A

Project Report on

MULTIPLE DISEASE PREDICTION SYSTEM USING MACHINE LEARNING

Submitted in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P S V S K VARUN KUMAR	19AK1A05I0
A TEJASWINI	19AK1A05H4
K VENKATA RAJU	19AK1A05I1
U SAI KRUPA	19AK1A05D9

Under the guidance of

Mr. D Sainath., MTech.
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur.
Three B. Tech Programmes (CSE, ECE & CIVIL) are accredited by NBA, New Delhi, Accredited by NAAC with 'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.)
Tirupati-517520.

2019-2023

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur.
Three B. Tech Programmes (CSE, ECE & CIVIL) are accredited by NBA, New Delhi, Accredited by NAAC with 'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.)
Tirupati-517520.

2019-2023

COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this is a bonafide record of the Project Report entitled, ***"Multiple Disease Prediction System using Machine Learning",*** *done by P S V S K Varun Kumar (19AK1A05I0), A Tejaswini (19AK1A05H4), K Venkata Raju (19AK1A05I1), U Sai Krupa (19AK1A05D9) is being submitted in partial fulfilment of the requirements for the award of the degree of* ***BACHELOR OF TECHNOLOGY*** *in* ***COMPUTER SCIENCE AND ENGINEERING*** *to the Annamacharya Institute of technology and Sciences, Tirupati, during the academic year 2022-23.*

Signature of the supervisor	Signature of Head of the Department
Mr D. Sainath, MTech.,	**Mr. B. Ramana Reddy, M.Tech.,(Ph.D)**
Assistant professor,	Associate professor and HOD
Department of CSE,	Department of CSE,
AITS, Tirupati	AITS, Tirupati

DATE:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur.
Three B. Tech Programmes (CSE, ECE & CIVIL) are accredited by NBA, New Delhi, Accredited by NAAC with 'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.)
Tirupati-517520.
2019-2023

COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the project titled **"Multiple Disease Prediction System using Machine Learning"** is a genuine project work carried out by us, in B.Tech (Computer Science and Engineering) course in Annamacharya Institute of Technology And Sciences and has not been submitted to any other course or university for the award of our degree by us.

P S V S K VARUN KUMAR	**19AK1A05I0**
A TEJASWINI	**19AK1A05H4**
K VENKATA RAJU	**19AK1A05I1**
U SAI KRUPA	**19AK1A05D9**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Dr. C. GANGI REDDY,** Hon'ble Secretary of AITS-Tirupati, for providing congenial atmosphere and encouragement.

We show gratitude to **Dr. C. NADHAMUNI REDDY**, **Principal** for having provided all the facilities and support.

We would like to thank **Mr. B. RAMANA REDDY., M.Tech(Ph.D), Associate professor & HOD**, **Computer Science and Engineering** for encouragement at various levels of our Project.

We thankful to our project coordinator **Mr. V.SAMBASIVA, Assistant Professor, CSE,** for his sustained inspiring guidance and cooperation throughout the process of this project.

We would like to express our sincere gratitude to our supervisor **Mr. D.Sainath, Assistant Professor**, Dept. of CSE, AITS, for his constant help, kind cooperation and encouragement in completing the work successfully.

We express our deep sense of gratitude and thanks to all the **Teaching** and **Non-Teaching Staff** of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our **Parents**, **Friends** and **Well wishers** who helped a lot in making the report of this project.

P S V S K VARUN KUMAR	**19AK1A05I0**
A TEJASWINI	**19AK1A05H4**
K VENKATA RAJU	**19AK1A05I1**
U SAI KRUPA	**19AK1A05D9**

CONTENTS

ABSTRACT

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 model we are proposing a system which used to predict multiple diseases by using Flask API. The system is used to analyse Diabetes analysis, Heart disease and breast cancer analysis, chronic kidney disease, liver disease 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 system 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.

LIST OF FIGURES

LIST OF SCREENS

LIST OF ABBREVIATIONS

Short form	abbreviation
al	Albumin
su	Sugar degree
rbc	Red blood cells
pc	Pus cell
pcc	Pus cell clumps
ba	Bacteria
bgr	Blood glucose
bu	Blood urea
sc	Cerum creatinine
pot	Potassium
wbc	White blood cells
htn	Hypertension
dm	Diabetes mellitus
cad	Coronery artery disease
pe	Pedal edema
ane	anemia
fbs	Fasting blood sugar
exang	Exercise induced angina
thal	thalassemia
bmi	Body mass index
alt	Alamine aminotransferase
ast	Aspartate aminotransferase

1. INTRODUCTION

1.1 Introduction

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. When anyone is currently afflicted with an illness, they must see a doctor, which is both time consuming and expensive. It can also be difficult for the user if they are out of reach of doctors and hospitals because the illness cannot be detected. 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 by using machine learning. In this system, we are going to analyze Diabetes, Heart, and liver disease analysis. Later many more diseases can be included. To implement multiple disease prediction systems, we are going to use machine learning algorithms. The importance of this system disease that lasts a long time or takes a long time to heal, and many chronic diseases cannot be cured but can only be managed with daily treatments. India, like all other nations, is undergoing significant social and economic shifts, which is causing a rapid rise in the prevalence of cardiovascular disease.

Many developed, developing, and developing countries, including India, are dealing with a wide range of chronic diseases, especially cardiovascular disease and diabetes, which could have serious consequences for global health, security, and economy. The rapid urbanisation and economic growth of today's world has resulted in a wide range of lifestyles. Chronic diseases are now a problem in all nations, with chronic disease afflicting one-third of the population in each. Chronic disease care is more expensive, and it is difficult for those who are sick. In the medical field, a large number of chronic disease datasets are gathered and processed analysis is that while analysing the diseases all the parameters which cause the disease is included so it is possible to detect the disease efficiently and more accurately.

The consumer will be able to determine the likelihood of a disease based on the symptoms given using Disease Predictor. People are always curious to learn new things, particularly as the use of the internet grows every day. When an issue occurs, people often want to look it up on the internet. Hospitals and physicians have less access to the internet than the general public. When people are

afflicted with an illness, they do not have many options. As a result, this system can be beneficial to people.

Chronic illness is data mining aids in disease early detection. Cardiovascular disease, diabetes, liver disease, Alzheimer's disease, and Parkinson's disease are the most expensive diagnosis diseases. 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. 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. So, if the above procedure can be done using an automated software that saves time and money, it could be better for the patient.

Machine learning algorithms such as decision trees, support vector machines, and neural networks can be trained on large datasets of medical records to learn patterns and correlations between various health parameters and disease outcomes. These algorithms can then be used to make predictions on new patient data, providing a probability estimate for each potential disease.

Multiple disease prediction systems can be used for a wide range of diseases, including heart disease, diabetes, liver disease, kidney disease, and more. By using machine learning algorithms, these systems can provide accurate and personalized predictions that can help healthcare professionals make informed decisions about patient care. Additionally, these systems can help reduce healthcare costs by identifying patients who may require more frequent monitoring or intervention.

3.3.2 Parameters

The parameters used in Heart disease prediction are as follows:

1. **Age:** It is divided into 4 ranges; Here Age is integer. Age is an important parameter for heart disease prediction using machine learning. As people age, their risk of developing heart disease increases.
2. **Sex:** Have only two variables: Male, Female. Men and women can have different risk factors for heart disease, so it is important to include sex as a predictor variable in machine learning models. Machine learning algorithms can use sex as a binary variable, where males are assigned a value of 1 and females are assigned a value of 0.
3. **Chest Pain:** There are 4 different levels of chest pain ranging from 0-3. 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. **tretstbps:** resting blood pressure (in mm Hg on admission to the hospital). 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. **Cholesterol:** serum cholesterol in mg/dl. 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. **fbs:** (fasting blood sugar> 120 mg/dl) (1-true: 0=false). Not producing enough of a bormone vected by yur ps (insulin) or not responding to insulin properly cases your body's 3iff gar levels to nse, increasing your risk of a heart attack.
7. **thalach:** maximum 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.
8. **exang;** exercise induced angina (1-yes: 0=no). 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 the chest but may spread to either or both of your shoulders, or your back, neck, jaw or arm.

9. **old peak:** ST Depression induced by Exercise relative to rest. A treadmill ECG stress test is considered abnormal when there is a horizontal or down-sloping ST-segment depression 21 mm at 60-80 ms after the J point.
10. **slope:** the slope of the peak exercise ST segment. The slope refers to the slope of the ST segment of an electrocardiogram (ECG) reading, which is a measure of the electrical activity of the heart.The slope of the ST segment can be categorized into three values: upsloping, flat, or downsloping.
11. **ca:** no. of major vessels (0-3) by fluoroscopy. Specifically, ca stands for the number of major vessels (0-3) colored by fluoroscopy. Fluoroscopy is a medical imaging technique that uses X-rays to obtain real-time images of the inside of the body. In machine learning models, the ca parameter is often included as a numeric variable. The number of major vessels colored by fluoroscopy can be an indicator of the extent and severity of coronary artery disease, which is a common type of heart disease.
12. **thal:** 3 = normal; 6= fixed defect; 7 = reversible defect. Thal stands for thalassemia, which is an inherited blood disorder that affects the production of hemoglobin. In machine learning models, the thal parameter is often included as a categorical variable with three possible values: normal, fixed defect, or reversible defect. These values correspond to the results of a nuclear stress test, which is a type of imaging test that is used to diagnose heart disease.

The parameters used in Diabetes disease prediction are as follows:

1. **Pregnancy:** the pregnancy parameter may be included as a binary variable indicating whether the individual is currently pregnant or not.
2. **Glucose:** It refers to the concentration of glucose (sugar) in the blood after a period of fasting, usually overnight. Elevated fasting blood glucose levels are a strong indicator of diabetes risk. The fasting blood glucose parameter is often included as a numeric variable.
3. **Blood pressure:** High blood pressure is a risk factor for both type 1 and type 2 diabetes. In addition, people with diabetes are at higher risk for high blood pressure and other cardiovascular diseases. Blood pressure is often included as a numeric variable, measured as systolic blood pressure (the top number) and diastolic blood pressure (the bottom number).
4. **Skin thickness:** It is measured by the thickness of a fold of skin at a particular location on the body, such as the triceps or subscapular region. A thicker skin fold measurement is often associated with higher levels of body fat, which is a risk factor for diabetes.In machine learning models for diabetes prediction, skin thickness is often included as a numeric variable.
5. **Insulin level:** Insulin is a hormone produced by the pancreas that regulates blood sugar levels by allowing glucose to enter cells for energy production. In machine learning models for diabetes prediction, insulin level is often included as a numeric variable.
6. **Body mass index:** BMI is a measure of body fat based on a person's height and weight, and is calculated by dividing weight in kilograms by height in meters squared. Higher BMI values are associated with an increased risk of developing diabetes, as well as other health problems such as cardiovascular disease and hypertension.In machine learning models for diabetes prediction, BMI is often included as a numeric variable.
7. **Diabetes degree function:** DPF is a measure of the family history of diabetes and is based on the genetic relationship between affected and unaffected family members. A higher DPF value indicates a stronger family history of diabetes and an increased risk of developing the disease.In machine learning models for diabetes prediction, DPF is often included as a numeric variable.
8. **Age:** age is a known risk factor for the development of type 2 diabetes, and the risk of developing the disease increases with age.In machine learning models for diabetes prediction, age is often included as a numeric variable.

The parameters used in Kidney disease prediction are as follows:

1. **Age:** It is divided into 4 ranges, here Age is integer. Age is an important parameter for kidney disease prediction using machine learning. As people age, their risk of developing kidney disease increases.
2. **Blood Pressure:** There are several BP parameters that can be used for kidney disease prediction some are, systolic blood pressure(SBP),Dialostic blood pressure(DBP),Pulse pressure(PP),Arterial Stiffness.
3. **Al:** Albumin is a protein that is produced by the liver and is normally present in the blood. However, in kidney disease, the kidneys may not be able to filter albumin properly, resulting in increased levels of albumin in the urine (albuminuria).
4. **SU:** The sugar degree parameter, also known as blood sugar level, is a measure of the amount of glucose in the blood. High levels of blood sugar, or hyperglycemia, can be a risk factor for kidney disease. When blood sugar levels are consistently high, it can damage the small blood vessels in the kidneys and affect their ability to function properly.
5. **Red Blood Cells:** The "rbc" parameter, which stands for red blood cells, is a commonly used parameter in the context of diagnosing and predicting kidney disease. In particular, a low red blood cell count (anemia) can be an indicator of kidney disease or damage.
6. **PC:** The "pus cell" parameter, which refers to the presence of white blood cells (leukocytes) in the urine, can be a useful diagnostic indicator for certain types of kidney disease, such as urinary tract infections or interstitial nephritis.
7. **BA:** The presence of bacteria in urine, also known as bacteriuria, can be a sign of a urinary tract infection (UTI) and can be a risk factor for kidney disease.
8. **BGR:** The "blood glucose random" parameter, which refers to the level of glucose in the blood at a random time, can be an important predictor variable in machine learning models for kidney disease prediction. Elevated blood glucose levels can be a sign of diabetes, which is a leading cause of kidney disease.
9. **BU:** BUN levels can provide information on how well the kidneys are functioning, as they are responsible for filtering urea and other waste products out of the blood.
10. **SC:** Creatinine is a waste product that is produced by muscles and excreted by the kidneys. If the kidneys are not functioning properly, the creatinine level in the blood will be elevated. Therefore, serum creatinine levels can be used to assess kidney function and detect kidney disease.

11. **DM:** Diabetes mellitus is a well-known risk factor for chronic kidney disease, and therefore it may be considered as a predictor variable in machine learning models for kidney disease prediction. Diabetes can cause damage to the small blood vessels in the kidneys, which can lead to kidney damage over time.
12. **POT:** Potassium is an important electrolyte that plays a critical role in various bodily functions, including muscle and nerve function, and the regulation of blood pressure. Abnormal potassium levels can be a sign of kidney disease, as the kidneys play a key role in maintaining the proper balance of electrolytes in the body.
13. **WC:** It refers to the number of white blood cells present in a cubic millimeter of blood and is an important indicator of the immune system's health. Abnormal levels of white blood cells can be indicative of various underlying conditions, such as infections, inflammation, or leukemia, and can also be a sign of kidney disease.
14. **HTN:** Hypertension, or high blood pressure, is a common risk factor for kidney disease. The high pressure in the blood vessels can damage the small blood vessels in the kidneys, leading to kidney damage over time.
15. **PE:** Pedal edema, or swelling in the legs and ankles, can be a sign of kidney disease. This is because the kidneys play a role in regulating fluid balance in the body, and damage to the kidneys can result in fluid buildup in the legs and ankles.
16. **ANE:** Anemia, or low levels of red blood cells or hemoglobin in the blood, is a common complication of kidney disease. The kidneys produce a hormone called erythropoietin that stimulates the production of red blood cells in the bone marrow. When the kidneys are damaged, they may not produce enough erythropoietin, leading to anemia.

The parameters used in Breast Cancer disease prediction are as follows:

1. **Fractal dimension:** fractal dimension is a parameter that can be used in breast cancer disease prediction using machine learning, and it may provide valuable information about the complexity of blood vessel branching patterns in breast tumors.
2. **Concavity:** Concavity is a measure of the degree to which the edges of a mass or lesion in the breast appear to be concave or rounded. In breast cancer, concavity can be used as a feature or parameter in predictive models, along with other clinical and imaging parameters, such as age, family history, mammogram results, ultrasound features, and genetic markers.
3. **Concavity worst:** Concavity worst is a measure of the degree to which the worst case concavity value of a mass or lesion in the breast appears to be concave or rounded. In breast cancer, concavity worst can be used as a feature or parameter in predictive models, along with other clinical and imaging parameters, such as age, family history, mammogram results, ultrasound features, and genetic markers.
4. **Compactness:** compactness has been found to be a significant predictor of breast cancer risk and can improve the accuracy of breast cancer prediction models. Compactness is a measure of the degree to which a mass or lesion in the breast is compact or densely packed. In breast cancer, compactness can be used as a feature or parameter in predictive models.
5. **Compactness worst:** Compactness worst is a measure of the degree to which the worst-case compactness value of a mass or lesion in the breast is compact or densely packed. In breast cancer, compactness worst can be used as a feature or parameter in predictive models, along with other clinical and imaging parameters, such as age, family history, mammogram results, ultrasound features, and genetic markers.
6. **Area and Parameter:** Area is a measure of the size of a mass or lesion in the breast, while perimeter is a measure of the length of its boundary. In breast cancer, area and perimeter can be used as features or parameters in predictive models, along with other clinical and imaging parameters, such as age, family history, mammogram results, ultrasound features, and genetic markers.
7. **Concave points:** It is important to note that concave points should not be used as the sole predictor of breast cancer risk, as other clinical and imaging parameters are also important in accurately predicting breast cancer risk.

The parameters used in Liver disease prediction are as follows:

1. **Total Bilirubin:** Bilirubin is a yellow pigment that is produced when red blood cells break down, Total bilirubin is a measure of the total amount of bilirubin in the blood, including both unconjugated and conjugated bilirubin
2. **Direct Bilirubin:** Direct bilirubin, also known as conjugated bilirubin, is a measure of the bilirubin that has been processed by the liver and excreted in the bile. In liver disease, direct bilirubin levels may be elevated due to impaired liver function, which can affect the processing and excretion of bilirubin.
3. **Albumin:** Albumin is a protein produced by the liver, and its levels in the blood can provide important information about liver function. In liver disease prediction using machine learning, albumin is often included as a feature or parameter in predictive models.
4. **Total proteins:** Total protein is another laboratory parameter that can be used in liver disease prediction using machine learning. Total protein is a measure of the total amount of protein in the blood, including albumin and other proteins.
5. **Globulin Ratio:** In liver disease, the globulin ratio may be increased due to changes in globulin levels, which can be caused by liver inflammation, infection, or cancer. An increased globulin ratio can be an indicator of liver disease and may be used as a feature or parameter in predictive models for liver disease. The globulin ratio is calculated by dividing the total protein level by the albumin level
6. **Alkaline 9ifferent9e:** Alkaline phosphatase (ALP) is a laboratory parameter that can provide important information about liver function and is often used in liver disease prediction using machine learning.
7. **ALT:** Alamine aminotransferase (ALT), also known as serum glutamate pyruvate transaminase (SGPT), is an important parameter used in liver disease prediction using machine learning. ALT is an enzyme found primarily in the liver, and elevated levels of ALT in the blood can indicate liver damage or disease.
8. **AST:** Aspartate aminotransferase (AST), also known as serum glutamic oxaloacetic transaminase (SGOT), is another important parameter used in liver disease prediction using machine learning. AST is an enzyme found in various tissues, including the liver, and elevated levels of AST in the blood can indicate liver damage or disease.

1.2 Existing System

A lot of analysis over existing systems in the health care industry considered only one disease at a time. For example, one system is used to analyse diabetes, another is used to analyse diabetes retinopathy, and another system is used to predict heart disease. Maximum systems focus on a particular disease. When an 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 diseases. In multiple diseases prediction system, a user can analyse more than one disease on a single website. The user doesn't need to traverse different places in order to predict whether he/she has a particular disease or not. In multiple diseases prediction system, the user needs to select the name of the particular disease, enter its parameters and just click on submit. The corresponding machine learning model will be invoked and it would predict the output and display it on the screen on the screen. There is a need to research and develop a system that will enable end users to predict chronic diseases without having to visit a physician or doctor for diagnosis. To identify various diseases by observing the symptoms of patients and applying various Machine Learning Models techniques. Machine Learning can improve the accuracy of predictions. There are numerous work that has been done related to disease prediction system using different Machine Learning algorithms and achieved different results for different methods in medical field. The multiple diseases predicition system using machine learning used Random forest algorithm to predict a disease on the basis of systems and to enable synchronized and well versed medical systems ensuring maximum patient satisfaction.

1.3 Disadvantages of Existing System

While machine learning-based disease predicting systems have the potential to revolutionize healthcare, they also have several disadvantages, including Limited Data, Disease prediction systems require large datasets to train the algorithms accurately. However, the availability of medical data can be limited in some cases. Furthermore, certain conditions may have lower prevalence rates, making it harder to gather enough data to create an accurate model. Bias, Machine learning models are only as unbiased as the data they are trained on. If the dataset is biased, the model may learn those biases, leading to inaccurate predictions. Interpretability, while machine learning models can make accurate predictions, they can be challenging to interpret. It can be difficult to determine how the model arrived at a particular prediction,

which can be problematic for clinicians trying to determine the best course of action for a patient. Dependence on technology, Machine learning-based disease predicting systems rely heavily on technology, and any issues with the system or the underlying technology can have significant consequences. Ethical concerns, The use of machine learning in healthcare raises ethical concerns about privacy, data ownership, and potential discrimination, particularly if the models are trained on biased datasets. There is a need for clear guidelines and regulations on the use of these systems in healthcare to address these concerns. Cost, Developing, deploying, and maintaining a machine learning-based disease prediction system can be expensive, particularly for healthcare systems with limited resources. Additionally, there may be ongoing costs associated with updating and refining the model as new data becomes available.

Overall, while machine learning-based disease predicting systems have the potential to transform healthcare, they also have several significant disadvantages that need to be considered before implementing them in clinical practice.

1.4 Proposed System

In multiple disease prediction, it is possible to predict more than one disease at a time. So the user doesn't need to traverse different sites in order to predict the diseases. We are taking three diseases that are Liver, Diabetes, and Heart. As all these diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms. And Django. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Django will invoke the corresponding model and returns the status of the patient. In order to make it less time consuming we are aiming at a more specific questionnaire which will be followed by the system. Our aim with this system is to be the connecting bridge between doctors and patients. The main feature will be the machine learning, in which we will be using algorithm Random Forest Algorithm which will help us in getting accurate predictions and Also, will find which algorithm gives a faster and efficient result by comparatively-comparing. Another feature that our system will comprise of is Doctor's Consultation. After delivering the results, our system will also suggest the user to get a doctors consultation on this report. By using this feature, we will not only address the other class of users i.e. the Doctors but we will also gain their trust in this system as in that this system is not affecting their business.

1.5 Advantages of Proposed System

Our project is stand on multiple disease prediction in accordance with symptoms entered by patient. The first task is to determine the problem statement. Then making the dataset ready to work on. After that we conceptualize our data using scatter plot, distribution graph, etc. by doing so we can find out anomalies, missing values, etc. on our data and make our dataset perfect for prediction. And finally, the main feature will be Machine Learning in which we will be using algorithm Random Forest which will predict accurate disease for early prediction and better patient care. For this model, we have used python as a platform to execute our Machine Learning algorithms. The patient can easily know the occurrence of the disease to them by just giving symptoms as input. With this multiple disease predicting system using machine learning for doctor's time will be saved and for patients both the time and money will be saved.

2.ANALYSIS

2.1 Introduction

System analysis is the process of analysing complex systems and breaking them down into smaller, more manageable components for better understanding. In the context of multiple disease prediction systems using machine learning (ML), system analysis is a crucial step in designing, developing, and deploying a reliable and accurate system.

Multiple disease prediction systems using ML are designed to predict the occurrence of multiple diseases based on various factors such as age, gender, medical history, and lifestyle. Such systems require the integration of various technologies such as machine learning algorithms, data preprocessing, and feature selection techniques.

A disease prediction system will predict diseases based on symptoms and some other parameters, which help doctors recognize patient health to improve the medical treatment given to the patient. Machine learning algorithms will detect the patterns of certain diseases with patient healthcare records and doctors can develop customized treatments and prescribe medicines for that specific disease in individual patients.

2.1.1 Literature Survey

1. Title : Multi-disease risk assessment using probabilistic graphical models.

Author: Farkhondeh

Description: This paper proposes a multi-disease risk assessment system based on probabilistic graphical models. The system is designed to predict the risk of developing multiple diseases simultaneously, including cardiovascular disease, diabetes, and hypertension, among others. The authors demonstrate the effectiveness of their system using real-world patient data.

2. Title: A hybrid approach for multiple disease prediction using machine learning techniques.

Author: Patel

Description: This paper proposes a hybrid approach that combines feature selection, clustering, and machine learning algorithms to predict the risk of multiple diseases. The authors demonstrate the effectiveness of their approach by predicting the likelihood of developing five diseases, including diabetes, hypertension, and cancer, among others.

3. Title: Disease Prediction Using Graph Machine Learning Based on Electronic Health Data: A Review of Approaches and Trends

Author: Haohui, Shahadat Uddin

Description: Electronic health data are computerised medical records for patients that contain information about healthcare entities. These data refer to a patient's diseases or conditions and are recorded in electronic systems, with the primary goal of delivering healthcare and related services. Electronic health data are rapidly being used for modelling and decision making in the healthcare research sector. These types of data are used for more than record-keeping in healthcare research, e.g., analysing healthcare utilisation, monitoring hospital care network effectiveness, developing predictive models for disease prediction.

4. Title: Machine learning-based predictive models for healthcare: A systematic review of methods and applications

Author: Kavakiotis

Description: The authors found that machine learning-based predictive models have the potential to improve diagnosis, treatment selection, and patient outcomes prediction in healthcare. They identified several challenges in the implementation of these models, such as data privacy, data bias, and interpretability. They also discussed the importance of validation and evaluation of these models in clinical settings.

2.2 Software Requirements Specification

A software requirements specification (SRS) is a detailed document that outlines the functional and non-functional requirements of a software system. In the case of a multiple disease prediction system using machine learning (ML), the SRS should outline the requirements necessary to develop a system that accurately predicts the likelihood of a patient having multiple diseases based on their levels of symptoms.

2.2.1 User Requirements

- PC, Mac or laptop with compatible processors.
    - Intel i3 & above
    - 1.6 GHz Processor is recommended
- At least 512 MB of free RAM should be available for the application.

- Internet connection required for managing and predicting.
- Microsoft Windows specific requirements:
    - Microsoft Windows 7 / 8 / 10 / 11.
    - MySql
    - One of the following development environment and platform for application and development.
        - Python 2.7 & above
        - Microsoft Visual Studio code 2012 or newer (for application development under python)
        - Jupyter Notebook

2.2.2 Functional Requirements & Non-Functional Requirements

Functional Requirements-

The functional requirements of a multiple disease prediction system using machine learning typically include the following:

- Data input and management: The system should allow for the input of patient data, such as demographic information, medical history, and diagnostic test results, and store and manage that data securely.
- Data pre-processing: The system should pre-process the data, including cleaning, normalization, and feature selection, to prepare it for analysis.
- Machine learning models: The system should incorporate various machine learning models, such as decision trees, random forest classifier and neural networks, that can be used to predict the likelihood of a patient having a specific disease.
- Training and testing: The system should allow for the training and testing of the machine learning models using the preprocessed data.
- Prediction output: The system should generate predictions about a patient's risk of developing a particular disease based on their input data and the trained machine learning models.
- Integration: The system should integrate with electronic health record systems and other medical databases to retrieve patient data and enable sharing of prediction results with healthcare providers.

Non-Functional Requirements-

- Performance: The system should be able to process large amounts of data quickly and efficiently, and provide accurate predictions within a reasonable amount of time.
- Reliability: The system should be reliable and consistent in its predictions, and be able to handle errors or unexpected data input without crashing.
- Scalability: The system should be able to handle increasing amounts of data as the user base grows, and be able to handle multiple users simultaneously.
- Usability: The system should be user-friendly and easy to navigate, with clear and concise instructions for inputting data and interpreting results.
- Maintainability: The system should be easy to maintain, with clear documentation and well-structured code that is easy to modify or update as needed.
- Security: The system should ensure the security and privacy of patient data, with robust security measures to prevent unauthorized access or data breaches.
- Compatibility: The system should be compatible with various hardware and software platforms, and able to integrate with existing healthcare systems and databases.
- Ethical considerations: The system should be developed with ethical considerations in mind, such as avoiding bias in the prediction models and protecting patient autonomy and privacy.

2.2.3 Software Requirements:

- Operating System – Windows 10
- Front End – HTML, CSS
- Back End – Python
- Database – MySql
- Spreadsheet – Microsoft Excel
- IDE – Visual Studio Code, Jupyter Notebook
- Framework – Flask

2.2.4 Hardware Components:

- Processor – i3
- Hard Disk – 1 GB
- Memory – 4 GB RAM

2.3: Flowchart

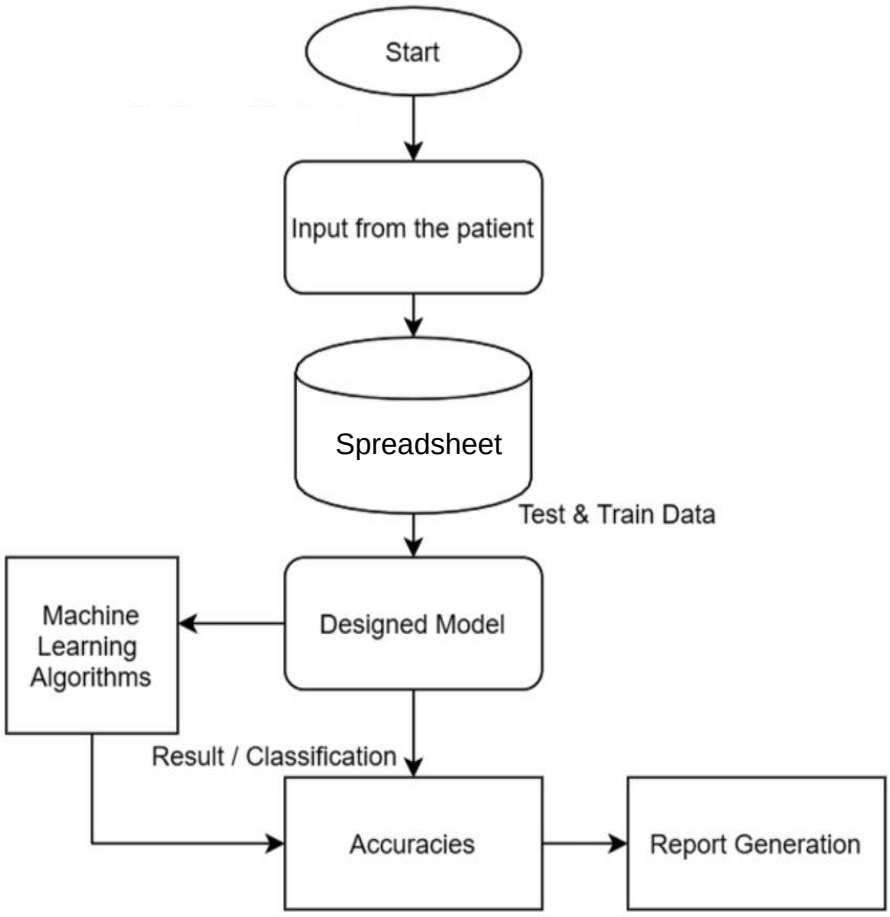


Fig 2.1: Flowchart for Attendance System

A flowchart for multiple disease prediction using machine learning can involve several steps, such as:

1. Data Collection: Gather a dataset that contains information on patients, including their symptoms, medical history, and diagnosis from any external source or visit any nearby hospital. In our project, we have used an external source for dataset.
2. Data Preprocessing: Clean and preprocess the dataset to remove any missing or irrelevant data. This can involve techniques like data normalization, imputation, and feature selection.

3. Feature Engineering: Extract relevant features from the dataset that can help to distinguish between different diseases. This can involve techniques like PCA, LDA, and feature selection algorithms.
4. Model Selection: Choose an appropriate machine learning algorithm to train the model on the preprocessed dataset. This can involve algorithms like logistic regression, decision trees, random forests, or neural networks. In our project, Random Forest classifier is used as it produces the best accuracies when compared to other algorithms.
5. Model Training: Train the model on the preprocessed dataset using the chosen machine learning algorithm i.e, Random Forest Classifier.
6. Model Validation: Validate the model by testing it on a separate dataset to evaluate its performance. This can involve techniques like cross-validation, hold-out validation, or k-fold validation.
7. Model Deployment: Deploy the model to make predictions on new patient data. This can involve creating a web application or API that takes patient data as input and returns predicted disease diagnoses as output.
8. Model Monitoring: Monitor the performance of the deployed model over time and update it as necessary to maintain its accuracy and reliability.

3. DESIGN

3.1 Introduction

A design is a plan or specification for the construction of an object of system or for the implementation of an activity or process, or the result of that plan or specification in the form of a prototype, product or process. The design expresses the process of developing a design. In some cases, the direct construction of an object without an explicit prior plan (such as in craftwork, some engineering. Coding, and graphic design) may also be considered to be a design activity. The design usually has to satisfy certain goals and constraints, may take into account aesthetic, functional, economic, or socio-political considerations, and is expected to interact with a certain environment. Major examples of designs include architectural blueprints, engineering drawings, processes etc.

The person who produces a design is called a designer, which is a term generally used for people who work professionally in one of the various design areas-usually specifying which area is being dealt with (such as a textile designer. Fashion designer, product designer, concept designer, website designer, but also others such as architects and engineers. A designer's sequence of activities is called a design process, possibly using design methods. The process of creating a design can be brief (a quick sketch) or lengthy and complicated, involving considerable research, negotiation, reflection, modelling, interactive adjustment, and re-design.

Designing a multiple disease prediction system using machine learning involves several steps. The first step is to define the problem statement, which involves identifying the target audience, the diseases to be predicted, and the performance metrics that will be used to evaluate the system.

The next step is to collect a large and diverse dataset of patient medical records and disease diagnoses. The dataset should be pre-processed to remove any missing or irrelevant data and to normalize the data to ensure that the machine learning algorithms can process it effectively.

After that, the dataset needs to be split into training, validation, and testing sets. The training set is used to train the machine learning models, while the validation set is used to tune the hyperparameters of the models. The testing set is used to evaluate the performance of the models and to compare them against each other.

Next, a variety of machine learning algorithms can be used to train the models, including decision trees, random forests, support vector machines, and neural networks. The models should be trained using a variety of techniques, such as bagging, boosting, and stacking, to improve their performance.

Once the models have been trained, they need to be evaluated using performance metrics such as accuracy, sensitivity, specificity, and F1-score. The models should be compared against each other, and the best-performing model should be selected for deployment.

The user interface of the system should be designed to be user-friendly and intuitive. The system should allow healthcare professionals to input patient data easily, and the system should provide real-time predictions. The user interface should also provide clear and concise information about the predicted diseases, and it should allow users to view and analyze the data in various ways.

Finally, the system should be regularly updated with new data to ensure that it continues to provide accurate predictions. This can be done by retraining the models with new data and by continuously improving the machine learning algorithms used by the system.

3.2 DFD/UML Diagrams

3.2.1 DFD Diagrams

A DFD is a graphical representation of how the data flows through a system. Developing a DFD is one of the first steps carried out when developing an information system. DFD displays details like the data that is coming in and going out of the system, how the data is travelled through the system and how the data will be stored in the system. But the DFD does not contain information about timing information of the processes. The main components included in a DFD are processes, data stores, data flow and external entities. When developing DFD diagrams, the context level DFD is drawn first. It displays how the entire system interacts with external data sources and data sinks. Next a Level 0 DFD is developed by expanding the context level DFD. Level 0 DFD contains details of the sub-systems within the system and how the data is flowing through them. It also contains details about the data stores required within the system. The entire working or the flow of the data can be divided into three groups for better understanding. They are- 1. DFD-L0 2. DFD-L1 3. DFD-L2.

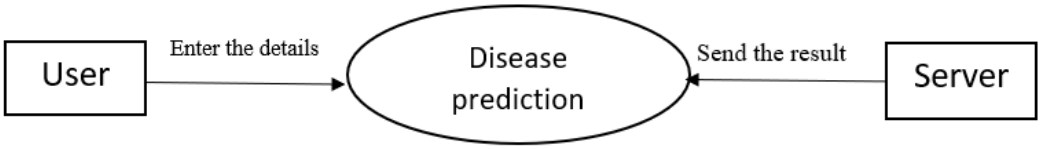


Fig 3.1: Level-0 DFD

This is the initial idea for the flow of the data. The data has to be flown from user to server and from server to the user for the prediction of the disease by entering details and sending the data. Communication is done between user and the server**.** Users, the main process, and data flow make up its parts. Also, the project concept is demonstrated using the single process visualization. DFD Level 0 shows the entities that interact with a system and defines the border between the system and its environment. The illustration presents the main process in a single node to introduce the project context. This context explains how the project works in just one look. The user feeds data into the system and then receives the output/report from it. In addition to this, you will perceive through the diagram that there is already the presence of data flow. Though the process is very general, the flow of data is clear. Nevertheless, just modify this diagram to meet the other requirements and include other matters regarding Multiple Disease Prediction.

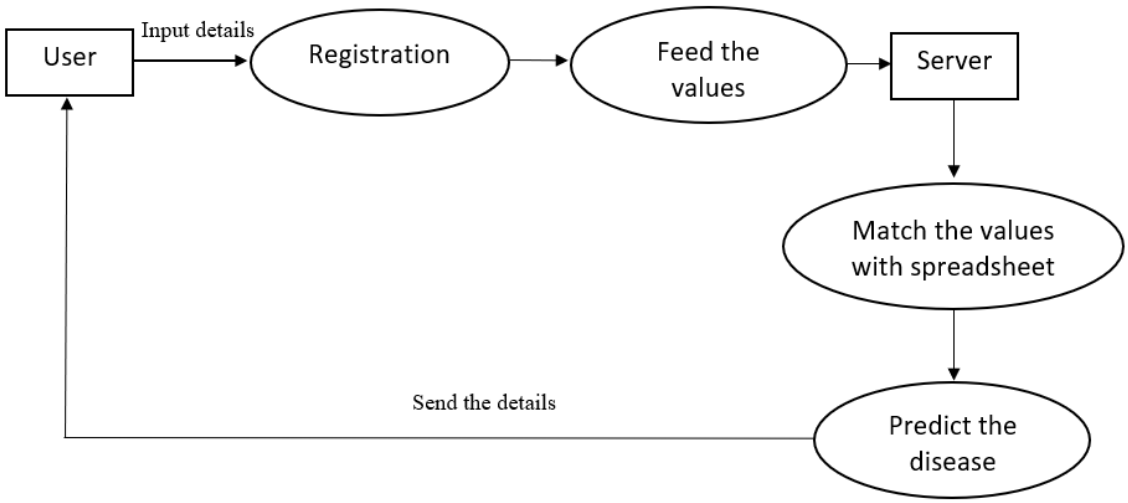


Fig 3.2: Level-1 DFD

This is the process or the idea where the data has been used to predict the disease by following several steps like registration (for new users), Feed the values(entering and storing values), Server(to store them), match the values(Finding probability) and finally predict the disease (Final result). The registered users can login to their account and can enter the values that is data and then can store them in the spreadsheet with the help of the server and then extract those values and find probability and then generate the report as similar to the newly registered users.

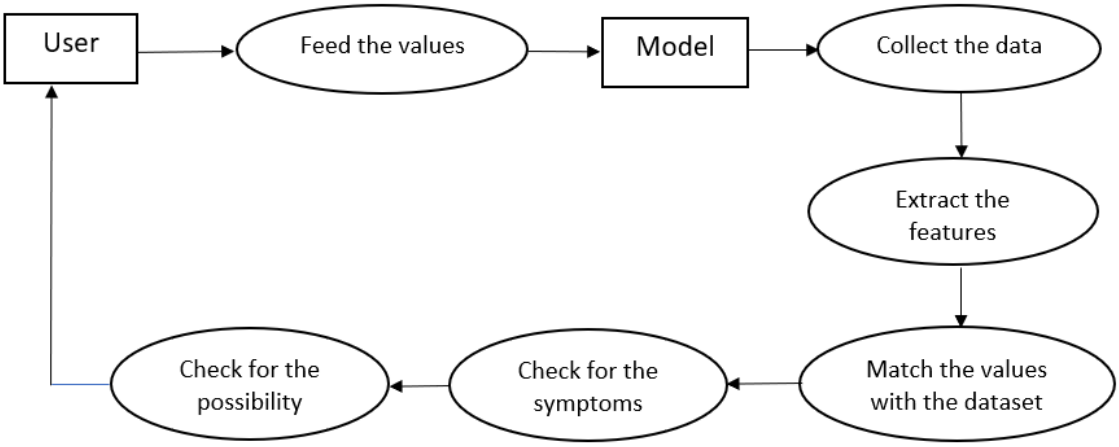


Fig 3.3: Level-2 DFD

The data which is flown from user to the server, there it undergoes matching for data from the user(input) and the data which we have i.e., datasets(train data). Finding probability between them by comparing the values and then generating the report. The DFD level-2 shows the processes involved in the machine learning system. The system starts with the input of patient data (symptoms and medical history) from the patient entity. The data is then processed through various stages, including data preprocessing, feature extraction, model selection, model training, model evaluation, model tuning, and model deployment. The preprocessed data is stored in the Preprocessed Data store, and the trained model is stored in the Trained Model store. The system then outputs the predicted disease diagnosis to the patient entity. Overall, the DFD level-2 provides a detailed illustration of how data flows through the machine learning system, from the input of patient data to the output of predicted disease diagnosis.

3.2.2 UML Diagrams

UML is a different language used in object oriented software design. UML provides capabilities to specify and visualize the components that make up a software system. UML diagrams mainly represent the structural view and the behavioural view of a system. Structural view of the system is represented using diagrams like class diagrams, composite structure diagrams, etc. Dynamic view of the system is represented using diagrams such as sequence diagrams, activity diagrams, etc. UML version 2.2 includes fourteen diagrams, which includes seven diagrams for representing the structural view and other seven representing the behavioural view. Among the seven behavioural diagrams, four diagrams can be used to represent interactions with the system. There are tools that can be used for UML modelling such as IBM Rational Rose.

Unified Modelling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. It offers a standard way to write a system's blueprints, including conceptual things such as processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.

Unified Modelling Language (UML) is a standardized general-purpose modelling language in the field of object-oriented software engineering. UML includes a set of graphic notation techniques to create visual models of object- oriented software systems. UML combines techniques from data modelling, object modelling, and component modelling and can be used throughout the seware development life-cycle and across 23ifferent implementation technology.

Modelling

There is a difference between a UMI model and the set of diagrams of syste A diagram is a partial graphic representation of system's model. The model also collains documentation that drives the model elements and diagrams (such as written use cases). UML diagrams represent two different views of a system model.

Static (or structural) view

This view emphasizes the static structure of the system using objects, attributes, operations, and relationships.

Ex: Class diagram, Composite Structure diagram.

Dynamic (or behavioral) view

This view emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects.

Ex Sequence diagram, activity diagram, state chart diagram

Goals of UML

1. A picture is worth a thousand words, this idiom absolutely fits describing UML Object-oriented concepts were introduced much earlier than UML. At that point of time, there were no standard methodologies to organize and consolidate the object-oriented development. It was then that UML came into picture.
2. There are a number of goals for developing UML but the most important is to define some general-purpose modelling language, which all modelers can use and it also needs to be made simple to understand and use.
3. UML diagrams are not only made for developers but also for business users, common.
4. people, and anybody interested to understand the system. The system can be a software or non-software system. It must be clear that UML is not a development method rather it accompanies processes to make it a successful system.
5. In conclusion, the goal of UML can be defined as a simple modelling mechanism to model all possible practical systems in today's complex environment.

In this project ,basic UML diagrams have been explained

- Use Case Diagram
- Sequence Diagram
- Activity Diagram
- Class diagram
- Component diagram

3.2.2.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagrams can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

The purpose of the use case diagrams is simply to provide the high-level view of the system and convey the requirements in laypeople's terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system. UML Use case diagrams are ideal for:

- Representing the goals of system-user interactions.
- Defining and organizing functional requirements in a system.
- Specifying the context and requirements of a system.
- Modelling basic flow of events in a use case.

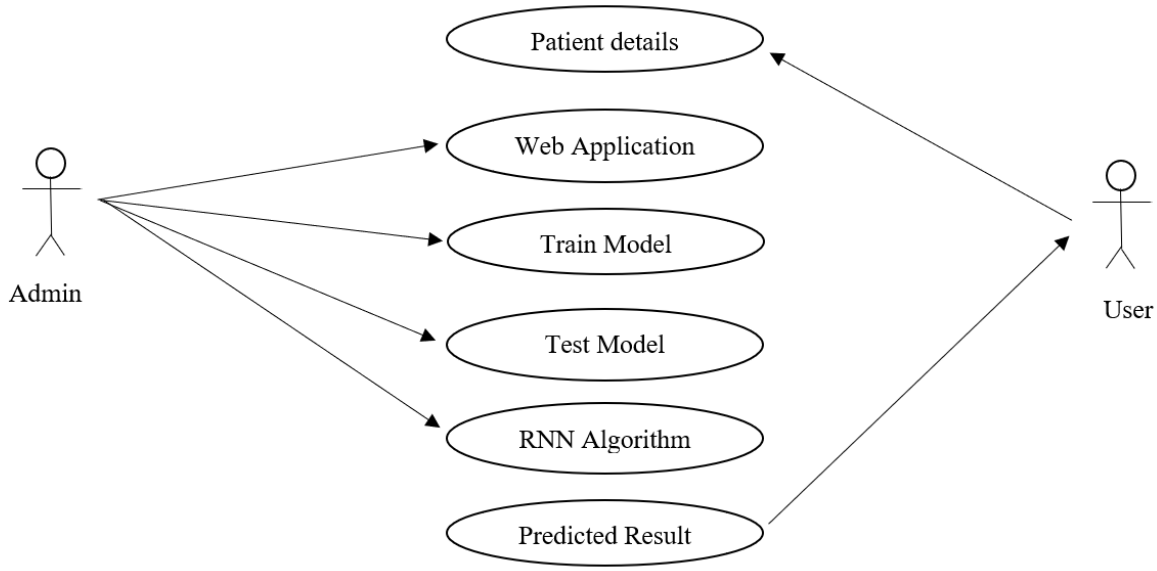


Fig 3.4: Use Case Diagram

3.2.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

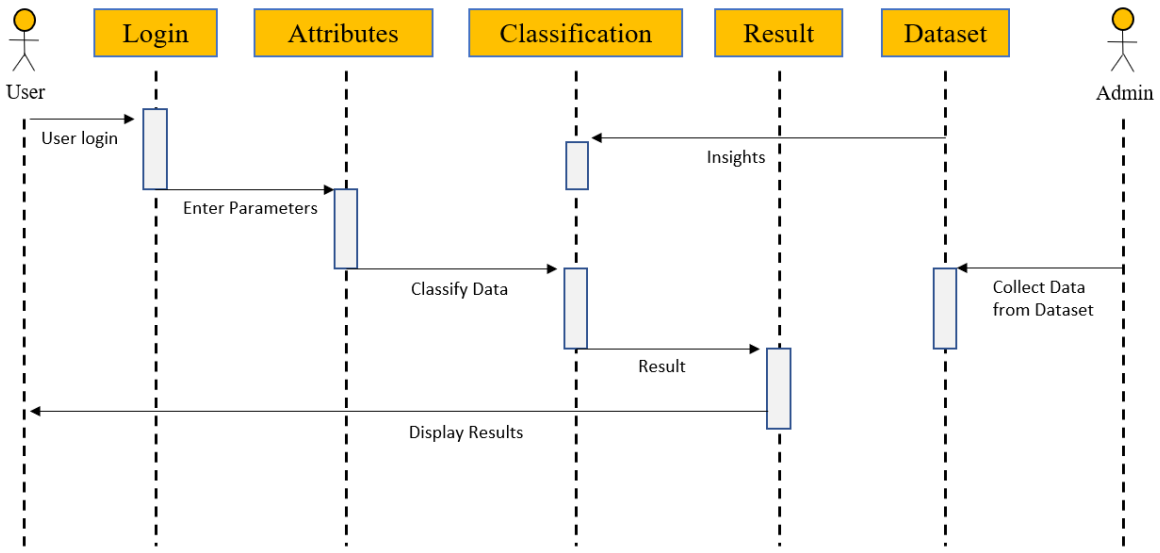


Fig 3.5: Sequence diagram

3.2.2.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the UMI activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

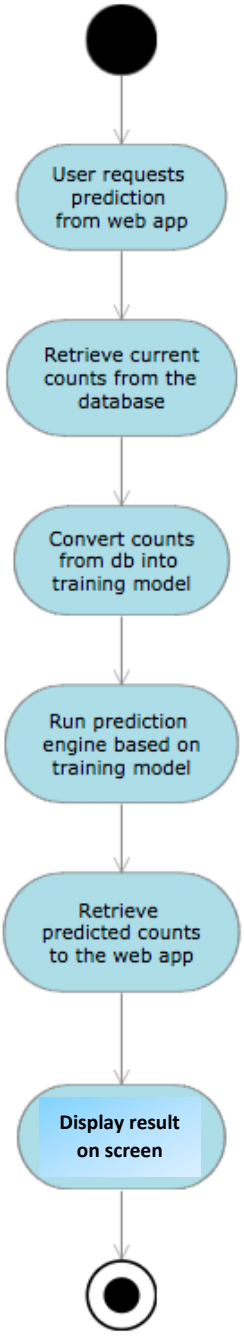


Fig 3.6: Activity Diagram

3.2.2.4 Class Diagram

In software engineering a class diagram in the unified modelling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations for methods), and the relationships among objects. The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for desailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.
- Class diagram helps construct the code for the software application development.

Fig 3.7: Class Diagram

3.2.2.5 Component Diagram

In the Unified modelling language (UML), a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements. System administrators can use component diagrams to plan ahead, using the view of the logical software components and their relationships on the system.

You can use component diagrams to show how software systems work at a high level, or you can use them to show how each component works at a lower level, like in a package.

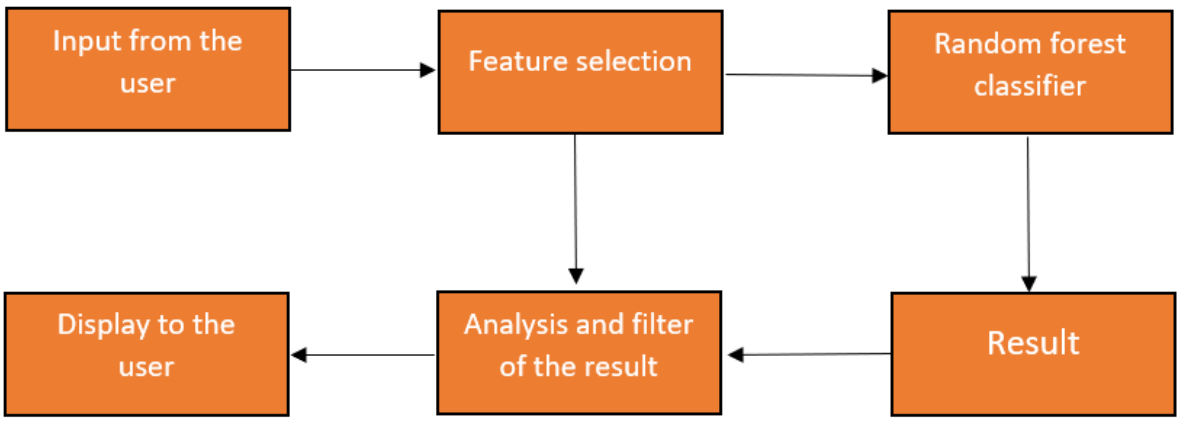


Fig 3.8: Component Diagram

3.3 Software Development Life Cycle (SDLC)

In our project we have used waterfall model as our software development cycle because of its step-by-step procedure while implementing system.

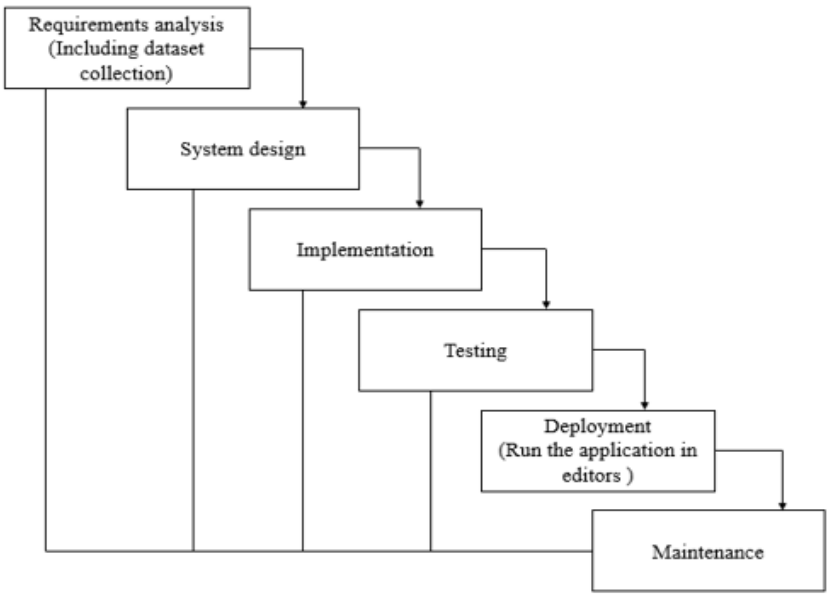


Fig. 3.9: Waterfall model

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design:** The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation:** The inputs from the system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after the testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues that come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Feasibility Study

The feasibility of the project is analysed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand for the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social feasibility:

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. IMPLEMENTATION & DETAILS

4.1 Introduction

A multiple disease predicting system using machine learning is a computational system that can predict the likelihood of a person having one or more diseases based on their medical history and other relevant factors. Machine learning algorithms are used to analyse the large amount of medical data and learn patterns that can be used to predict the disease risk. The implementation of such a system involves several steps, Firstly, the data must be collected and processed. This involves cleaning the data, removing any outliers missing values, and normalizing the data if necessary. The data can come from a variety of sources, such as electronic health records, medical imaging, or genetic testing. Once the data has been prepared, the next step is to train the machine learning algorithm. this typically works by analysing large amount of medical data from patients with different health condition, symptoms, and medical histories. The system uses advanced machine learning algorithms to identify patterns and correlations in the data that may indicate the presence or likelihood of specific diseases.

The system may also be updated periodically as new medical data becomes available to ensure that it remains accurate and up-to-date. This involves feeding the algorithm a large dataset of medical records and associated disease outcomes, and allowing it to learn patterns in the data. The algorithm can be trained using a variety of supervised learning techniques, such as logistic regression, decision trees, or neural networks. After the algorithm has been trained, it can be tested and evaluated using a separate dataset of medical records that it has not seen before. This is important to ensure that the algorithm is able to generalize well to new data and is not simply memorizing the training set. Once the algorithm has been evaluated and found to be accurate, it can be deployed as a predictive tool for healthcare professionals. This may involve integrating the algorithm into an electronic health record system, or providing it as a standalone web-based tool that can be accessed by clinicians. Overall, the implementation of a multiple disease predicting system using machine learning is a complex process that requires expertise in data collection, preprocessing, machine learning, and healthcare.

4.1.1 How Does Multiple Disease Predicting System Works?

A multiple disease predicting system using machine learning typically works by analyzing large amounts of data from various sources, such as medical records, patient symptoms, lab results, and demographic information. The system then uses this data to build predictive models that can identify patterns and correlations between different factors and the presence of specific diseases.

The process of building a disease prediction model typically involves several steps, including data collection, data cleaning, feature engineering, model selection, and model training. During data collection, the system gathers relevant data from various sources and formats it for analysis. In the data cleaning step, the system removes any irrelevant or inaccurate data and resolves any inconsistencies or errors.

Overall, a multiple disease predicting system using machine learning relies on sophisticated algorithms and large amounts of data to identify patterns and correlations that can help predict the likelihood of specific diseases. By analyzing a wide range of factors, these systems can potentially improve early diagnosis and treatment of diseases, leading to better patient outcomes.

4.2 Explanation of Key Features

The procedure is as follows:

1. In this research work with data with attributes are observable and then all of them are floating data. And there's a decision class/class variable. This data was collected from Kaggle machine learning repository.
2. In this research 70% data use for train model and 30% data use for testing purpose.
3. Random Forest is used as Classifier.
4. In the classification report we were able to find out the desired result.
5. In this analysis the result depends on some part of this research. However, which algorithm gives the best true positive, false positive, true negative, and false negative are the best algorithms in this analysis.

4.3 Methods Of Implementation

4.3.1 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used in data science and machine learning projects because it provides an interactive environment for data analysis and visualization.

In the context of a multiple disease prediction system using machine learning, Jupyter Notebook can be used to develop and implement the predictive models. You can use Python, along with machine learning libraries such as Scikit-learn, TensorFlow, Keras, and PyTorch, to build the predictive models.

You can also use data preprocessing libraries such as NumPy and Pandas to clean and prepare the disease datasets for machine learning. Once the data is prepared, you can use machine learning algorithms such as Logistic Regression, Random Forest, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) to train and test the predictive models.

With Jupyter Notebook, you can easily create visualizations and graphs to analyze and interpret the disease datasets. You can use visualization libraries such as Matplotlib and Seaborn to create plots, histograms, scatter plots, and other types of visualizations.

Jupyter Notebook also allows you to document your code and results in a narrative format, making it easy to share your work with others. You can export your Jupyter Notebook as a PDF or HTML file, or you can share it on GitHub or other online platforms.

Overall, Jupyter Notebook provides a powerful and flexible environment for developing and implementing multiple disease prediction systems using machine learning.

4.3.2 Python IDE

Python IDE (Integrated Development Environment) is a software application that provides a comprehensive environment for developing, testing, and debugging Python code. In the context of a multiple disease prediction system using machine learning, Python IDE can be used to write and execute Python code for data analysis, data preprocessing, and machine learning model development.

There are several Python IDEs available for data science and machine learning projects, including:

1. PyCharm: PyCharm is a popular Python IDE developed by JetBrains. It provides features such as code analysis, debugging, and code completion, making it an ideal choice for machine learning projects.
2. Spyder: Spyder is an open-source Python IDE that is designed for scientific computing and data analysis. It includes features such as a code editor, variable explorer, and a debugger.
3. Visual Studio Code: Visual Studio Code is a lightweight and cross-platform IDE that supports many programming languages, including Python. It provides features such as debugging, code completion, and syntax highlighting.

Visual Studio code is used for implementing our project "Multiple disease prediction system using machine learning".

4.3.3 Flask

- Flask is a web framework for building web applications in Python. It is lightweight and easy to use, making it a popular choice for building web applications, including those related to multiple disease prediction systems using machine learning.
- In the context of a multiple disease prediction system, Flask can be used to build a web interface for users to interact with the machine learning model. For example, users can input their medical data, such as age, gender, and medical history, into a web form, and Flask can use this data to make predictions using the machine learning model.
- Flask provides several features that make it well-suited for building web applications, including:
    1. Routing: Flask provides a routing system that allows developers to define the URLs for different pages of the web application.
    2. Templates: Flask includes a template engine that allows developers to create HTML templates for the web application.
    3. Sessions: Flask provides a session management system that allows developers to store user data between requests.
    4. Integration with machine learning libraries: Flask can easily integrate with machine learning libraries such as scikit-learn and TensorFlow to build and deploy machine learning models.

Overall, Flask is a powerful tool for building web applications for multiple disease prediction systems using machine learning. It is easy to use, flexible, and can be integrated with other Python libraries and tools.

4.3.4 Visual Studio Code

Visual Studio Code (VS Code) is a popular open-source code editor that provides a range of features and extensions that can be helpful for implementing a multiple disease prediction system using machine learning. Here are some tips for using VS Code:

1. Install the necessary extensions: VS Code has a wide range of extensions that can be helpful for data analysis and machine learning. Some useful extensions for implementing a multiple disease prediction system might include Python, Jupyter Notebook, and the various machine learning libraries such as scikit-learn, Tensorflow, or PyTorch.
2. Create a project directory: Organize your project by creating a directory with subdirectories for data, scripts, and other necessary files. Use version control tools like Git to track changes and collaborate with others.
3. Write Python code: Use the Python extension in VS Code to write Python code for data preprocessing, feature selection, model selection, and training. Use Jupyter Notebooks to perform exploratory data analysis and prototype the machine learning models.
4. Debug and test: VS Code has built-in debugging tools that can help you troubleshoot issues in your code. Use the debugger to step through your code and identify errors. Use testing frameworks like pytest or unittest to ensure that your code is working correctly.
5. Deploy the model: Use VS Code to create a web application or API for deploying the trained model. Use web frameworks like Flask or Django to create a user interface and handle input and output data.
6. Use VS Code's machine learning tools: VS Code provides various machine learning tools that can be helpful for implementing a multiple disease prediction system. For example, you can use the IntelliSense feature to get code suggestions and auto-completion for machine learning functions, or use the built-in terminal to execute Python scripts.

Overall, VS Code can be a powerful tool for implementing a multiple disease prediction system using machine learning, providing a range of features and extensions that can help you write, test, and deploy your code efficiently.

4.3.5 RANDOM FOREST ALGORITHM

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It isbased on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

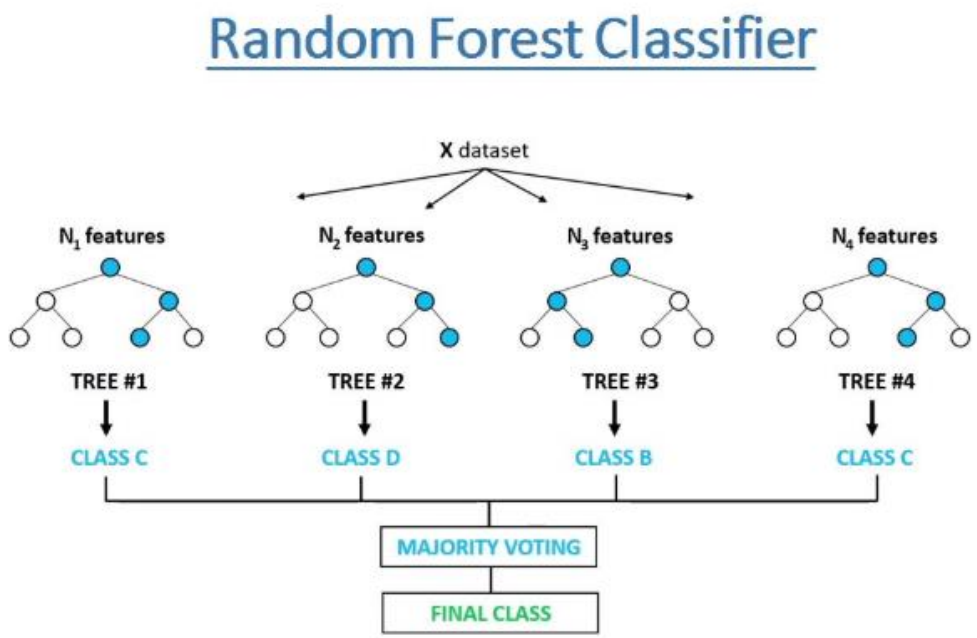


Figure 4.1: Random Forest Algorithm Flow Chart

Note: To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.

Assumptions for Random Forest:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

The predictions from each tree must have very low correlations.

Why use Random Forest:

Below are some points that explain why we should use the Random Forest algorithm:

- Takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps:

- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points (Subsets).
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.
- Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Advantages of Random Forest:

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

4.4 Sample Code

cancer.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("F:/projects/multiple-disease-prediction-final/Dataset/cancer.csv")
data.head()
data.info()
data.isnull().sum()
data['diagnosis'].unique()
dataset = data
dataset['diagnosis'].replace(["M","B"],[1,0],inplace=True)
del dataset['Unnamed: 32']
dataset.head()
dataset.groupby('diagnosis').size()
sns.set_style('whitegrid')
sns.countplot(x = 'diagnosis', data = dataset)
plt.figure(figsize=(25,25))
sns.heatmap(dataset.corr(),annot=True)
```

```
dataset.drop(['id','fractal_dimension_mean','texture_se','smoothness_se','symmetry_se'],axis
=1,inplace=True)

plt.figure(figsize=(25,25))

sns.heatmap(dataset.corr(),annot=True)

X = dataset.drop('diagnosis',axis=1)

y = dataset['diagnosis']

from sklearn.model_selection import train_test_split as tts

X_train,X_test,y_train,y_test = tts(X,y,test_size=0.2,random_state=42)

print("Train set size ",X_train.shape,y_train.shape)

print("Test set size ",X_test.shape,y_test.shape)

from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=20)

model.fit(X_train,y_train)

from sklearn.metrics import confusion_matrix , accuracy_score

predict = model.predict(X_test)

confusion_matrix(y_test,predict)

print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100,2)}")

import pickle

pickle.dump(model,open("cancer.pkl",'wb'))
```

diabetes.py

```
import numpy as np

import pickle

import pandas as pd

# Load the data set

df = pd.read_csv("F:/projects/multiple-disease-prediction-final/Dataset/diabetes.csv")

# remane the DiabetesPredictionFunction as DPF for making our work easier

df = df.rename(columns = {'DiabetesPredictionFunction':'DPF'})

# Replacing the 0 values from ['Glucose','BloodPressure','SkinThickness','Insulin','BMI'] by
NaN(Not a number)

df_copy = df.copy(deep=True)
```

```
df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] =
df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0,np.NaN)
# Replacing NaN value by mean ,median depending upon distribution
df_copy['Glucose'].fillna(df_copy['Glucose'].mean(),inplace=True)
df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(),inplace=True)
df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(),inplace=True)
df_copy['Insulin'].fillna(df_copy['Insulin'].median(),inplace=True)
df_copy['BMI'].fillna(df_copy['BMI'].median(),inplace=True)
# Model Building
from sklearn.model_selection import train_test_split
X = df.drop(columns='Outcome')
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,random_state=0)
# Create Random Forest Model
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 20)
classifier.fit(X_train,y_train)
from sklearn.metrics import confusion_matrix , accuracy_score
predict = classifier.predict(X_test)
confusion_matrix(y_test,predict)
print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}")
# Creating a pickel file for the classifier
filename = 'diabetes-prediction-rfc-model.pkl'
pickle.dump(classifier,open(filename,'wb'))
```

heart.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("F:/projects/multiple-disease-prediction-final/Dataset/heart.csv")
data.head()
```

```
data.info()
data.isnull().sum()
data.describe()
print(data.target.value_counts())
sns.set_style('whitegrid')
sns.countplot(x = 'target', data = data)
categorical_variable = []
continuous_variable = []
for column in data.columns:
  if(len(data[column].unique())<=10):
    categorical_variable.append(column)
  else:
    continuous_variable.append(column)
categorical_variable
continuous_variable
dataset = data.copy()
dataset.head()
corr = data.corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr,annot=True)
X = dataset.drop(columns='target',axis=1)
y = dataset['target']
print("Shape of dataset ",dataset.shape)
print("Shape of X ", X.shape)
print("Shape of Y ",y.shape)
print(X.columns)
from sklearn.model_selection import train_test_split as tts
from sklearn.ensemble import RandomForestClassifier
X_train,X_test ,y_train,y_test = tts(X,y,test_size=0.2,random_state = 42)
model1 = RandomForestClassifier(n_estimators=20)
model1.fit(X_train,y_train)
pred1 = model1.predict(X_test)
```

```
pred1[:10]
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, pred1)
from sklearn.metrics import accuracy_score
print(f"Accuracy of model is {round(accuracy_score(y_test, pred1)*100, 2)}%")
from sklearn.model_selection import RandomizedSearchCV
classifier = RandomForestClassifier(n_jobs=-1)
from scipy.stats import randint
param_dist={'max_depth':[3,5,10,None],
            'n_estimators':[10,100,200,300,400,500],
            'max_features':randint(1,31),
             'criterion':['gini','entropy'],
             'bootstrap':[True,False],
             'min_samples_leaf':randint(1,31),
            }
search_clfr = RandomizedSearchCV(classifier, param_distributions = param_dist, n_jobs=
-1, n_iter = 40, cv = 9)
search_clfr.fit(X_train, y_train)
params = search_clfr.best_params_
score = search_clfr.best_score_
print(params)
print(score)
claasifier=RandomForestClassifier(n_jobs=1,n_estimators=400,bootstrap=False,criterion='g
ini',max_depth=5,max_features=3,min_samples_leaf= 7)
classifier.fit(X_train,y_train)
pred2 = classifier.predict(X_test)
confusion_matrix(y_test,pred2)
print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%")
import pickle
pickle.dump(classifier,open('heart.pkl','wb'))
```

kidney.py

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read_csv("F:/projects/multiple-disease-prediction-
final/Dataset/kidney_disease.csv")

data.head()

print(data.shape)

data['classification'].unique()

data.info()

data.classification = data.classification.replace("ckd\t","ckd")

data.classification.unique()

data.drop('id', axis = 1, inplace = True)

data.head()

data.isnull().sum()

data_drop = data.dropna(axis=0)

print(f"Data shape before deletion: {data.shape}")

print(f"Data shape after deletion: {data_drop.shape}")

data_drop.head()

for i in data_drop['wc']:

  print(i)

data_drop['wc']=data_drop['wc'].replace(["\t6200","\t8400"],[6200,8400])

for i in data_drop['wc']:

  print(i)

data_drop.info()

data_drop.isnull().any()

data_drop['pcv'] = data_drop['pcv'].astype('int')

data_drop['wc'] = data_drop['wc'].astype('int')

data_drop['rc'] = data_drop['rc'].astype('float')

data_drop.info()

object_data_drop = data_drop.select_dtypes(include='object')
```

```
object_data_drop.head()
dictonary = {
    "rbc":{
        "normal":0,
        "abnormal":1,
    },
    "pc":{
        "normal":0,
        "abnormal":1,
    },
    "pcc":{
        "present":1,
        "notpresent":0,
    },
    "ba":{
        "notpresent":0,
        "present":1,
    },
    "htn":{
        "yes":1,
        "no":0,
    },
    "dm":{
        "yes":1,
        "no":0,
    },
    "cad":{
        "yes":1,
        "no":0,
    },
    "appet":{
        "good":1,
```

```
        "poor":0,
    },
    "pe":{
        "yes":1,
         "no":0,
    },
    "ane":{
        "yes":1,
        "no":0,
    }
}
data_drop = data_drop.replace(dictonary)
data_drop.head()
plt.figure(figsize=(15,15))
sns.heatmap(data_drop.corr(),annot=True,fmt=".2f",linewidths=0.6)
data_drop.corr()
X = data_drop.drop(["sg","sod","hemo","pcv","rc","appet","classification"],axis=1)
y = data_drop['classification']
X.columns
from sklearn.model_selection import train_test_split as tts
X_train,X_test,y_train,y_test = tts(X,y,test_size=0.2,random_state=42)
print(f"Size of X_train: {X_train.size}")
print(f"Size of X_test: {X_test.size}")
print(f"Size of y_train: {y_train.size}")
print(f"Size of y_test: {y_test.size}")
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train,y_train)
from sklearn.metrics import confusion_matrix , accuracy_score
predict = model.predict(X_test)
confusion_matrix(y_test,predict)
```

```
print(f"Accuracy of model is {round(accuracy_score(y_test, model.predict(X_test))*100, 2)}%")

import pickle

pickle.dump(model,open("kidney.pkl","wb"))
```

liver.py

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import pickle

import warnings

warnings.simplefilter("ignore")

data=pd.read_csv("F:/projects/multiple-disease-prediction-final/Dataset/indian_liver_patient.csv")

data.head()

data['Albumin_and_Globulin_Ratio'].mean()

data.info()

data.isnull().sum()

data.Dataset.unique()

data['Dataset'] = data['Dataset'].replace([2,1],[1,0])

data['Dataset'].head(10)

data['Albumin_and_Globulin_Ratio']= data['Albumin_and_Globulin_Ratio'].fillna(0.9470639032815201)

data.isnull().sum()

data.describe()

data = pd.get_dummies(data,columns=['Gender'],drop_first=True)

data.head()

plt.figure(figsize=(20,14))

sns.heatmap(data.corr(),annot=True)

data.corr()

X = data.drop("Dataset",axis=1)

y = data['Dataset']
```

```
print(X.shape)
print(y.shape)
X.columns
from sklearn.model_selection import train_test_split as tts
X_train,X_test,y_train,y_test = tts(X,y,test_size=0.1,random_state=42)
print("Train shape", X_train.shape," ",y_train.shape,"\n")
print("Test shape", X_test.shape," ",y_test.shape)
from sklearn.ensemble import RandomForestClassifier
model1 = RandomForestClassifier(n_estimators=20)
model1.fit(X_train,y_train)
from sklearn.metrics import confusion_matrix,accuracy_score
# Printing confusion metrics
confusion_matrix(y_test,model1.predict(X_test))
print(f"Accuracy is {round(accuracy_score(y_test, model1.predict(X_test))*100,2)}")
from sklearn.metrics import confusion_matrix , accuracy_score
predict = model1.predict(X_test)
confusion_matrix(y_test,predict)
# Adding Second Model
from xgboost import XGBClassifier
model2=XGBClassifier(n_estimators=1000,learning_rate=0.075,max_depth=3,early_stopping_rounds=10,verbose=False)
model2.fit(X_train,y_train)
# Printing accuracy score
confusion_matrix(y_test,model1.predict(X_test))
print(f"Accuracy is {round(accuracy_score(y_test, model2.predict(X_test))*100,2)}")
pickle.dump(model1,open("liver.pkl",'wb'))
```

4.5 Output Screens

- **Screen1: Login Page**

The login page consits of two fields Email, Password which are necessary for users to use the system for predicting the specific disease. For new users, they have to register to use the system. For that, they have to click on “Create a Account”.

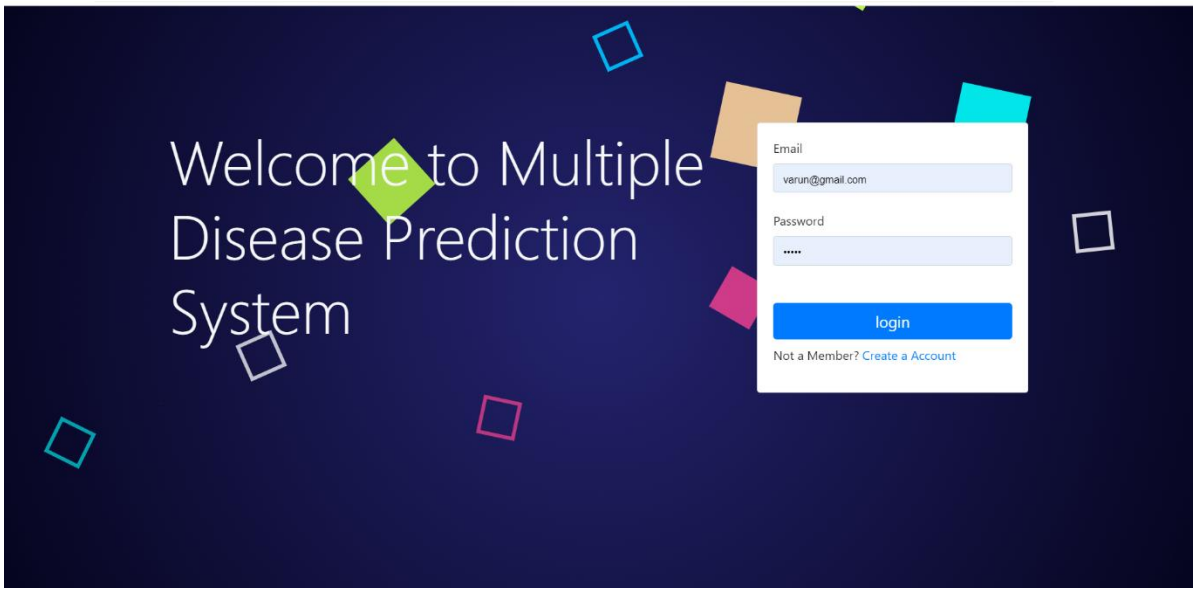


Fig 4.2: Login Screen

- **Screen2: Registration Page**

The registration page consists of three fields Name, Email and Password. If the existing user enters the details of them, it will show a message like ”User already exist”. So, make sure to enter the details into fields and remember.

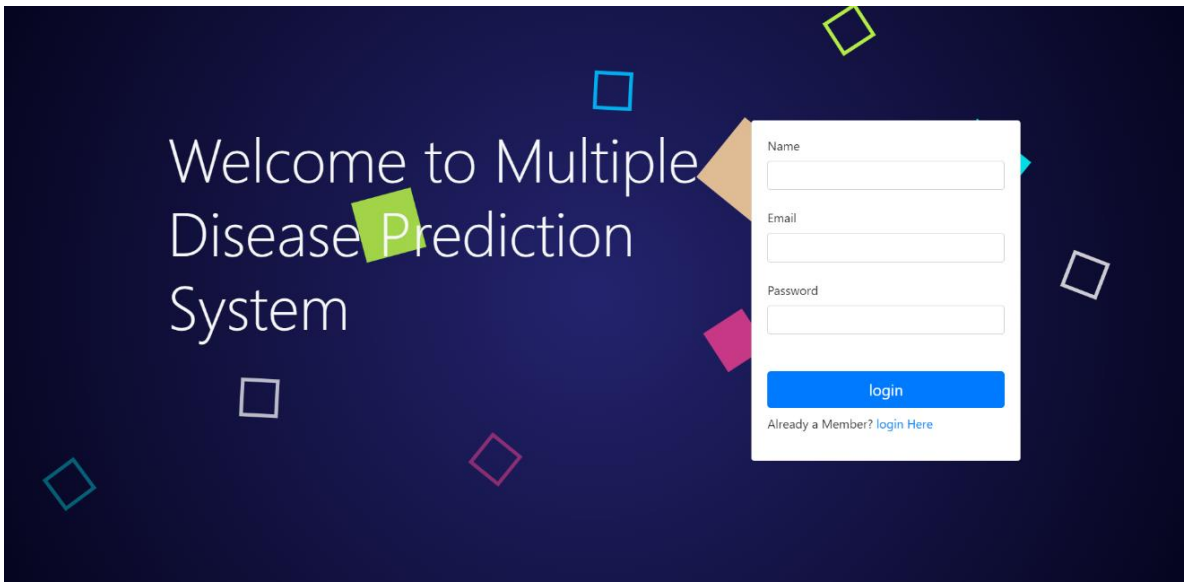


Fig 4.3: Registration Screen

Screen3: Diabetes disease Prediction Page

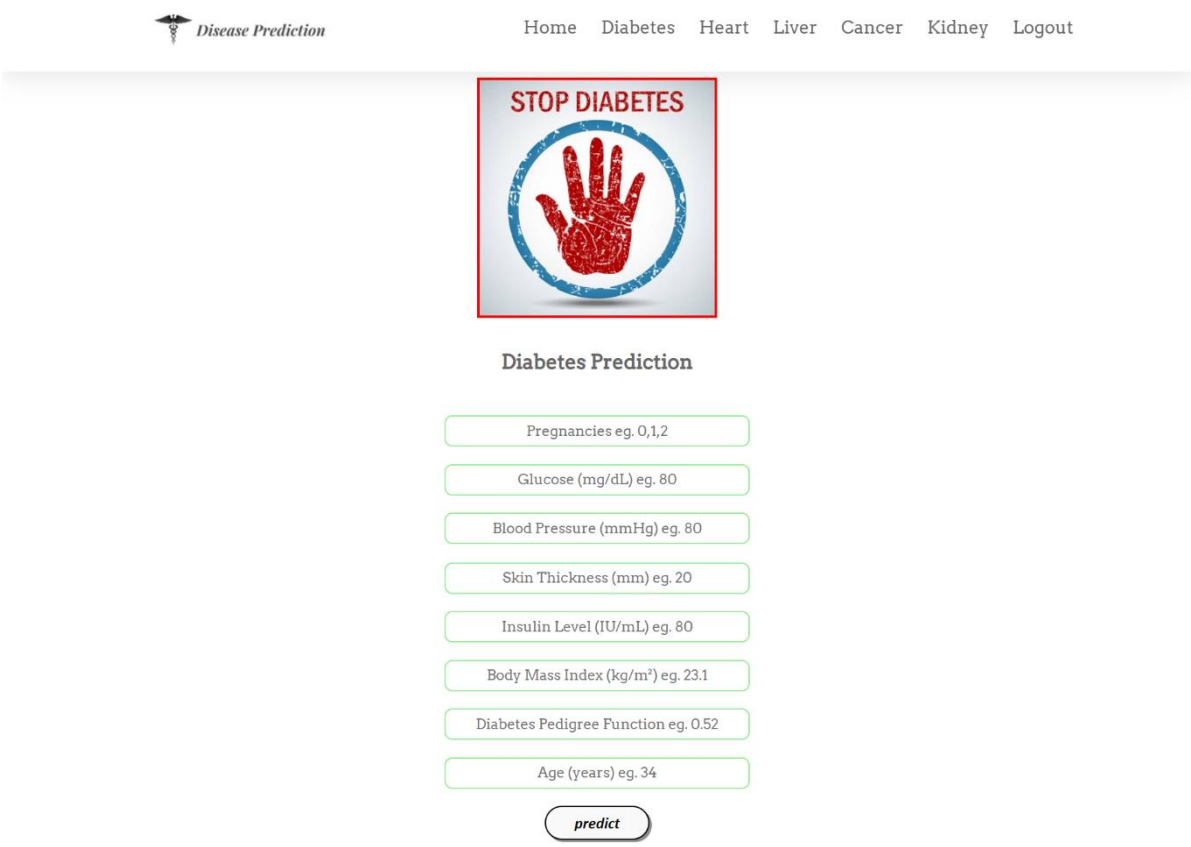


Fig 4.4: Diabetes disease prediction

This is the interface of diabetes disease prediction page. The user has to enter the values of parameters in the respective fields such as number of pregnancies, glucose level, blood pressure, thickness of the skin, insulin level, BMI value, DPF, Age. After entering the values, there is an option called Predict. Thus, whenever the user clicks on the predict button, the model evaluates the result based on the given dataset and used model for designing the above system. A testcase has been designed for this model as shown in the below diagram which predicts the result as “The user is not diagnosed with disease. They are healthy”.

Disease Prediction

Home Diabetes Heart Liver Cancer Kidney Logout

Diabetes Prediction

0

80

90

40

90

24

0.76

38

predict

Status

GREAT!! You are healthy

Home

Screen4: Heart disease Prediction Page

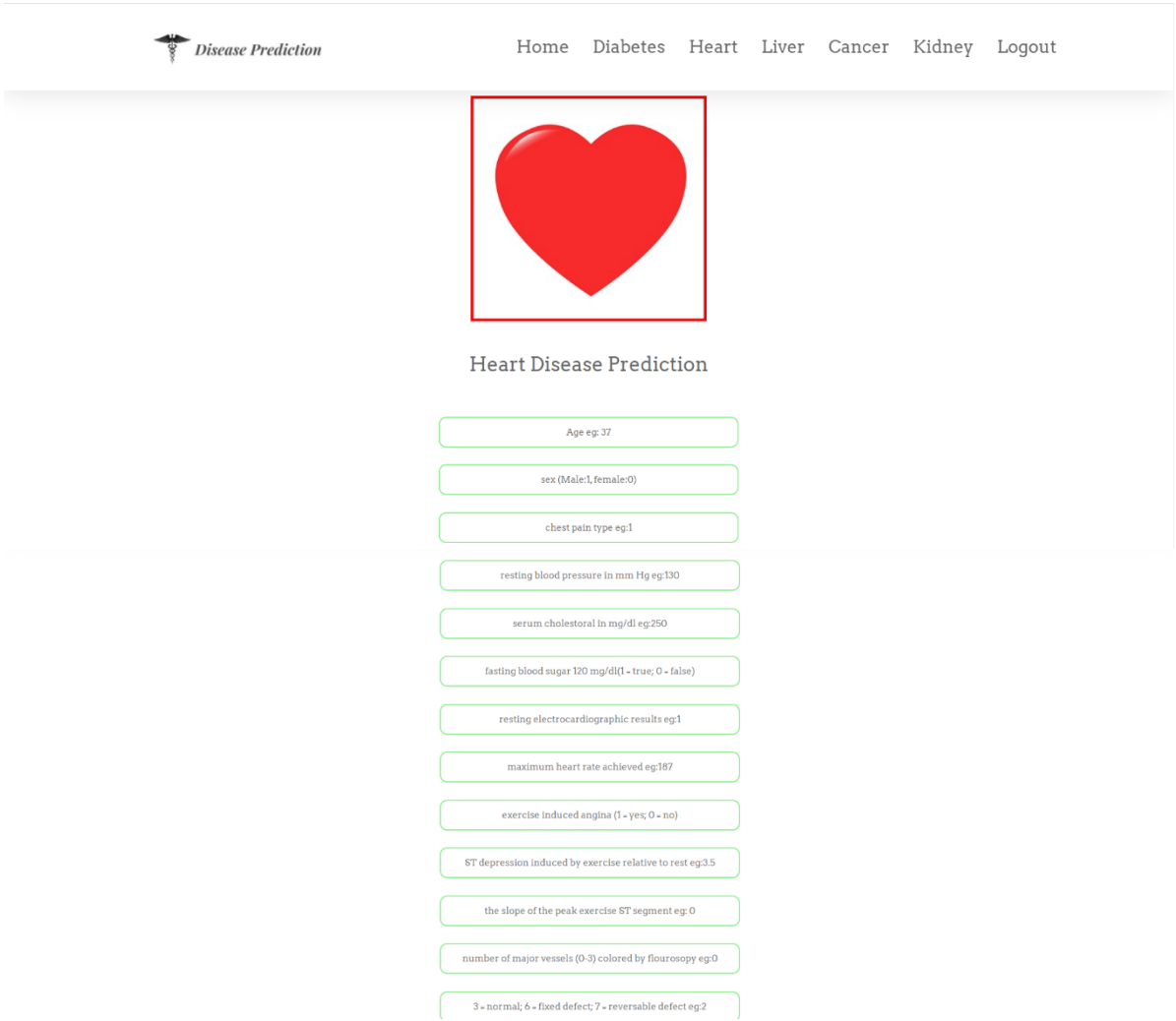


Fig. 4.5: Heart disease prediction

This is the interface of heart disease prediction page. The user has to enter the values of parameters in the respective fields such as Age, Gender, type of chest pain, blood pressure, serum cholesterol, blood sugar, max hear tare achieved, etc. After entering the values, there is an option called Predict. Thus, whenever the user clicks on the predict button, the model evaluates the result based on the given dataset and used model for designing the above system. A testcase has been designed for this model as shown in the below diagram which predicts the result as “The user is suffering from the disease. They not are healthy”.

Disease Prediction

Home Diabetes Heart Liver Cancer Kidney Logout

Heart Disease Prediction

63

1

3

145

233

1

0

150

0

2.3

0

0

1

Status

SORRY!!! You are suffering from disease 😔, please visit a doctor.

Home

Screen5: Liver disease Prediction Page

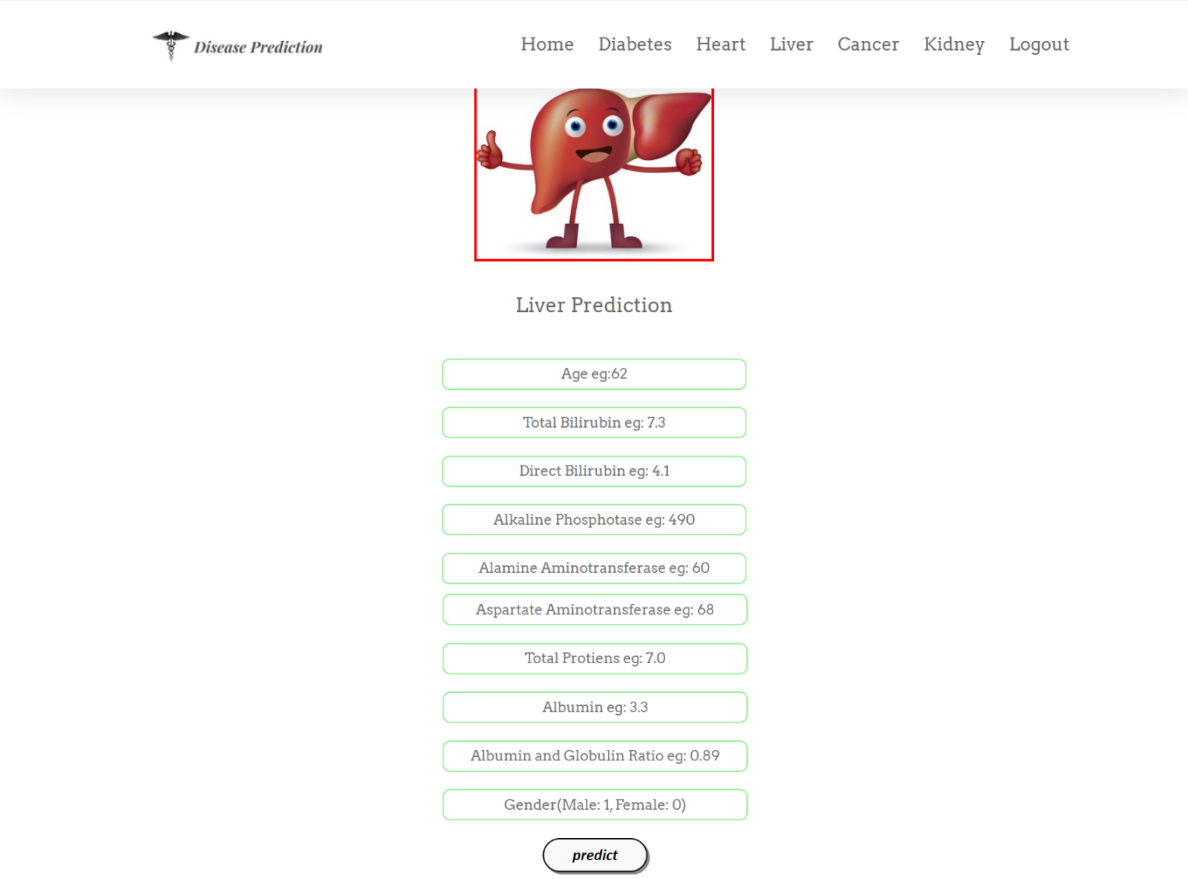
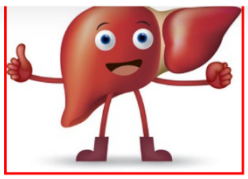


Fig. 4.6: Liver disease prediction

This is the interface of liver disease prediction page. The user has to enter the values of parameters in the respective fields such as Age, Gender, total bilirubin, alkaline phosphatase, aminotransferase, total proteins, albumin, albumin and globulin ratio, etc. After entering the values, there is an option called Predict. Thus, whenever the user clicks on the predict button, the model evaluates the result based on the given dataset and used model for designing the above system. A testcase has been designed for this model as shown in the below diagram which predicts the result as “The user is suffering from the disease. They not are healthy”.

Disease Prediction

Home Diabetes Heart Liver Cancer Kidney Logout



Liver Prediction

17

0.9

0.3

202

22

19

7.4

4.1

1.2

1

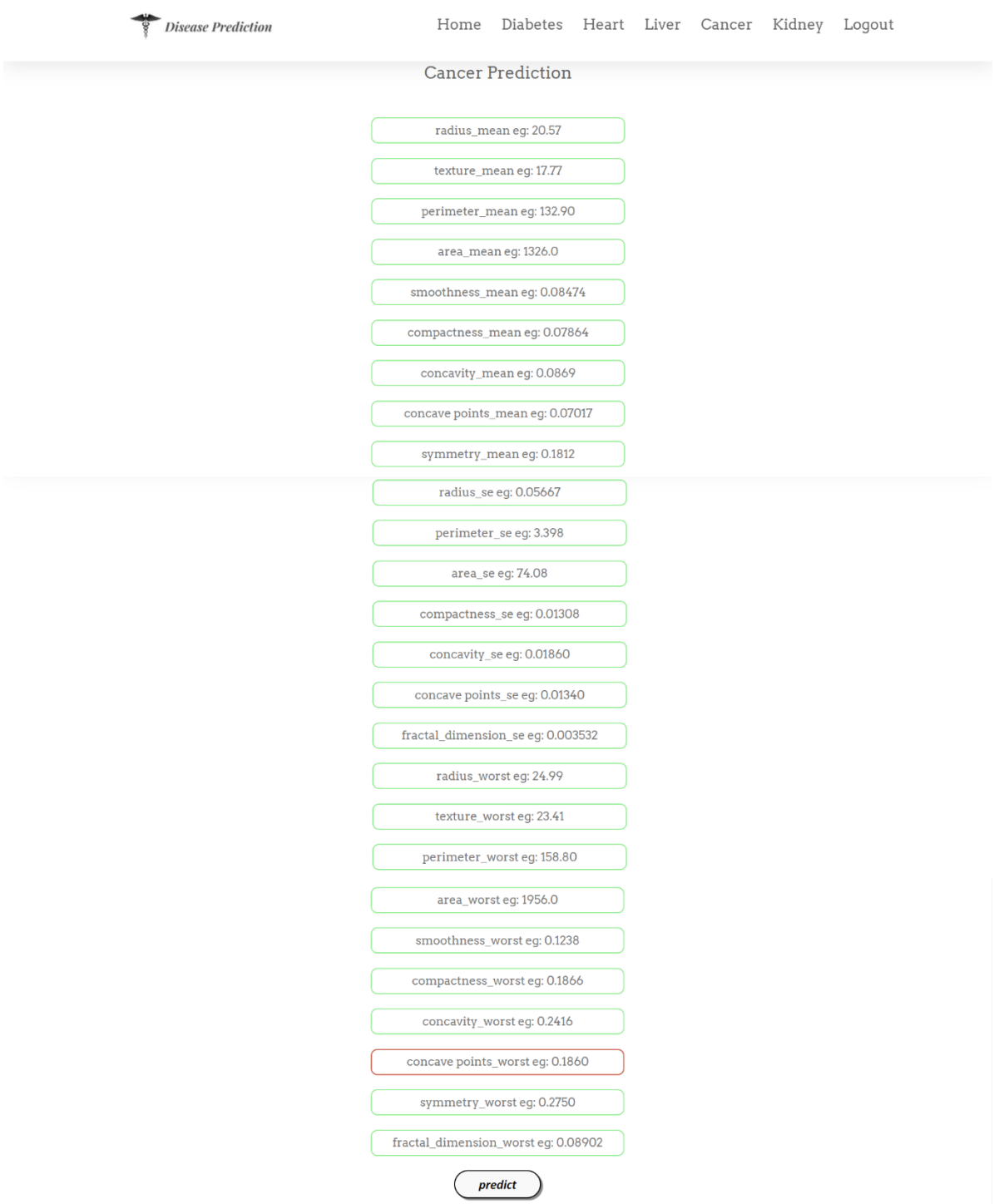
predict

Status

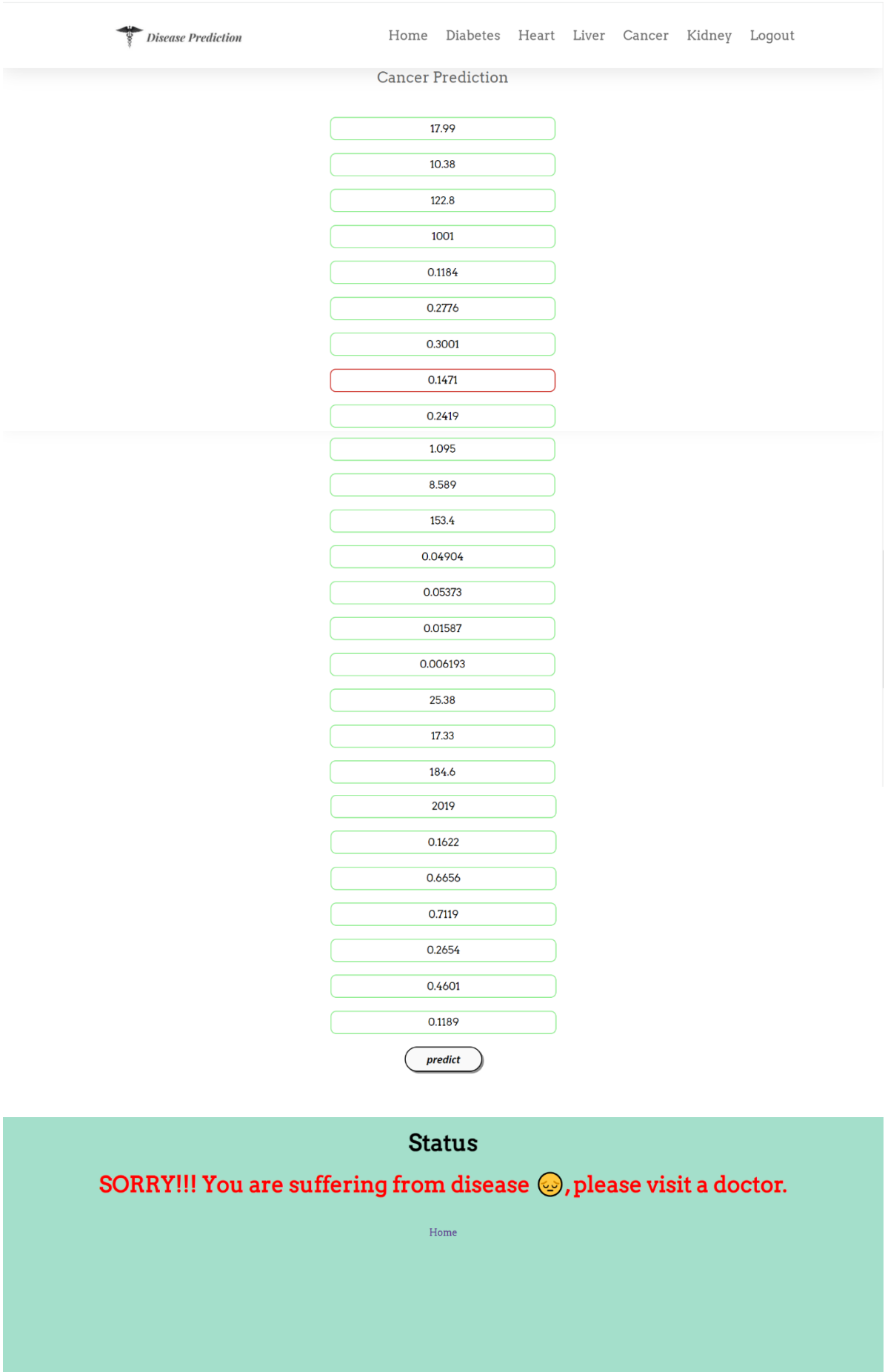
SORRY!!! You are suffering from disease 😔, please visit a doctor.

Home

Screen6: Cancer disease Prediction Page



This is the interface of liver disease prediction page. The user has to enter the values of parameters in the respective fields After entering the values, there is an option called Predict. Thus, whenever the user clicks on the predict button, the model evaluates the result based on the given dataset and used model for designing the above system.


Disease Prediction
Home
Diabetes
Heart
Liver
Cancer
Kidney
Logout
Cancer Prediction
17.99
10.38
122.8
1001
0.1184
0.2776
0.3001
0.1471
0.2419
1.095
8.589
153.4
0.04904
0.05373
0.01587
0.006193
25.38
17.33
184.6
2019
0.1622
0.6656
0.7119
0.2654
0.4601
0.1189
predict
Status
SORRY!!! You are suffering from disease 😔, please visit a doctor.
Home

Screen7: Kidney disease Prediction Page

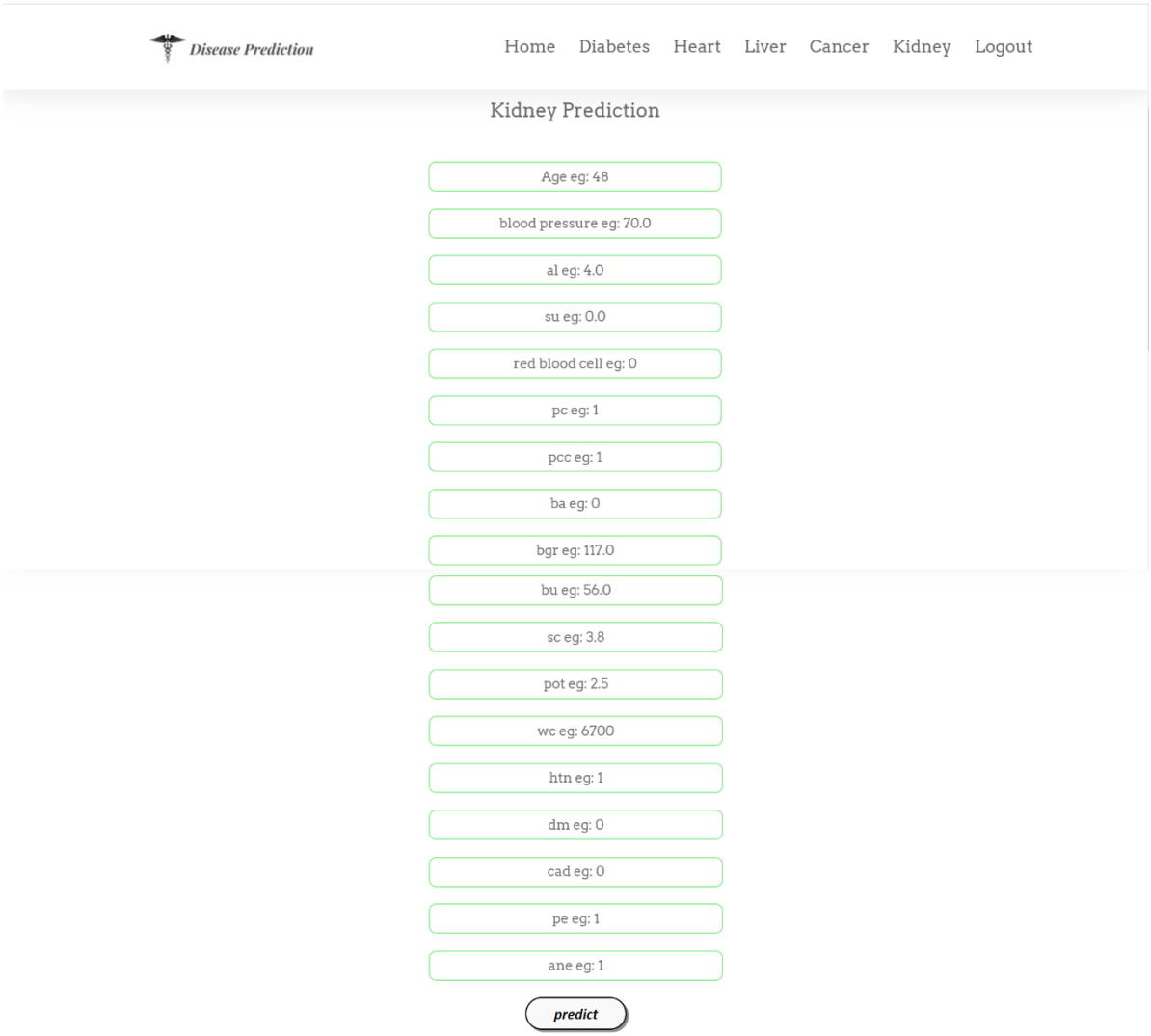
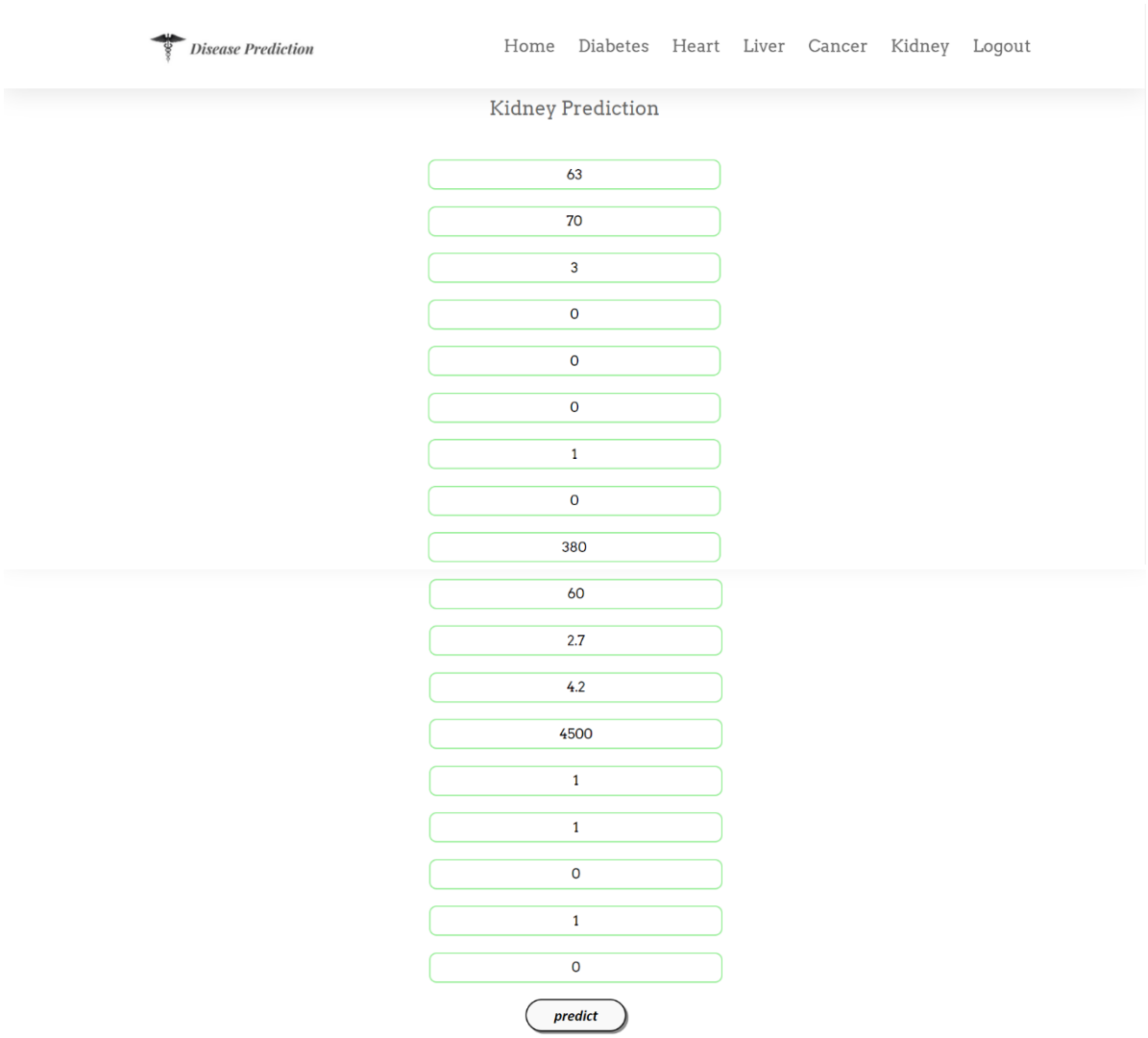
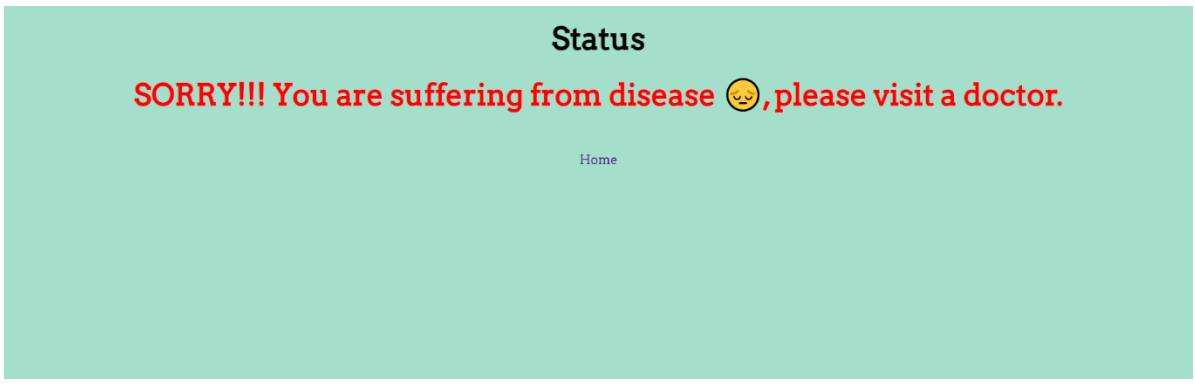


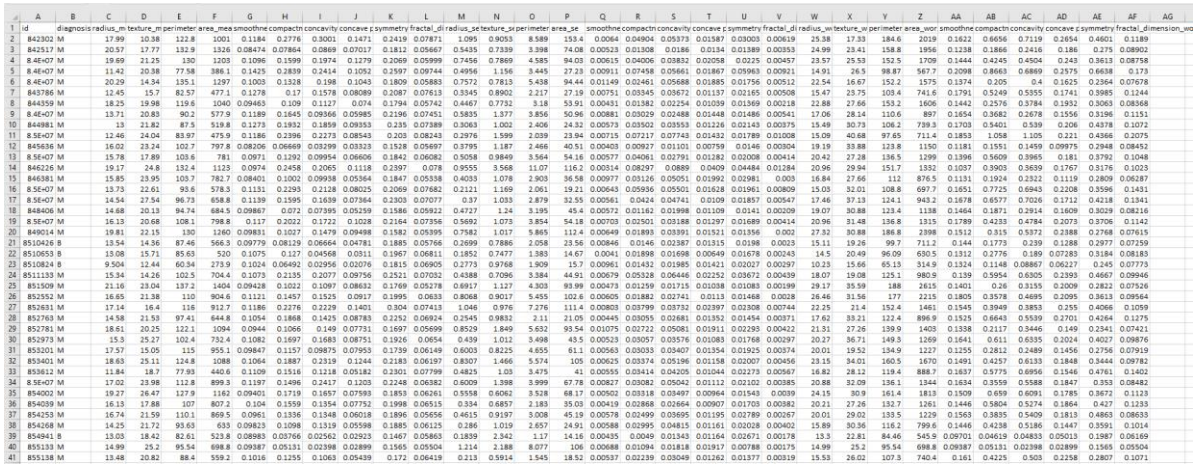
Fig. 4.8: Kidney disease prediction

This is the interface of kidney disease prediction page. The user has to enter the values of parameters in the respective fields such as Age, blood pressure, al, su, RBC count, pc, pcc, ba, etc. After entering the values, there is an option called Predict. Thus, whenever the user clicks on the predict button, the model evaluates the result based on the given dataset and used model for designing the above system. A testcase has been designed for this model as shown in the below diagram which predicts the result as “The user is suffering from the disease. They not are healthy”.


Disease Prediction
Home
Diabetes
Heart
Liver
Cancer
Kidney
Logout
Kidney Prediction
63
70
3
0
0
0
1
0
380
60
2.7
4.2
4500
1
1
0
1
0
predict


Status
SORRY!!! You are suffering from disease 😔, please visit a doctor.
Home

4.6 Datasets



id	diagnosis	radius_m	texture_m	perimeter	area_mea	smoothne	compactn	concavity	concave p	symmetry	fractal_di	radius_se	texture_se	perimeter	area_se	smoothne	compactn	concavity	concave p	symmetry	fractal_di	radius_wo	texture_wo	perimeter	area_wor	smoothne	compactn	concavity	concave p	symmetry	fractal_dimension_wo
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.0064	0.04904	0.05373	0.01587	0.03003	0.00619	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.00523	0.01308	0.0186	0.0134	0.01389	0.00353	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902
8.48E+07	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.00457	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758
8.48E+07	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.00921	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173
8.48E+07	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.00512	22.54	16.67	152.2	1575	0.1374	0.205	0.4	0.1625	0.2364	0.07678
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.00508	15.47	23.75	103.4	741.6	0.1791	0.5249	0.5355	0.1741	0.3985	0.1244
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.00431	0.01582	0.02254	0.01039	0.01369	0.00218	22.88	27.66	153.2	1606	0.1442	0.2576	0.3784	0.1932	0.3063	0.08368
8.48E+07	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.00881	0.03029	0.02488	0.01448	0.01486	0.00541	17.06	28.14	110.6	897	0.1654	0.3682	0.2678	0.1556	0.3196	0.1151
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.00573	0.03502	0.03553	0.01226	0.02143	0.00375	15.49	30.73	106.2	739.3	0.1703	0.5401	0.539	0.206	0.4378	0.1072
8.5E+07	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.00715	0.07217	0.07743	0.01432	0.01789	0.01008	15.09	40.68	97.65	711.4	0.1853	1.058	1.105	0.221	0.4366	0.2075
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.00403	0.00927	0.01101	0.00759	0.0146	0.00304	19.19	33.88	123.8	1150	0.1181	0.1551	0.1459	0.09975	0.2948	0.08452
8.5E+07	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.00577	0.04061	0.02791	0.01282	0.02008	0.00414	20.42	27.28	136.5	1299	0.1396	0.5609	0.3965	0.181	0.3792	0.1048
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.00514	0.08297	0.0889	0.0409	0.04484	0.01284	20.96	29.94	151.7	1332	0.1037	0.3903	0.3639	0.1767	0.3176	0.1023
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.00977	0.03126	0.05051	0.01992	0.02981	0.003	16.84	27.66	112	876.5	0.1131	0.1924	0.2322	0.1119	0.2809	0.06287
8.5E+07	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.00643	0.05936	0.05501	0.01628	0.01961	0.00809	15.03	32.01	108.8	697.7	0.1651	0.7725	0.6943	0.2208	0.3596	0.1431
8.5E+07	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.00561	0.0424	0.04741	0.0109	0.01857	0.00547	17.46	37.13	124.1	943.2	0.1678	0.6577	0.7026	0.1712	0.4218	0.1341
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.00572	0.01162	0.01998	0.01109	0.0141	0.00209	19.07	30.88	123.4	1138	0.1464	0.1871	0.2914	0.1609	0.3029	0.08216
8.5E+07	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.00703	0.02501	0.03188	0.01297	0.01689	0.00414	20.96	31.48	136.8	1315	0.1789	0.4233	0.4784	0.2073	0.3706	0.1142
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.00649	0.01893	0.03391	0.01521	0.01356	0.002	27.32	30.88	186.8	2398	0.1512	0.315	0.5372	0.2388	0.2768	0.07615
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.00846	0.0146	0.02387	0.01315	0.0198	0.0023	15.11	19.26	99.7	711.2	0.144	0.1773	0.239	0.1288	0.2977	0.07259
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.0041	0.01898	0.01698	0.00649	0.01678	0.00243	14.5	20.49	96.09	630.5	0.1312	0.2776	0.189	0.07283	0.3184	0.08183
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.00961	0.01432	0.01985	0.01421	0.02027	0.00297	10.23	15.66	65.13	314.9	0.1324	0.1148	0.08867	0.06227	0.245	0.07773
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.00679	0.05328	0.06446	0.02252	0.03672	0.00439	18.07	19.08	125.1	980.9	0.139	0.5954	0.6305	0.2393	0.4667	0.09946
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.00473	0.01259	0.01715	0.01038	0.01083	0.00199	29.17	35.59	188	2615	0.1401	0.26	0.3155	0.2009	0.2822	0.07526
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.00605	0.01882	0.02741	0.0113	0.01468	0.0028	26.46	31.56	177	2215	0.1805	0.3578	0.4695	0.2095	0.3613	0.09564
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.00803	0.03799	0.03732	0.02397	0.02308	0.00744	22.25	21.4	152.4	1461	0.1545	0.3949	0.3853	0.255	0.4066	0.1059
852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.00445	0.03055	0.02681	0.01352	0.01454	0.00371	17.62	33.21	122.4	896.9	0.1525	0.6643	0.5539	0.2701	0.4264	0.1275
852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.00422	21.31	27.26	139.9	1403	0.1338	0.2117	0.3446	0.149	0.2341	0.07421
852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498	43.5	0.00523	0.03057	0.03576	0.01083	0.01768	0.00297	20.27	36.71	149.3	1269	0.1641	0.611	0.6335	0.2024	0.4027	0.09876
853201	M	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655	61.1	0.00563	0.03033	0.03407	0.01354	0.01925	0.00374	20.01	19.52	134.9	1227	0.1255	0.2812	0.2489	0.1456	0.2756	0.07919
853401	M	18.63	25.11	124.8	1088	0.1064	0.1887	0.2319	0.1244	0.2183	0.06197	0.8307	1.466	5.574	105	0.00625	0.03374	0.05196	0.01158	0.02007	0.00456	23.15	34.01	160.5	1670	0.1491	0.4257	0.6133	0.1848	0.3444	0.09782
853612	M	11.84	18.7	77.93	440.6	0.1109	0.1516	0.1218	0.05182	0.2301	0.07799	0.4825	1.03	3.475	41	0.00555	0.03414	0.04205	0.01044	0.02273	0.00567	16.82	28.12	119.4	888.7	0.1637	0.5775	0.6956	0.1546	0.4761	0.1402
8.5E+07	M	17.02	23.98	112.8	899.3	0.1197	0.1496	0.2417	0.1203	0.2248	0.06382	0.6009	1.398	3.999	67.78	0.00827	0.03082	0.05042	0.01112	0.02102	0.00385	20.88	32.09	136.1	1344	0.1634	0.3559	0.5588	0.1847	0.353	0.08482
854002	M	19.27	26.47	127.9	1162	0.09401	0.1719	0.1657	0.07593	0.1853	0.06261	0.5558	0.6062	3.528	68.17	0.00502	0.03318	0.03497	0.00964	0.01543	0.0039	24.15	30.9	161.4	1813	0.1509	0.659	0.6091	0.1785	0.3672	0.1123
854039	M	16.13	17.88	107	807.2	0.104	0.1559	0.1354	0.07752	0.1998	0.06515	0.334	0.6857	2.183	35.03	0.00419	0.02868	0.02664	0.00907	0.01703	0.00382	20.21	27.26	132.7	1261	0.1446	0.5804	0.5274	0.1864	0.427	0.1233
854253	M	16.74	21.59	110.1	869.5	0.0961	0.1336	0.1348	0.06018	0.1896	0.05656	0.4615	0.9197	3.008	45.19	0.00578	0.02499	0.03695	0.01195	0.02789	0.00267	20.01	29.02	133.5	1229	0.1563	0.3835	0.5409	0.1813	0.4863	0.08633
854268	M	14.25	21.72	93.63	633	0.09823	0.1098	0.1319	0.05598	0.1885	0.06125	0.286	1.019	2.657	24.91	0.00588	0.02995	0.04815	0.01161	0.02028	0.00402	15.89	30.36	116.2	799.6	0.1446	0.4238	0.5186	0.1447	0.3591	0.1014
854941	B	13.03	18.42	82.61	523.8	0.08983	0.03766	0.02562	0.02923	0.1467	0.05863	0.1839	2.342	1.17	14.16	0.00435	0.0049	0.01343	0.01164	0.02671	0.00178	13.3	22.81	84.46	545.9	0.09701	0.04619	0.04833	0.05013	0.1987	0.06169
855133	M	14.99	25.2	95.54	698.8	0.09387	0.05131	0.02398	0.02899	0.1565	0.05504	1.214	2.188	8.077	106	0.00688	0.01094	0.01818	0.01917	0.00788	0.00175	14.99	25.2	95.54	698.8	0.09387	0.05131	0.02398	0.02899	0.1565	0.05504
855138	M	13.48	20.82	88.4	559.2	0.1016	0.1255	0.1063	0.05439	0.172	0.06419	0.213	0.5914	1.545	18.52	0.00537	0.02239	0.03049	0.01262	0.01377	0.00319	15.53	26.02	107.3	740.4	0.161	0.4225	0.503	0.2258	0.2807	0.1071

Fig. 4.9: Cancer disease Dataset

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
59	1	2	150	212	1	1	157	0	1.6	2	0	2	1
51	1	2	110	175	0	1	123	0	0.6	2	0	2	1

Fig. 4.10: Diabetes disease Dataset

Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesP	Age
6	148	72	35	0	33.6	0.627	50
1	85	66	29	0	26.6	0.351	31
8	183	64	0	0	23.3	0.672	32
1	89	66	23	94	28.1	0.167	21
0	137	40	35	168	43.1	2.288	33
5	116	74	0	0	25.6	0.201	30
3	78	50	32	88	31	0.248	26
10	115	0	0	0	35.3	0.134	29
2	197	70	45	543	30.5	0.158	53
8	125	96	0	0	0	0.232	54
4	110	92	0	0	37.6	0.191	30
10	168	74	0	0	38	0.537	34
10	139	80	0	0	27.1	1.441	57
1	189	60	23	846	30.1	0.398	59
5	166	72	19	175	25.8	0.587	51
7	100	0	0	0	30	0.484	32
0	118	84	47	230	45.8	0.551	31
7	107	74	0	0	29.6	0.254	31
1	103	30	38	83	43.3	0.183	33
1	115	70	30	96	34.6	0.529	32
3	126	88	41	235	39.3	0.704	27
8	99	84	0	0	35.4	0.388	50
7	196	90	0	0	39.8	0.451	41

Fig. 4.11: Heart disease Dataset

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48	80	1.02	1	0		normal	notpresent	notpresent	121	36	1.2			15.4	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	7	50	1.02	4	0		normal	notpresent	notpresent		18	0.8			11.3	38	6000		no	no	no	good	no	no	ckd
2	62	80	1.01	2	3	normal	normal	notpresent	notpresent	423	53	1.8			9.6	31	7500		no	yes	no	poor	no	yes	ckd
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	51	80	1.01	2	0	normal	normal	notpresent	notpresent	106	26	1.4			11.6	35	7300	4.6	no	no	no	good	no	no	ckd
5	60	90	1.015	3	0			notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no	ckd
6	68	70	1.01	0	0		normal	notpresent	notpresent	100	54	24	104	4	12.4	36			no	no	no	good	no	no	ckd
7	24		1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1			12.4	44	6900	5	no	yes	no	good	yes	no	ckd
8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9			10.8	33	9600	4	yes	yes	no	good	no	yes	ckd
9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	no	yes	ckd
10	50	60	1.01	2	4		abnormal	present	notpresent	490	55	4			9.4	28			yes	yes	no	good	no	yes	ckd
11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	yes	no	ckd
12	68	70	1.015	3	1		normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	yes	no	ckd
13	68	70						notpresent	notpresent	98	86	4.6	135	3.4	9.8				yes	yes	yes	poor	yes	no	ckd
14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	yes	no	ckd
15	40	80	1.015	3	0		normal	notpresent	notpresent	76	162	9.6	141	4.9	7.6	24	3800	2.8	yes	no	no	good	no	yes	ckd
16	47	70	1.015	2	0		normal	notpresent	notpresent	99	46	2.2	138	4.1	12.6				no	no	no	good	no	no	ckd
17	47	80						notpresent	notpresent	114	87	5.2	139	3.7	12.1				yes	no	no	poor	no	no	ckd
18	60	100	1.025	0	3		normal	notpresent	notpresent	263	27	1.3	135	4.3	12.7	37	11400	4.3	yes	yes	yes	good	no	no	ckd
19	62	60	1.015	1	0		abnormal	present	notpresent	100	31	1.6			10.3	30	5300	3.7	yes	no	yes	good	no	no	ckd
20	61	80	1.015	2	0	abnormal	abnormal	notpresent	notpresent	173	148	3.9	135	5.2	7.7	24	9200	3.2	yes	yes	yes	poor	yes	yes	ckd
21	60	90						notpresent	notpresent		180	76	4.5		10.9	32	6200	3.6	yes	yes	yes	good	no	no	ckd
22	48	80	1.025	4	0	normal	abnormal	notpresent	notpresent	95	163	7.7	136	3.8	9.8	32	6900	3.4	yes	no	no	good	no	yes	ckd
23	21	70	1.01	0	0		normal	notpresent	notpresent										no	no	no	poor	no	yes	ckd
24	42	100	1.015	4	0	normal	abnormal	notpresent	present		50	1.4	129	4	11.1	39	8300	4.6	yes	no	no	poor	no	no	ckd
25	61	60	1.025	0	0		normal	notpresent	notpresent	108	75	1.9	141	5.2	9.9	29	8400	3.7	yes	yes	no	good	no	yes	ckd
26	75	80	1.015	0	0		normal	notpresent	notpresent	156	45	2.4	140	3.4	11.6	35	10300	4	yes	yes	no	poor	no	no	ckd
27	69	70	1.01	3	4	normal	abnormal	notpresent	notpresent	264	87	2.7	130	4	12.5	37	9600	4.1	yes	yes	yes	good	yes	no	ckd
28	75	70		1	3			notpresent	notpresent	123	31	1.4							no	yes	no	good	no	no	ckd
29	68	70	1.005	1	0	abnormal	abnormal	present	notpresent		28	1.4			12.9	38			no	no	yes	good	no	no	ckd
30		70						notpresent	notpresent	93	155	7.3	132	4.9					yes	yes	no	good	no	no	ckd
31	73	90	1.015	3	0		abnormal	present	notpresent	107	33	1.5	141	4.6	10.1	30	7800	4	no	no	no	poor	no	no	ckd
32	61	90	1.01	1	1		normal	notpresent	notpresent	159	39	1.5	133	4.9	11.3	34	9600	4	yes	yes	no	poor	no	no	ckd
33	60	100	1.02	2	0	abnormal	abnormal	notpresent	notpresent	140	55	2.5			10.1	29			yes	no	no	poor	no	no	ckd
34	70	70	1.01	1	0	normal		present	present	171	153	5.2							no	yes	no	poor	no	no	ckd
35	65	90	1.02	2	1	abnormal	normal	notpresent	notpresent	270	39	2			12	36	9800	4.9	yes	yes	no	poor	no	yes	ckd

Fig. 4.12: Kidney disease Dataset

Age	Gender	Total_Bilir	Direct_Bili	Alkaline_P	Alamine_A	Aspartate	Total_Prot	Albumin	Albumin_a	Dataset
65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	1
62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
62	Male	7.3	4.1	490	60	68	7	3.3	0.89	1
58	Male	1	0.4	182	14	20	6.8	3.4	1	1
72	Male	3.9	2	195	27	59	7.3	2.4	0.4	1
46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	1
26	Female	0.9	0.2	154	16	12	7	3.5	1	1
29	Female	0.9	0.3	202	14	11	6.7	3.6	1.1	1
17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	2
55	Male	0.7	0.2	290	53	58	6.8	3.4	1	1
57	Male	0.6	0.1	210	51	59	5.9	2.7	0.8	1
72	Male	2.7	1.3	260	31	56	7.4	3	0.6	1
64	Male	0.9	0.3	310	61	58	7	3.4	0.9	2
74	Female	1.1	0.4	214	22	30	8.1	4.1	1	1
61	Male	0.7	0.2	145	53	41	5.8	2.7	0.87	1
25	Male	0.6	0.1	183	91	53	5.5	2.3	0.7	2
38	Male	1.8	0.8	342	168	441	7.6	4.4	1.3	1
33	Male	1.6	0.5	165	15	23	7.3	3.5	0.92	2
40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
51	Male	2.2	1	610	17	28	7.3	2.6	0.55	1
51	Male	2.9	1.3	482	22	34	7	2.4	0.5	1
62	Male	6.8	3	542	116	66	6.4	3.1	0.9	1
40	Male	1.9	1	231	16	55	4.3	1.6	0.6	1
63	Male	0.9	0.2	194	52	45	6	3.9	1.85	2

Fig. 4.13: Liver disease Dataset

4.7 Result analysis

The accuracies of each model in the designed system are diabetes model: 79.22%, breast cancer model: 95.61%, heart model: 81.97%, kidney model: 98%, liver model: 79.66%.

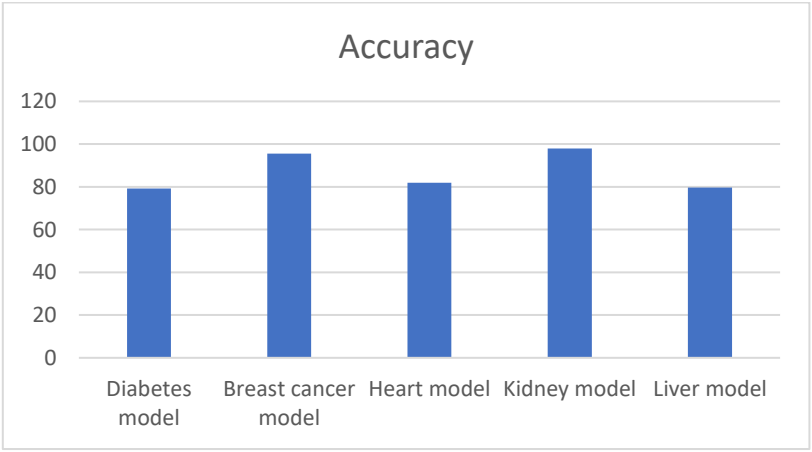


Fig. 4.14: Accuracy

5. TESTING & VALIDATION

5.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. It is the major quality measure employed during software development. Testing is the exposure of the system to trial input to see whether it produces correct output. It is a process, which reveals errors in the program. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

5.1.1 Testing steps:

- Validate raw feature data and engineered feature data.
- Debug a ML model to make the model work.
- Implement tests that simplify debugging.
- Optimize a working ML. model.
- Monitor model metrics during development, launch, and production.

There are various types of Testing:

- Unit Testing:

  Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program.
- Integration Testing:

  All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules.
- System Testing:

  It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.
- Acceptance Testing:

  It is performed with realistic data of the client to demonstrate that the software is working satisfactorily.

5.1.2 Testing Methods:

Testing is a process of executing a program to find out errors. If testing is conducted 23 successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

- White Box Testing:

  It is a test case design method that uses the control structures of the procedural design to derive test cases. In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. Using this testing a Software Engineer can derive the following test cases: Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity. White Box testing attempts to find errors in the following categories:

  - Guarantee that all independent paths have been executed.
  - Execute all logical decisions on their true and false Sides.
  - Execute all loops at their boundaries and within their operational bounds
  - Execute internal data structures to ensure their validity.

- Black Box Testing:

  It is a test case design method used on the functional requirements of the software. In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program It will help a software engineer to derive sets of input conditions that will exercise all the functional requirements of the program. Black Box testing attempts to find errors in the following categories:

  - Incorrect or missing functions
  - Interface errors
  - Errors in data structure or external database access
  - Performance errors
  - Initialization and termination errors

5.1.3 Testing Approach:

Testing can be done in two ways:

• Bottom-up approach

• Top-down approach

- Bottom-up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

- Top-down approach:

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module 17 shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

5.2 Design of Test Cases and Scenarios

Test Scenario	Test steps	Test Data	Expected results	Actual results	Pass/Fail
Check user login with valid data	1. Go to http://127.0.0.1:5000 2. Enter Email 3. Enter Password 4. Click Login	Email=varun@gmail.com Password=12345	User should be redirected to home page.	As expected	Pass
Check user login with invalid data	1. Go to http://127.0.0.1:5000 2. Enter Email 3. Enter Password 4. Click Login	Email=varun@gmail.com Password=varun	User should not be redirected to home page.	As expected	Pass

Table 5.1: Test Cases for Login

Lets consider the Test cases for diabetes and liver diseases.

Diabetes disease prediction test cases

Test case-1:

Input- Pregnancies : 0

Glucose : 35 mg/dL

Blood pressure : 120 mmHg

Skin thickness : 15 mm

Insulin level : 60 IU/mL

Body mass index : 25.3 kg/m^2

Diabetes pedigree function : 0.55

Age: 30

Expected output- The patient is not diagnosed with the disease.

Actual output- The patient is not diagnosed with the disease.

Status- Pass

Test case-2:

Input- Pregnancies : 2

Glucose : 124 mg/dL

Blood pressure : 135 mmHg

Skin thickness : 56 mm

Insulin level : 125 IU/mL

Body mass index : 45 kg/m^2

Diabetes pedigree function : 2.65

Age: 56

Expected output- The patient is diagnosed with the disease.

Actual output- The patient is diagnosed with the disease.

Status- Pass

Liver disease prediction test cases

Test case-1:

Input- Age : 65

Total Bilirubin : 0.7

Direct Bilirubin : 0.1

Alkaline phosphotase : 187

Alamine Aminotransferase : 16

Aspartate Aminotransferase : 18

Total protiens : 6.8

Albumin : 3.3

Albumin and globulin ratio : 0.9

Gender : 0

Expected output- The patient is effected with the disease.

Actual output- The patient is effected with the disease.

Status- Pass

Test case-2:

Input- Age : 17

Total Bilirubin : 0.9

Direct Bilirubin : 0.3

Alkaline phosphotase : 202

Alamine Aminotransferase : 22

Aspartate Aminotransferase : 19

Total protiens : 7.4

Albumin : 4.1

Albumin and globulin ratio : 1.2

Gender : 1

Expected output- The patient is not effected with the disease.

Actual output- The patient is not effected with the disease.

Status- Pass

5.3 Validation

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

6. CONCLUSION & FUTURE ENHANCEMENT

6.1 Conclusion

Multiple disease prediction using machine learning is a promising approach to revolutionize the healthcare industry. With the rapid increase in medical data and the advancements in machine learning algorithms, this approach can enable early and accurate diagnosis of multiple diseases, thereby improving patient outcomes and reducing healthcare costs.

Machine learning algorithms can learn from vast amounts of data to identify complex patterns and relationships between various features and diseases. This allows for the development of more accurate prediction models, which can provide healthcare professionals with a more comprehensive view of a patient's health status.

However, the successful implementation of multiple disease prediction using machine learning requires addressing several challenges, including privacy concerns, data bias, and transparency in decision-making processes. Moreover, further research is necessary to investigate the integration of different types of medical data, including genetic, environmental, and lifestyle data.

Overall, multiple disease prediction using machine learning holds tremendous potential for the healthcare industry, and it is expected to play a crucial role in improving the quality and efficiency of healthcare in the future.

6.2 Future Enhancement

There are many possible improvements that could be explore to diversify the research by discovering and considering extra features. Due to time boundation, the following work required to be performed in future. There is plan to use more classification techniques/ methods, different discretization techniques, multiple classifiers voting methods. Would like to use different rules such as association rule and various algorithms like logistic regression and clustering algorithms. In future, willing to make use of filter-based feature selection methods in order to achieve more appropriate as well as functional result.

7.BIBILOGRAPHY

7.1 References

[1] A.S. Monto, S. Gravenstein, M. Elliott, M. Colopy, J. Schweinle, Clinical signs and symptoms predicting inuenza infection, Archives of internal medicine 160(21), 3243 (2000)

[2] International Journal of Scientific Research in Computer Science, E., & IJSRCSEIT, I. T. (2019). Generic Disease Prediction using Symptoms with Supervised Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. https://doi.org/10.32628/CSEIT1952297.

[3] Automatic Heart Disease Prediction Using Feature Selection And Data Mining Technique Le Ming Hung,a, Tran Ding,Journal of Computer Science and Cybernetics, V.34, N.1 (2018), 3347 DOI: 10.15625/1813- 9663/34/1/12665

[4] Balasubramanian, Satyabhama, and Balaji Subramanian. "Symptom based disease prediction in medical system by using Kmeans algorithm." International Journal of Advances in Computer Science and Technology 3.

[5] Pingale, Kedar, et al. "Disease Prediction using Machine Learning." (2019).Mr. Chala Beyene, 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

[6] Automatic Heart Disease Prediction Using Feature Selection And Data Mining Technique Le Ming Hung,a, Tran Ding,Journal of Computer Science and Cybernetics, V.34, N.1 (2018), 3347 DOI: 10.15625/1813- 9663/34/1/12665

[7] Fowler, M.J. Diabetes: Magnitude and Mechanisms. Clin. Diabetes 2007, 25, 25–28.

[8] DeWitt, D.E.; Hirsch, I.B. Outpatient insulin therapy in type 1 and type 2 diabetes mellitus: Scientific review. JAMA 2003, 289, 2254–2264.