

AN APPROACH TO FORECASTING MULTIPLE MALADIES UTILIZING MACHINE LEARNING ALGORITHMS



A PROJECT REPORT

Submitted by

VARUN KRISHNAN V **(19104117)**
SURIYASANKAR P **(19104108)**
NANDHAKUMAR M **(19104069)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PAAVAI ENGINEERING COLLEGE
(AUTONOMOUS)

MAY 2023

BONA FIDE CERTIFICATE

Certified that this project report “**AN APPROACH TO FORECASTING MULTIPLE MALADIES UTILIZING MACHINE LEARNING ALGORITHMS**” is the bona fide work of **VARUN KRISHNAN V (19104117)**, **SURIYASANKAR P (19104108)**, **NANDHAKUMAR M (19104069)** who have carried out the project work under my supervision.

SIGNATURE

**Dr.D.BANUMATHY, M.E., Ph.D.,
HEAD OF THE DEPARTMENT**

Department of Computer Science and
Engineering
Paavai Engineering College,
Namakkal – 637018.

SIGNATURE

**Mrs.P.RENUKADEVI, M.E., (Ph.D.),
SUPERVISOR – INTERNAL**

Department of Computer Science and
Engineering
Paavai Engineering College,
Namakkal – 637018.

Submitted for End Semester Examinations held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We **VARUN KRISHNAN V (19104117)**, **SURIYASANKAR P (19104108)**, **NANDHAKUMAR M (19104069)** hereby declare that the project report titled **AN APPROACH TO FORECASTING MULTIPLE MALADIES UTILIZING MACHINE LEARNING ALGORITHMS** done by me under the guidance of **Mrs.P.RENUKADEVI,M.E.,(Ph.D.)**, (Internal Supervisor) at **PAAVAI ENGINEERING COLLEGE** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering. Certified further that, to the best of my knowledge, the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

1.

2.

DATE:

3.

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENTS

A great deal of arduous work and effort has been spent in implementing this project work. Several special people have guided us and have contributed significantly to this work and so this becomes obligatory to record our thanks to them.

We express my profound gratitude to our honourable Chairman, **Shri. CA.N.V.NATARAJAN, B.Com., F.C.A.,** and also to our Correspondent, **Smt. MANGAI NATARAJAN, M.Sc.,** for providing all necessary facilities for the successful completion of my project.

We wish to express my sincere thanks to our respected **Director-Administration, Dr.K.K.RAMASAMY, M.E., Ph.D.,** for all the blessing and help provided during the period of project work.

We would like to thank our respected **Principal, Dr.M.PREM KUMAR, M.E., Ph.D.,** for allowing us to do this project and providing required time to complete the same.

We extend our sincere thanks to **Dr.D.BANUMATHY, M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for being the Project Coordinator and for the able guidance and useful suggestions, which helped us for completing project work in time.

We would like to extend my sincere thanks to **Mrs.P.RENUKADEVI,M.E.,(Ph.D.),** Associate Professor, Department of Computer Science and Engineering for being a Project Coordinator and Supervisor and for giving an opportunity to do this project and also for her inspiring guidance, generous help and support.

We would like to extend our sincere thanks to **all our department staff members and our parents** for their advice and encouragement to do the project work with full interest and enthusiasm.

ABSTRACT

Nowadays, machine learning is utilized across various industries, including healthcare, where it holds notable importance. One way to enhance sufferer satisfaction is by applying machine learning techniques to healthcare. The "An Approach to forecasting multiple maladies utilizing machine learning algorithms" system is based on predictive modelling that considers symptoms entered by the sufferer to forecast the disease. While many existing machine learning applications for health analytics focus on only one disease, such as diabetes, cancer, or skin diseases, there is a lack of a unified system capable of predicting multiple diseases. This article proposes a system capable of predicting diabetes and heart disease. To achieve more accurate predictions of heart disease risk levels, the project proposes the use of a support vector machine (SVM) as the backbone of computational diagnostic tools. The SVM model is a promising classification method that can predict medication adherence in Cardiovascular disease (CVD) patients, thereby helping differentiate sufferers and making evidence-based decisions possible. To predict early diabetes risk, bagging and boosting methods using DTB algorithms were applied to experimental data. Besides this the admin basically the doctor can also able to train their own model using our web application which is created on top of Flask. Ultimately, this project can help improve the health of many people by allowing their condition to be monitored closely, and necessary precautions can be taken to increase their life expectancy.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF ABBREVIATIONS	iii
	LIST OF TABLES	iv
1	INTRODUCTION	1
	1.1 Project Introduction	1
	1.2 Project Objectives	6
2	LITERATURE SURVEY	8
3	SYSTEM ANALYSIS	12
	3.1 Existing System	12
	3.2 Disadvantages	13
	3.3 Proposed System	14
	3.4 Advantages	15
4	SYSTEM ENVIRONMENT	16
	4.1 Hardware Specification	16
	4.2 Software Specification	16
	4.3 Project Problem Statement	16
	4.4 Software Description	16

CHAPTER NO.	TITLE	PAGE NO.
5	SYSTEM DESIGN	22
	5.1 Multi Disease UI Diagnosis Model	22
	5.2 Diabetes Model Train and Build	22
	5.3 Heart Disease Model Train and Build	32
6	SYSTEM TESTING	42
	6.1 Usability Testing	42
	6.2 Functional Testing	42
	6.3 Integration Testing	43
	6.4 Test Cases	43
	6.4.1 Signup/Registration Test Case	43
	6.4.2 Login Test Case	43
7	CONCLUSION	45
	7.1 Conclusion	45
	7.2 Future Scope	45
8	APPENDIX	46
	8.1 Source Code	46
	8.2 Screenshots	55
9	REFERENCES	59

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Segmentation of heart	2
1.2	Glucose conversion from food	3
1.3	Working of machine learning	5
1.4	Types of machine learning	5
5.1	Process of diabetes prediction	24
5.2	Data preprocessing in diabetes prediction	25
5.3	Confusion matrix of diabetes prediction	27
5.4	Prediction model for diabetes	30
5.5	Workflow for heart disease prediction	33
5.6	System architecture for heart disease prediction	38
8.1	Signup form	55
8.2	Login form	55
8.3	User information	56
8.4	Diabetes prediction	56
8.5	Heart disease prediction	57
8.6	Data exploration	58
8.7	Suggestion and Prescription	58

LIST OF ABBREVIATIONS

ACRONYMS	ABBREVIATION
CVD	Cardio Vascular Diseases
SVM	Support Vector Machine
AI	Artificial Intelligence
ML	Machine Learning
MMH	Maximum Marginal Hyperplane
KNN	K-Nearest Neighbour
FFN	Feed Forward Networks
HTTPS	Hypertext Transform Protocol Secure
API	Application Programming Interface
UI	User Interface
ECG	Electrocardiogram
BMI	Body Mass Index

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
5.1	Diabetes dataset model	23
5.2	Effect of Selected factors on diabetes	25
5.3	Database Description	33
5.4	Heart disease dataset model	39
5.5	Precision and recall	40
5.6	Comparison of various algorithms for prediction	40
6.1	Signup page testing	43
6.2	Login page testing	44

CHAPTER 1

INTRODUCTION

1.1 PROJECT INTRODUCTION

Healthcare has always been a critical aspect of our lives, and predicting multiple illnesses accurately is a challenging task that requires extensive research and innovative techniques. With the rise of machine learning algorithms, there has been a significant shift towards developing accurate predictive models for various health conditions. In this project, we present an approach to forecasting multiple maladies utilizing machine learning algorithms. The prevalence of chronic diseases such as heart disease, diabetes, and cancer has been on the rise in recent years, resulting in an increasing burden on healthcare systems worldwide. Early detection and prevention of these diseases are essential for improving public health outcomes and reducing healthcare costs. The proposed approach aims to address this challenge by developing an efficient and accurate model for predicting multiple illnesses. Our methodology involves data collection, feature selection, model training, and performance evaluation to develop an efficient and accurate model for predicting multiple illnesses. Data from various sources, such as electronic health records and patient-generated data, are collected and processed to identify relevant features that can be used to predict multiple maladies. The selected features are then used to train machine learning algorithms such as random forests, logistic regression, and artificial neural networks. The performance of the models is evaluated using various metrics such as accuracy, precision, recall, and F1-score.

In this digital world, data is an asset, and enormous data was generated in all the fields. Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry. Many of the existing models are concentrating on one disease per analysis. Like one analysis for diabetes analysis, one for cancer analysis, one for skin diseases like that. There is no common system present that can analyze more than one disease at a time. Thus, we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict multiple diseases

During a lot of analysis over existing systems in health care analysis considered only one disease at a time. For example, is used to analyse diabetes, article is used to analyse diabetes is used to predict heart disease . Maximum articles focus on a particular disease. When any organization wants to analyse their patient's health reports then they have to deploy many models. The approach in the existing system is useful to analyse only particular disease. Now a day's mortality got increased due to exactly not identifying exact disease. Even the patient got cured from one disease may be suffering from another disease. In real life, I faced that situation. My father got cured from the accident. My father got discharged from hospital but after a few days he got expired. Internally suffering from heart issues which is not identified. Like this many instances observed in many people's life stories.

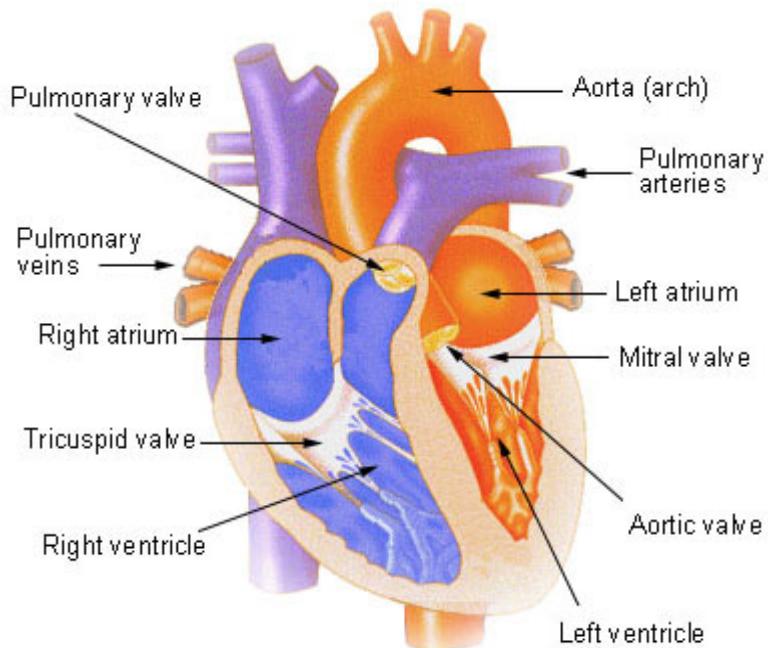


Figure 1.1 Segmentation of Heart

Analyzing the disease. Due to that may be not possible to identify the diseases which will be caused because of that disease. For example, due to diabetes, there may be chance of heart disease, neuropathy, retinopathy, hearing loss, and dementia.

Structure of the Heart

The human heart is a four-chambered muscular organ, shaped and sized roughly like a man's closed fist with two-thirds of the mass to the left of midline. The heart is enclosed in a pericardial sac that is lined with the parietal layers of a serous membrane. The visceral layer of the serous membrane forms the epicardium.

Layers of the Heart Wall

Three layers of tissue form the heart wall. The outer layer of the heart wall is the epicardium, the middle layer is the myocardium, and the inner layer is the endocardium.

Chambers of the Heart

The internal cavity of the heart is divided into four chambers:

- Right atrium
- Right ventricle
- Left atrium
- Left ventricle

The two atria are thin-walled chambers that receive blood from the veins. The two ventricles are thick-walled chambers that forcefully pump blood out of the heart. Differences in thickness of the heart chamber walls are due to variations in the amount of myocardium present, which reflects the amount of force each chamber is required to generate.

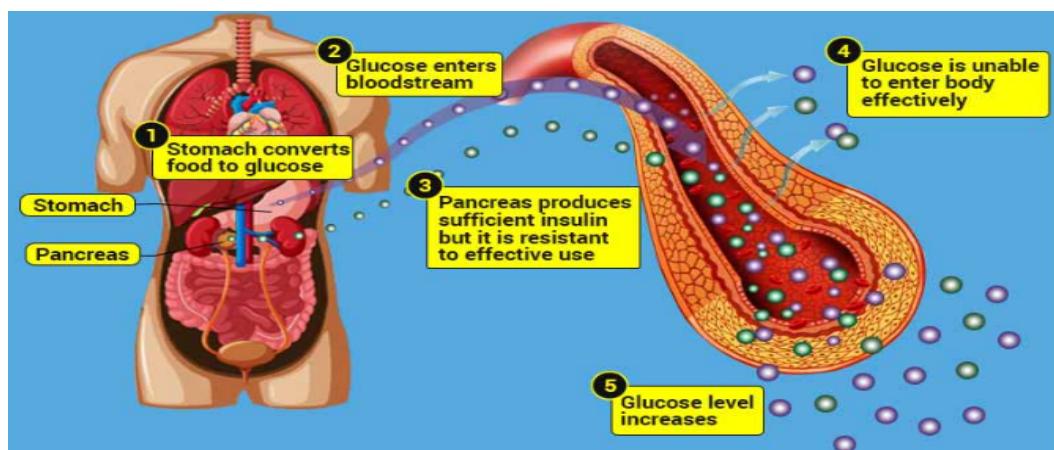


Figure 1.2 Glucose conversion from food

Problem Identified

It's a major challenge in the medical or healthcare industries to offer the highest quality services to all patients, and only those who can afford it can benefit from it. There is a vast amount of healthcare data available that is not being mined in a more efficient and reliable manner to uncover secret knowledge for successful decision-making. There are instruments available which can predict heart disease but either it is expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

Machine Learning

Machine learning is a branch of AI. Other tools for reaching AI include rule-based engines, evolutionary algorithms, and Bayesian statistics. While many early AI programs, like IBM's Deep Blue, which defeated Garry Kasparov in chess in 1997, were rule-based and dependent on human programming, machine learning is a tool through which computers have the ability to teach themselves, and set their own rules. In 2016, Google's DeepMind beat the world champion in Go by using machine learning—training itself on a large data set of expert moves.

Machine Learning Work

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model. The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set. This is just a very high-level example as there are many factors and other steps involved.

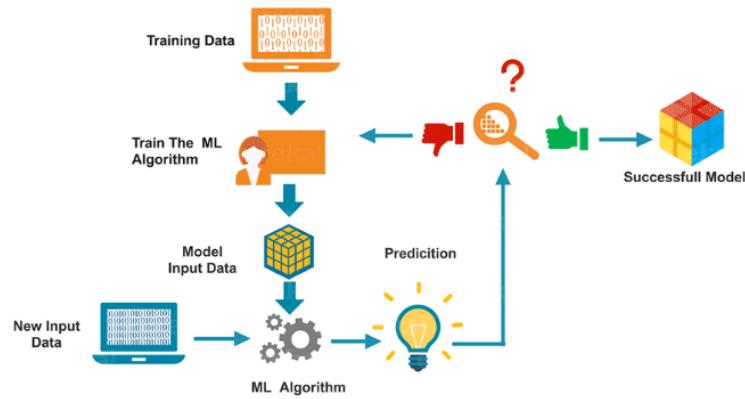


Figure 1.3 Working of Machine Learning

Types of Machine Learning

Machine learning is sub-categorized to three types:

- Supervised Learning – Train Me!
- Unsupervised Learning – I am self-sufficient in learning
- Reinforcement Learning – My life My rules! (Hit & Trial)

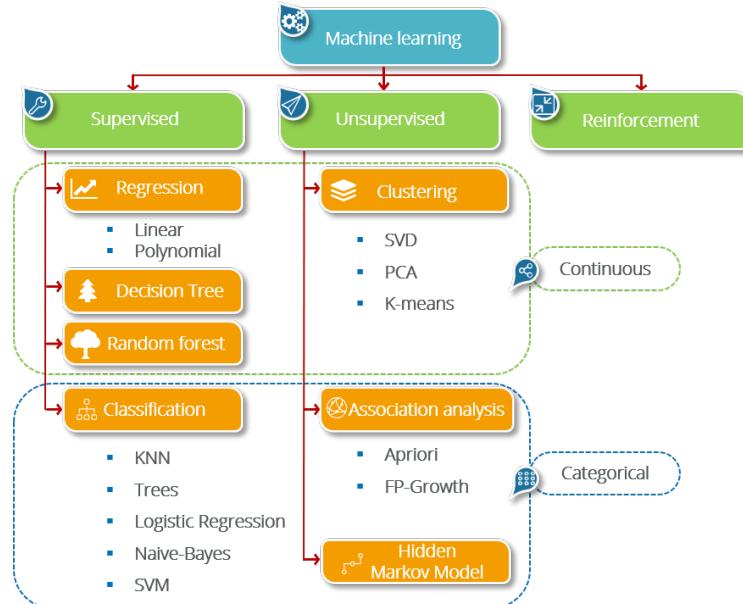


Figure 1.4 Types of machine learning

Supervised Learning: More Control, Less Bias

Supervised machine learning algorithms apply what has been learned in the past to new data using labelled examples to predict future events. By analysing a known training dataset, the learning algorithm produces an inferred function to predict output values. The system can provide targets for any new input after sufficient training. It can also compare its output with the correct, intended output to find errors and modify the model accordingly.

Unsupervised Learning: Speed and Scale

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. At no point does the system know the correct output with certainty. Instead, it draws inferences from datasets as to what the output should be.

Reinforcement Learning: Rewards Outcomes

Reinforcement machine learning algorithms are a learning method that interacts with its environment by producing actions and discovering errors or rewards. The most relevant characteristics of reinforcement learning are trial and error search and delayed reward. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context to maximize its performance. Simple reward feedback — known as the reinforcement signal — is required for the agent to learn which action is best.

1.2 PROJECT OBJECTIVES

The objective of this project is to develop a system based on machine learning algorithms that can accurately predict multiple diseases using patient symptoms as input. The proposed system aims to address the limitations of existing machine learning applications for health analytics, which typically focus on only one disease. The project focuses on predicting diabetes and heart disease as a starting point, using a support vector machine (SVM) as the backbone of the computational diagnostic tool for accurate prediction of heart disease risk levels. The model is also expected to help differentiate patients, so that evidence-based decisions can be made and patients can be treated appropriately. Additionally, bagging and boosting methods using DTB algorithms were

applied to experimental data to predict early diabetes risk, with random forest classifications chosen for the bagging method. The project's long-term goal is to expand to other diseases such as skin diseases, fever analysis, and many others. By analyzing all disease-causing parameters, the machine learning algorithms used in this project can identify the maximum effect caused by the disease and provide early interventions, which can help many people by monitoring their conditions and taking necessary precautions to increase life expectancy. Overall, the objective of this project is to provide a unified system that can predict multiple diseases accurately and efficiently, which can ultimately improve public health outcomes and reduce healthcare costs.

CHAPTER 2

LITERATURE SURVEY

Title: An Approach to detect multiple diseases using machine learning algorithm

Authors: Indukuri Mohit, K.Santhosh Kumar, Avula Uday Kumar Reddy, Badhagouni Suresh Kumar

There are multiple techniques in machine learning that can in a variety of industries, do predictive analytics on large amounts of data. Predictive analytics in healthcare is a difficult endeavour, but it can eventually assist practitioners in making timely decisions regarding patients' health and treatment based on massive data. Diseases like Breast cancer, diabetes, and heart- related diseases are causing many deaths globally but most of these deaths are due to the lack of timely check-ups of the diseases. The above problem occurs due to a lack of medical infrastructure and a low ratio of doctors to the population. The statistics clearly show the same, WHO recommended, the ratio of doctors to patients is 1:1000 whereas India's doctor-to- population ratio is 1:1456, this indicates the shortage of doctors. The diseases related to heart, cancer, and diabetes can cause a potential threat to mankind, if not found early. Therefore, early recognition and diagnosis of these diseases can save a lot of lives. This work is all about predicting diseases that are harmful using machine learning classification algorithms. In this work, breast cancer, heart, and diabetes are included. To make this work seamless and usable by the mass public, our team made a medical test web application that makes predictions about various diseases using the concept of machine learning. In this work, our aim to develop a disease-predicting web app that uses the concept of machine learning-based predictions about various diseases like Breast cancer, Diabetes, and Heart diseases.

Title: Multi Disease Prediction Model by using Machine Learning and Flask API

Authors: Akkem Yaganteeswarudu

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. Like one analysis if for diabetes analysis, one for cancer analysis, one for skin diseases like that. There is no common system where one

analysis can perform more than one disease prediction. In this article proposing a system which used to predict multiple diseases by using Flask API. In this article used to analyse Diabetes analysis, Diabetes Retinopathy analysis, Heart disease and breast cancer analysis. Later other diseases like skin diseases, fever analysis and many more diseases can be included. To implement multiple disease analysis used machine learning algorithms, tensorflow and Flask API. Python pickling is used to save the model behaviour and python unpickling is used to load the pickle file whenever required. The importance of this article analysis in while analysing the diseases all the parameters which causes the disease is included so it possible to detect the maximum effects which the disease will cause. For example for diabetes analysis in many existing systems considered few parameters like age, sex, bmi, insulin, glucose, blood pressure, diabetes pedigree function, pregnancies, considered in addition to age, sex, bmi, insulin, glucose, blood pressure, diabetes pedigree function, pregnancies included serum creatinine, potassium, GlasgowComaScale, heart rate/pulse Rate, respiration rate, body temperature, low density lipoprotein (LDL), high density lipoprotein (HDL), TG (Triglycerides).

Final models behaviour will be saved as python pickle file. Flask API is designed. When user accessing this API, the user has to send the parameters of the disease along with disease name. Flask API will invoke the corresponding model and returns the status of the patient. The importance of this analysis to analyse the maximum diseases, so that to monitor the patient's condition and warn the patients in advance to decrease mortality ratio.

Title: Alumni Info-Com Management with Distinct Classification of Data

Authors: Sasikumar R, Haritha B, Borshiya Vincy T, Kamali M

The Alumni Info-Com Management System is able to manage alumni data of a college and provide easy access to the same. Alumni of college stay in touch with their immediate friends and it is hard to stay connected with college mates. Contact between alumni develops business connections and to gain insight in a new field. Current students will be initially given a student login id. Access to the system can help them to seek help in their projects or for placements. This single system can satisfy almost every

requirement of the alumni. Usually, alumni associations are organized in colleges, but may also be organized in a place where the alumni can meet each other. Despite the fact that there are many existing systems in colleges to maintain the alumni information, they are manual and more time consuming to current students to reach out their alumni and maintaining the privacy of the alumni. To overcome these issues, we proposed a web application which allows alumni to update their information and students can connect with them and can view the filtered events posted by alumni and admin through support vector machine algorithm.

Title: Multiple Disease Prediction System

Authors: Ankush Singh, Ashish Yadav, Saloni Shah, Prof. Renuka Nagpure

Machine learning and Artificial Intelligence are playing a huge role in today's world. From self-driving cars to medical fields, we can find them everywhere. The medical industry generates a huge amount of patient data which can be processed in a lot of ways. So, with the help of machine learning, we have created a Prediction System that can detect more than one disease at a time. Many of the existing systems can predict only one disease at a time and that too with lower accuracy. Lower accuracy can seriously put a patient's health in danger. We have considered three diseases for now that are Heart, Liver, and Diabetes and in the future, many more diseases can be added. The user has to enter various parameters of the disease and the system would display the output whether he/she has the disease or not. This project can help a lot of people as one can monitor the persons' condition and take the necessary precautions thus increasing the life expectancy.

Title: Multiple disease prediction using Machine learning algorithms

Authors: K. Arumugam, Mohd Naved, Priyanka P. Shinde, Orlando Leiva-Chauca, Antonio Huaman-Osorio, Tatiana Gonzales-Yanac

Data mining for healthcare is an interdisciplinary field of study that originated in database statistics and is useful in examining the effectiveness of medical therapies. Machine learning and data visualization Diabetes-related heart disease is a kind of heart disease that affects diabetics. Diabetes is a chronic condition that occurs when

the pancreas fails to produce enough insulin or when the body fails to properly use the insulin that is produced. Heart disease, often known as cardiovascular disease, refers to a set of conditions that affect the heart or blood vessels. Despite the fact that various data mining classification algorithms exist for predicting heart disease, there is inadequate data for predicting heart disease in a diabetic individual. Because the decision tree model consistently beat the naive Bayes and support vector machine models, we fine-tuned it for best performance in forecasting the likelihood of heart disease in diabetes individuals.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Naive Bayes

The Naïve Bayes Classifier technique is mainly applicable when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. Naïve Bayes model recognizes the characteristics of patients with heart disease. It shows the probability of each input attribute for the predictable state. Naive Bayes or Bayes' Rule is the basis for many machine learning and data mining methods. The rule (algorithm) is used to create models with predictive capabilities. It provides new ways of exploring and understanding data. Many of existing analysis involved analysing particular disease. When a user wants to analyse diabetes needs to use one analysis and same user wants to analyse heart disease then user has to use one more model. This is a time taking process. And also if any user having more than one disease but in existing system if it is able to predict only one disease then there is a chance of mortality rate increase due to not able to predict the other disease in advance.

Random Forest

Random Forest is essentially an ensemble of unpruned classification trees. It gives excellent performance on a number of practical problems, largely because it is not sensitive to noise in the data set, and it is not subject to overfitting. It works fast, and generally exhibits a substantial performance improvement over many other tree-based algorithms. Random forests are built by combining the predictions of several trees, each of which is trained in isolation. There are three main choices to be made when constructing a random tree.

Active Contour Model (ACM) filter

Active contour models (ACMs) have been widely applied to image segmentation since their introduction. ACMs can obtain closed object contours as segmentation results, which can be conveniently used for shape analysis and recognition. The active contours can utilize various types of prior knowledge, such as image intensity distribution information, boundary shape information, and texture information, to obtain accurate results for object boundaries in image analysis.

3.2 DISADVANTAGES

- One algorithm may work well on a specific dataset while it cannot show a good performance on some others.
- So, selecting a suitable algorithm for a specific dataset is a big challenge in bioinformatics.
- Consequently, selecting good feature selection or classification algorithms is also a big challenge in this field.
- ML/DM algorithms commonly need massive datasets to be trained.
- These datasets must be inclusive and unbiased with high quality.
- Datasets also need time to be collected.
- RF does not improve the classification accuracy.
- It shows a higher computational cost and training time.
- The classification performance is slightly worst.
- All these models cannot be generalized enough to produce quality prediction about tissues diagnosis in the rest of the patients.
- ANN prone to data-overfitting.
- lack of transparency in the “hidden layers” of neural networks

3.3 PROPOSED SYSTEM

In multi disease model prediction, it is possible to predict more than one disease at a time. So user no need to traverse many models to predict the diseases. It will reduce time and also due to predicting multiple diseases at a time there is a chance of reducing mortality rate. The proposed framework employs data mining techniques to detect Chronic diseases early. Machine learning is the process of programming computers to improve their output based on examples or previous data. The study of computer systems that learn from data and experience is known as machine learning. Training and Testing are the two stages of the machine learning algorithm. Prediction of a disease based on the signs and medical history of the patient Machine learning has been a stumbling block for decades .

x2 statistical model for features refinement

A chi-square test is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other.

Support Vector Machine

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables. The SVM is a learning algorithm for classification. It tries to find the optimal separating hyper plane such that the expected classification error for unseen patterns is minimized. For linearly non-separable data the input is mapped to high-dimensional feature space where they can be separated by a hyper plane. This projection into high dimensional feature space is efficiently performed by using kernels. More precisely, given a set of training samples and the corresponding decision values -1, 1 the SVM aims to find the best separating hyper

plane given by the equation $WTx+b$ that maximizes the distance between the two classes. An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

3.4 ADVANTAGES:

- It may result in early detection that leads to a decrease in mortality rate.
- ML can provide a priori probability of disease and use this probability to selectively target patients for angiography. This can save in cost and time for other patients. The side effects of angiography are also eliminated for them.
- Using DL can extract hidden patterns in the collected data. This may lead to finding new methods for early detection in many diseases like CAD.
- This is a novel method that builds on current research to derive quick and precise diagnostics.
- The method significantly outperforms other published research in this area due to its superior accuracy.
- intention is to accurately classify the presence of diseases.

CHAPTER 4

SYSTEM ENVIRONMENT

4.1 HARDWARE SPECIFICATION

- Processors:
 - Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
- Disk space: 320 GB
- Operating systems: Windows® 10, macOS*, and Linux*

4.2 SOFTWARE SPECIFICATION

- Server Side : Python 3.7.4(64-bit) or (32-bit)
- Client Side : HTML, CSS, Bootstrap
- IDE : Visual Studio Code
- Back end : MySQL 5.
- Server : Wampserver
- OS : Windows 10 64 –bit or Ubuntu 18.04 LTS “Bionic Beaver”

4.3 PROJECT PROBLEM STATEMENT

To create a machine learning model and to integrate it into a web application in such a way that any user can able to predict the multiple disease under a same application. Besides this admin can able able to train the model with the updated dataset.

4.4 SOFTWARE DESCRIPTION

Python 3.7.4

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public

License (GPL). This tutorial gives enough understanding on Python programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQtetc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more..

Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON,

Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda's library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Visualization is the central part of Seaborn which helps in exploration and understanding of data. Seaborn offers the following functionalities:

- Dataset oriented API to determine the relationship between variables.

- Automatic estimation and plotting of linear regression plots.
- It supports high-level abstractions for multi-plot grids.
- Visualizing univariate and bivariate distribution.

Scikit Learn

Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license. Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

MySQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database. MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish

company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

The Apache Web Server

In addition to PHP, MySQL, JavaScript, and CSS, there's actually a fifth hero in the dynamic Web: the web server. In the case of this book, that means the Apache web server. We've discussed a little of what a web server does during the HTTP server/client exchange, but it actually does much more behind the scenes. For example, Apache doesn't serve up just HTML files—it handles a wide range of files, from images and Flash files to MP3 audio files, RSS (Really Simple Syndication) feeds, and more. Each element a web client encounters in an HTML page is also requested from the server, which then serves it up. But these objects don't have to be static files, such as GIF images. They can all be generated by programs such as PHP scripts. That's right: PHP can even create images and other files for you, either on the fly or in advance to serve up later. To do this, you normally have modules either precompiled into Apache or PHP or called up at runtime. One such module is the GD library (short for Graphics Draw), which PHP uses to create and handle graphics. Apache also supports a huge range of modules of its own. In addition to the PHP module, the most important for your purposes as a web programmer are the modules that handle security. Other examples are the Rewrite module, which enables the web server to handle a varying range of URL types and rewrite them to its own internal requirements, and the Proxy module, which you can use to serve up often-requested pages from a cache to ease the load on the server. Later in the book, you'll see how to actually use some of these modules to enhance the features provided by the core technologies we cover. About Open Source Whether or not being open source is the reason these technologies are so popular has often been debated, but PHP, MySQL, and Apache are the three most commonly used tools in their categories.

WampServer

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database. WAMP Server is a reliable web development software program that lets you create web apps with MySQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription.

Bootstrap 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers - Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project. Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Flask

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application.

CHAPTER 5

SYSTEM DESIGN

5.1 MULTI DISEASE UI DIAGNOSIS MODEL

In this module we developed UI-based interface designed to be used by doctors, nurses and hospital administration staff, Patient – untrained in data science – to predict, forecast and recommendation System. In order to facilitate operations, a simple user interface was designed using the Flask Framework. This interface can accurately infer the edge contour, assisting users in segmentation. The module should be able to collect data from a user's blood sugar readings, insulin dosages, and other relevant health metrics. This module developing the user interface, data collection and analysis, and reporting features.

5.2 DIABETES MODEL TRAIN AND BUILD

Dataset Collection

The datasets consist of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **BloodPressure:** Diastolic blood pressure (mm Hg)
- **SkinThickness:** Triceps skin fold thickness (mm)
- **Insulin:** 2-Hour serum insulin (mu U/ml)
- **BMI:** Body mass index (weight in kg/(height in m)²)
- **DiabetesPedigreeFunction:** Diabetes pedigree function
- **Age:** Age (years)
- **Outcome:** Class variable (0 or 1)

Number of Observation Units: 768

Variable Number: 9

Data Set Acquisition

Diabetes Lab Test Sample Data

In this module, we load the CSV file format Diabetes Lab Test Sample Data to our web screening tool. The Diabetes Disease Dataset consists of 9 features for 400 people7 features are numerical, while 8 are categorical which is stored in CSV file format. The dataset contains 768 observables with eight feature variables and one target variable. RangeIndex: 768 entries, 0 to 767. Data columns (total 9 columns).

Table 5.1: Diabetes dataset model

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

Dataset Pre-processing

Before feeding the data into the Prediction model, following data cleaning and preprocessing steps are performed

- Checking null values and filling using forward fill method
- Converting data into different cases
- Standardizing the data using mean and standard deviation
- Splitting the dataset into training and testing sets

1. Formatting: The data set is used for implementation is taken from UCI repository, it may contain certain attributes whose names are not clear in the (dataset name) also contain certain unrelated attribute which is not useful for the greater performance of proposed work . An attribute name as “Thal” has been removed from dataset by using following command in R, Dataset\$Thal<-Null

2. Cleaning: This part of pre-processing belongs to remove or fixing of missing out entry in the data frame. Row containing these incomplete columned to be removed also for

removing certain redundant entries in data frame this step is recommend

3. Sampling: Sampling is also done on the dataset to enhance the performance of the algorithm on sample data set may lead algorithm to take longer time.

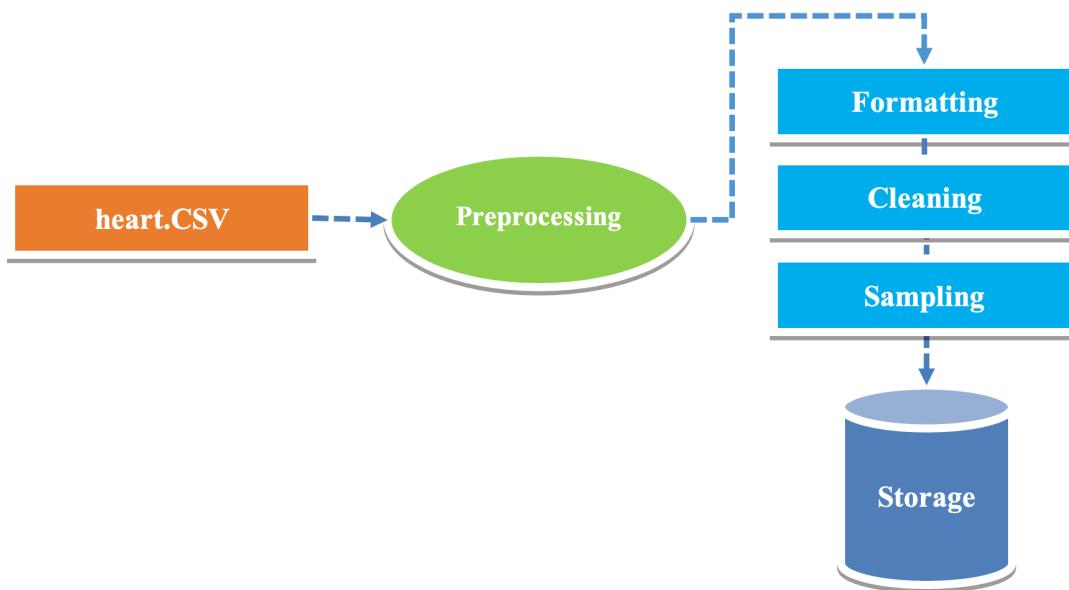


Figure 5.1 Process of Diabetes prediction

Clustering

The pre-processed data is clustered using clustering algorithms like K-Nearest Neighbour (KNN) to cluster relevant data in database. KNN algorithm provides a simplest way to classify the given dataset using k no. of clusters. We define k centroids, one for each cluster. In this algorithm we choose k no. of clusters (kfixed), find the centroids and distance between the objects to centroids, then we group them based on minimum distance loop is generated and iterated until there are no more changes to be done to change the centroids position after every loop. The main aim of this algorithm is to minimize the objective function

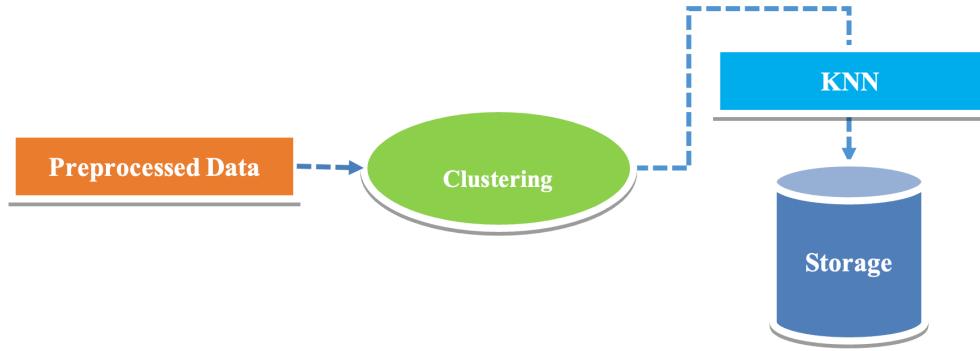


Figure 5.2 Data preprocessing in Diabetes prediction

Feature Selection

- the effect of selected factors on diabetes using logistic regression. It shows that age, education, BMI, systolic BP, diastolic BP, direct cholesterol, and total cholesterol are statistically significant factors for diabetes disease at 5% level of significance and the rest of the factors are insignificant. These seven factors are used for ML-based system to classify and predict of diabetes disease.
- Effect of selected factors on the diabetes using logistic regression

Table 5.2: Effect of Selected factors on diabetes

Factors	OR	p value	95% CI for OR	
			Lower	Upper
Age (years)	1.055	< 0.001	1.047	1.064
Gender				
Female (ref)	1.000			
Male	1.270	0.086	0.967	1.670

Factors	OR	p value	95% CI for OR	
			Lower	Upper
Race				
<u>Black (ref)</u>	<u>1.000</u>			
Hispanic	0.746	0.206	0.470	1.167
Mexican	0.858	0.465	0.567	1.290
Other	1.109	0.623	0.732	1.670
White	0.469	0.263	0.358	0.619
Education				
8th grade (ref)	1.000			
9–11th grade	0.577	0.005	0.394	0.844
College grad	0.641	0.019	0.441	0.932
High school	0.746	0.010	0.526	1.060
Some college	0.786	0.030	0.559	1.112
Marital status				

Data Visualization

The correlation between each column are visualized using heatmap. From the output, the lighter colors indicate more correlation. We notice the correlation between pairs of features, like age and pregnancies, or BMI and skin thickness, etc. In feature selection, irrelevant features are eliminated and the most important or relevant features are applied to the network. Thus, if we supply all features to SVM, some features may be noisy and if they are learned in the training process, they may degrade generalization of the network although the network will show good performance on the training data. That is why large number of features are also considered one of the main causes of over fitting.

Thus, searching out optimal subset of features by eliminating noisy features can help SVM to show good performance on both training and testing data. In this module, we use X^2 statistical model to eliminate irrelevant features.

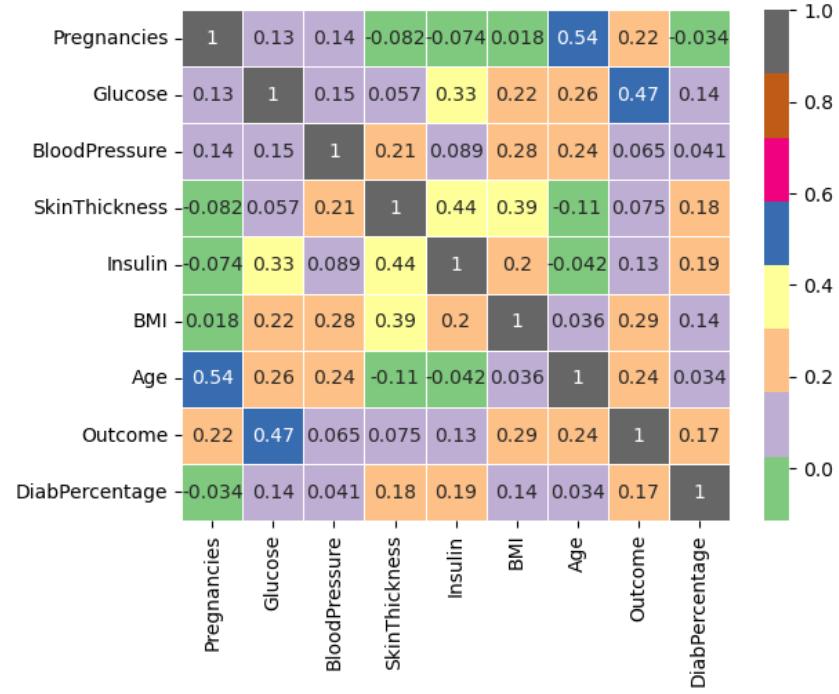


Figure 5.3 Confusion matrix of diabetes prediction

In feature selection, irrelevant features are eliminated and the most important or relevant features are applied to the network. Thus, if we supply all features to SVM, some features may be noisy and if they are learned in the training process, they may degrade generalization of the network although the network will show good performance on the training data. That is why large number of features are also considered one of the main causes of over fitting. Thus, searching out optimal subset of features by eliminating noisy features can help SVM to show good performance on both training and testing data. In this module, we use X^2 statistical model to eliminate irrelevant features.

Diabetes Disease Classification

In this module, we used Ensemble learning as the classifiers. Ensemble learning is a successful paradigm of machine learning which merges a set of learners instead of using a single learner to predict unknown target attributes. In this structure, all output values obtained from each learner are combined by using a voting mechanism to make final class label prediction. The main goal of ensemble learning is to form a strong classifier consisting of multiple learners to obtain more accurate classification results. Ensemble learning techniques are mainly group under four categories: bagging, boosting, voting, and stacking. In this study, bagging and boosting approaches which are widely preferred ensemble learning methods are implemented on the experimental data and compared with each other.

Bagging

Bagging, for bootstrap aggregation, is a frequently used ensemble technique that creates multiple training sets using a bootstrap method. In the bootstrap method, multiple training sets are constructed by choosing random and repeatable samples from the original dataset. After creating training subsets, multiple learning models are generated by training each learner in the ensemble structure with these subsets. Last, the predictions obtained from each model are aggregated to obtain the final decision.

- Random forest: the random forest algorithm, proposed by Leo Breiman and Adele Cutler, is a bagging algorithm which generates a forest with multiple decision trees. It classifies the unknown target attribute of a new sample by taking a majority vote over all the decision trees in the forest.

Boosting

In the boosting method, the aim is to acquire a strong classifier from a set of weak learners. According to this approach, the samples in the training set are reweighted during the learning phase to generate multiple learners.

- AdaBoost: AdaBoost (adaptive boosting), introduced by Yoav Freund and Robert Schapire, is a boosting algorithm that earned Gödel Prize in 2013. In this algorithm, the weight of misclassified samples in the training set is increased in

each iteration. Thus, the chance of selecting misclassified samples for the training set is increased and more samples are classified correctly.

SVM (Support Vector Machine) is a class of FFN (Feed Forward Networks) like outspread premise Function systems, established by vapnik. Support vector machine can be used for design order additionally nonlinear regression. SVM is nearly accurate execution of the arrangement of compositional hazard decrease. This standard is relying upon the truth of the blunder rate of a learning machine on the test information is compassed by the expansion of the preparation mistake rate and term that relies upon the vapnik chervonenkis (VC) measurement. The SVM can offer fine speculation accomplishment on design order debate. The SVM algorithm was developed for prediction by using an ε -insensitive loss function

Prediction

Diabetes Predictor

The model should learn to predict instances as either having diabetes disease or not having it, based on the input features.

- **Input Diabetes Target Value**

In this module the user of the system input the diabetes disease attributes values to the system through web interface.

- **Prediction**

In this module, the eigen vectors is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, eigen space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric.

The eigen distance between points p and q is the length of the line segment connecting them: $p \cdot q$. In Cartesian coordinates, if $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ are two points in Euclidean n space, then the distance from p to q, or from q to p is given by the following

heterogeneous value difference metric.

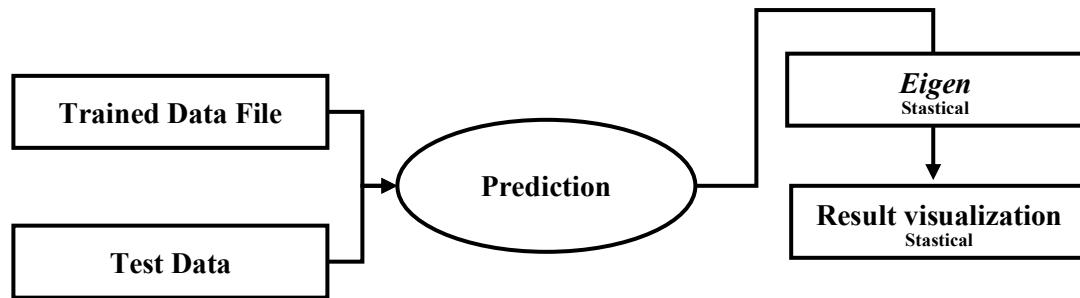


Figure 5.4 Prediction model for diabetes.

In this module, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric

5.2.6 END USER MODULE

1 Patient or Doctor or User

- Register
- Login
- Add Biographic and Demographic Information
- Test Diabetes
- Receive recommendation
- Logout

2. Admin

- Login
- Import Dataset
- Train the Model
- Customize Recommendation

- Track user accounts
- Report
- Logout

Performance Analysis

Valuate the classification model: Evaluate the performance of the classification model on the testing set using metrics such as accuracy, precision, recall, and F1 score. You can also use cross-validation to estimate the generalization performance of the model.

If the actual value of the target in the dataset is True and the classifier predicts it as such, then the prediction is a True Positive (TP). On the contrary, if the classifier predicts it as False, then the prediction is a False Negative (FN). Similarly, if the actual value of the target in the dataset is False and the classifier predicts it as such, then the prediction is True Negative (TN). On the contrary, if the classifier predicts it as True, then the prediction is False Positive (FP).

Finding out how the developed predictive model performs becomes easy with the help of the confusion matrix.

The following metrics are used to evaluate the proposed model.

Accuracy (Acc) is the percentage of the correct predictions that a classifier has made compared with the actual values of the target in the testing phase.

$$Acc = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100\%.$$

Sensitivity (Sens) gives information about the percentage of True Positives that are correctly classified during the test.

$$Sens = \frac{TP}{(TP + FN)} * 100\%.$$

Specificity (Spec) gives information about of True Negatives that are correctly classified during the test.

$$\text{Spec} = \frac{\text{TN}}{(\text{TN} + \text{FP})} * 100\%.$$

Precision (Pre) is the percentage of instances that a classifier has labelled as positive with respect to the total predictive positives (the exactness of a classifier).

$$\text{Pre} = \frac{\text{TP}}{(\text{TP} + \text{FP})} * 100\%.$$

F1-score shows the harmonic mean of precision and recall.

$$F1 - \text{score} = \frac{2 * \text{TP}}{(2 * \text{TP} + \text{FN} + \text{FP})} * 100\%.$$

5.3 HEART DISEASE MODEL TRAIN AND BUILD

Dataset Collection

Diabetes analysis initially Pima Indian Diabetes Dataset, the data set which was acquired from a hospital in Frankfurt, Germany are used. For diabetic retinopathy over 150 GB image data from the UCI machine learning repository are used. For heart disease analysis Cleveland, Hungarian and Switzerland heart disease patient's data sets are used. And for cancer disease prediction used Heart disease Data Set which is available in machine learning repository. In the current analysis in addition to those data sets used other live data sets by visiting corresponding hospitals. The importance of this analysis is by consulting the doctors collected the necessary parameters which will cause the disease and also due to that disease any other disease likely to occur. After doing this analysis there is a chance of reducing mortality ratio because if can able to predict the maximum diseases chances of occurring so that can warn the patients in advance for treatment

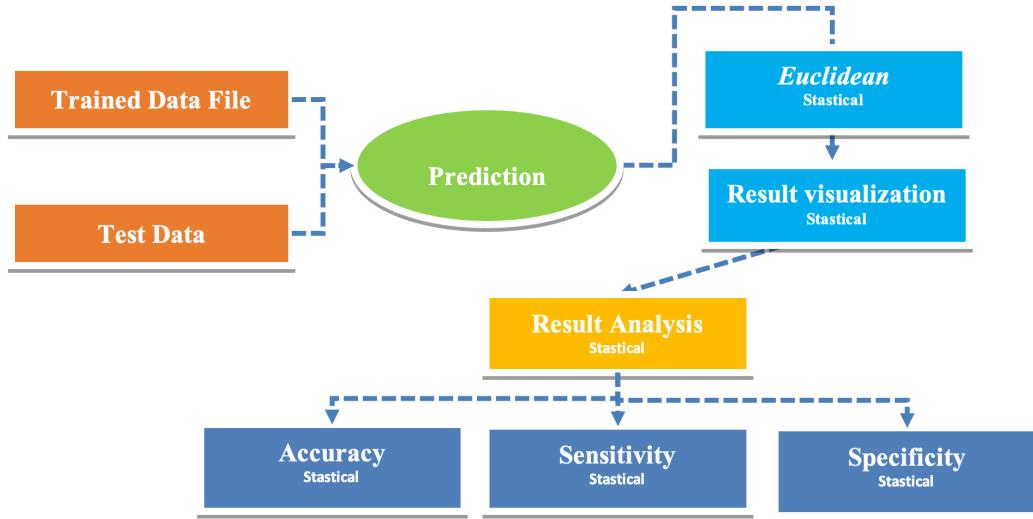


Figure 5.5 Workflow for Heart disease prediction

Table 5.3: Database Description

Field	Description	Range and Values
Age	Age of the patient	0-100 in years
Sex	Gender of the patient	0-1 (1:Male 0:Female)
Chest Pain	Type of chest pain	1-4 (1: Typical Angina, 2: Atypical Angina, 3: Non-anginal, 4: Asymptotic)
Resting Blood Pressure	Blood pressure during rest	mm Hg
Cholesterol	Serum Cholesterol	mg / dl
Fasting Blood Sugar	Blood sugar content before food intake if >120 mg/dl	0-1 (0: False, 1: True)
ECG	Resting Electrocardiographic results	0-1 (0: Normal, 1: Having ST-T wave)
Max Heart Rate	Maximum heart beat rate.	Beats/min
Exercise Induced Angina	Has pain been induced by	0-1 (0: No, 1: Yes)

	exercise	
Old Peak	ST depression induced by exercise relative to rest	0-4
Slope of Peak Exercise	Slope of the peak exercise ST segment	1-3 (1: Up sloping, 2: Flat, 3: Down sloping)
Ca	Number of vessels colored by fluoroscopy	0-3
Thal		3- normal 6-Fixed Defect 7- Reversible Defect
Num	Diagnostics of Heart Disease	0-1 (0: <50% Narrowing 1: >50% Narrowing)

The dataset consists of 303 individuals data. There are 14 columns in the dataset, which are described below.

1. **Age:** displays the age of the individual.
2. **Sex:** displays the gender of the individual using the following format :
 1 = male
 0 = female
3. **Chest-pain type:** displays the type of chest-pain experienced by the individual using the following format :
 1 = typical angina
 2 = atypical angina
 3 = non — anginal pain
 4 = asymptotic
4. **Resting Blood Pressure:** displays the resting blood pressure value of an individual in mmHg (unit)
5. **Serum Cholesterol:** displays the serum cholesterol in mg/dl (unit)

6. **Fasting Blood Sugar**: compares the fasting blood sugar value of an individual with 120mg/dl.
If fasting blood sugar > 120mg/dl then : 1 (true)
else : 0 (false)
7. **Resting ECG** : displays resting electrocardiographic results
0 = normal
1 = having ST-T wave abnormality
2 = left ventricular hypertrophy
8. **Max heart rate achieved** : displays the max heart rate achieved by an individual.
9. **Exercise induced angina** :
1 = yes
0 = no
10. **ST depression induced by exercise relative to rest**: displays the value which is an integer or float.
11. **Peak exercise ST segment** :
1 = upsloping
2 = flat
3 = downsloping
12. **Number of major vessels (0–3) colored by fluoroscopy** : displays the value as integer or float.
13. **Thal** : displays the thalassemia :
3 = normal
6 = fixed defect
7 = reversible defect
14. **Diagnosis of heart disease** : Displays whether the individual is suffering from heart disease or not :
0 = absence
1, 2, 3, 4 = present.

Why these parameters:

In the actual dataset, we had 76 features but for our study, we chose only the above 14 because:

1. **Age:** Age is the most important risk factor in developing cardiovascular or heart diseases, with approximately a tripling of risk with each decade of life. Coronary fatty streaks can begin to form in adolescence. It is estimated that 82 percent of people who die of coronary heart disease are 65 and older. Simultaneously, the risk of stroke doubles every decade after age 55.
2. **Sex:** Men are at greater risk of heart disease than pre-menopausal women. Once past menopause, it has been argued that a woman's risk is similar to a man's although more recent data from the WHO and UN disputes this. If a female has diabetes, she is more likely to develop heart disease than a male with diabetes.
3. **Angina (Chest Pain):** Angina is chest pain or discomfort caused when your heart muscle doesn't get enough oxygen-rich blood. It may feel like pressure or squeezing in your chest. The discomfort also can occur in your shoulders, arms, neck, jaw, or back. Angina pain may even feel like indigestion.
4. **Resting Blood Pressure:** Over time, high blood pressure can damage arteries that feed your heart. High blood pressure that occurs with other conditions, such as obesity, high cholesterol or diabetes, increases your risk even more.
5. **Serum Cholesterol:** A high level of low-density lipoprotein (LDL) cholesterol (the "bad" cholesterol) is most likely to narrow arteries. A high level of triglycerides, a type of blood fat related to your diet, also ups your risk of a heart attack. However, a high level of high-density lipoprotein (HDL) cholesterol (the "good" cholesterol) lowers your risk of a heart attack.
6. **Fasting Blood Sugar:** Not producing enough of a hormone secreted by your pancreas (insulin) or not responding to insulin properly causes your body's blood sugar levels to rise, increasing your risk of a heart attack.
7. **Resting ECG:** For people at low risk of cardiovascular disease, the USPSTF concludes with moderate certainty that the potential harms of screening with resting

or exercise ECG equal or exceed the potential benefits. For people at intermediate to high risk, current evidence is insufficient to assess the balance of benefits and harms of screening.

8. **Max heart rate achieved:** The increase in cardiovascular risk, associated with the acceleration of heart rate, was comparable to the increase in risk observed with high blood pressure. It has been shown that an increase in heart rate by 10 beats per minute was associated with an increase in the risk of cardiac death by at least 20%, and this increase in the risk is similar to the one observed with an increase in systolic blood pressure by 10 mm Hg.
9. **Exercise induced angina:** The pain or discomfort associated with angina usually feels tight, gripping or squeezing, and can vary from mild to severe. Angina is usually felt in the center of your chest but may spread to either or both of your shoulders, or your back, neck, jaw or arm. It can even be felt in your hands.
 - o Types of Angina
 - a. Stable Angina / Angina Pectoris
 - b. Unstable Angina
 - c. Variant (Prinzmetal) Angina
 - d. Microvascular Angina.
10. **Peak exercise ST segment:** A treadmill ECG stress test is considered abnormal when there is a horizontal or down-sloping ST-segment depression ≥ 1 mm at 60–80 ms after the J point. Exercise ECGs with up-sloping ST-segment depressions are typically reported as an ‘equivocal’ test. In general, the occurrence of horizontal or down-sloping ST-segment depression at a lower workload (calculated in METs) or heart rate indicates a worse prognosis and higher likelihood of multi-vessel disease. The duration of ST-segment depression is also important, as prolonged recovery after peak stress is consistent with a positive treadmill ECG stress test.

System Architecture:

The system architecture for predicting heart disease using SVM typically includes data collection from trustworthy sources, pre-processing to remove any inconsistencies or missing values, feature selection to identify the most relevant input variables, SVM model training, and model performance evaluation. The input data is pre-

processed and translated to numerical values before feature selection is performed to identify the most significant variables influencing heart disease outcomes. The selected features are utilised to train an SVM model, which can determine whether or not a patient has heart disease. The model is assessed using performance metrics such as accuracy, precision, and recall, and iterative testing and training can be done to further refine it.

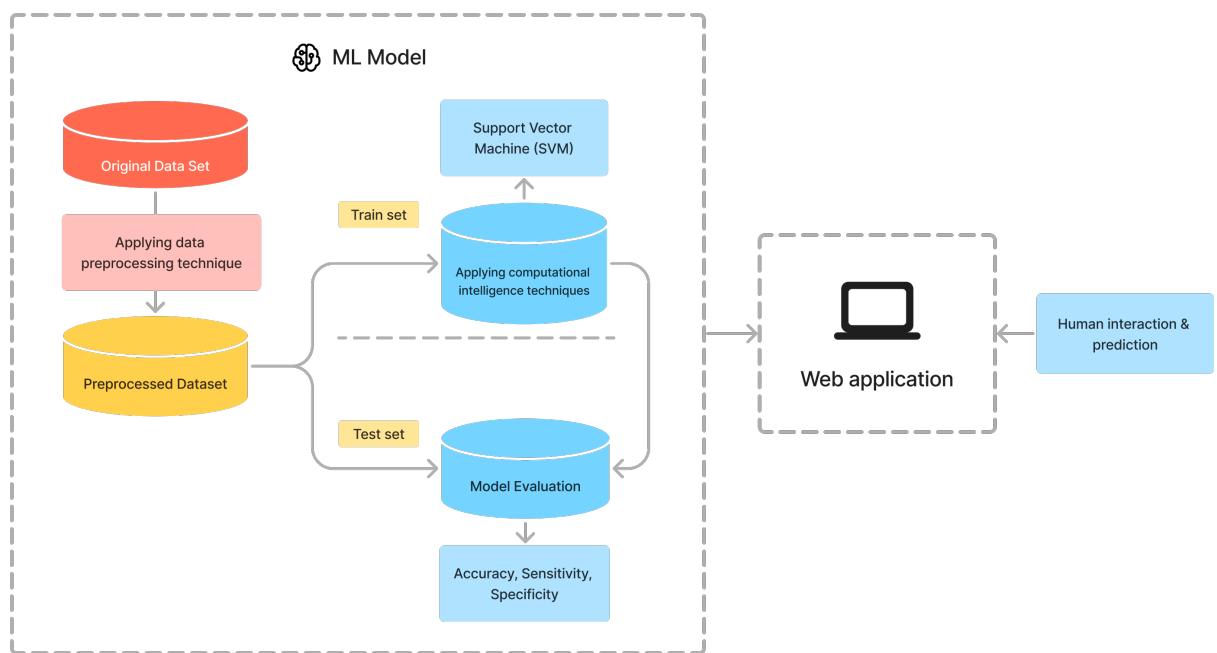


Figure 5.6 System architecture for Heart disease prediction

Data preparation

There are several machine learning algorithms available to determine if a person has heart disease or not. In this study, we apply the Support Vector Machine technique to forecast people's cardiac illnesses. We use the SVM method for the prediction process since it is more accurate than other machine learning techniques. The accuracy of the algorithm is depicted graphically. The graphical results of the provided data are displayed below.

As per the standards training set and testing set are prepared. By using

Scikit learn train_test_split method the data has been divided into two sets one is of 70% of the dataset for training and the other is of 30% of the dataset for testing. Example: heart_feature_training, heart_feature_testing, heart_label_training, heart_label_testing = train_test_split (heart_features, heart_label, test_size=0.3, random_state=0).

Table 5.4 : Heart disease dataset model

age	sex	cp	tresp	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
67	0	3	115	564	0	2	160	0	1.6	2	0	7	0
65	0	3	140	417	1	2	157	0	0.8	1	1	3	0
56	0	4	134	409	0	2	150	1	1.9	2	2	7	2
63	0	4	150	407	0	2	154	0	4	2	3	7	4
62	0	4	140	394	0	2	157	0	1.2	2	0	3	0
65	0	3	160	360	0	2	151	0	0.8	1	0	3	0
57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
55	1	4	132	353	0	0	132	1	1.2	2	1	7	3
55	0	2	132	342	0	0	166	0	1.2	1	0	3	0
43	0	4	132	341	1	2	136	1	3	2	0	7	2
58	0	3	120	340	0	0	172	0	0	1	0	3	0
57	1	4	110	335	0	0	143	1	3	2	1	7	2
64	1	3	140	335	0	0	158	0	0	1	0	3	1
61	0	4	130	330	0	2	169	0	0	1	0	3	1
63	1	4	130	330	1	2	132	1	1.8	1	3	7	3
55	0	4	180	327	0	1	117	1	3.4	2	0	3	2
59	1	4	170	326	0	2	140	1	3.4	3	0	7	2

Machine Learning

SVM works by generating a hyperplane that divides data into classes. The hyperplane is chosen to optimise the margin between the classes. SVM may also employ the kernel method to convert input data into a higher dimensional space, perhaps improving class separation.

The dataset for heart disease prediction using SVM is separated into training and testing sets. The SVM algorithm is trained on the labelled data using the

training set, where each patient's characteristics such as age, gender, blood pressure, cholesterol levels, and other parameters are used to predict the chance of heart disease. The SVM model's hyperparameters are modified using approaches such as grid search, which aids in optimising the model's performance.

After trained, the model is used to generate predictions on the testing set. Metrics such as accuracy, precision, recall, and F1 score are used to assess the SVM model's performance. These metrics give information about the model's ability to properly categorise patients as having or not having heart disease.

Model Analysis:

Once the model is ready then we have to analyze the model against precision, recall, accuracy and specificity. The below chart show the comparison of SVM with all other algorithms as well.

Table 5.5 : Precision and Recall

Class Label	Precision	Recall	Support
Yes	0.92	0.9	93
No	0.91	0.89	72
Average	0.91	0.89	165

Table 5.6: Comparison of various machine learning algorithms for heart disease

Algorithm	Accuracy	Sensitivity	Specificity
ANN	85.30%	83.75%	75.73%
Navie Bayes	81.14%	61.03%	70.11%
RIPPER	81.08%	86.25%	75.82%
Decision Support	79.05%	83.12%	74.26%
SVM	85.97%	90.10%	77.20%
KNN	84.12%	56.87%	71.21%

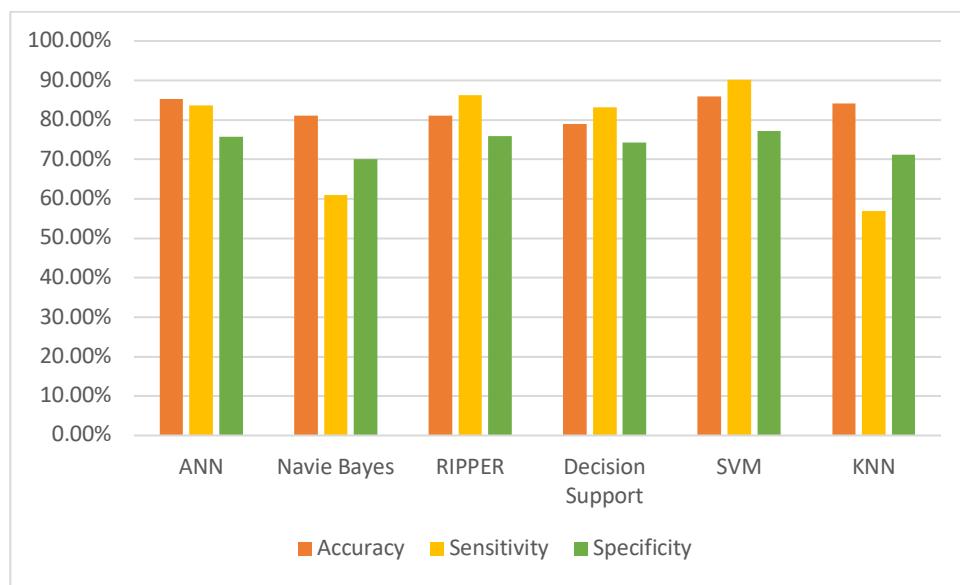


Figure 5.7: Comparison of various machine learning algorithms for prediction of heart disease.

CHAPTER 6

SYSTEM TESTING

In this phase of methodology, testing was carried out on the several application modules. Different kind of testing was done on the modules which are described in the following sections. Generally, tests were done against functional and non-functional requirements of the application following the test cases. Testing the application again and again helped it to become a reliable and stable system.

6.1 USABILITY TESTING

This was done to determine the usability of the application that was developed. This helped to check whether the application would be easy to use or what pitfalls would the users come through. This was used to determine whether the application is user friendly. It was used to ascertain whether a new user can easily understand the application even before interacting with it so much. The major things checked were: the system flow from one page to another, whether the entry points, icons and words used were functional, visible and easily understood by user.

6.2 FUNCTIONAL TESTING:

Functional Testing is defined as a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application. Functional tests were done based on different kind of features and modules of the application and observed that whether the features are met actual project objectives and the modules are hundred percent functional. Functional tests, as shown in the following Table-1 to Table-5, were done based on use cases to determine success or failure of the system implementation and design. For each use case, testing measures were set with results being considered successful or unsuccessful. Below are the tables which are showing some of the major test cases along with their respective test results.

6.3 INTEGRATION TESTING:

Integration testing is often done in concert with unit testing. Through integration testing, QA professionals verify that individual modules of code work together properly as a group. Many modern applications run on microservices, self-contained applications that are designed to handle a specific task. These microservices must be able to communicate with each other, or the application won't work as intended. Through integration testing, testers ensure these components operate and communicate together seamlessly.

6.4 TEST CASES:

6.4.1 SIGNUP/REGISTRATION TEST CASE

Table 8.1: Signup page testing

Identifier	Test Case-1
Test Case	Signup
Description	To register new account in the application.
Pre-requisite	1) Username and email must not exist previously.
Test procedure	1) Select Sign Up from the menu. 2) Fill in username, email, and password and retype password accordingly. 3) Click on Sign Up button
Expected Result	1) User can register to the application successfully. 2) Username, email and password stored in the user table in the database.
Pass/Fail	Pass

6.4.2 LOGIN TEST CASE

Table 6.2: Login page testing

Identifier	Test Case-2
Test Case	Login
Description	To login new account in the application
Pre-requisite	1) Registration must be done previously.
Test procedure	1) Select Log In from themenu. 2) Fill in username and passwordaccordingly. 3) Click on Log In button.
Expected Result	1) User can login to the applicationsuccessfully. 2) User should access the application features which areallowed
Pass/Fail	Pass

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

In conclusion, the proposed system for predicting multiple diseases using machine learning algorithms offers a promising solution for enhancing healthcare outcomes. With the ability to predict both diabetes and heart disease, the system offers a unified approach to disease prediction that is currently lacking in many existing applications. The use of a support vector machine (SVM) in predicting heart disease risk levels is particularly noteworthy, as it offers a reliable classification method that can predict medication adherence in CVD patients, enabling evidence-based decision-making. The application of DTB algorithms in predicting early diabetes risk also shows promising results, with high precision, recall, accuracy, and F1 score metrics achieved. The availability of a web application that allows doctors to train their own models further enhances the practicality of this system. Overall, the system has the potential to improve the health outcomes of patients by enabling early diagnosis and appropriate interventions, ultimately leading to increased life expectancy.

7.2 FUTURE SCOPE

In the future we can add more diseases in the existing API. We can try to improve the accuracy of prediction to decrease the mortality rate. Try to make the system user-friendly and provide a chatbot for normal queries. In future research, we would investigate other enhanced methods that would further raise the diagnostic accuracy of the deep learning model by utilizing deep learning based on morphologic class pattern predictions in order to further enhance the performances of the SVM models for heart disease diagnoses in patients worldwide.

CHAPTER 8

APPENDIX

8.1 SOURCE CODE

index.html

```
# main.py
import os
import base64
import io
import math
from flask import Flask, render_template, Response, redirect, request, session, abort,
url_for
import mysql.connector
import hashlib
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.feature_selection import VarianceThreshold
constant_filter = VarianceThreshold(threshold=0)
import csv
import codecs
from flask import (jsonify, request)
mydb = mysql.connector.connect(
host="localhost",
user="root",
password="",
charset="utf8",
database="multi_disease"
```

```

)
app = Flask(__name__)
##session key
app.secret_key = 'abcdef
#####
UPLOAD_FOLDER = 'upload'
ALLOWED_EXTENSIONS = { 'csv'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
#####
@app.route('/', methods=['GET', 'POST'])
def index():
msg=""
if request.method=='POST':
uname=request.form['uname']
pwd=request.form['pass']
cursor = mydb.cursor()
cursor.execute('SELECT * FROM patient WHERE uname = %s AND pass = %s',
(uname, pwd))
account = cursor.fetchone()
if account:
session['username'] = uname
return redirect(url_for('pat_home'))
else:
msg = 'Incorrect username/password! or access not provided'
return render_template('index.html',msg=msg)
@app.route('/login', methods=['GET', 'POST'])
def login():
msg=""
```

```

if request.method=='POST':
    uname=request.form['uname']
    pwd=request.form['pass']
    cursor = mydb.cursor()
    cursor.execute('SELECT * FROM admin WHERE username = %s AND password = %s',
    (uname, pwd))
    account = cursor.fetchone()
    if account:
        session['username'] = uname
        return redirect(url_for('admin'))
    else:
        msg = 'Incorrect username/password! or access not provided'
        return render_template('login.html',msg=msg)
@app.route('/forgot', methods=['GET', 'POST'])
def forgot():
    msg=""
    if request.method=='POST':
        uname=request.form['uname']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM patient WHERE uname = %s', (uname, ))
        account = cursor.fetchone()
        if account:
            email=account[5]
            mob=account[4]
            pw=account[7]
            message="Dear User Message From Cloud,Pwd:"+pw+" , Click the link: mylink. By
SMSWAY IOTCLD"
            #params = urllib.parse.urlencode({'token': 'b81edee36bcef4ddbaa6ef535f8db03e', 'credit':

```

```

2,      'sender':      'IOTCLD',      'message':message,      'number':str(mob),
'templateid':'1207162443831712783'})
#url = "http://pay4sms.in/sendsms/?%os" % params
#with urllib.request.urlopen(url) as f:
#  print(f.read().decode('utf-8'))
#  print("sent"+str(mob))
msg="Password has sent.."
else:
msg = 'Incorrect username'
return render_template('forgot.html',msg=msg)
@app.route('/forgot2', methods=['GET', 'POST'])
def forgot2():
msg=""
if request.method=='POST':
uname=request.form['uname']
cursor = mydb.cursor()
cursor.execute('SELECT * FROM doctor WHERE uname = %s', (uname, ))
account = cursor.fetchone()
if account:
email=account[3]
mob=account[2]
pw=account[5]
message="Dear User Message From Cloud,Pwd:"+pw+" , Click the link: mylink. By
SMSWAY IOTCLD"
#params = urllib.parse.urlencode({'token': 'b81edee36bcf4ddbaa6ef535f8db03e', 'credit':
2,      'sender':      'IOTCLD',      'message':message,      'number':str(mob),
'templateid':'1207162443831712783'})
#url = "http://pay4sms.in/sendsms/?%os" % params
#with urllib.request.urlopen(url) as f:

```

```

# print(f.read().decode('utf-8'))
# print("sent"+str(mob))
msg="Password has sent.."
else:
    msg = 'Incorrect username'
    return render_template('forgot2.html',msg=msg)
@app.route('/register', methods=['GET', 'POST'])
def register():
    msg=""
    mycursor = mydb.cursor()
    mycursor.execute("SELECT max(id)+1 FROM patient")
    maxid = mycursor.fetchone()[0]
    if maxid is None:
        maxid=1
    if request.method=='POST':
        name=request.form['name']
        gender=request.form['gender']
        dob=request.form['dob']
        mobile=request.form['mobile']
        email=request.form['email']
        uname=request.form['uname']
        pass1=request.form['pass']
        cursor = mydb.cursor()
        sql = "INSERT INTO patient(id,name,gender,dob,mobile,email,uname,pass) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
        val = (maxid,name,gender,dob,mobile,email,uname,pass1)
        cursor.execute(sql, val)
        mydb.commit()
        print(cursor.rowcount, "Registered Success")

```

```

result="sucess"
if cursor.rowcount==1:
    return redirect(url_for('index'))
else:
    msg='Already Exist'
    return render_template('/register.html',msg=msg)

@app.route('/login_doc', methods=['GET', 'POST'])
def login_doc():
    msg=""
    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM doctor WHERE uname = %s AND pass = %s', (uname, pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname
            return redirect(url_for('doc_home'))
        else:
            msg = 'Incorrect username/password! or access not provided'
    return render_template('login_doc.html',msg=msg)

@app.route('/reg_doc', methods=['GET', 'POST'])
def reg_doc():
    msg=""
    mycursor = mydb.cursor()
    mycursor.execute("SELECT max(id)+1 FROM doctor")
    maxid = mycursor.fetchone()[0]

```

```

if maxid is None:
    maxid=1
if request.method=='POST':
    name=request.form['name']
    mobile=request.form['mobile']
    email=request.form['email']
    uname=request.form['uname']
    pass1=request.form['pass']
    cursor = mydb.cursor()
    sql = "INSERT INTO doctor(id,name,mobile,email,uname,pass) VALUES (%s, %s, %s,
    %s, %s, %s)"
    val = (maxid,name,mobile,email,uname,pass1)
    cursor.execute(sql, val)
    mydb.commit()
    print(cursor.rowcount, "Registered Success")
    result="sucess"
    if cursor.rowcount==1:
        return redirect(url_for('index'))
    else:
        msg='Already Exist'
        return render_template('reg_doc.html',msg=msg)
@app.route('/pat_home', methods=['GET', 'POST'])
def pat_home():
    msg=""
    if 'username' in session:
        uname = session['username']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM patient WHERE uname = %s', (uname, ))
        data = cursor.fetchone()

```

```

return render_template('pat_home.html',msg=msg, data=data)

@app.route('/sugg', methods=['GET', 'POST'])

def sugg():

    msg=""

    if 'username' in session:

        result=""

        per=0

        if request.method=='POST':

            name=request.form['name']

            gender=request.form['gender']

            age=request.form['age']

            height=request.form['height']

            weight=request.form['weight']

            glucose=request.form['glucose']

            bp=request.form['bp']

            gg=int(glucose)

            bb=int(bp)

            print(gg)

            print(bb)

            dv = pd.read_csv('static/dataset/diabetes.csv')

            if ss2[1]>=g1 and ss2[1]<=g2 and ss2[2]>=b1 and ss2[2]<=b2:

                result="1"

                print(ss2[1])

                print(ss2[2])

                print(ss2[6])

                if ss2[6]<1:

                    per=ss2[6]*100

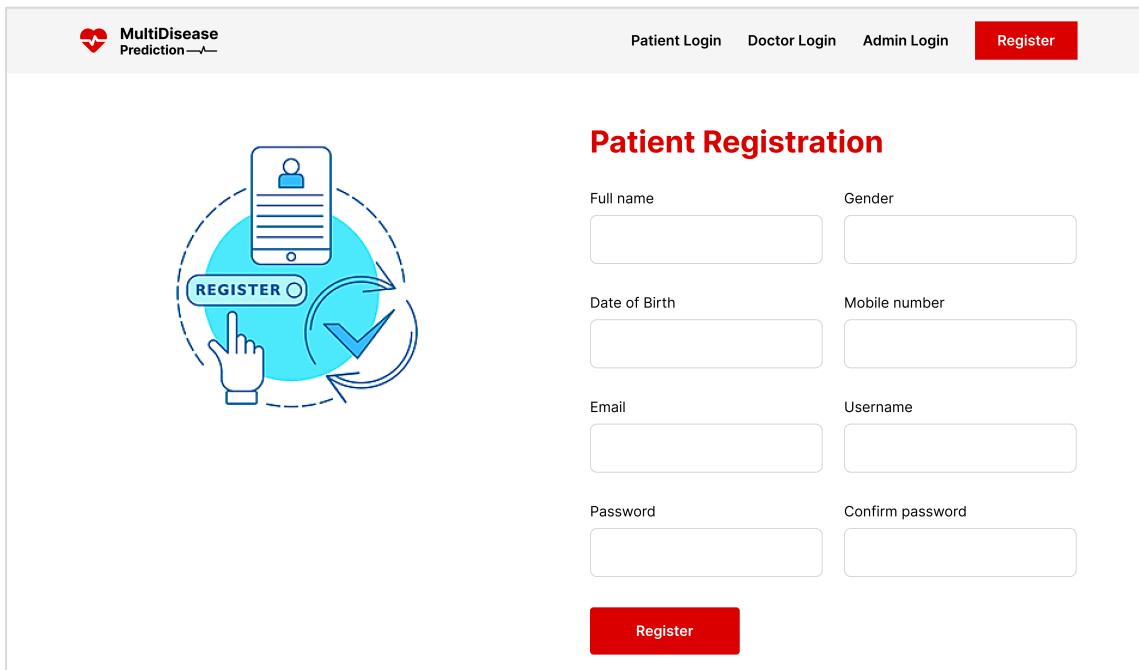
                else:

                    per=randint(20,54)

```

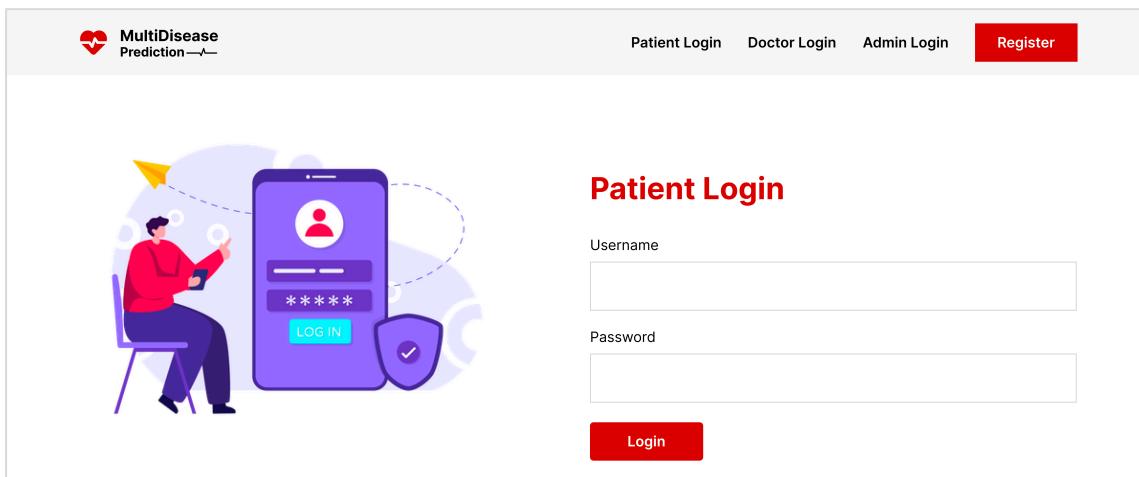
```
break
else:
    result="2"
    print(per)
    print(result)
    return render_template('test.html',act=act,per=per,result=result)
@app.route('/logout')
def logout():
    # remove the username from the session if it is there
    session.pop('username', None)
    return redirect(url_for('index'))
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

8.2 SCREENSHOTS



The screenshot shows the 'Patient Registration' page of the MultiDisease Prediction application. At the top, there is a logo with a heart icon and the text 'MultiDisease Prediction'. To the right are links for 'Patient Login', 'Doctor Login', 'Admin Login', and a red 'Register' button. On the left, there is a circular illustration of a smartphone displaying a user profile icon, with a hand pointing to a 'REGISTER' button and a checkmark icon. The main form area has the title 'Patient Registration' in bold red font. It contains six input fields arranged in two columns: 'Full name' and 'Gender', 'Date of Birth' and 'Mobile number', 'Email' and 'Username', and 'Password' and 'Confirm password'. Below these fields is a red 'Register' button.

Figure 8.1 Signup form –
User creates the account by filling the fields of registration form



The screenshot shows the 'Patient Login' page of the MultiDisease Prediction application. At the top, there is a logo with a heart icon and the text 'MultiDisease Prediction'. To the right are links for 'Patient Login', 'Doctor Login', 'Admin Login', and a red 'Register' button. On the left, there is a circular illustration of a person sitting at a desk, interacting with a smartphone that displays a login screen with a user profile icon, a password field with asterisks, and a 'LOG IN' button. The main form area has the title 'Patient Login' in bold red font. It contains two input fields: 'Username' and 'Password'. Below these fields is a red 'Login' button.

Figure 8.2 Login form –
A user fill the details to login to the application to get prediction

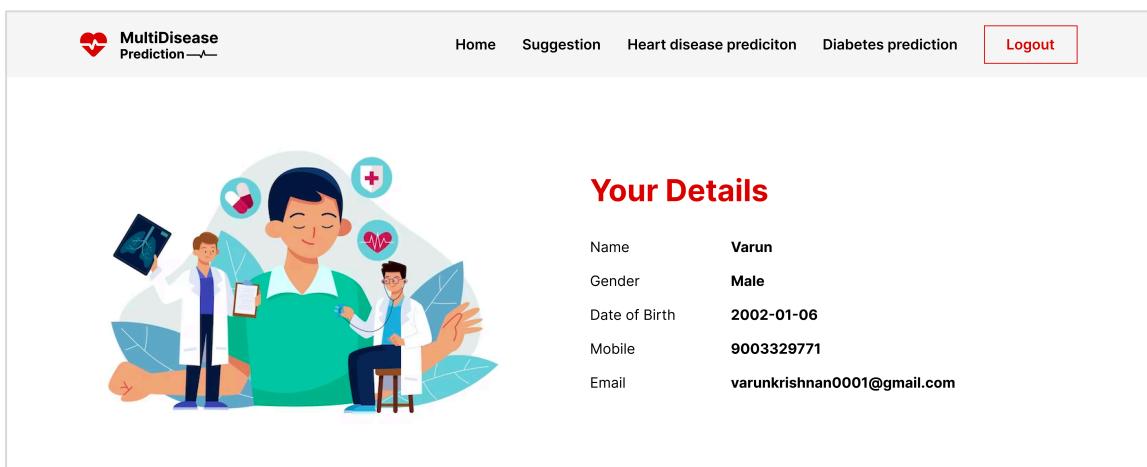


Figure 8.3 User information –

A user can view his details after login

The screenshot shows the 'Diabetes prediction' section of the website. On the left, there is a sidebar titled 'Tips to stay away from diabetes' which lists the following tips:

- Maintain a healthy weight
- Eat a healthy diet
- Exercise regularly
- Avoid smoking and limit alcohol consumption
- Get enough sleep
- Manage stress
- Monitor your blood sugar levels

Below the sidebar is a decorative illustration related to diabetes management. On the right, the heading 'Diabetes Prediction' is displayed in red. Below it, there is a form with fields for Name, Gender, Age, Height(cm), Weight(kg), Glucose level, and Blood pressure. A note at the bottom states: 'Resting blood pressure (in mm Hg on admission to the hospital) (94 to 200)'. A red button labeled 'Predict diabetes →' is located at the bottom right.

Figure 8.4: Diabetes prediction –

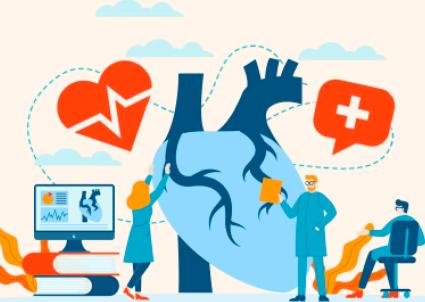
User Fill the form with respective value to get the diabetes prediction


**MultiDisease
Prediction**

[Home](#) [Suggestion](#) [Heart disease prediction](#) [Diabetes prediction](#) [Logout](#)

Tips to stay away from heart disease

- Maintain a healthy weight
- Eat a healthy diet
- Exercise regularly
- Avoid smoking and limit alcohol consumption
- Get enough sleep
- Manage stress
- Monitor your blood sugar levels
- Control high blood pressure
- Limit sodium intake
- Manage Diabetes



Heart Disease Prediction

Name	<input type="text"/>			
Gender	<input type="text"/>	Age	<input type="text"/>	
Height(cm)	<input type="text"/>		Weight (kg)	<input type="text"/>
Blood pressure	<input type="text"/>		Serum Cholesterol	<input type="text"/>
Resting blood pressure (in mm Hg on admission to the hospital) (94 to 200)				Serum Cholesterol in mg/dl - (126 to 564)
Fasting blood sugar	<input type="text"/>		Max heart rate	<input type="text"/>
>120mg/dl		71 to 202		
Rest ECG results	<input type="text"/>		Exercise included Angina	<input type="text"/>
Slope of peak exercise	<input type="text"/>		ST depression (0 to 6.2)	<input type="text"/>
Thal	<input style="width: 100px; height: 30px; margin-bottom: 5px;" type="text"/> <div style="border: 1px solid #ccc; width: 100px; height: 15px; background-color: #f0f0f0;"></div>		Number of major vessels	<input type="text"/>
<input style="background-color: red; color: white; border: 1px solid red; padding: 5px;" type="button" value="Predict disease →"/>				
No symptoms found for heart disease and your BMI is 22.09. You are healthy.				

Figure 8.5: Heart disease prediction–
User input the values to get the heart disease prediction

Data Exploration													
56.0	0.0	4.0	134.0	409.0	0.0	2.0	150.0	1.0	1.9	2.0	2.0	7.0	2.0
63.0	0.0	4.0	150.0	407.0	0.0	2.0	154.0	0.0	4.0	2.0	3.0	7.0	4.0
62.0	0.0	4.0	140.0	394.0	0.0	2.0	157.0	0.0	1.2	2.0	0.0	3.0	0.0
65.0	0.0	3.0	160.0	360.0	0.0	2.0	151.0	0.0	0.8	1.0	0.0	3.0	0.0
57.0	0.0	4.0	120.0	354.0	0.0	0.0	163.0	1.0	0.6	1.0	0.0	3.0	0.0
55.0	1.0	4.0	132.0	353.0	0.0	0.0	132.0	1.0	1.2	2.0	1.0	7.0	3.0
55.0	0.0	2.0	132.0	342.0	0.0	0.0	166.0	0.0	1.2	1.0	0.0	3.0	0.0
43.0	0.0	4.0	132.0	341.0	1.0	2.0	136.0	1.0	3.0	2.0	0.0	7.0	2.0
58.0	0.0	3.0	120.0	340.0	0.0	0.0	172.0	0.0	0.0	1.0	0.0	3.0	0.0
57.0	1.0	4.0	110.0	335.0	0.0	0.0	143.0	1.0	3.0	2.0	1.0	7.0	2.0
64.0	1.0	3.0	140.0	335.0	0.0	0.0	158.0	0.0	0.0	1.0	0.0	3.0	1.0

Figure 8.6: Data exploration for diabetes –
Admin use the CSV file to train the model


**MultiDisease
Prediction**

[Home](#) [Suggestion](#) [Heart disease prediciton](#) [Diabetes prediction](#) [Logout](#)

Suggestion & Prescription

S.No.	Suggestions	Prescription	Date
1	Do exercise daily	-	27/04/2023

Figure 8.7: Suggestion & Prescription –
One will get the suggestions and prescriptions form doctor in suggestions tab

CHAPTER 9

REFERENCES

- [1] Aidaroos, K. Bakar, A. & Othman, Z. (2012). Medical Data Classification with Naive Bayes Approach. *Information Technology Journal*.
- [2] Aidaroos, KM. A. B. (n.d.). 2012. Medical Data Classification with Naive Bayes Approac.
- [3] Ashok kumarD.wivedi “Evaluate the performance of different machine learning techniques for prediction of heart disease using ten-fold cross-validation” Springer 17 September 2016.
- [4] ChalaBeyene Prof. Pooja Kamat “Survey on Prediction and Analysis the Occurrence of Heart Disease Using Data Mining Techniques” International Journal of Pure and Applied Mathematics 2018. [7] R. Sharmila S. Chellammal “A conceptual method to enhance the prediction of heart diseases using the data techniques” International Journal of Computer Science and Engineering May 2018.
- [5] Darcy A. Davis N. V.-L. (2008). Predicting Individual Disease Risk Based on Medical History.
- [6] JyotiSoni Ansari, U. Sharma, D. & Soni,S. (2011). Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction.
- [7] Marjia Sultana Afrin Haider “Heart Disease Prediction using WEKA tool and 10-Fold cross-validation” The Institute of Electrical and Electronics Engineers March 2017.
- [8] MeghaShahi, R. Kaur Gurm “Heart Disease Prediction System using Data Mining Techniques” Orient J. Computer Science Technology vol.6 2017 pp.457-466.
- [9] NisharBanu, MA.; Gomath B.; (2013). Disease Predicting System Using Data Mining Techniques.
- [10] Polaraju, K. Durga Prasad, D. “Prediction of Heart Disease using Multiple Linear Regression Model” International Journal of Engineering Development and Research Development ISSN:2321-9939 2017.
- [11] SeemaShedole, S., KumariDeepika, “Predictive analytics to prevent and control chronic disease”, <https://www.researchgate.net/punlication/316530782>, January 2016.