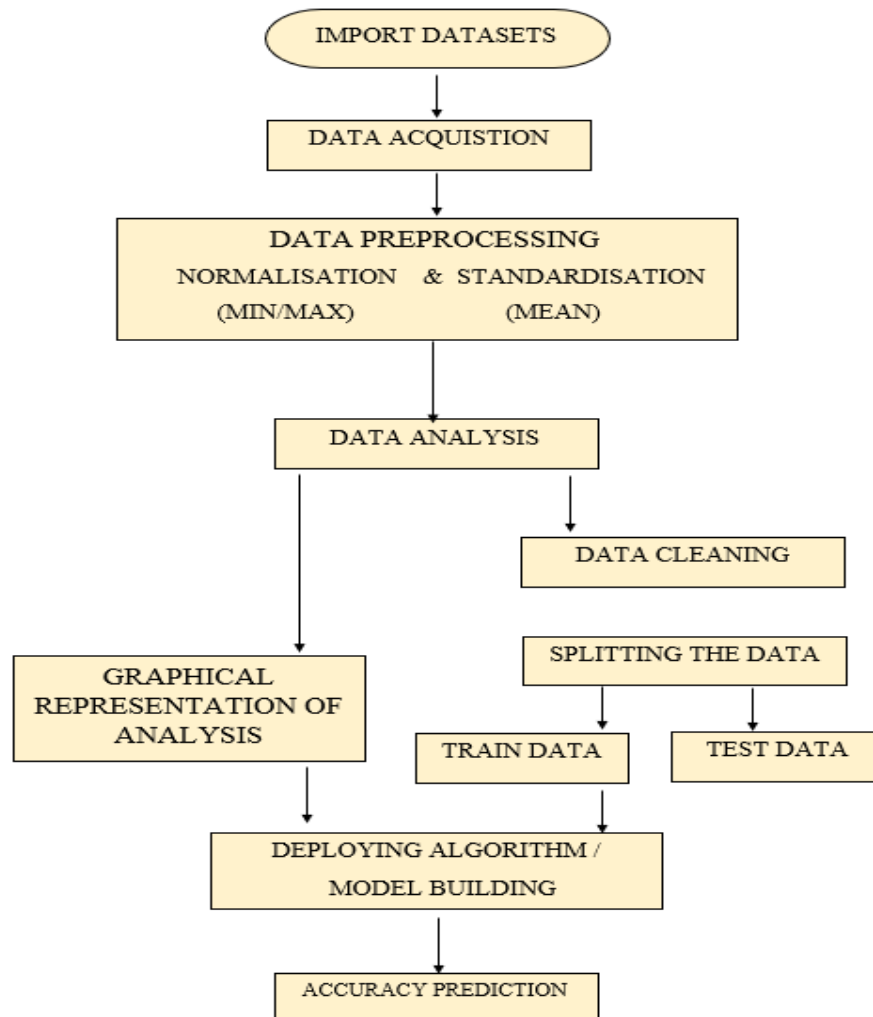


Fig 3.1 Work Flow Diagram

Data collection

Data used in this project is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which I already know the target answer is called labelled data.

Data pre-processing

Pre-processing is the process of three important and common steps as follows:

- **Formatting:** It is the process of putting the data in a legitimate way that it would be suitable to work with. Format of the data files should be formatted according to the need. Most recommended format is .csv files.
- **Cleaning:** Data cleaning is a very important procedure in the path of data science as it constitutes the major part of the work. It includes removing missing data and complexity with naming category and so on. For most of the data scientists, Data Cleaning continues of 80% of work.
- **Sampling:** This is the technique of analyzing the subsets from whole large datasets, which could provide a better result and help in understanding the behavior and pattern of data in an integrated way

Data visualization

Data Visualization is the method of representing the data in a graphical and pictorial way, data scientists depict a story by the results they derive from analyzing and visualizing the data. The best tool used is Tableau which has many features to play around with data and fetch wonderful results.

Feature extraction

Feature extraction is the process of studying the behavior and pattern of the analyzed data and draw the features for further testing and training. Finally, my models are trained using the Classifier algorithm. I used to classify module on Natural Language Toolkit library on Python. I used the labelled dataset gathered. The rest of my labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random Forest. These algorithms are very popular in text classification tasks.

Evaluation model

Evaluation is an essential part of the model development process. It helps to find the best model that represents our data and how well the selected model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can effortlessly generate overoptimistically and over fitted models. To avoid overfitting, evaluation methods such as hold out and cross-validations are used to test to evaluate model performance. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is well-defined as the proportion of precise predictions for the test data. It can be calculated easily by mathematical calculation i.e., dividing the number of correct predictions by the number of total predictions.

3.7 ALGORITHMS USED

3.7.1 Logistic Regression

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

3.7.2 KNN

The K-nearest neighbor (KNN) algorithm is one of the simplest and earliest classification algorithms. It can be thought a simpler version of an NB classifier. Unlike the NB technique, the KNN algorithm does not require to consider probability values. The 'K' is the KNN algorithm is the number of nearest neighbors considered to take 'vote' from. The selection of different values for 'K' can generate different classification results for the same sample object.

3.7.3 SVM

Support vector machine (SVM) algorithm can classify both linear and non-linear data. It first maps each data item into an n -dimensional feature space where n is the number of features. It then identifies the hyperplane that separates the data items into two

classes while maximizing the marginal distance for both classes and minimizing the classification errors. The marginal distance for a class is the distance between the decision hyperplane and its nearest instance which is a member of that class. More formally, each data point is plotted first as a point in an n -dimension space (where n is the number of features) with the value of each feature being the value of a specific coordinate. To perform the classification, we then need to find the hyperplane that differentiates the two classes by the maximum margin.

3.7.4 Random Forest

A random forest (RF) is an ensemble classifier and consisting of many DTs similar to the way a forest is a collection of many trees. DTs that are grown very deep often cause overfitting of the training data, resulting a high variation in classification outcome for a small change in the input data. They are very sensitive to their training data, which makes them error-prone to the test dataset. The different DTs of an RF are trained using the different parts of the training dataset. To classify a new sample, the input vector of that sample is required to pass down with each DT of the forest. Each DT then considers a different part of that input vector and gives a classification outcome. The forest then chooses the classification of having the most 'votes' (for discrete classification outcome) or the average of all trees in the forest (for numeric classification outcome). Since the RF algorithm considers the outcomes from many different DTs, it can reduce the variance resulted from the consideration of a single DT for the same dataset.

Steps for Implementation:

- Initialize the classifier to be used.
- Train the classifier: All classifiers in scikit-learn uses a fit (X, y) method to fit the model(training) for the given train data X and train label y .
- Predict the target: Given a non-label observation X , the predict(X) returns the predicted label y .
- Evaluate the classifier model

3.8 MODULES

The project contains three parts:

- 3.8.1 Dataset Collection- We had collected datasets from Kaggle notebooks. The dataset contains the symptoms and the corresponding disease. It contains 303 rows.
- 3.8.2 Train and test the model- We had used four classification algorithms named KNN algorithm, Logistic regression, Random Forest and SVM algorithm to train the dataset. After training, we had tested the model and found the prediction of disease with maximum accuracy.
- 3.8.3 Hyperparameter tuning- Hyperparameters cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Following are the steps to do this project (use Jupyter Notebook):

- A)** Collect the dataset.
- B)** Import the necessary libraries.
- C)** Visualize the dataset.
- D)** Train the dataset using LR, KNN, RF, SVM.
- E)** Test the model and find the accuracy score
- F)** Tune the model
- G)** Based on the scores predict which algorithm is best for prediction.

3.9 Web Application

To Develop a Web based Application which will predict heart related diseases considering various cardiovascular attributes. Using technology stacks HTML, CSS Animations, Heroku, Data Science, Machine Learning.

The aim is the system developed will be a web application which will take input the Heart measures of the user and based on those conditions and predict whether the user may suffer from heart disease or not. The main objectives are to predict heart disease through a web app, analyze all different attributes related to cardiac body parts and check how they affect heart system and to visualize different parameters associated with heart disease. This proposed method is a web application which will input the attributes of Heart of the patient and predict whether it may develop any heart syndrome or not. The Dataset used here contains the age, gender, chest pain types, cholesterol, resting ecg results, fasting blood pressure, maximum heart rate achieved etc.

The Frontend application of the system consist of the attributes of the dataset which the user has to input based on is/her current cardiac conditions.

The Frontend is developed using HTML and CSS Animations and has been designed in a user-friendly manner. The Backend of the system is designed using Heroku module and it contains the saved file of the best working ML algorithm in the dataset. The prediction is done by applying the selected algorithm and on the data entered by the user. The designed Web Application will help users in estimating future heart diseases based on their current cardiac conditions.

3.10 Heroku

Heroku is a widely relied-upon platform as a service offering that enables developers to carry out hassle-free application deployment, scaling, and management. This platform offers support for a wide range of programming languages such as Java, Ruby, PHP, Node.js, Python, Scala, and Clojure. Heroku runs applications through virtual containers known as Dynos.

Advantages

Users can get started with the free tier of Heroku, which does come with some restrictions and fundamental functionality. It is a suitable plan for those who intend to create a basic cloud app.

Developers can opt for this plan initially before committing to the platform. You can use it for testing the features of Heroku and figuring out if it's a suitable fit for your application needs.

As a leading platform as a service offering, Heroku is known for its ease-of-use. The free version of Heroku is best suited for small development projects.

Developers can also opt for several different tier packages that are more suitable for large businesses' complicated requirements. The intuitive Heroku platform dashboard helps users perform easy scaling, management, and application monitoring.

Heroku focuses on helping developers hone their skills to create feature-rich applications. The experience is favourable for developers as they access some useful tools to accelerate core development processes.

They are free from having to perform various repetitive tasks and can concentrate on critical development actions. Heroku's vast array of features can breathe life into any application development project.

Heroku takes away the burden of server management and gives developers access to an easy-to-use web UI. One-click deployments are also quite convenient for developers once an application is running.

They can create new software releases conveniently in the form of release branches and quickly deploy branches across different environments. The ease of running Heroku makes it a convenient choice for application development.

Heroku PaaS relies on a Dyno for running the code written by developers. Each dyno is fundamentally a container encapsulating resources such as memory and CPU and application code and related dependencies.

Heroku users can then choose to create any number of independent application copies once it has been prepared, a process known as horizontal scaling. Applications that receive an unprecedented influx of traffic can be handled through the one-click creation of hundreds of app instances.

Heroku's auto-scaling feature helps it easily detect traffic spikes and create more Dynos accordingly.

Heroku offers developers a secure way of developing applications due to its set of security features. This PaaS platform reduces developers' requirement to push constant security patches, which can be a hassle, especially in more sophisticated web applications. Heroku ensures an optimal level of security for servers, application code and prevents any possible issues.

CHAPTER 4

RESULTS AND DISCUSSION

4.1. PERFORMANCE ANALYSIS

Coming to the performance it works in a time rate of 1 second per statement and code implied. Duplicated and similar lookalike data's can be removed efficiently too. The performance of a predictive model is calculated and compared by choosing the right metrics. So, it is very crucial to choose the right metrics for a particular predictive model in order to get an accurate outcome. It is very important to evaluate proper predictive models because various kinds of data sets are going to be used for the same predictive model.

Algorithm	Precision	Recall	F-measure	Accuracy
KNN	0.845	0.823	0.835	89.62%
Logistic Regression	0.857	0.882	0.869	92.25%
Random Forest	0.867	0.882	0.909	86.16%
SVM	0.837	0.911	0.873	88.25%

Fig 4.1: Accuracies obtained

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

Disease is getting worse, and people are getting used to it. The sooner we know an infection, the sooner it tends to be restored. With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. The motivation for the study was to find the most efficient ML algorithm for detection of heart diseases. This study compares the accuracy score of Support Vector Machine, Logistic Regression, Random Forest and K Nearest Neighbor algorithms for predicting heart disease using UCI machine learning repository dataset.

Data sets are repeatedly cleaned, trained, tested to achieve maximum accuracy. To get the best result, visual implants, imaging and biopsy can also be done with different analyzes to achieve the best results among you.

Technology has evolved into a stage where even emotional illnesses can be diagnosed using simple code. All techniques were cleaned, trained and tested from start to end. In all of this, all codes compared to their previous academics have found a better result by predicting heart disease gaining a percentage close to 93 percent. Experimentation with all these three coding categories is essential for producing a successful outcome.

Awareness needs to be made so that people are aware of the difficulties they are facing so that they can get tested early. Diseases especially cancer with minor symptoms should be evaluated early.

At all times, we train the database, without the slightest equilibrium or overcrowding we achieve output that is overlooked at least in small variations. This means that strong achievements in this field and new algorithm will still be available to make it a better place and more relevant to the medical field.

In future the work can be enhanced by developing a web application based on the Random Forest algorithm or SVM including tuning as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting the heart disease effectively and efficiently.

REFERENCES

- [1] Rajdhan Apurb, Agarwal Avi, Sai Milan, Ravi Dundigalla, Ghuli Poonam." Heart Disease Prediction using Machine Learning" INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY
- [2] Shah K, Patel H, Sanghvi D, Shah M (2020) A comparative analysis of logistic regression, random Forest and KNN models for the text classification. Augmented Human Research 5(1):1–16K.
- [3] J. Krishnan Santana; S. Geetha "Prediction of Heart Disease Using Machine Learning Algorithms". 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)Publisher: IEEE
- [4] "Prediction of Heart Disease using Machine Learning Algorithms" Krishnan J Santhana and S Geetha ICIICT |Year :2019| Conference Paper | Publisher: IEEE.
- [5] "Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective heart disease prediction using hybrid machine learning techniques." IEEE Access 7(2019): 81542-81554.
- [6] Savita ganga sharma (2020)." A Review on machine learning techniques for prediction of cardiovascular diseases". 2020 Sixth international conference on parallel, Distributed and Grid computing.
- [7] Ashish Chhabbi, Lakhan Ahuja, Sahil Ahir and Y. K. Sharma, "Heart Disease Prediction Using Data Mining Techniques", *IJRAT Special Issue National Conference "NCPC-2016"*, pp. 104-106, March 2016.
- [8] "Santhana Krishnan. J, Geetha S. "Prediction of Heart Disease Using Machine Learning Algorithms", 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT).
- [9] National Health Council, 'Heart HealthScreenings',2017. [Online]Available: http://www.heart.org/HEARTORG/Conditions/HeartHealthScreenings_UCM_428687_Article.jsp#.WnsOAeeYPIV
- [10] P. Hamsagayathri; S. Vigneshwaran," Symptoms Based Disease Prediction Using Machine Learning Techniques",IEEE 2021

APPENDICES

A. SOURCE CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
# Other libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.naive_bayes import BernoulliNB
from sklearn.naive_bayes import MultinomialNB
# Machine Learning
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
df = pd.read_csv(r'C:\Users\nayan\OneDrive\Desktop\Nayana\heart.csv')
df.head()
df.shape
df.columns
df.describe()
df.isnull().sum()
```

```

print(df.info())
rcParams['figure.figsize'] = 20, 20
plt.matshow(df.corr())
plt.yticks(np.arange(df.shape[1]), df.columns)
plt.xticks(np.arange(df.shape[1]), df.columns)
plt.colorbar()
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(), annot=True, cmap='terrain')
sns.pairplot(data=df)
df.hist(figsize=(12,12), layout=(5,3));
df.plot(kind='box', subplots=True, layout=(5,3), figsize=(12,12))
plt.show()
sns.catplot(data=df, x='sex', y='age', hue='target', palette='husl')
sns.barplot(data=df, x='sex', y='chol', hue='target', palette='spring')
df['sex'].value_counts()
df['target'].value_counts()
df['thal'].value_counts()
sns.countplot(x='sex', data=df, palette='husl', hue='target')
sns.countplot(x='target', palette='BuGn', data=df)
sns.countplot(x='ca', hue='target', data=df)
df['ca'].value_counts()
sns.countplot(x='thal', data=df, hue='target', palette='BuPu' )
sns.countplot(x='thal', hue='sex', data=df, palette='terrain')
df['cp'].value_counts()
sns.countplot(x='cp' ,hue='target', data=df, palette='rocket')
sns.countplot(x='cp', hue='sex', data=df, palette='BrBG')
sns.boxplot(x='sex', y='chol', hue='target', palette='seismic', data=df)
sns.barplot(x='sex', y='cp', hue='target', data=df, palette='cividis')
sns.barplot(x='sex', y='thal', data=df, hue='target', palette='nipy_spectral')
sns.barplot(x='target', y='ca', hue='sex', data=df, palette='mako')

```

```

sns.barplot(x='sex', y='oldpeak', hue='target', palette='rainbow', data=df)
df['fbs'].value_counts()
sns.barplot(x='fbs', y='chol', hue='target', data=df,palette='plasma' )
sns.barplot(x='sex',y='target', hue='fbs',data=df)
gen = pd.crosstab(df['sex'], df['target'])
print(gen)
gen.plot(kind='bar', stacked=True, color=['green','yellow'], grid=False)
temp=pd.crosstab(index=df['sex'],
                  columns=[df['thal']],
                  margins=True)
temp
temp.plot(kind="bar",stacked=True)
plt.show()
temp=pd.crosstab(index=df['target'],
                  columns=[df['thal']],
                  margins=True)
temp
temp=pd.crosstab(index=df['target'],
                  columns=[df['thal']],
                  margins=True)
temp
chest_pain = pd.crosstab(df['cp'], df['target'])
chest_pain
chest_pain.plot(kind='bar', stacked=True, color=['purple','blue'], grid=False)
categorical_val = []
continous_val = []
for column in df.columns:
    if len(df[column].unique()) <= 10:
        categorical_val.append(column)
    else:

```

```

        continous_val.append(column)
categorical_val
plt.figure(figsize=(15, 15))
for i, column in enumerate(categorical_val, 1):
    plt.subplot(3, 3, i)
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Have Heart Disease =
NO', alpha=0.6)
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Have Heart Disease =
YES', alpha=0.6)
    plt.legend()
    plt.xlabel(column)
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
StandardScaler = StandardScaler()
columns_to_scale = ['age','trestbps','chol','thalach','oldpeak']
df[columns_to_scale] = StandardScaler.fit_transform(df[columns_to_scale])
plt.figure(figsize=(15, 15))
for i, column in enumerate(continous_val, 1):
    plt.subplot(3, 2, i)
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Have Heart Disease =
NO', alpha=0.6)
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Have Heart Disease =
YES', alpha=0.6)
    plt.legend()
    plt.xlabel(column)
df.head()
X= df.drop(['target'], axis=1)
y= df['target']
X_train, X_test,y_train, y_test=train_test_split(X,y,test_size=0.3,random_state=40)

```

```

print('X_train-', X_train.size)
print('X_test-',X_test.size)
print('y_train-', y_train.size)
print('y_test-', y_test.size)
#LogisticRegression
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
model1=lr.fit(X_train,y_train)
prediction1=model1.predict(X_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,prediction1)
cm
sns.heatmap(cm, annot=True,cmap='BuPu')
TP=cm [0][0]
TN=cm [1][1]
FN=cm [1][0]
FP=cm [0][1]
print ('Accuracy:', (TP+TN)/(TP+TN+FN+FP))
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction1))
#RandomForest
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
model3 = rfc.fit(X_train, y_train)
prediction3 = model3.predict(X_test)
confusion_matrix(y_test, prediction3)
accuracy_score(y_test, prediction3)
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction3))

```

```

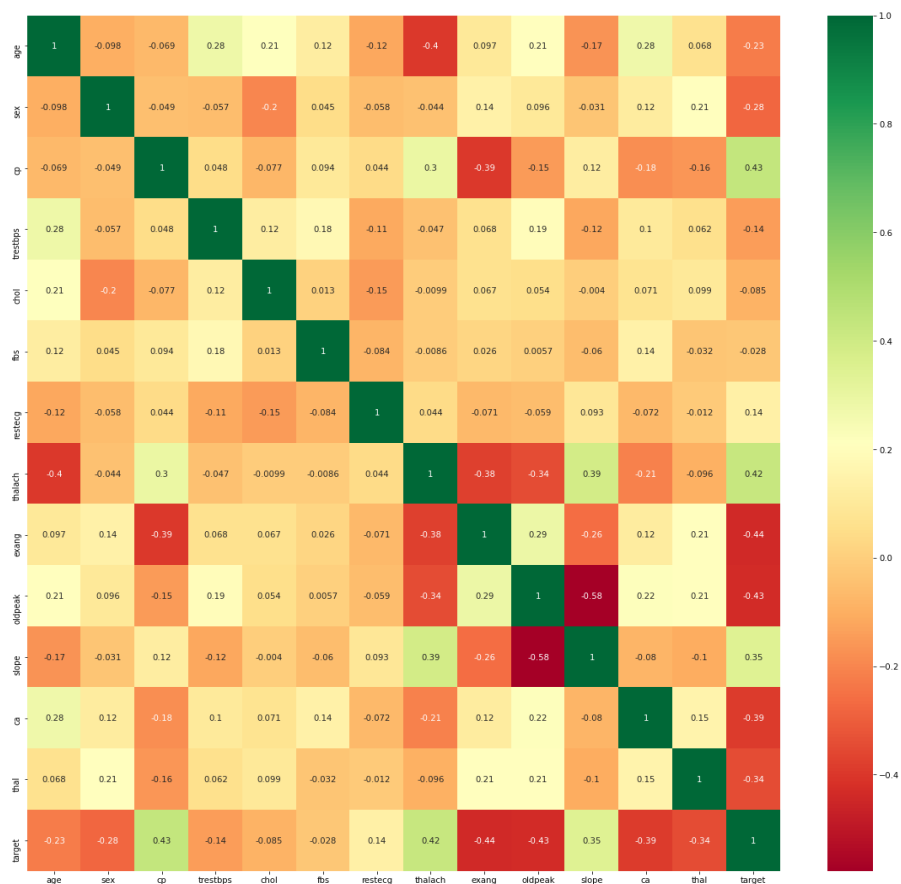
#SupportVectorMachine
from sklearn.svm import SVC
svm=SVC()
model4=svm.fit(X_train,y_train)
prediction4=model4.predict(X_test)
cm4= confusion_matrix(y_test,prediction4)
accuracy_score(y_test, prediction4)
print(classification_report(y_test, prediction4))
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier()
model6 = KNN.fit(X_train, y_train)
prediction5 = model6.predict(X_test)
cm6= confusion_matrix(y_test, prediction5)
cm6
accuracy_score(y_test, prediction5)
print(classification_report(y_test, prediction5))
print('KNN :', accuracy_score(y_test, prediction5))
print('lr :', accuracy_score(y_test, prediction1))
print('rfc :', accuracy_score(y_test, prediction3))
print('SVM :', accuracy_score(y_test, prediction4))

```

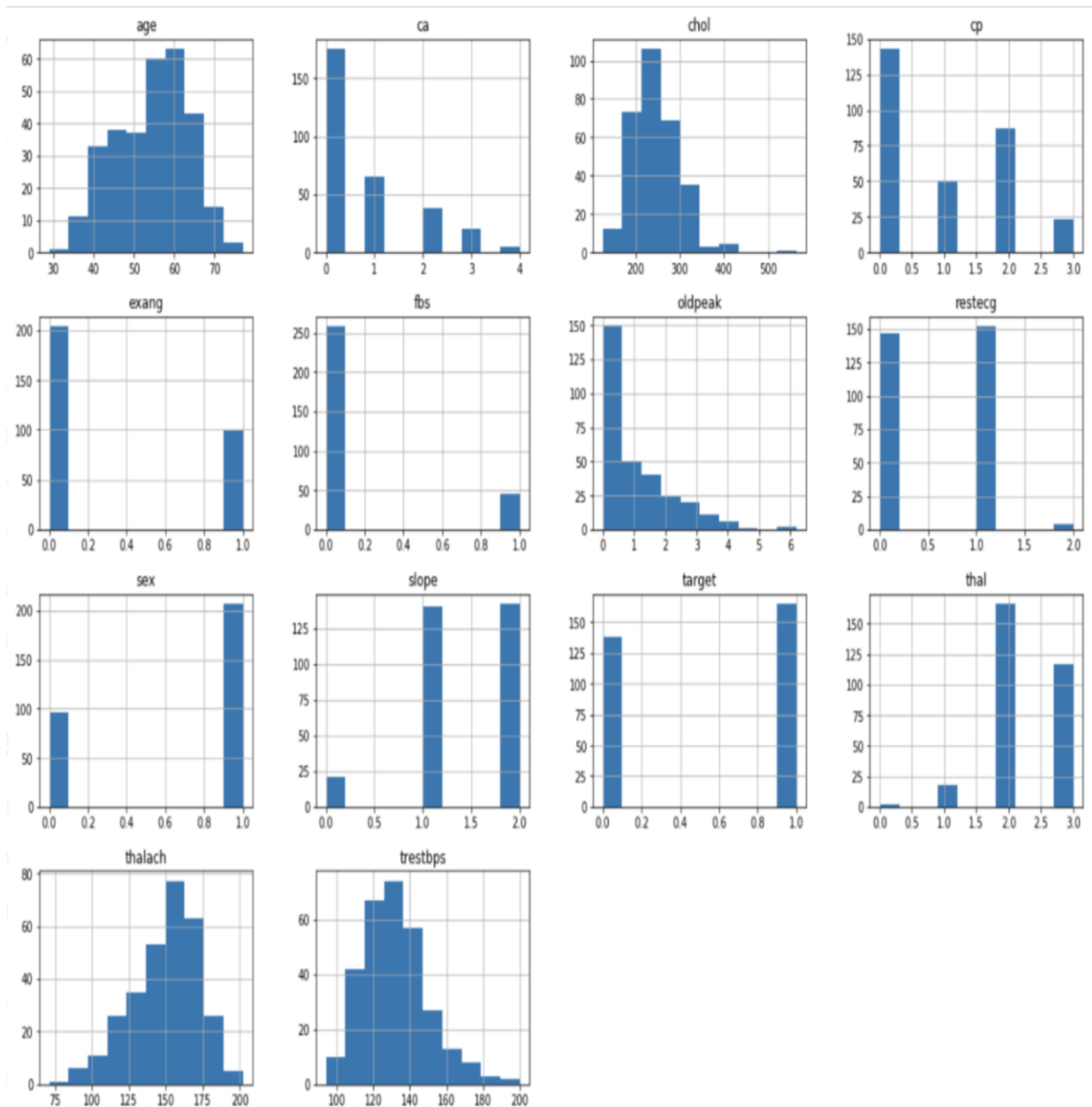
B. SCREENSHOTS

	age	trestbps	chol	thalach	oldpeak	target	sex_0	sex_1	cp_0	cp_1	...	slope_2	ca_0	ca_1	ca_2	ca_3	ca_4	thal_0	thal_1	thal_2	th
0	0.952197	0.763956	-0.256334	0.015443	1.087338	1	0	1	0	0	...	0	1	0	0	0	0	0	1	0	
1	-1.915313	-0.092738	0.072199	1.633471	2.122573	1	0	1	0	0	...	0	1	0	0	0	0	0	0	1	
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912	1	1	0	0	1	...	1	1	0	0	0	0	0	0	1	
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705	1	0	1	0	1	...	1	1	0	0	0	0	0	0	1	
4	0.290464	-0.663867	2.082050	0.583939	-0.379244	1	1	0	1	0	...	1	1	0	0	0	0	0	0	1	

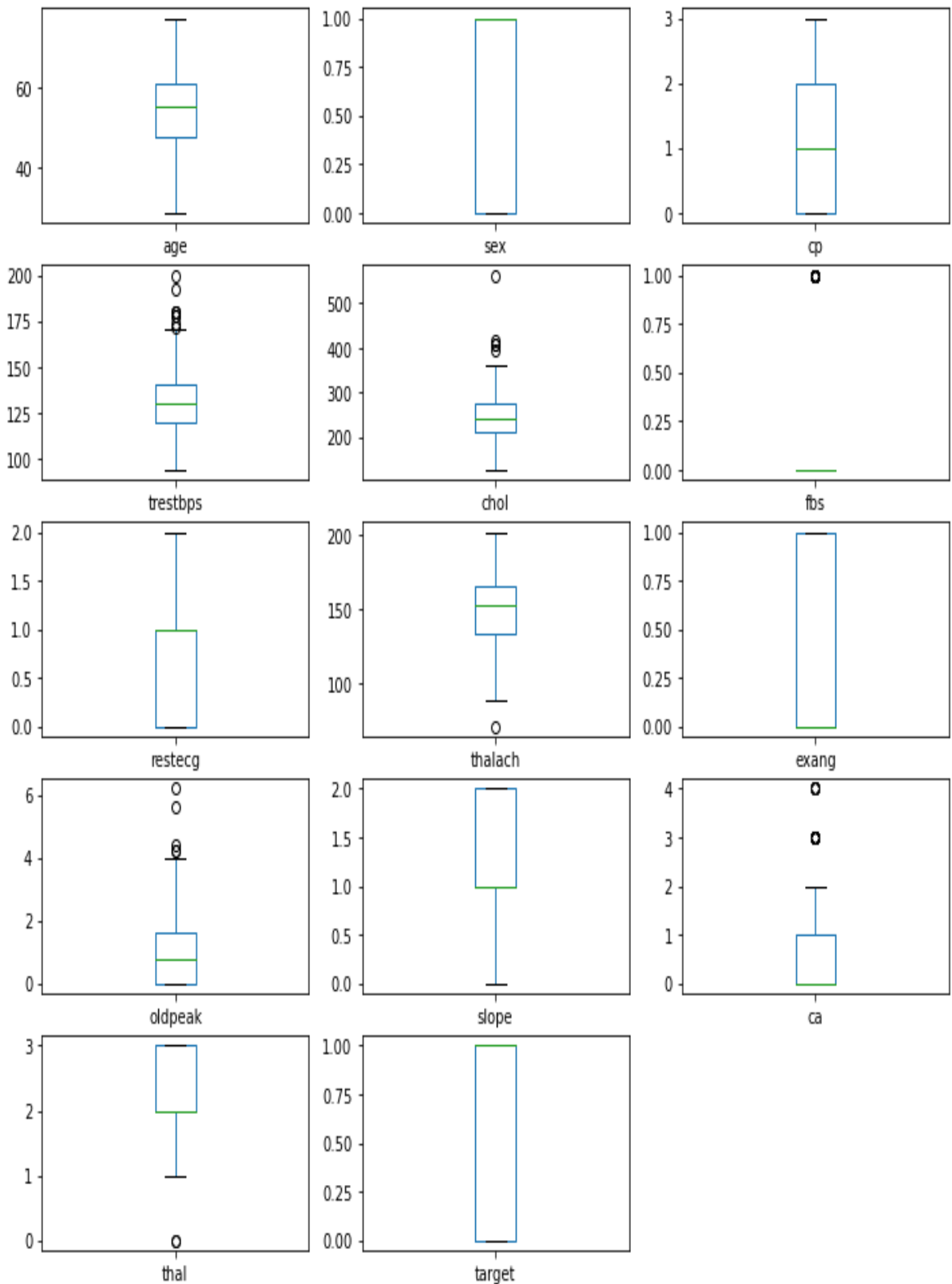
B.1 Dataset



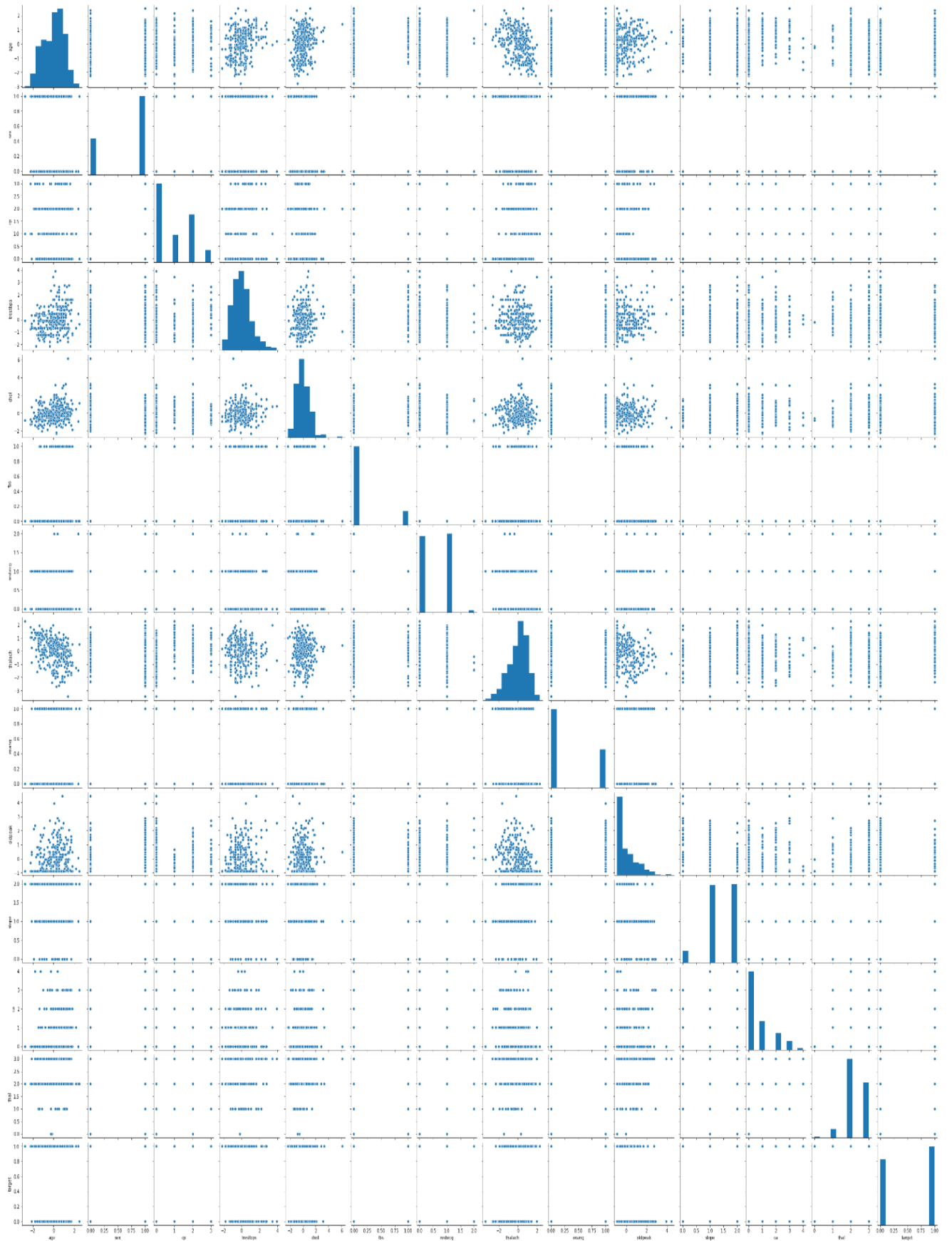
B.2. Correlation Matrix



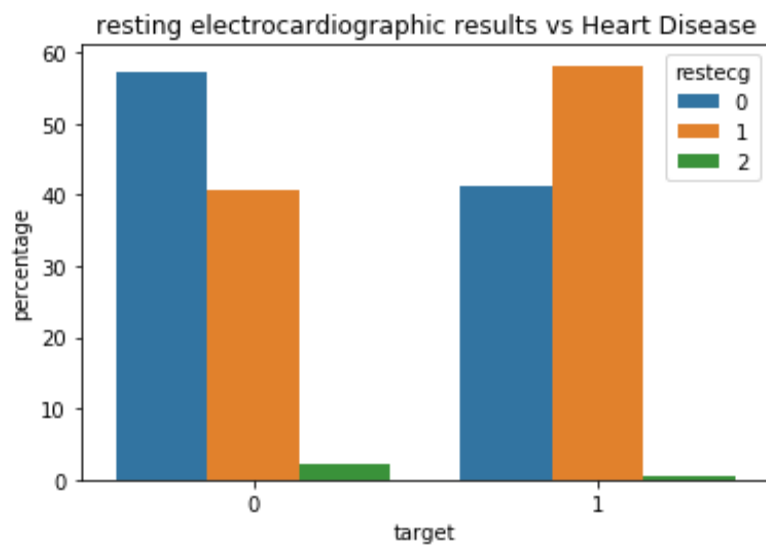
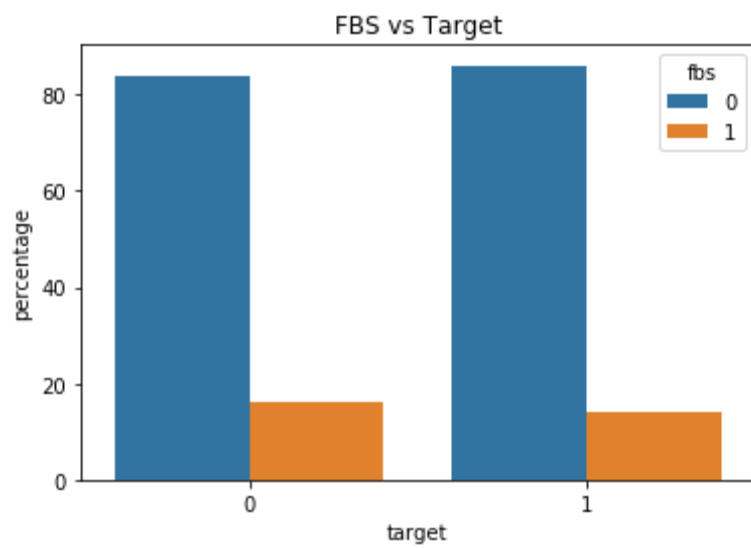
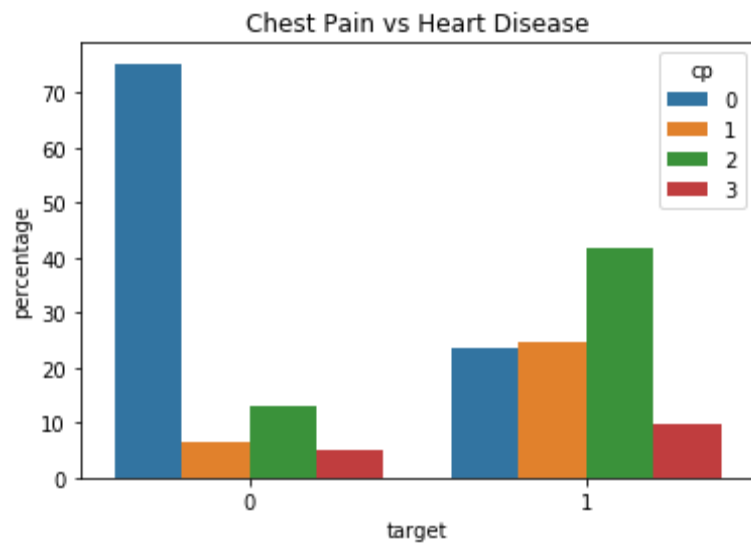
B.3 Histogram

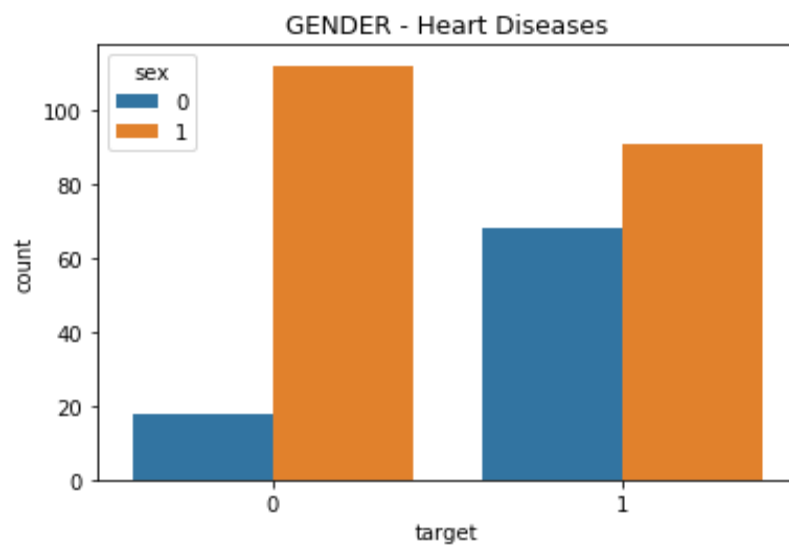
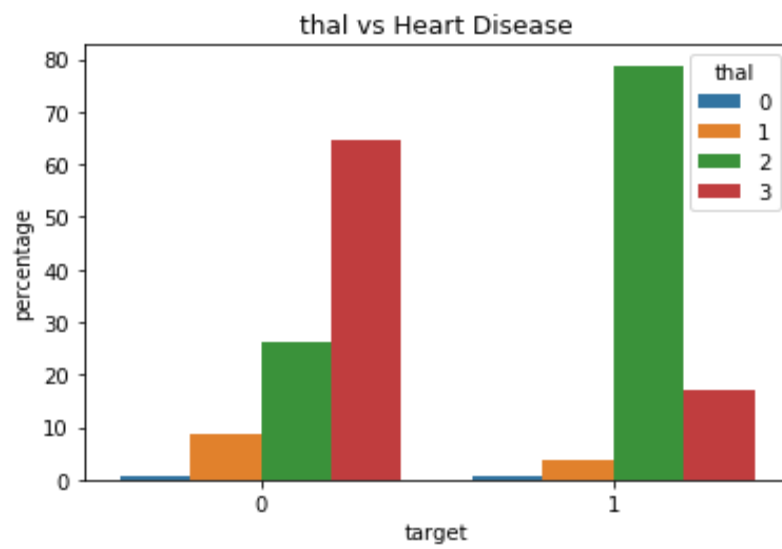
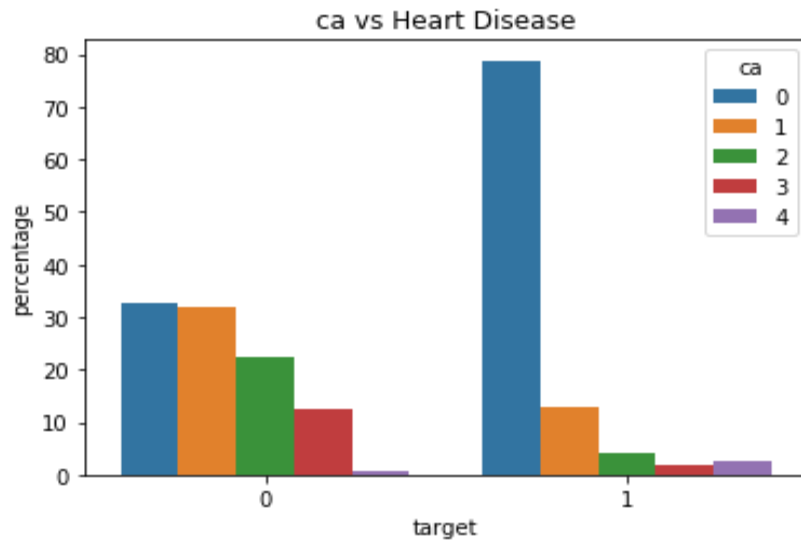


B.4 Box plot

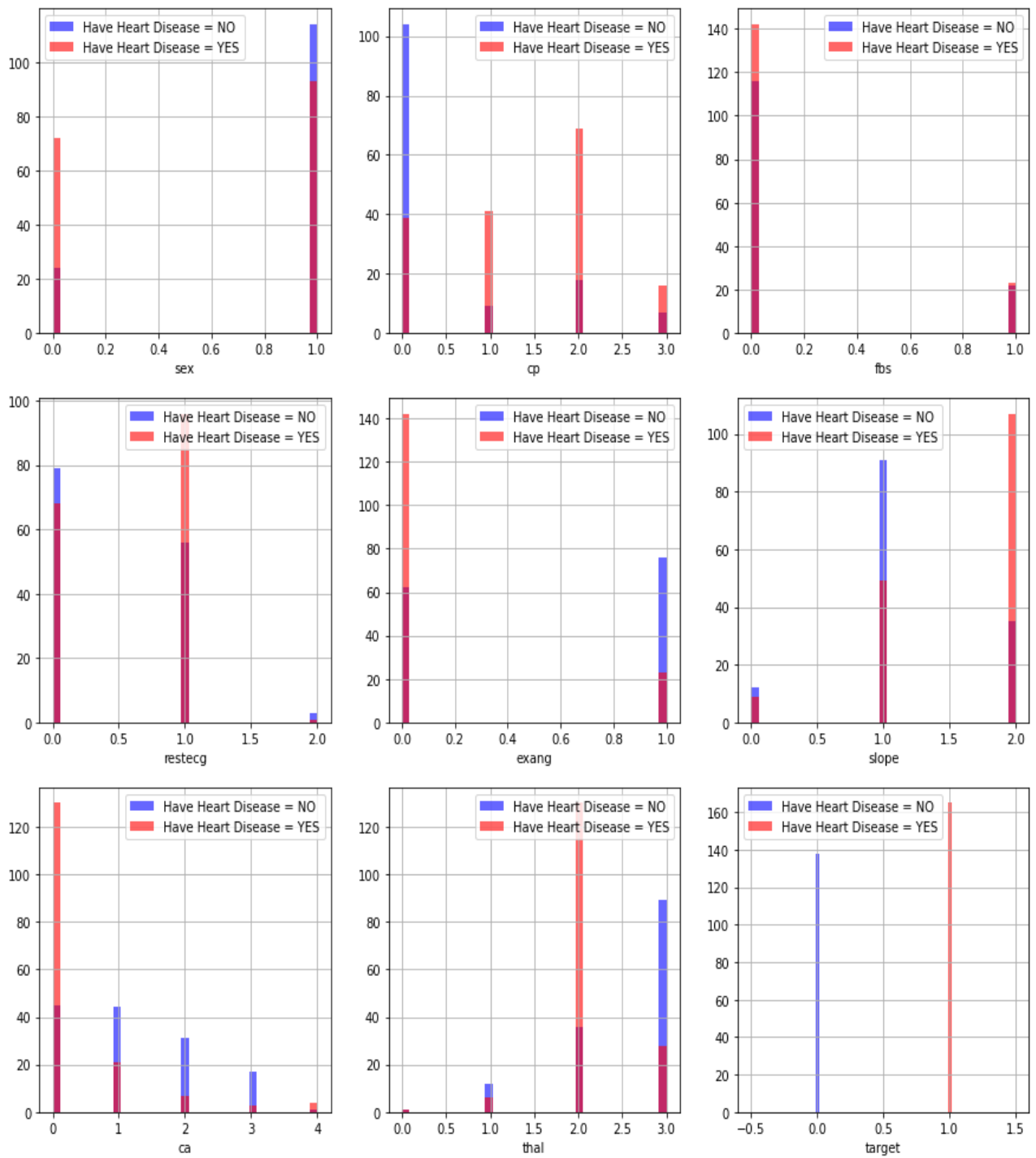


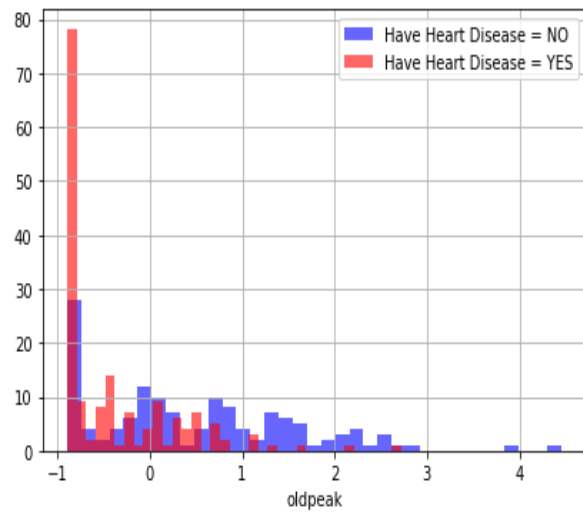
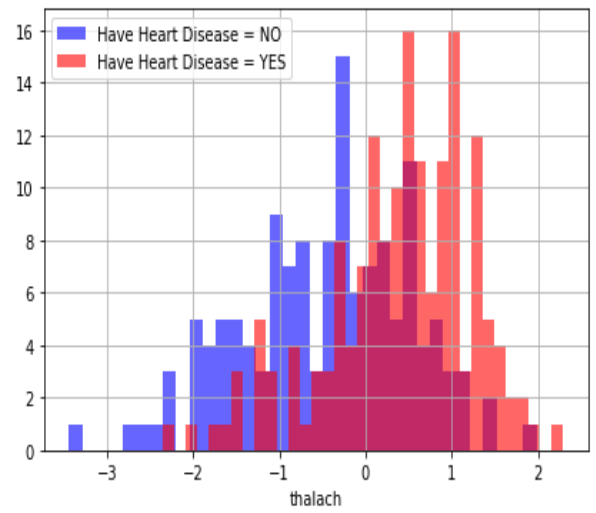
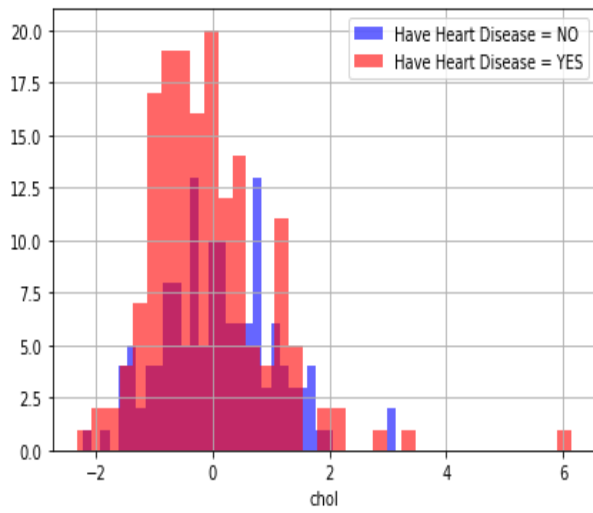
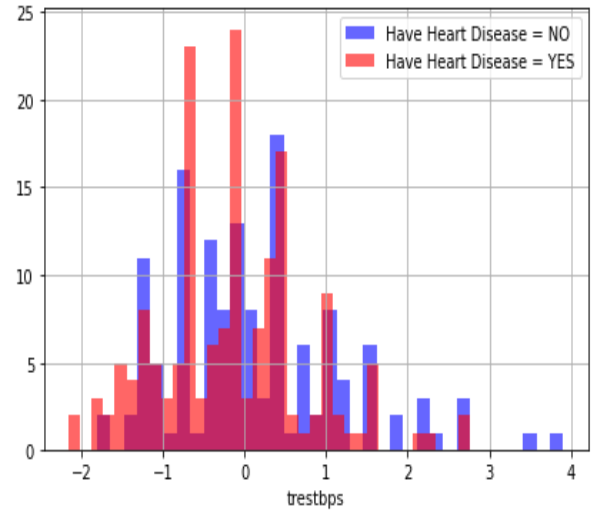
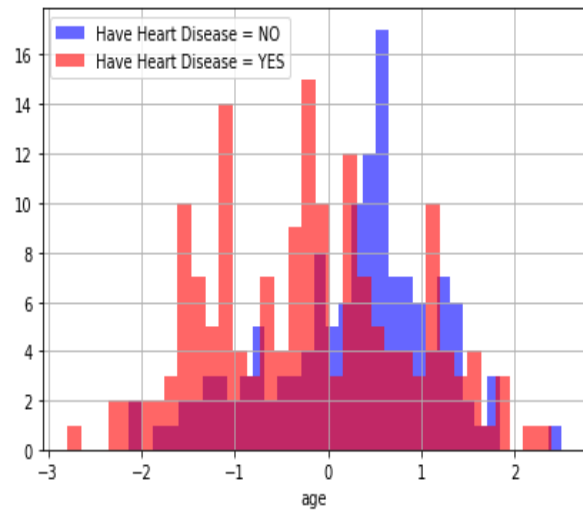
B.5 Seaborn Visualization





B.6 Bar plot Representation





B.7 Graph plot

	precision	recall	f1-score	support
0	0.92	0.90	0.91	40
1	0.92	0.94	0.93	51
accuracy			0.92	91
macro avg	0.92	0.92	0.92	91
weighted avg	0.92	0.92	0.92	91

B.8 LR Classification Report

	precision	recall	f1-score	support
0	0.82	0.80	0.81	40
1	0.85	0.86	0.85	51
accuracy			0.84	91
macro avg	0.83	0.83	0.83	91
weighted avg	0.83	0.84	0.83	91

B.9 KNN Classification Report

	precision	recall	f1-score	support
0	0.94	0.82	0.88	40
1	0.88	0.96	0.92	51
accuracy			0.90	91
macro avg	0.91	0.89	0.90	91
weighted avg	0.90	0.90	0.90	91

B.10 SVM Classification Report

	precision	recall	f1-score	support
0	0.88	0.88	0.88	40
1	0.90	0.90	0.90	51
accuracy			0.89	91
macro avg	0.89	0.89	0.89	91
weighted avg	0.89	0.89	0.89	91

B.11 RF Classification Report

```
print('KNN :', accuracy_score(y_test, prediction5))
print('lr :', accuracy_score(y_test, prediction1))

print('rfc :', accuracy_score(y_test, prediction3))

print('SVM :', accuracy_score(y_test, prediction4))
```

```
KNN : 0.8791208791208791
lr : 0.9230769230769231
rfc : 0.8901098901098901
SVM : 0.9010989010989011
```

B.12 Accuracy table

Heart Disease Predictor

A Machine Learning Web Application that predicts chances of having heart Disease or not.

Age	Your age..
Sex	---select option---
Chest Pain Type	---select option---
Resting Blood Pressure	A number in range [94-200] mmHg
Serum Cholesterol	A number in range [126-564] mg/dl
Fasting Blood Sugar	---select option---
Resting ECG Results	---select option---
Max Heart Rate	A number in range [71-202] bpm
Exercise-induced Angina	---select option---
ST depression	ST depression, typically in [0-6.2]
Serum Cholesterol	A number in range [126-564] mg/dl
Fasting Blood Sugar	---select option---
Resting ECG Results	---select option---
Max Heart Rate	A number in range [71-202] bpm
Exercise-induced Angina	---select option---
ST depression	ST depression, typically in [0-6.2]
slope of the peak exercise ST segment	---select option---
Number of Major vessels	Typically in [0-4]
Thalassemia	---select option---

Predict

Heart Disease Predictor

A Machine Learning Web App, Built with Flask, Deployed using Heroku.

Prediction: **Oops! You have Chances of Heart Disease.**

B.13 Web application of disease prediction

C. RESEARCH PAPER PUBLICATION





SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in



National Conference on Computational Intelligence and Communication Networks



NCCICN

24TH – 25TH MARCH 2022



Certificate of Presentation

*This is to certify that **Dr./Mr./Ms. SRIJA SAHA**, of **Sathyabama Institute of Science and Technology**, has presented a paper entitled **"Heart Disease Analysis and Prediction using Machine Learning"**, in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala

Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan

Convener
Professor & Head, CSE

Dr. S. Vigneshwari

Convener
Professor & Head, CSE

D. PLAGARISM REPORT

The screenshot displays a web browser window with the URL `quetext.com/results/e4e5e3a3db8b9079af4a`. The Quetext logo is in the top left, and a "New Search" button is in the top right. The main content area is split into two columns. The left column contains an abstract text snippet: "Abstract— Over the years we have seen that the spread of disease is a global problem. Accurate and timely analysis of any health-related problems is essential to treat and treat them early but with this growing number it becomes more difficult. Machine learning strategies have been accelerating the field of health with more research and methodologies. Machine learning strategies can contribute significantly to the process of predicting and diagnosing diseases early, becoming a field of research and proving to be a viable option in the health sector. The research paper focuses on the prediction of diseases caused in Heart, predicting the likelihood of a person having a specific disease that can lead to heart attack based on various medical factors and related parameters. We set up 4 different predictive". The right column shows a circular progress indicator at 4%, with a tooltip indicating "2 matches from 2 sources". Below this, two sources are listed, each with a 50% match score: `www.igi-global.com` (book > multinational-electronic-health-records-interoperabil...) and `www.ijsdr.org` (papers > USDR2008043.pdf). A red "Feedback" button is on the far right. A blue footer bar at the bottom states "Only the first 500 words were checked" and includes a green "Signup to search more" button.

quetext

New Search

4%

2 matches from 2 sources

50% www.igi-global.com
www.igi-global.com > book > multinational-electronic-health-records-interoperabil...

50% www.ijsdr.org
www.ijsdr.org > papers > USDR2008043.pdf

Feedback

Only the first 500 words were checked [Signup to search more](#)