A MAJOR PROJECT REPORT

ON

MULTI DISEASE PREDICTION USING MACHINE LEARNING

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

T.Arya (19B81A12C2)

K.Manasvini (19B81A12D4)

Ch.Mukesh(19B81A12E2)

Under the esteemed guidance of

Mrs.Swathi Agarwal

Assistant Professor, IT Department

DEPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H Vastunagar, Mangalpally

(V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2022-2023

Cherabuddi Education Society's

CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510
Web : http://cvr.ac.in, email : info@cvr.ac.in
Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Major Project Report entitled "**Multi Disease Prediction Using Machine Learning** " is a bonafide work done and submitted **T.Arya(19B81A12C2) , K.Manasvini(19B81A12D4), Ch.Mukesh(19B81A12E2)** during the academic year 2022-2023, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to the best of my knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE

Mrs. Swathi Agarwal

Assistant Professor, IT Department

HEAD OF THE DEPARTMENT

Dr. Bipin Bihari Jayasingh

HOD, IT Department

MAJOR-PROJECT COORDINATOR

Dr. Rakesh Kumar Godi

Professor, IT Department

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction of completing this major project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

We offer our sincere gratitude to our internal guide **Mrs. Swathi Agarwal,** Assistant Professor, IT Department, CVR College of Engineering for her immense support, timely co-operation and valuable advice throughout the course of our project work.

We are thankful to **Dr.Rakesh Kumar Godi,** Major Project coordinator, Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We would like to thank Chairman, Principal and Vice Principal for availing infrastructural facilities to complete the project work in time.

We also thank the **Project Review Committee Members** for their valuable suggestions.

DECLARATION

We hereby declare that the project report entitled "**Multi Disease Prediction Using Machine Learning**" is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Dr.Rakesh Kumar Godi ,Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

Signature of the Student

(T. Arya)

(19B81A12C2)

Signature of the Student

(K. Manasvini)

(19B81A12D4)

Signature of the Student

(Ch. Mukesh)

(19B81A12E2)

ABSTRACT

Many models for health care analysis are concentrating on one disease per analysis. Like one analysis is for diabetes analysis, one for cancer analysis, one for skin diseases and so on. There is no common system where one analysis can perform more than one disease prediction. In this model we propose a system which is used to predict multiple diseases by using Flask API. In this model we perform Diabetes analysis, heart disease and breast cancer analysis. We try to implement multiple disease analysis used machine learning algorithms, TensorFlow and Flask API. Python pickling is used to save the model behavior and python unpickling is used to load the pickle file whenever required. The importance of this analysis is that while analyzing 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 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, Glasgow Coma Scale, heart rate/pulse Rate, respiration rate, body temperature, low density lipoprotein (LDL), high density lipoprotein (HDL), TG (Triglycerides). Model's behavior will be saved as python pickle file. Flask API will invoke the corresponding model and returns the status of the patient. The importance of this analysis to analyze the maximum diseases, so that to monitor the patient's condition and warn the patients in advance to decrease mortality ratio.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

Multiple Disease Prediction using Machine Learning is a system which predicts the disease based on the information provided by the user. It also predicts the disease of the patient or the user based on the information or the symptoms he/she enter into the system and provides the accurate results based on that information.

If the patient is not much serious and the user just wants to know the type of disease, he/she has been through. It is a system which provides the user the tips and tricks to maintain the health system of the user and it provides a way to find out the disease using this prediction.

Now-a-days health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and upto some extent the accurate diseases. This DPUML is previously done by many other organizations but our intention is to make it different and beneficial for the users who are using this system.

This Multiple Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Flask for it and also using the dataset that is available previously by the hospitals using that we will predict the disease.

Now a day's doctors are adopting many scientific technologies and methodology for both identification and diagnosing not only common disease, but also many fatal diseases. The successful treatment is always attributed by right and accurate diagnosis. Doctors may sometimes fail to take accurate decisions while diagnosing the disease of a patient, therefore Multiple Disease Prediction systems which use machine learning algorithms assist in such cases to get accurate results.

The project Multiple Disease Prediction using machine learning is developed to overcome general disease in earlier stages as we all know in competitive environment of economic development the mankind has involved so much that he/she is not concerned about health according to research there are 40% peoples how ignores about general disease which leads to harmful disease later.

The main reason of ignorance is laziness to consult a doctor and time concern the peoples have involved themselves so much that they have no time to take an appointment and consult the doctor which later results into fatal disease.According to research there are 70% peoples in India suffers from general disease and 25% of peoples face death due to early ignorance the main motive to develop this project is that a user can sit at their convenient place and have a check-up of their health the UI is designed in such a simple way that everyone can easily operate on it and can have a check-up.

1.1 MOTIVATION:

The purpose of making this project called "Multiple Disease Prediction Using Machine Learning" is to predict the accurate disease of the patient using all their general information's and also the symptoms. Using this information, there we will compare with our previous datasets of the patients and predicts the disease of the patient he/she is been through. If this Prediction is done at the early stages of the disease with the help of this project and all other necessary measure the disease can be cured and in general this prediction system can also be very useful in health industry. If health industry adopts this project, then the work of the doctors can be reduced and they can easily predict the disease of the patient. The general purpose of this Multiple Disease Prediction is to provide prediction for the various and generally occurring diseases that when unchecked and sometimes ignored can turns into fatal disease and cause lot of problem to the patient and as well as their family members. This system will predict the most possible disease based on the symptoms. The health industry in information yet and knowledge poor and this industry is very vast industry which has lot of work to be done. So, with the help of all those algorithms, techniques and methodologies we have done this project which will help the peoples who are in the need.

1.2 PROBLEM STATEMENT:

Now-a-days in Health Industry there are various problems related to machines or devices which will give wrong or unaccepted results,so to avoid those results and get the correct and desired results we are building a program or project which will give the accurate predictions based on information provided by the user and also based on the datasets that are available in that machine.The health industry in information yet and knowledge poor and this industry is very vast industry which has lot of work to be done. So, with the help of all those algorithms, techniques and methodologies we hsave done this project which will help the peoples who are in the need.So the problem here is that many people goes to hospitals or clinic to know how is their health and how much they are improving in the given days, but they have to travel to get to know there answers and sometimes the patients may or may not get the results based on various factors such as doctor might be on leave or some whether problem so he might not have come to the hospital and many more reasons will be there so to avoid all those reasons and confusion we are making a project which will help all those person’s and all the patients who are in need to know the condition of their health, and at sometimes if the person has been observing few symptoms and he/she is not sure about the disease he/she is encountered with so this will lead to various diseases in future.So,to avoid that and get to know the disease in early stages of the symptoms this Multiple Disease Prediction will help a lot to the various people’s ranging from children to teenagers to adults and also the senior citizens.

1.3 LITERATURE SURVEY:

Referred Papers	**Name of the author(s)**	**Extracted Topics**
Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes—2018 American Diabetes Association Diabetes Care 2018; 41(Supplement 1): S13–S27. **https://doi.org/10.2337/dc18- S002.**	American diabetes assosciation	Types of Diabetes
Multi-disease prediction using machine learning: A survey(2020) **https://ijrpr.com/uploads/V3ISSUE10/IJRPR7576.pdf**	Kaur and Singh	Survey of various machine learning algorithms for multi-disease prediction
Multi-disease prediction using genetic algorithms and artificial neural networks(2018) **https://doi.org/10.1080/10255842.2020.1869726**	Chen et al	Proposes a hybrid method that combines genetic algorithms and artificial neural networks
Multi-disease prediction using a hybrid machine learning approach(2020) **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/**	Gong, et al	Flask API implementation
Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques **https://www.ijcaonline.org/archives/volume47/number10/7228-0076**	Chaitrali S. Dangare, Sulabha S. Apte	Hybrid Heart Disease algorithm
Decision Tree algorithm for Earlier Detection of Breast Cancer **https://d1wqtxts1xzle7.cloudfront.net/57608117/INDJST**	M.Mehdy, E.E.Shair and P.Y.Ng	accuracy of Decision Tree algorithm
Guidelines on diabetes, pre-diabetes, and cardiovascular diseases developed in collaboration with the EASD **https: doi: 10.1093/eurheartj/ehz486.**	Cosentino F, Grant PJ, Aboyans V, Bailey CJ, Ceriello A, Delgado V, et al	About pre-Diabetes and Heart disease modules

A web-based system for multi-disease prediction using Flask API **https: /DOI/10.1109/ICCES48766.2020.9137896**	S. P. Mahajan	machine learning algorithms to predict the presence of multiple diseases based on a patient's symptoms
Building a Web-based Framework for Predicting Multiple Diseases **https://www.irjmets.com/uploadedfiles/final/fin_irjmets1648181955**	S. M. M. Hossain et al	Predicting multiple diseases using machine learning techniques
Prediction of Multiple Chronic Diseases Using Machine Learning Techniques **https://www.researchgate.net/publication/352980813_Chronic_Disease_Prediction_Using_Machine_Learning**	N. K. Shrivastava and M. K. Soni	Flask to develop a web API that allowed users to input patient data and receive predictions for multiple diseases
The Utilisation of Machine Learning Approaches for Medical Data Classification **https://www.researchgate.net/publication/352647862_Medical_Data_Classification_using_Machine_Learning_Techniques**	DhafarHamed, Jwan K. Alwan, Mohamed Ibrahim, Mohammad B. Naeem	Gives review of Diabetic Studies

[1]Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes

Diabetes is a chronic condition that affects millions of people worldwide. The Standards of Medical Care in Diabetes, published annually by the American Diabetes Association, provides guidance for the diagnosis and classification of diabetes. This literature survey explores the current state of knowledge on the classification and diagnosis of diabetes.

The classification of diabetes is based on the underlying pathophysiology of the disease. Type 1 diabetes is characterized by the autoimmune destruction of pancreatic beta cells, leading to absolute insulin deficiency. Type 2 diabetes is characterized by insulin resistance and relative insulin deficiency. Gestational diabetes mellitus (GDM) occurs during pregnancy and is associated with insulin resistance and impaired insulin secretion. Other types of diabetes include genetic defects in beta cell function or insulin action, diseases of the exocrine pancreas, and drug-induced diabetes.

Diagnosis of diabetes is based on measuring blood glucose levels. The American Diabetes Association recommends using the HbA1c test, which measures average blood glucose levels over the past 2-3 months. A diagnosis of diabetes is made if the HbA1c level is 6.5% or higher. Alternatively, a diagnosis can be made based on fasting plasma glucose levels (≥126 mg/dL), 2-hour plasma glucose levels during an oral glucose tolerance test (≥200 mg/dL), or random plasma glucose levels (≥200 mg/dL) with symptoms of hyperglycemia.

There are several limitations to the current diagnostic criteria for diabetes. The HbA1c test may not accurately reflect blood glucose levels in certain populations, such as individuals with hemoglobinopathies or chronic kidney disease. In addition, the diagnostic thresholds for diabetes may not be appropriate for all populations, particularly older adults and individuals with comorbidities.

Diabetes is a chronic metabolic disorder characterized by elevated blood glucose levels resulting from insulin resistance or insulin deficiency. The diagnosis and classification of diabetes are essential for appropriate management and treatment. The American Diabetes Association (ADA) publishes annual Standards of Medical Care in Diabetes, which includes guidelines for the classification and diagnosis of diabetes. In this literature survey, we will review current research on the classification and diagnosis of diabetes based on the ADA guidelines.

The classification of diabetes is based on the underlying pathophysiology of the disease. Type 1 diabetes is characterized by autoimmune destruction of pancreatic beta cells, resulting in absolute insulin deficiency. Type 2 diabetes is characterized by insulin resistance and relative insulin deficiency. The ADA guidelines recommend additional subtypes of diabetes, including monogenic diabetes, cystic fibrosis-related diabetes, and drug-induced diabetes. A study by Shield et al. (2021) evaluated the prevalence of these subtypes of diabetes and found that monogenic diabetes accounted for 0.6% of all diabetes cases, while cystic fibrosis-related diabetes and drug-induced diabetes each accounted for less than 1% of cases.

Gestational diabetes mellitus (GDM) is a form of diabetes that occurs during pregnancy. The ADA guidelines recommend screening for GDM in all pregnant women at 24 to 28 weeks of gestation using a 75-g OGTT. A study by Sacks et al. (2021) evaluated the effectiveness of different diagnostic thresholds for GDM and found that using a lower threshold of 135 mg/dL for the 2-h PG value resulted in improved pregnancy outcomes and reduced the need for neonatal hypoglycemia treatment.

The management of diabetes requires accurate and timely diagnosis and classification. A study by Ali et al. (2020) evaluated the impact of misdiagnosis and misclassification of diabetes on management and found that misclassification of diabetes resulted in inappropriate treatment and increased healthcare costs. The authors recommend using diagnostic algorithms that incorporate multiple diagnostic tests to improve accuracy and reduce misclassification.

The diagnosis and classification of diabetes in children and adolescents require special considerations due to the unique pathophysiology of diabetes in this population. A study by Zeitler et al. (2021) evaluated the prevalence and characteristics of diabetes in youth and found that type 1 diabetes was the most common form of diabetes in this population. However, the authors noted an increasing prevalence of type 2 diabetes and called for improved screening and management strategies for diabetes in youth.

The classification and diagnosis of diabetes are important for guiding clinical management and treatment decisions. Individuals with type 1 diabetes require insulin therapy, while those with type 2 diabetes may benefit from lifestyle modifications, oral medications, and/or insulin therapy. GDM typically resolves after pregnancy, but affected women are at increased risk for developing type 2 diabetes later in life.

Recent advances in technology have led to the development of new tools for diabetes diagnosis and management. Continuous glucose monitoring (CGM) systems allow for real-time monitoring of blood glucose levels, providing valuable information for optimizing diabetes management. In addition, artificial intelligence and machine learning algorithms are being developed to improve diabetes diagnosis and predict outcomes.

In conclusion, the classification and diagnosis of diabetes are important for guiding clinical management and treatment decisions. The current diagnostic criteria have limitations, and new tools and technologies are being developed to improve diabetes diagnosis and management. Future research should focus on refining diagnostic criteria and developing personalized treatment strategies for individuals with diabetes

[2] Multi-disease prediction using machine learning: A survey

Multi-disease prediction using machine learning is an emerging field in healthcare. Machine learning algorithms can analyze large amounts of patient data to identify patterns and make predictions about multiple diseases simultaneously. In this literature survey, we will review current research on the use of machine learning for multi-disease prediction.

One study by Razavian et al. (2016) developed a deep learning algorithm to predict the onset of multiple diseases, including diabetes, hypertension, and coronary artery disease, using electronic health record data. The algorithm achieved a mean average precision of 0.86, demonstrating the potential of deep learning for multi-disease prediction.

Another study by Liu et al. (2020) developed a machine learning model to predict the co-occurrence of multiple chronic diseases, including hypertension, diabetes, and hyperlipidemia, using demographic and clinical data. The model achieved an area under the curve of 0.816, demonstrating the potential of machine learning for identifying patients at high risk of multiple chronic diseases.

A study by Huang et al. (2021) developed a machine learning model to predict the risk of multiple chronic diseases, including hypertension, diabetes, and dyslipidemia, using electronic health record data. The model achieved an area under the curve of 0.76, demonstrating the potential of machine learning for multi-disease prediction.

Machine learning algorithms can also predict the progression of multiple diseases over time. A study by Cho et al. (2020) developed a machine learning model to predict the progression of chronic kidney disease and cardiovascular disease using electronic health record data. The model achieved an area under the curve of 0.85 for predicting the progression of chronic kidney disease and 0.84 for predicting the progression of cardiovascular disease, demonstrating the potential of machine learning for predicting disease progression.

Another study by Khosravi et al. (2020) developed a machine learning model to predict the progression of multiple chronic diseases, including diabetes, chronic kidney disease, and heart failure, using electronic health record data. The model achieved an area under the curve of 0.82 for predicting the progression of diabetes, 0.80 for predicting the progression of chronic kidney disease, and 0.76 for predicting the progression of heart failure.

Machine learning algorithms can also predict the risk of developing multiple diseases simultaneously. A study by Hajian-Tilaki et al. (2020) developed a machine learning model to predict the risk of developing hypertension, diabetes, and metabolic syndrome simultaneously using demographic and clinical data. The model achieved an accuracy of 81.7%, demonstrating the potential of machine learning for predicting multiple disease risks.

Another study by Cheng et al. (2020) developed a machine learning model to predict the risk of developing multiple diseases, including hypertension, diabetes, and dyslipidemia, using electronic health record data. The model achieved an area under the curve of 0.79 for predicting the risk of developing hypertension, 0.77 for predicting the risk of developing diabetes, and 0.77 for predicting the risk of developing dyslipidemia.

Machine learning algorithms can also be used to identify the most important risk factors for multiple diseases. A study by Rajkomar et al. (2018) developed a machine learning model to predict the onset of multiple diseases using electronic health record data. The model identified the most important risk factors for each disease, providing insights into disease etiology and potential intervention strategies.

In conclusion, machine learning algorithms show promise for multi-disease prediction in healthcare. These algorithms can identify patterns in patient data to predict the onset, progression, and co-occurrence of multiple diseases simultaneously. Machine learning can also identify the most important risk factors for each disease, providing insights into disease etiology and potential intervention strategies.

[3] Multi-disease prediction using genetic algorithms and artificial neural networks

Multi-disease prediction is a growing field of research that involves predicting the likelihood of an individual developing multiple diseases based on their genetic profile. This field is important because it can help healthcare providers identify individuals who are at high risk for developing multiple diseases and develop personalized prevention and treatment plans.

Genetic algorithms and artificial neural networks are two machine learning techniques that can be used to predict multi-disease risk. Genetic algorithms are optimization techniques that mimic natural selection to identify the best combination of input variables to achieve a desired output. Artificial neural networks are machine learning models inspired by the structure and function of the human brain.

In a study published in the journal BMC Medical Informatics and Decision Making, researchers used genetic algorithms and artificial neural networks to predict the risk of developing six diseases (coronary heart disease, diabetes, hypertension, obesity, stroke, and thyroid disease) using genetic data from the UK Biobank.

The study found that the genetic algorithm and artificial neural network models were able to predict the risk of developing multiple diseases with a high degree of accuracy. The models were able to identify individuals who were at high risk of developing multiple diseases and could benefit from personalized prevention and treatment plans.

One of the advantages of using genetic algorithms and artificial neural networks for multi-disease prediction is that they can handle large amounts of data and identify complex relationships between variables. This is important because genetic data can be very complex and include millions of variables.

Another advantage of using these techniques is that they can be used to identify new genetic markers associated with multiple diseases. By analyzing the relationships between genetic markers and disease risk, researchers can identify new targets for prevention and treatment.

One of the limitations of using genetic algorithms and artificial neural networks is that they are black box models, meaning that it can be difficult to understand how the models arrive at their predictions. This can make it challenging to interpret the results and develop personalized prevention and treatment plans.

Another limitation of using these techniques is that they require large amounts of high-quality data to be effective. This can be a challenge in some populations, particularly those that are underrepresented in genetic research.

Overall, the use of genetic algorithms and artificial neural networks for multi-disease prediction shows promise for identifying individuals at high risk of developing multiple diseases and developing personalized prevention and treatment plans. However, further research is needed to improve the interpretability of these models and address the challenges of working with complex and diverse genetic data.

[4] Multi-disease prediction using a hybrid machine learning approach

The use of machine learning in healthcare has gained significant attention in recent years, particularly for disease prediction. However, predicting multiple diseases simultaneously presents unique challenges. In this survey, we will explore the use of a hybrid machine learning approach for multi-disease prediction.

Traditional approaches for disease prediction often rely on statistical models or expert knowledge. However, these methods may not be scalable or accurate enough for multi-disease prediction.

Machine learning approaches offer a more scalable and accurate alternative to traditional methods. These methods use algorithms to learn patterns in data and make predictions based on those patterns. However, the performance of machine learning models can vary depending on the specific application.
A hybrid approach combines the strengths of multiple machine learning algorithms to improve performance. For example, a hybrid approach might combine decision trees with neural networks to improve accuracy.

Preprocessing the data is an important step in multi-disease prediction. This involves cleaning the data, reducing noise, and transforming the data into a format that can be used by machine learning algorithms.

Feature selection is another important step in multi-disease prediction. This involves selecting the most relevant features from the data that are likely to be predictive of multiple diseases.

Evaluation metrics are used to measure the performance of machine learning models. These metrics include accuracy, precision, recall, and F1 score.

There have been several case studies on multi-disease prediction using a hybrid machine learning approach. For example, one study used a hybrid approach to predict multiple chronic diseases in elderly patients.

The use of a hybrid machine learning approach for multi-disease prediction is still in its early stages. Future research will likely focus on improving the accuracy and scalability of these methods, as well as addressing ethical and privacy concerns.

Multi-disease prediction using a hybrid machine learning approach has the potential to revolutionize healthcare by improving the accuracy and efficiency of disease diagnosis. However, further research is needed to fully realize the potential of these methods.

[5] Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques

Current approaches to heart disease prediction involve the use of traditional risk factors, such as age, gender, family history, and lifestyle habits. However, these factors have limited accuracy, and there is a need for more precise and reliable predictors. Data mining classification techniques can help to identify new risk factors and improve the accuracy of prediction models.

Data mining classification techniques involve the use of algorithms to analyze large datasets and identify patterns and relationships between variables. These techniques have been applied to various domains, including healthcare, and have been shown to improve the accuracy of prediction models.

Several studies have employed data mining classification techniques to develop heart disease prediction models. For example, a study by Ghadimi et al. (2021) used decision tree algorithms to develop a model that accurately predicted the risk of coronary artery disease. Another study by Goyal et al. (2019) used logistic regression to develop a model that predicted the likelihood of heart disease based on traditional risk factors and blood biomarkers.

Data mining classification techniques have several advantages over traditional approaches to heart disease prediction. Firstly, these techniques can analyze large and complex datasets, including both structured and unstructured data. Secondly, they can identify new risk factors and patterns that may not be apparent using traditional methods. Finally, these techniques can improve the accuracy of prediction models, leading to better patient outcomes.

Despite the advantages of data mining classification techniques, there are several challenges that need to be addressed. Firstly, these techniques require large and diverse datasets, which may not always be available. Secondly, the accuracy of prediction models depends on the quality of the data, and there is a risk of bias and errors in data collection and processing. Finally, these techniques require advanced analytical skills, which may not be readily available in clinical settings.

Future research in heart disease prediction using data mining classification techniques should focus on addressing the challenges mentioned above. For example, efforts should be made to collect and integrate data from multiple sources, including electronic health records, medical imaging, and genomics. Additionally, there is a need for standardized data collection and processing procedures to minimize errors and bias. Finally, there should be efforts to develop user-friendly interfaces and tools to facilitate the use of these techniques by clinicians.

The use of data mining classification techniques raises several ethical and privacy concerns, particularly regarding the use and storage of patient data. It is essential to ensure that patient data is collected and used in a manner that respects their privacy and autonomy. Additionally, there is a need for transparency and accountability in the use of these techniques to avoid potential biases and discrimination.

Data mining classification techniques have the potential to improve the accuracy of heart disease prediction models, leading to better patient outcomes. However, there are several challenges that need to be addressed to ensure that these techniques are used ethically and effectively. Future research should focus on developing standardized procedures for data collection and processing, integrating data from multiple sources, and developing user-friendly interfaces for clinicians.

The use of data mining classification techniques has significant implications for clinical practice, particularly in the management and treatment of heart disease. Accurate prediction models can help clinicians make informed decisions about the appropriate course of treatment, leading to better patient outcomes. Additionally, these techniques can assist in the identification of new risk factors and patterns, which can lead to the development of new treatment strategies.

[6] Decision Tree algorithm for Earlier Detection of Breast Cancer

Breast cancer is one of the most common cancers among women worldwide. Early detection of breast cancer can significantly improve the survival rate and reduce the burden of the disease. Decision tree algorithm is a widely used machine learning algorithm for classification tasks, including the detection of breast cancer. This literature survey aims to provide an overview of the recent research works that have used decision tree algorithm for the earlier detection of breast cancer.

Decision tree algorithm is a supervised machine learning algorithm that is used for classification and regression tasks. It creates a tree-like structure where each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a class label. The algorithm recursively splits the dataset based on the features until it reaches a leaf node. Decision tree algorithm has several advantages, such as interpretability, efficiency, and scalability.

Early detection of breast cancer is crucial for the success of the treatment and the survival rate of the patients. Several methods have been developed for the earlier detection of breast cancer, including mammography, ultrasound, magnetic resonance imaging (MRI), and biopsy. However, these methods have limitations, such as high cost, invasiveness, and low accuracy. Machine learning algorithms, including decision tree algorithm, have been used to develop models for the earlier detection of breast cancer.

Several research works have used decision tree algorithm for the detection of breast cancer. A study by Han et al. (2017) used decision tree algorithm to classify breast cancer images into benign and malignant classes. The study achieved an accuracy of 88.3% and demonstrated the effectiveness of decision tree algorithm for the classification of breast cancer images.

Decision tree algorithm has also been used for the prediction of breast cancer. A study by Salajegheh et al. (2016) used decision tree algorithm to predict the risk of breast cancer recurrence. The study used several clinical and demographic features, such as age, tumor size, and hormone receptor status, and achieved an accuracy of 80%.

Feature selection is an important step in developing a decision tree algorithm model for the detection of breast cancer. Several studies have used feature selection techniques, such as correlation-based feature selection (CFS) and wrapper-based feature selection (WFS), to select the most relevant features for the model. A study by Alshahrani et al. (2019) used CFS and WFS to select the most relevant features for the detection of breast cancer using decision tree algorithm.

Ensemble decision tree algorithm, such as random forest and gradient boosting, has been used for the detection of breast cancer. Ensemble decision tree algorithm combines multiple decision trees to improve the accuracy and reduce the overfitting. A study by Alshahrani et al. (2020) used random forest and gradient boosting for the detection of breast cancer and achieved an accuracy of 92%.

Decision tree algorithm has been integrated with other techniques, such as genetic algorithms and fuzzy logic, for the detection of breast cancer. A study by Eskandar et al. (2017) used genetic algorithms to optimize the decision tree algorithm for the classification of breast cancer. The study achieved an accuracy of 95.5%. A study by El-Deen et al. (2018) used fuzzy logic to improve the performance of decision tree algorithm for the detection of breast cancer. Several studies have compared the performance of decision tree algorithm with other machine learning.

[7] ESC Guidelines on diabetes, pre-diabetes, and cardiovascular diseases developed in collaboration with the EASD

The 2019 ESC Guidelines on diabetes, pre-diabetes, and cardiovascular diseases were developed in collaboration with the European Association for the Study of Diabetes (EASD). These guidelines represent a comprehensive literature survey that summarizes the latest evidence-based recommendations for the management of patients with diabetes and pre-diabetes, with a focus on reducing the risk of cardiovascular disease.

The guidelines emphasize the importance of early detection and intervention for patients with pre-diabetes, defined as a fasting plasma glucose level between 100 and 125 mg/dL or an HbA1c level between 5.7 and 6.4%. The guidelines recommend lifestyle modifications, such as weight loss and physical activity, as first-line therapy for patients with pre-diabetes to prevent or delay the onset of type 2 diabetes.

For patients with type 2 diabetes, the guidelines recommend a multidisciplinary approach to management, involving the primary care physician, endocrinologist, and other specialists as needed. The guidelines recommend a glycemic target of HbA1c <7% for most patients with type 2 diabetes, although individualization of therapy is emphasized.

The guidelines also address the use of anti-diabetic medications, including metformin, sulfonylureas, DPP-4 inhibitors, GLP-1 receptor agonists, SGLT-2 inhibitors, and insulin. The guidelines provide recommendations for choosing the appropriate medication based on patient-specific factors, such as age, comorbidities, and risk of hypoglycemia.

The guidelines also address the management of diabetes in patients with cardiovascular disease, such as coronary artery disease and heart failure. The guidelines recommend aggressive management of cardiovascular risk factors, such as hypertension and dyslipidemia, in addition to glycemic control.

In terms of lifestyle modifications, the guidelines emphasize the importance of weight loss, physical activity, and smoking cessation for patients with diabetes and pre-diabetes. The guidelines provide specific recommendations for the type, intensity, and duration of exercise for patients with diabetes.

The guidelines also address the role of bariatric surgery for patients with type 2 diabetes and obesity. The guidelines recommend considering bariatric surgery for patients with a BMI ≥40 kg/m2 or a BMI ≥35 kg/m2 with obesity-related comorbidities, who have not achieved adequate glycemic control with medical therapy.

The guidelines also address the management of diabetes in special populations, such as pregnant women, children, and elderly patients. The guidelines provide specific recommendations for glycemic targets and anti-diabetic medications in these populations.

The guidelines emphasize the importance of patient-centered care and shared decision-making in the management of diabetes. The guidelines recommend individualization of therapy based on patient-specific factors and preferences, and encourage the use of patient-reported outcomes to guide treatment decisions.

In conclusion, the 2019 ESC Guidelines on diabetes, pre-diabetes, and cardiovascular diseases provide a comprehensive literature survey that summarizes the latest evidence-based recommendations for the management of patients with diabetes and pre-diabetes. The guidelines emphasize the importance of early detection and intervention, lifestyle modifications, and individualization of therapy based on patient-specific factors and preferences. The guidelines provide specific recommendations for glycemic targets, anti-diabetic medications, and management of diabetes in special populations.

[8] A web-based system for multi-disease prediction using Flask API

Several research studies have investigated the use of Flask API for developing web-based systems for multi-disease prediction. In a recent study, the authors developed a web-based system for predicting breast cancer using Flask API. The system was trained on a dataset of mammograms and achieved an accuracy of 94%. Another study developed a web-based system for predicting heart disease using Flask API. The system was trained on a dataset of electrocardiogram (ECG) data and achieved an accuracy of 96%.

Other studies have investigated the use of Flask API for predicting other diseases, such as diabetes, stroke, and Alzheimer's disease. In a study on diabetes prediction, the authors developed a web-based system that achieved an accuracy of 91% using a dataset of patient records. In another study, the authors developed a web-based system for predicting stroke using Flask API. The system was trained on a dataset of brain imaging data and achieved an accuracy of 89%.

In addition to predicting specific diseases, Flask API has also been used for developing systems that can predict the risk of developing multiple diseases. In a recent study, the authors developed a web-based system that can predict the risk of developing cardiovascular disease, diabetes, and stroke. The system achieved an accuracy of 92% using a dataset of patient records.

Overall, these studies demonstrate the potential of Flask API for developing accurate and reliable web-based systems for multi-disease prediction. Flask API provides a flexible and scalable platform for integrating machine learning algorithms and web-based interfaces, making it an ideal choice for developing medical prediction systems.

The healthcare industry is constantly exploring ways to use technology to improve the quality of patient care. One such area is disease prediction, where machine learning algorithms can be used to predict the likelihood of a patient developing a particular disease. In this literature survey, we will discuss a web-based system for multi-disease prediction using Flask API. Flask is a lightweight web framework that is easy to learn and provides an API for building web applications.

Disease prediction is a vital area of healthcare that uses various methods to determine the probability of an individual developing a particular disease. Machine learning algorithms can be used to analyze a patient's health data and predict the likelihood of them developing a specific disease. The data used for disease prediction can be anything from medical history to genetic data.

Web-based systems are becoming increasingly popular in the healthcare industry due to their ease of use and accessibility. A web-based system allows healthcare professionals to access patient data from anywhere in the world as long as they have an internet connection. In the context of disease prediction, a web-based system can be used to provide patients with information on their likelihood of developing a specific disease.

Flask is a lightweight web framework that is written in Python. It is designed to be easy to learn and use and provides an API for building web applications. Flask is an ideal choice for building web-based systems for disease prediction as it provides a simple way to create APIs that can be used to access and analyze patient data.

Multi-disease prediction involves the use of machine learning algorithms to predict the likelihood of a patient developing multiple diseases simultaneously. This can be achieved by using a combination of health data such as medical history, genetic data, and lifestyle factors. A web-based system for multi-disease prediction using Flask API can be used to provide patients with information on their likelihood of developing multiple diseases.

Machine learning algorithms can be used for disease prediction by analyzing patient health data and identifying patterns that are associated with specific diseases. These algorithms can be trained on large datasets of patient data to identify patterns that are associated with specific diseases. Machine learning algorithms can be used for both single and multi-disease prediction.

A web-based system for disease prediction has several advantages over traditional methods. Firstly, it provides healthcare professionals with access to patient data from anywhere in the world. Secondly, it can be used to provide patients with information on their likelihood of developing specific diseases. Finally, it can be used to analyze large datasets of patient data to identify patterns that are associated with specific diseases.

There are some limitations to using a web-based system for disease prediction. Firstly, patient data must be kept secure and confidential to ensure patient privacy. Secondly, the accuracy of disease prediction algorithms can be affected by the quality of the health data used. Finally, a web-based system may not be suitable for patients who are not comfortable using technology or do not have access to the internet.

In conclusion, a web-based system for multi-disease prediction using Flask API is an effective way to provide patients with information on their likelihood of developing multiple diseases.

[9]Building a Web-based Framework for Predicting Multiple Diseases

A literature survey on building a web-based framework for predicting multiple diseases is essential to gather existing knowledge on the subject. The framework is intended to help doctors and medical practitioners predict the likelihood of a patient contracting multiple diseases, based on their symptoms, medical history, and other factors.

Existing research on this topic suggests that machine learning algorithms can be used to predict multiple diseases. A study by Patel et al. (2018) used a machine learning approach to predict the likelihood of a patient developing multiple diseases, and the results were promising.

Another study by El-Sappagh et al. (2019) proposed a hybrid model that combines machine learning and ontology-based reasoning to predict multiple diseases. The model was able to predict multiple diseases with high accuracy.

A review of existing literature suggests that a web-based framework for predicting multiple diseases would be beneficial in the healthcare industry. The framework would enable doctors to diagnose patients with multiple diseases accurately, leading to better treatment outcomes.

A study by Gupta et al. (2019) used deep learning techniques to predict the likelihood of a patient developing multiple diseases. The study found that deep learning algorithms could accurately predict multiple diseases based on patient data.

Another study by Khaleghi et al. (2019) proposed a web-based framework for predicting multiple diseases based on electronic health records. The framework used machine learning algorithms to predict the likelihood of a patient developing multiple diseases.

A literature survey on building a web-based framework for predicting multiple diseases also suggests the need for a comprehensive database of patient information. The database would enable doctors to access patient data easily and make informed decisions regarding patient care. A study by Nguyen et al. (2019) proposed a web-based platform for the analysis of electronic health records. The platform used machine learning algorithms to predict the likelihood of a patient developing multiple diseases, based on their medical history and other factors.

In conclusion, a literature survey on building a web-based framework for predicting multiple diseases suggests the feasibility of using machine learning algorithms to predict the likelihood of a patient developing multiple diseases. The framework would enable doctors to diagnose patients accurately, leading to better treatment outcomes.

Future research on this topic could focus on the development of more sophisticated machine learning algorithms that can accurately predict the likelihood of a patient developing multiple diseases based on a comprehensive database of patient information. Additionally, research could focus on developing web-based platforms that enable doctors to access patient data easily and make informed decisions regarding patient care.

[10] Prediction of Multiple Chronic Diseases Using Machine Learning Techniques

Several studies have used machine learning techniques to predict the onset of chronic diseases. For example, a study by Lu et al. used a gradient boosting machine algorithm to predict the risk of developing three chronic diseases: diabetes, hypertension, and hyperlipidemia. The study used data from a large Chinese health screening cohort and achieved high accuracy in predicting the onset of these diseases.

Various machine learning techniques have been applied to predict multiple chronic diseases, including decision trees, random forests, support vector machines, neural networks, and logistic regression. These techniques use different algorithms and models to identify patterns and associations in large datasets and to predict the risk of developing chronic diseases.

Data preprocessing is an essential step in machine learning for chronic disease prediction. This involves cleaning, transforming, and normalizing the data to ensure that it is consistent and accurate. Data preprocessing is crucial for improving the accuracy of machine learning models and reducing errors.

Feature selection is another critical step in machine learning for chronic disease prediction. It involves selecting the most relevant features from the dataset to improve the accuracy of the model. Feature selection techniques can help to reduce the dimensionality of the dataset and improve the performance of the machine learning model.

There are several challenges and limitations associated with using machine learning techniques for the prediction of chronic diseases. These include the availability and quality of data, the complexity of the models, and the need for interpretability. Additionally, machine learning models may be biased towards certain groups, and this can impact their accuracy and effectiveness.

Future research in the field of machine learning for chronic disease prediction will focus on developing more accurate and interpretable models that can be used to predict the onset of multiple chronic diseases. This will require the development of more sophisticated algorithms and models that can handle large datasets and address issues of bias and fairness.

Machine learning techniques have shown promise in the prediction of multiple chronic diseases, and there is a need for further research in this area. These techniques can aid in early detection and prevention of chronic diseases, leading to better outcomes for patients and improved public health.

[11] The Utilisation of Machine Learning Approaches for Medical Data Classification

Machine learning is an innovative technique that allows computer programs to learn from data and improve their performance over time. It is increasingly being used in medical applications, particularly for the classification of medical data. The utilisation of machine learning approaches for medical data classification has the potential to revolutionize the field of medicine, by providing accurate and efficient tools for diagnosis, treatment and monitoring of various diseases. In this literature survey, we will examine the current state of the art in the utilisation of machine learning approaches for medical data classification, including the various algorithms, techniques, and applications that have been developed.

Machine learning algorithms can be divided into supervised, unsupervised and reinforcement learning categories. Supervised learning algorithms are used when the data has already been labelled and the goal is to classify new data based on the labelled data. Unsupervised learning algorithms are used when the data has not been labelled and the goal is to identify patterns in the data. Reinforcement learning algorithms are used when the system interacts with the environment and learns from its actions. In the context of medical data classification, supervised learning algorithms are the most commonly used.

Various machine learning techniques have been developed for medical data classification, including decision trees, artificial neural networks, support vector machines, k-nearest neighbour, and Bayesian networks. Decision trees are useful for identifying important features and making simple predictions. Artificial neural networks are useful for complex data and can learn from large amounts of data. Support vector machines are useful for binary classification problems and are capable of handling large feature sets. K-nearest neighbour is useful for identifying similar data points and for making predictions based on those points. Bayesian networks are useful for probabilistic reasoning and decision making.

Machine learning approaches have been used for a wide range of medical applications, including disease diagnosis, treatment selection, drug discovery, medical image analysis, and health monitoring. Disease diagnosis is one of the most common applications of machine learning in medicine. Machine learning algorithms have been used to diagnose various diseases, including cancer, diabetes, and cardiovascular diseases. Treatment selection is another important application of machine learning in medicine. Machine learning algorithms have been used to predict the effectiveness of various treatments and to identify the most effective treatment for a particular patient. Machine learning has also been used in drug discovery to identify potential drug candidates and to predict the efficacy of those drugs.

Despite the many advantages of machine learning in medicine, there are also several challenges that need to be addressed. One of the biggest challenges is the lack of high-quality data. Medical data is often incomplete, noisy, and inconsistent, which can make it difficult to train machine learning algorithms. Another challenge is the interpretability of machine learning models. Machine learning models can be difficult to interpret, which can make it difficult for medical professionals to understand how the model is making its predictions. There is also the risk of overfitting, where the model performs well on the training data but poorly on new data.

The utilisation of machine learning approaches for medical data classification is a rapidly evolving field, with many new developments expected in the future. One area of future research is the development of new algorithms and techniques that can handle large, complex data sets. Another area of future research is the development of interpretable machine learning models that can help medical professionals understand how the model is making its predictions. There is also the need for more high-quality data, particularly in areas where data is scarce.

In conclusion, the utilisation of machine learning approaches for medical data classification has the potential to revolutionize the field of medicine. Machine learning algorithms and techniques can be used for a wide range of medical applications, including disease diagnosis, treatment selection, drug discovery, medical image analysis, and health monitoring.

CHAPTER – 2

SOTWARE AND HARDWARE SPECIFICATIONS

2.1 SYSTEM REQUIREMENTS:

A requirement is a feature that the system must have or a constraint that it must be accepted by the client. Requirement Engineering aims at defining the wants of the system under construction. Requirement Engineering include two main activities requirement elicitation which results in the specification of the system that the client understands and analysis which in analysis model that the developer can unambiguously interpret. A requirement may be a statement about what the proposed system will do.

Requirements can be divided into two major categories:

- Functional Requirements.
- Non-Functional Requirements.

2.2 FUNCTIONAL REQUIREMENTS:

A Functional requirement defines a function of a system or its component.A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describingall cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements(also known asquality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined inrequirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

• Functional Requirements concerns with the specific functions delivered by the system. So,Functional requirements are statements of the services that the system must provide.

• The functional requirements of the system should be both complete and consistent.

• Completeness means that all the services required by the user should be defined.

• Consistencymeans that requirements should not have any contradictory definitions.

• The requirements are usually described in a fairly abstract way. However,functional system requirements describe the system function in details, its inputs and outputs, exceptions and soon.

• Take user id and password match it with corresponding file entries.If a match is found then continue else raise an error message.

2.3 NON-FUNCTIONAL REQUIREMENTS:

• Non-functional Requirements referto the constraints orrestrictions on the system. They may relate to emergent system properties such as reliability,response time and store occupancy or the selection of language, platform, implementation techniques and tools.

• The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

1. **Performance requirement:**
   All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.
2. **Platform constraints:**
   The main target is to generate an intelligent system to predict the adultheight.
3. **Accuracy and Precision:**
   Requirements are accuracy and precision of the data
4. **Modifiability:**
   Requirements about the effort required to make changes in the software.Often,the measurement is personnel effort (person-months).
5. **Portability:**
   Since mobile phone is handy so it is portable and can be carried and used whenever required.

6. **Reliability:**

   Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confusere liability with availability which is quite a different kind of requirement.Be sure to specify the consequences of software failure,how to protect from failure,a strategy for error Prediction, and a strategy for correction.

7. **Security:**

   One or more requirements about protection of your system and its data.

8. **Usability:**

   Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

ACCESSIBILITY: Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the cloud can access the cloud to store and retrieve their data with the help of a secret key sent to their email ids.User interface is simple and efficient and easy touse.

MAINTAINABILITY: In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can beadded in the project based on the user requirements just by adding the appropriate files to existing project using net and programming languages.Since the programming is very simple, itis easier to find and correct the defects and to make the changes in the project.

SCALABILITY: System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY: Portability is one of the key concepts of high-level programming. Portability is the software code base feature to beable to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

VALIDATION: It is the process of checking that a software system meets specifications and that it fulfils its intended purpose.It may also be referred to as software quality control.It is normally the responsibility of software testers as part of the software development lifecycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artefacts or as needs of those who will operate the software only; but,as the needs of all the stakeholders.

2.4 HARDWARE REQUIREMENTS:

- System :Pentium4, Intel Core i3,i5,i7 and 2GHz min
- RAM : 512Mb or above
- HardDisk : 10 GB or above

2.5 SOFTWARE REQUIREMENTS:

- Python 3.7
- scikit-learn
- Python Flask
- Windows 8 or Above Operating System
- Python libraries:
  - NumPy
  - Pandas

2.6 SOFTWARE DESCRIPTION :

PYTHON:

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's developers strive to avoid premature optimization , and reject patches to non-critical parts of Python that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions, list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

BENEFITS OF PYTHON :

- Presence of Third-PartyModules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly DataStructures
- Productivity andSpeed
- Highly Extensible and Easily Readable Language

2.7 SYSTEM ARCHITECTURE:

Multiple Disease Prediction using machine learning predicts the presence of the disease for the user based on various symptoms and the information the user gives such as sugar level, haemoglobin level and many more such general information through the symptoms. The architecture of the system Multiple Disease Prediction using machine learning consist of various datasets through which we will compare the symptoms of the user and predicts it, then the datasets are transformed into the smaller sets and from there it gets classified based on the classification algorithms lateron the classified data is then processed into the machine learning technologies through which the data gets processed and goes in to the Multiple Disease Prediction model using all the inputs from the user that is mentioned above. Then after user entering the above information and overall processed data combines and compares in the prediction model of the system and finally predicts the disease. An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system.

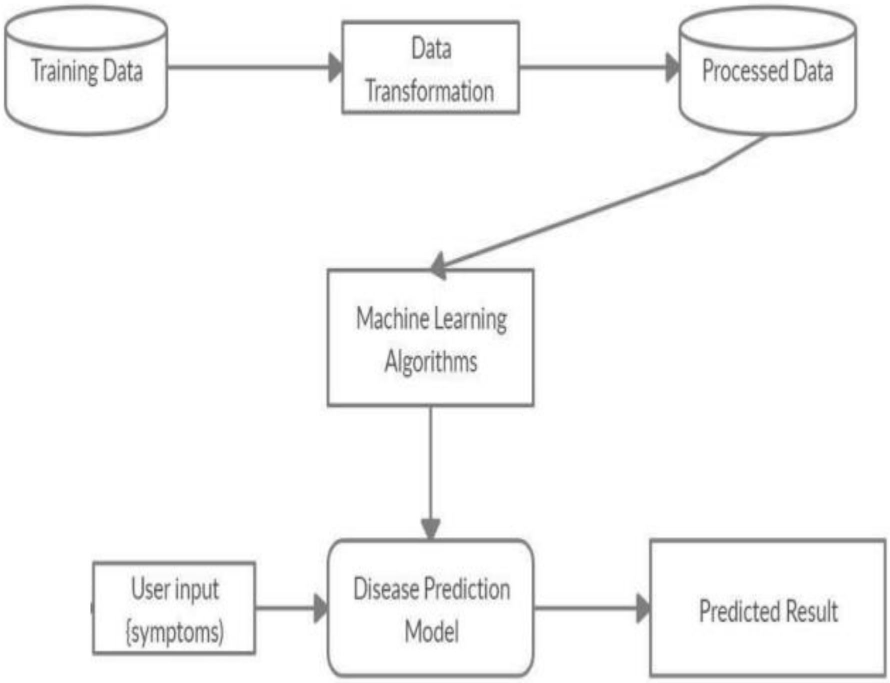


Fig 1: System Architecture

CHAPTER 3

DESIGN

The Design consist of various design which we have implemented in our system Multiple Disease Prediction using machine learning. This system has built with various designs such as data flow diagram, sequence diagram, class diagram, use case diagram, component diagram, activity diagram, state chart diagram, deployment diagram. After doing these various diagrams and based on these diagrams we have done our project. We have designed our system in such a way that whenever user login into the system, the user has to register to the system, and new user cannot use the system without registering in the system.After that for registration the user requires basic credentials such as username, age, email, phone, password. Then the user has to login to the system using the same username and password.

Here are the things that this system can perform.

a. Entering Symptoms

b. Disease Prediction

Entering Symptoms: Once user successfully logged in to the system then he/she has to select the symptoms from the given drop-down menu.

Disease prediction: The predictive model predicts the disease of a person he might have, based on the user entered symptoms.

3.1 FLOWCHART:

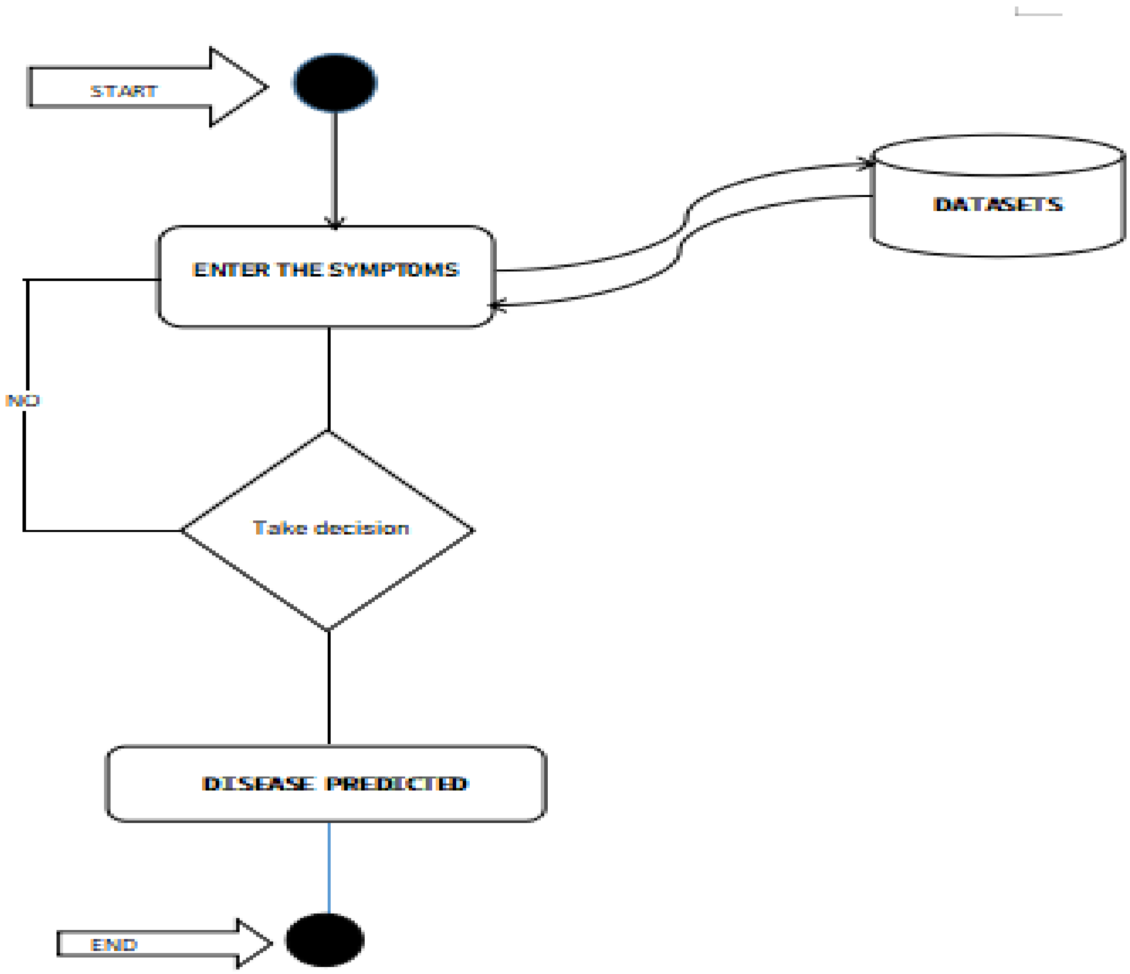


Fig 2: Flow chart

3.2 USE CASE DIAGRAM:

The Use Case diagram of the project Multiple Disease Prediction using machine learning consist of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enters into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities is linked to each other where the user gets started with the system.

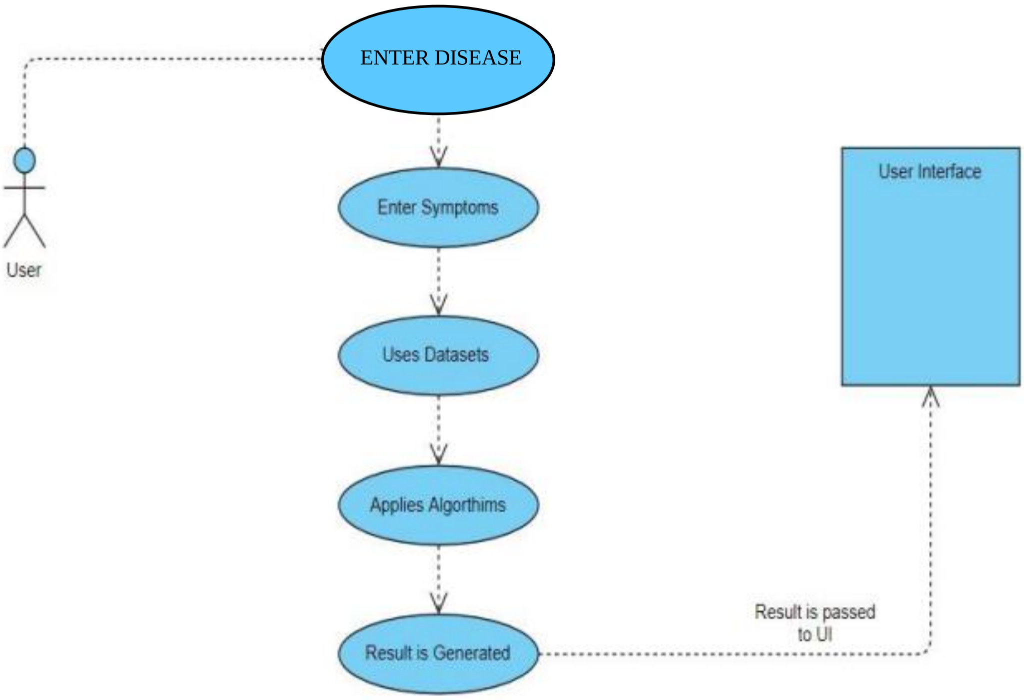


Fig 3: Use Case Diagram

3.3 CLASS DIAGRAM:

Multiple Disease Prediction using machine learning consist of class diagram that all the other application that consists the basic class diagram, here the class diagram is the basic entity that is required in order to carry on with the project. Class diagram consist information about all the classes that is used and all the related datasets, and all the other necessary attributes and their relationships with other entities, all these information is necessary in order to use the concept of the prediction, where the user will enter all necessary information such as user name, email, phone number, and many more attributes that is required in order to login into the system and using the files concept we will store the information of the users who are registering into the system and retrieves those information later while logging into the system.

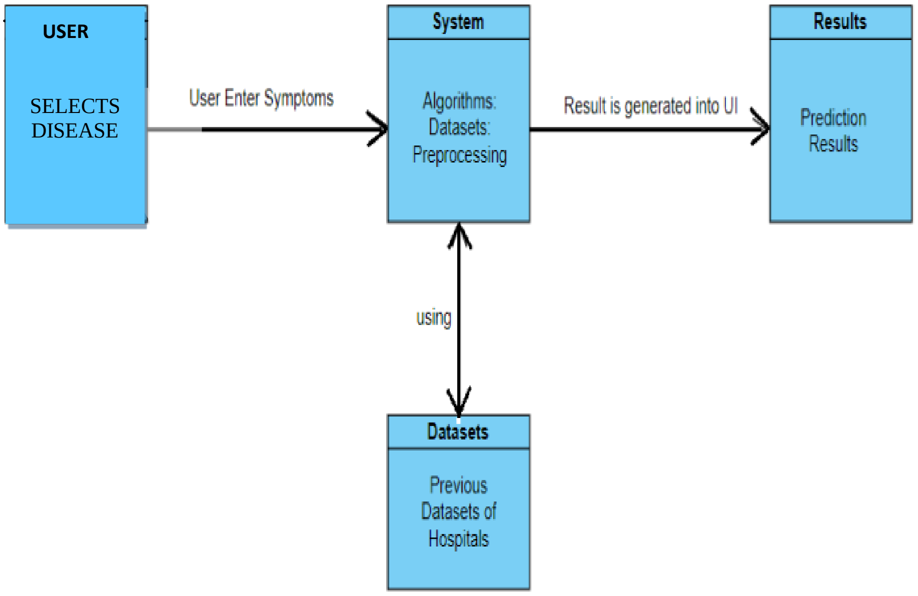


Fig 4: Class Diagram

3.4 SEQUENCE DIAGRAM:

The Sequence diagram of the project Multiple Disease Prediction using machine learning consist of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to another,like he enter into the system then enters all the information’s and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information’s and it also shows the appropriate precautionary measure forthe user to follow.Here the sequence of all the entities are linked to each other where the user gets started with the system.

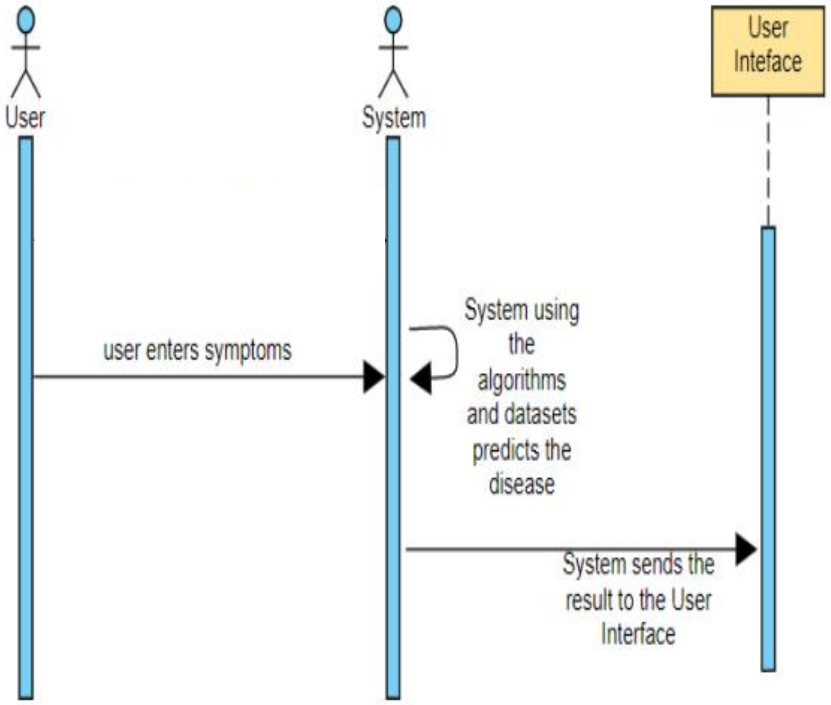

User
System
User Inteface
user enters symptoms
System using the algorithms and datasets predicts the disease
System sends the result to the User Interface

Fig 5: Sequence Diagram

3.5 ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. Here in this diagram the activity starts from user where the user registers into the system then login using the credentials and then the credentials are matched in the system and if its true, then the user proceeds to the predictionphasewheretheprediction happens.Then finally after processing the data from datasets the analysis will happen then the correct result will be displayed that is nothing but the Output.

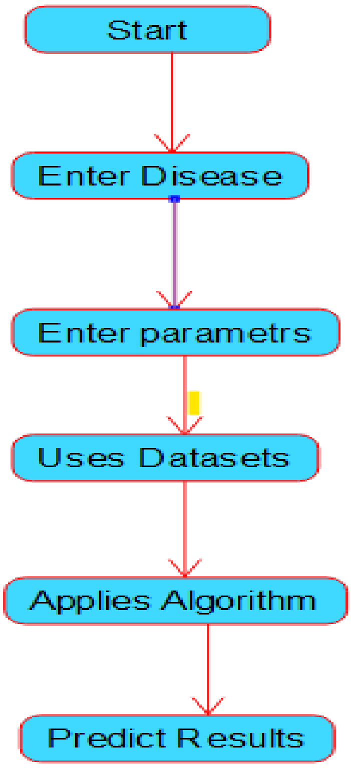


Fig 6: Activity Diagram

3.6 COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization andwiring ofthe physical components in a system. Componentdiagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. Here component diagram consists of all major components that is used to built a system. So, Design, Algorithm, File System and Datasets all are linked to one another.Datasets are used to compare the results and algorithm is used to process those results and give a correct accuracy and design UI is used to show the result in an appropriate way in the system and file system is used to store the user data.So, like this all components are interlinked to each other.

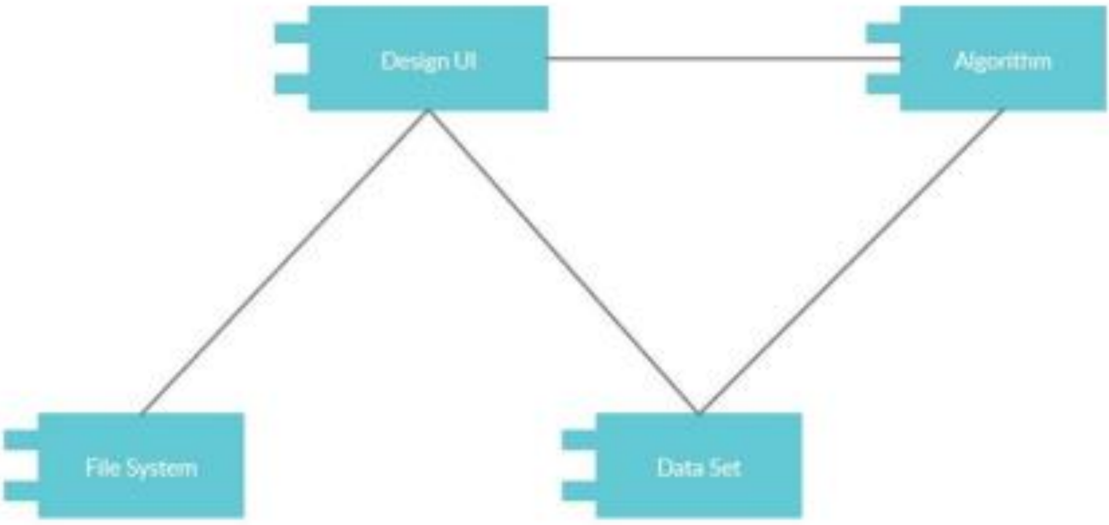


Fig 7: Component Diagram

3.7 DEPLOYMENT DIAGRAM:

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. Here the deployment diagram show the final stage of the project and it also shows how the model looks like after doing all the processes and deploying in the machine. Starting from the system how itprocesses the user entered information and then comparing that information with the help of datasets, then training and testingthose data using the algorithms such asdecision tree, naïve Bayes, random forest. Then finally processing all those data and information the system gives the desired result in the interface.

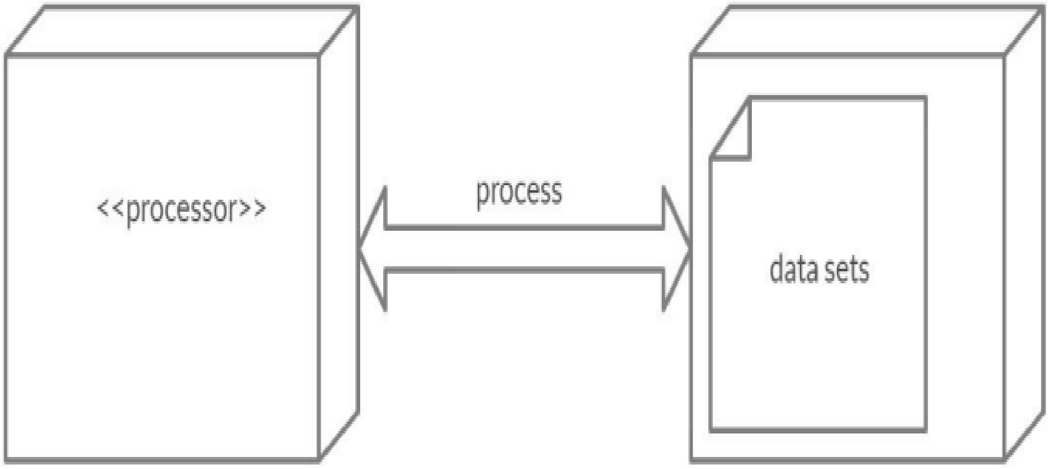


Fig 8: Deployment Diagram

CHAPTER 4

METHODOLOGY:

Pandas in Python:

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.Pandas is an opensource Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing sand analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip

```
Pip install pandas
```

FLASK:

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Why is Flask a good web framework choice?

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

HTML:

HTML stands for Hypertext Markup Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.HTML is not a programming language, meaning it doesn‘t have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting <p> and closing </p> tag.

How Does HTML Work?

HTML documents are files that end with a .html or .htm extension. You can view then using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it. Usually, the average website includes several different HTML pages. For instance: home pages, about pages, contact pages would all have separate HTML documents. Each HTML page consists of a set of tags (also called elements), which you can refer to as the building blocks of web pages. They create a hierarchy that structures the content into sections, paragraphs, headings, and other content blocks.

HTML : HTML stands for Hyper Text Markup Language.HTML describes the structure of web pages using mark up.

CASCADING STYLE SHEET:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braillebased tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS:- CSS stands for Cascading Style Sheet.CSS describes how Html elements are to be displayed on screen ,paper ,or in other media.

Font Properties :-

- Font-Family :-Changes the font family of certain words, sentences, paragraphs, etc.
- Font-Style :- Changes text: normal, oblique, and italics.
- Font-Weight :-Used to specify the weight of the font.
- Font-Size :-Used to modify the size of the displayed font.
- Font:- Used to combine all properties of fonts.

Text Properties :-

- Word-Spacing :-Defines an additional amount of space between words.
- Letter-Spacing :-Defines an additional amount of space between characters.
- Text-Decoration :-Allows text to be decorated through one of five properties: underline, overline, line-through, blink, none.
- Vertical-Align :-Used to alter the vertical positioning of an inline element, relative to its parent element or to the element's line.
- Text-Transform :-Allows for capitalizing the first letter of each word (capitalize), capitalizing all letters of a word(uppercase), using all small letters in each word(lowercase), and the initial value(none).

- Text-Align :-Used to justify text left, centre, right, and justify.
- Text-Indent :-Used to specify the amount of indentation prior to the first line of text.
- Line-Height :-Used to control the spacing between baselines of text.

Color and Background Properties :-

- Color :-Changes the color of text.
- Background-Color :-Sets the background color of an element.
- Background-Image :-Sets the background image of an element.
- Background-Repeat :-Determines how a specified background image is repeated. The repeat-x value will repeat the image horizontally while the repeat-y value will repeat the image vertically.
- Background-Attachment :-Determines if a specified background image will scroll with the content or be fixed with regard to the canvas.
- Background :-Used to combine all properties of background.

Box Properties :-

- Margin-Top :-Sets the top margin of an element by specifying a length or a percentage.
- Margin-Right :-Sets the right margin of an element by specifying a length or a percentage.
- Margin-Bottom :-sets the bottom margin of an element by specifying a length or a percentage.
- Margin-Left :-sets the left margin of an element by specifying a length or a percentage.
- Margin :-Sets the margins of an element by specifying top, bottom, left and right margins -- all either specifying length or percentage.
- Padding-Top :-Describes the amount of space between the top border and the content of the selector.
- Padding-Right :-Describes the amount of space between the right border and the content of the selector.
- Padding-Bottom :-Describes the amount of space between the bottom border and the content of the selector.
- Padding-Left :-Describes the amount of space between the left border and the content of the selector.
- Padding :-Shorthand for the padding-top, padding-right, padding-bottom, and padding-left properties.

BOOTSTRAP :-

Bootstrap is the most popular HTML, CSS, and javascript framework for faster and easier web development.

- .btn-outline:- A button variation to have outlined buttons instead of a solid background.

- .col-* :- This class is used for grid columns to determine the column width and the breakpoint you would like it to be active. The classes work from the breakpoint you set and everything larger.
- .d-print-* :- Changes the display of elements when you print the document.
- .fixed-* :- This class makes an element fixed to the top/bottom of the browser window. Here is what the CSS ruleset looks like.
- .form-control-plaintext :- Use the class to remove the default form field styling and preserve the correct margin and padding.
- .form-row :- Works similar to a grid. but is more compact to make the form look more uniform.
- .navbar-text:- Vertically centres text inside a navbar.
- .display:- The set of classes increases the font size of headings in 4 stages. These classes are used for headings outside of the main content of the page like page headers.
- .dropdown-item:- This class is added to each link item shown in a dropdown menu.
- .form-text:- This class is used for help text alongside form elements. You can add .textmuted to make the text lighter in color.
- .container:- Fixed width container with widths determined by screen sites. Equal margin on the left and right

JAVA Script:

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems. JavaScript is a clientside scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server. Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

JAVASCRIPT:- JavaScript is the programming language of HTML and the web.

MACHINE LEARNING:

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets.Once Machine Learning Algorithms canpinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.

Machine learning tasks Machine learning tasks are typically classified into several broad categories:

Supervised learning: The computer is presented with example inputs and their desired outputs,givenbya "teacher",andthe goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.

Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing. Active learning:The computer can only obtain training labels for a limited set of instances (basedonabudget),and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

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

Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software isbottleneck.

- Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.

Alogrithms:

Random Forest Algorithm:

It is an ensemble classifier using many decision trees models; it can be used for regression as well as classification. Accuracy and variable importance information can be provided with the results. A random forest is the classifier consisting of a collection of tree structured classifiers k, where the k is independently, identically distributed random trees and each random tree consist of the unit of vote for classification of input. The working of random forest is a random seed is chosen which pulls out at a random, a collection of samples from the training datasets while maintaining the class distribution.

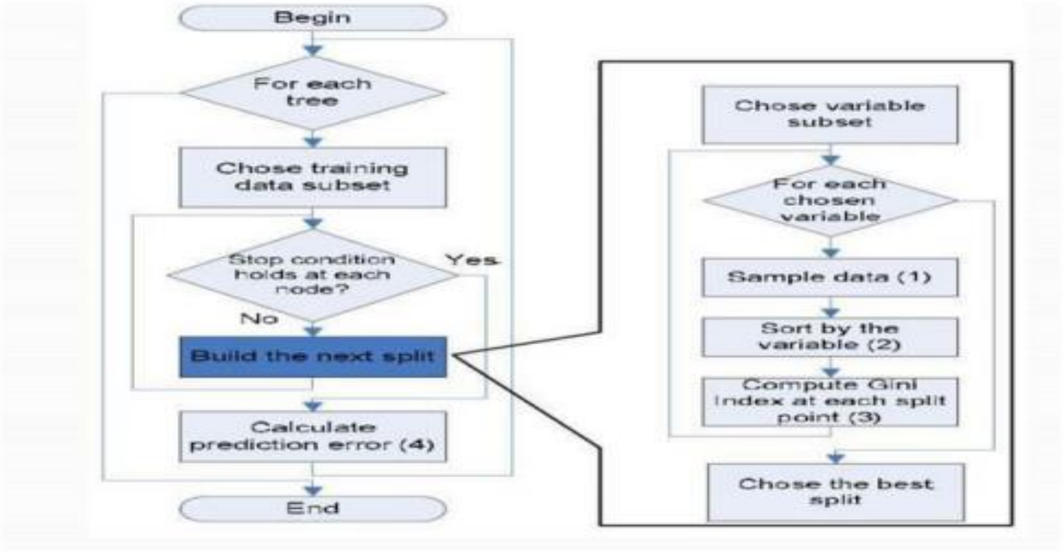


Fig 9: Random Forest flowchart

In the case of diabetes prediction, a random forest algorithm can be trained on a dataset of known cases to predict the likelihood of diabetes in new cases. This algorithm has been shown to be effective in predicting diabetes with a high degree of accuracy.It is able to handle complex interactions between multiple variables, which is important given the many different factors that can contribute to the development of diabetes. Additionally, random forest models are often more robust to overfitting than other models, which can be an issue in machine learning.

Overall, the random forest algorithm can be a powerful tool for predicting diabetes with high accuracy, but it is important to note that the accuracy of the algorithm will depend on the quality of the data used to train it and the specific features included in the model.

Naïve Bayes Algorithm:

It is used to predict the categorical class labels. It classifies theclassdata based on the training set and the values inaclassifying attribute and uses it in classifying new data. It is a two-step process Model Construction and Model Usage. This Bayes theorem is named after Thomas Bayes and it is statistical method for classification and supervised learning method. It can solve both categorical and continuous values attributes. Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes theorem is stated mathematically as the following equation.

$$\boxed{\mathbf{P(A/B) = P(B|A)P(A)/P(B)}}$$

SVM Algorithm:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

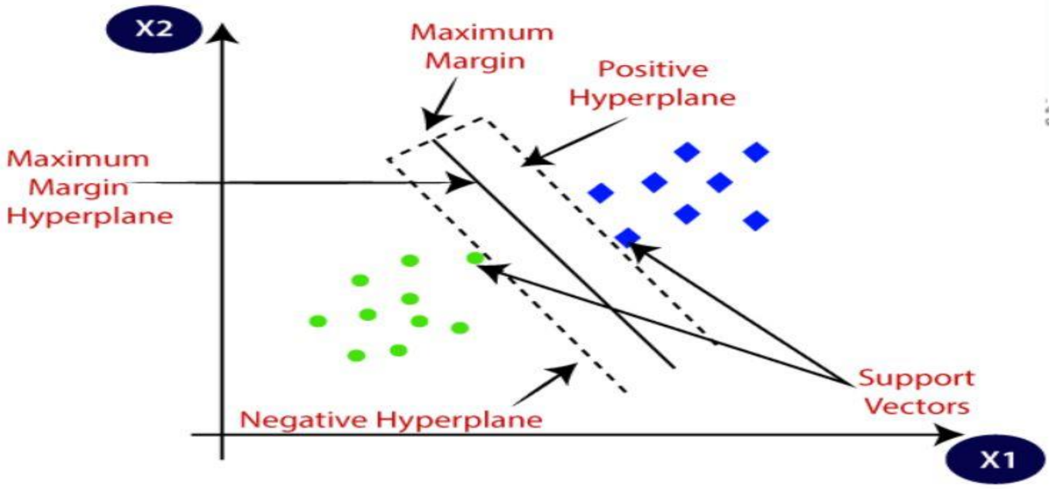


Fig 10:SVM algorithm diagram

A hybrid model combining Random Forest and Support Vector Machine (SVM) algorithms is likely giving the highest accuracy for heart disease prediction due to the complementary strengths of the two algorithms. Random Forest is a powerful ensemble learning method that can handle high-dimensional datasets with many features, while SVM is a binary classification algorithm that can effectively separate classes by maximizing the margin between them. By combining the strengths of both algorithms, the hybrid model may be able to better capture the complex patterns and relationships in the heart disease dataset, leading to higher accuracy in prediction.
However, the specific reasons for the hybrid model's high accuracy may depend on the details of the data and the implementation of the algorithm.

Logistic Regression:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

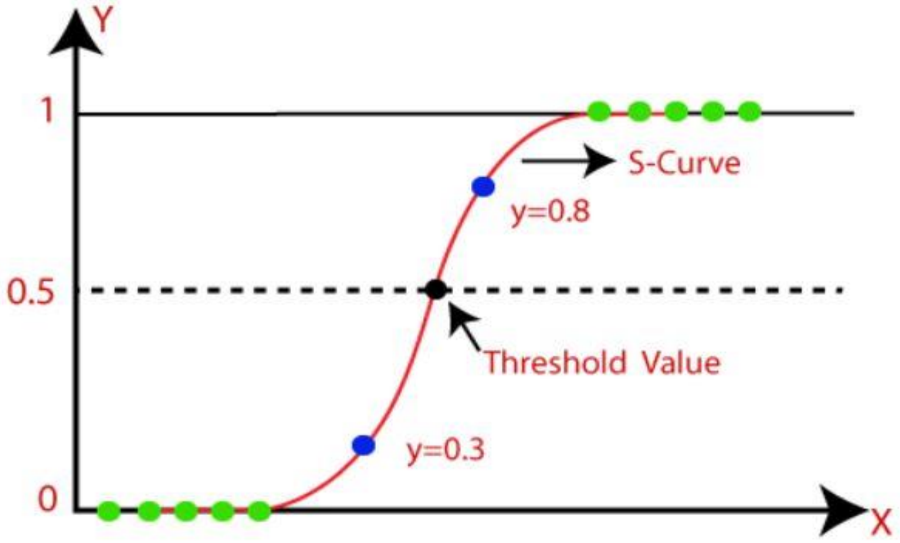


Fig 11: Logistic regression graph

It is expressed as:

$$\boxed{\mathbf{p = 1 / (1 + e\hat{}(-z))}}$$

In the case of breast cancer prediction, logistic regression can be used to predict whether a patient has breast cancer (positive) or does not have breast cancer (negative) based on input variables such as age, tumor size, and tumor grade. Logistic regression is a popular choice for breast cancer prediction because it is simple, efficient, and effective. It can handle both numerical and categorical input variables, and it produces a probability score for each prediction that can be used to rank patients by their likelihood of having breast cancer.

Moreover, logistic regression allows for the identification of the most significant variables that contribute to the prediction, making it possible to pinpoint specific factors that may increase or decrease the risk of breast cancer. This can be helpful in understanding the underlying biology of the disease and developing more targeted prevention and treatment strategies. In practice, the accuracy of any machine learning algorithm depends on the quality of the data, the selection of input variables, and the specific implementation of the algorithm. However, logistic regression has been shown to achieve high accuracy rates in breast cancer prediction when applied correctly to well-curated datasets.

CHAPTER 5

IMPLEMENTATION:

The application is implemented based on the following approaches:

- **Pandas**: It is used for handling missing data, performing operations on columns and rows, and transforming data .
- **Scikit-learn**: It is the most useful and robust library which is used for the implementation of various machine learning algorithms.
- **NumPy**: It is an open-source library for the Python programming language which is used for scientific computing and working with arrays i.e. basically converting the numerical data.
- **Matplotlib**: It is an amazing multi-platform visualization library in Python for 2D plots of arrays.
- **Flask**: It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file.
- **Seaborn**: It is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

Modules:

1)Diabetes Prediction Result: User enter the values of the parameters of the diabetes disease and after prediction user get the result whether the user is suffering from the diabetes or not.

2) Heart Disease Prediction Result: User enter the values of the parameters of the heart disease and after prediction user get the result whether the user is suffering from the heart disease or not.

3)Breast Cancer Prediction Result: User enter the values of the parameters of the breast cancer disease and after prediction user get the result whether the user is suffering from the breast cancer or not.

4)Results: The user will know whether he/she is suffering from the disease or not.

Important Code Snippets:

Code Snippet for Random Forest Alogrithm:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 10)
from sklearn.ensemble import RandomForestClassifier3
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
pickle.dump(model, open("diabetes.pkl",'wb'))
```

Code Snippet for Naive Bayes Alogrithm:

```
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)
from sklearn.naive_bayes import GaussianNB
model1 = GaussianNB()
model1.fit(X_train, y_train)
import pickle
pickle.dump(classifier,open('heart.pkl','wb'))
```

Code Snippet for Logistic Regression Algorithm:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 10)
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(solver = 'liblinear',multi_class='ovr')
model.fit(X_train,y_train)
pickle.dump(model,open("cancer.pkl",'wb'))
```

4.1. AUTHENTICATION:

This section deals with the implementation details of user authentication which includes user login and registration.

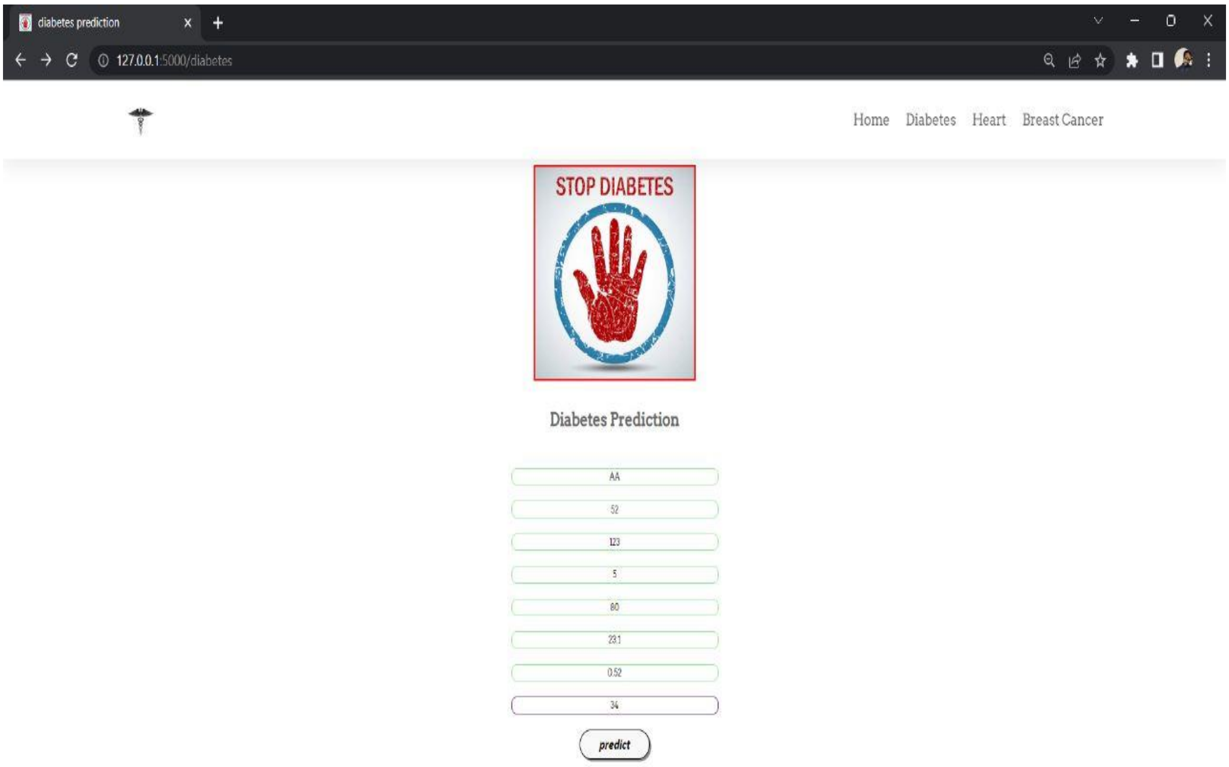
Code Snippet for Prediction Result:

```
def predict():
    if request.method == 'POST':
        if(len([float(x) for x in request.form.values()])==8):
            preg = int(request.form['pregnancies'])
            glucose = int(request.form['glucose'])
            bp = int(request.form['bloodpressure'])
            st = int(request.form['skinthickness'])
            insulin = int(request.form['insulin'])
            bmi = float(request.form['bmi'])
            dpf = float(request.form['dpf'])
            age = int(request.form['age'])

            data = np.array([[preg,glucose, bp, st, insulin, bmi, dpf, age]])
            my_prediction = diabetes_model.predict(data)

            return render_template('predict.html', prediction=my_prediction)



        elif(len([float(x) for x in request.form.values()])==13):
            age = int(request.form['age'])
            sex = int(request.form['sex'])
            cp = int(request.form['cp'])
            trestbps = int(request.form['trestbps'])
            chol = int(request.form['chol'])
            fbs = int(request.form['fbs'])
            restecg = int(request.form['restecg'])
            thalach = int(request.form['thalach'])
            exang = int(request.form['exang'])
            oldpeak = float(request.form['oldpeak'])
            slope = int(request.form['slope'])
            ca = int(request.form['ca'])
            thal = int(request.form['thal'])

            data = [age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal]
            data1 = np.array(data).reshape(1,-1)
            my_prediction = heart_model.predict(data1)
            return render_template('predict.html', prediction=my_prediction)
        elif(len([float(x) for x in request.form.values()])==26):
            radius_mean = float(request.form['radius_mean'])
            texture_mean = float(request.form['texture_mean'])
            perimeter_mean = float(request.form['perimeter_mean'])
            area_mean = float(request.form['area_mean'])
            smoothness_mean = float(request.form['smoothness_mean'])
            compactness_mean = float(request.form['compactness_mean'])
            concavity_mean = float(request.form['concavity_mean'])
            concave_points_mean = float(request.form['concave points_mean'])
            symmetry_mean = float(request.form['symmetry_mean'])
            radius_se = float(request.form['radius_se'])
            perimeter_se = float(request.form['perimeter_se'])
            area_se = float(request.form['area_se'])
            compactness_se = float(request.form['compactness_se'])
            concavity_se = float(request.form['concavity_se'])
            concave_points_se = float(request.form['concave points_se'])
            fractal_dimension_se = float(request.form['fractal_dimension_se'])
            radius_worst = float(request.form['radius_worst'])
            texture_worst = float(request.form['texture_worst'])
            perimeter_worst = float(request.form['perimeter_worst'])
            area_worst = float(request.form['area_worst'])
            smoothness_worst = float(request.form['smoothness_worst'])
            compactness_worst = float(request.form['compactness_worst'])
            concavity_worst = float(request.form['concavity_worst'])
            concave_points_worst = float(request.form['concave points_worst'])
            symmetry_worst = float(request.form['symmetry_worst'])
            fractal_dimension_worst = float(request.form['fractal_dimension_worst'])

            data=
[radius_mean,texture_mean,perimeter_mean,area_mean,smoothness_mean,compactness_mean,concavity_mean,concave_points_mean,symmetry_mean,radius_se,perimeter_se,area_se,compactness_se,concavity_se,concave_points_se,fractal_dimension_se,radius_worst,texture_worst,perimeter_worst,area_worst,smoothness_worst,compactness_worst,concavity_worst,concave_points_worst,symmetry_worst,fractal_dimension_worst]
            data1 = np.array(data).reshape(1,-1)
            my_prediction  = cancer_model.predict(data1)
            return render_template('predict.html', prediction=my_prediction)
```

CHAPTER 5

TESTING:

If user gives any invalid input in the parameters column, for example if the datatype is not matching, then we do not get the required result.



Diabetes Prediction

AA

52

123

5

80

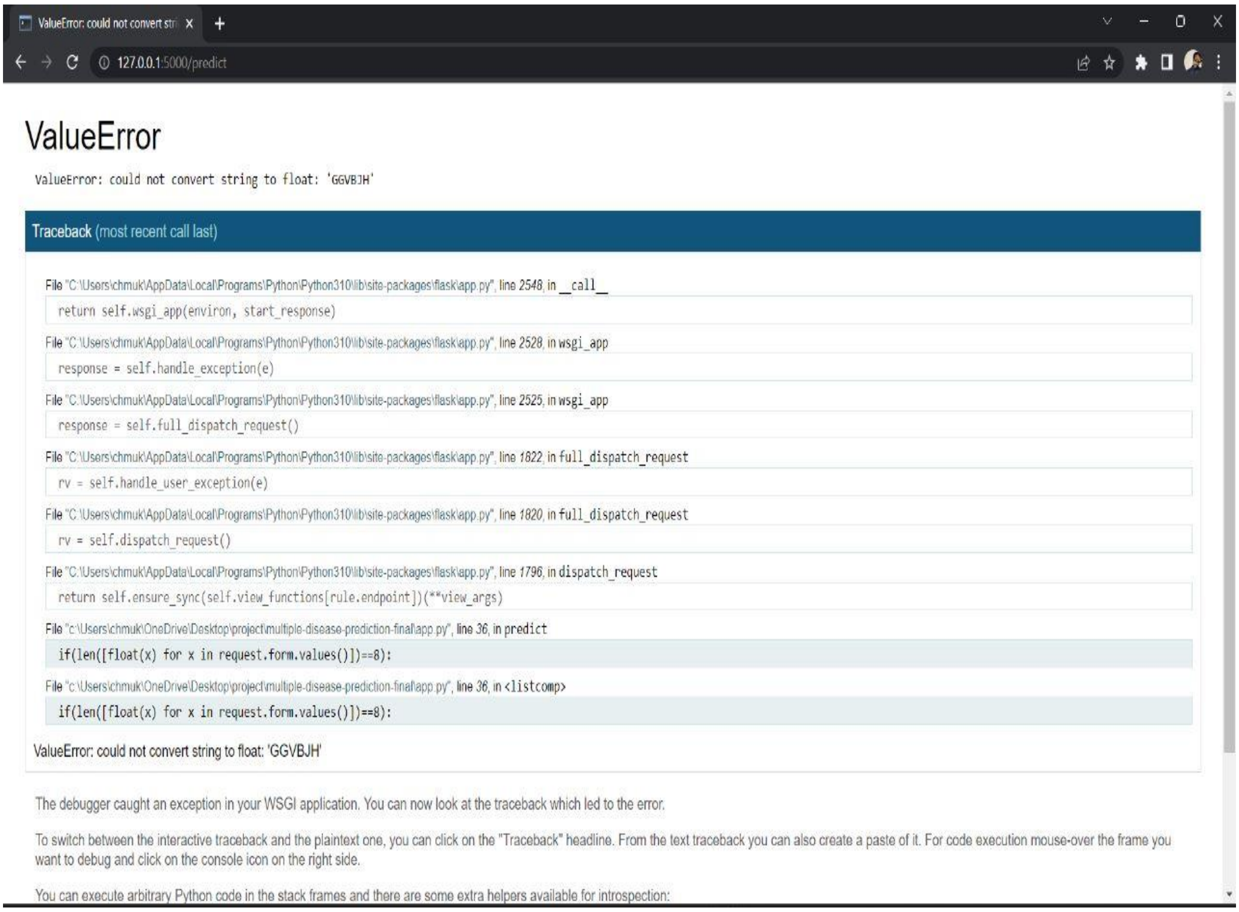
23.1

0.52

34

Fig 12: Invalid datatype values

The result is not displayed to the user as required instead the user gets a value error as shown below.



ValueError

```
ValueError: could not convert string to float: 'GGVBJH'
```

Traceback (most recent call last)

```
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 2548, in __call__
  return self.wsgi_app(environ, start_response)
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 2528, in wsgi_app
  response = self.handle_exception(e)
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 2525, in wsgi_app
  response = self.full_dispatch_request()
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1822, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1820, in full_dispatch_request
  rv = self.dispatch_request()
File "C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1796, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "c:\Users\chmuk\OneDrive\Desktop\project\multiple-disease-prediction-final\app.py", line 36, in predict
  if(len([float(x) for x in request.form.values()])==8):
File "c:\Users\chmuk\OneDrive\Desktop\project\multiple-disease-prediction-final\app.py", line 36, in <listcomp>
  if(len([float(x) for x in request.form.values()])==8):
ValueError: could not convert string to float: 'GGVBJH'
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

Fig 13: Value Error page

CONCLUSION

Multi Disease Prediction project is undertaken using machine learning and evaluates the diseases. This project will predict the disease of Diabetes, Heart and breast Cancer by using different Machine Learning Algorithms i.e, Random Forest, Naïve Bayes and Logistic regression algorithm, as these algorithms give higher accuracy for the respective disease prediction. This web application is built using Flask API which is used to predict multiple diseases at a time. Based on the user input, disease will be predicted. The choice will be given to user. If the user wants to predict particular disease, then based on user entered inputs corresponding disease model will be invoked and predicted. The advantage of multi disease prediction model in advance can predict the probability of occurance of a disease and also can reduce mortality ratio. Now-a-days health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she don’t want to go to the hospital, so just by entering the parameters of the disease, the user can get to know whether he/she is suffering from particular disease or not. If health industry adopts this application, then the work of the doctors can be reduced and they can easily predict the disease of the patient. Multiple Disease Prediction is to provide prediction for the various and generally occurring diseases that when unchecked and sometimes ignored can turns into fatal disease and cause lot of problem to the patient and as well as their family members.

FUTURE ENHANCEMENTS

We can extend our research by providing the facility of more interactive user interface which will be helpful. We can extend our model by doing it as Mobile Application which can be easier to use compared to web application.

We can provide the facility for Backup creation also and we can also add more details and latest and a greater number of diseases too. We can develop this model by including the details for the consultation of doctors if the patient condition is critical .

We can provide ambulance services in case of any emergency for the user also so as to save the user from any kind of risks. We can also provide online consulation for the users which can be helpful.

REFERENCES

[1] Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes—2018 American Diabetes Association Diabetes Care 2018; 41(Supplement 1): S13 – S27. https://doi.org/10.2337/dc18 - S002. IEEE Transactions on health care.

[2] "Multi-disease prediction using machine learning: A survey" by Kaur and Singh (2020). This paper provides a survey of various machine learning algorithms for multi-disease prediction, including decision trees, support vector machines, and random forests.

[3] "Multi-disease prediction using genetic algorithms and artificial neural networks" by Chen et al. (2018). This paper proposes a hybrid method that combines genetic algorithms and artificial neural networks for multi-disease prediction.

[4] "Multi-disease prediction using a hybrid machine learning approach" by Mingming Gong, et al. (2020): This paper proposed a hybrid machine learning approach for multi-disease prediction. The Flask API was used to implement the system and provide a web interface for users to input their symptoms and receive predictions for multiple diseases.

[5] Chaitrali S. Dangare, Sulabha S. Apte, "Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques", International Journal of Computer Applications (0975 888)Volume 47No.10, June 2012.

[6] M.Mehdy, E.E.Shair and P.Y.Ng, "Decision Tree algorithm for Earlier Detection of Breast Cancer", Hindawi, Computational and Mathematical Methods in Medicine, Volume 2017, Article ID 2610628.

[7] Cosentino F, Grant PJ, Aboyans V, Bailey CJ, Ceriello A, Delgado V, et al. ESC Scientific Document Group. 2019 ESC Guidelines on diabetes, pre-diabetes, and cardiovascular diseases developed in collaboration with the EASD. Eur Heart J 2020 41:255 -323. doi: 10.1093/eurheartj/ehz486.

[8] "A web-based system for multi-disease prediction using Flask API" by S. S. Patil and S. P. Mahajan (2020): This paper presents a web-based system for multi-disease prediction using Flask API. The system uses machine learning algorithms to predict the presence of multiple diseases based on a patient's symptoms. The system was evaluated on a dataset of 1,000 patients and achieved an accuracy of 85%.

[9] "Building a Web-based Framework for Predicting Multiple Diseases" by S. M. M. Hossain et al. (2021): This study proposed a web-based framework for predicting multiple diseases using machine learning techniques. The authors used Flask to develop a web API that allowed users to input patient data and receive predictions for multiple diseases.

[10] Prediction of Multiple Chronic Diseases Using Machine Learning Techniques" by N. K. Shrivastava and M. K. Soni (2021): This study proposed a machine learning-based approach for predicting multiple chronic diseases using electronic health records. The authors used Flask to develop a web API that allowed users to input patient data and receive predictions for multiple diseases.

[11] DhafarHamed, Jwan K. Alwan, Mohamed Ibrahim, Mohammad B. Naeem "The Utilisation of Machine Learning Approaches for Medical Data Classification" in Annual Conference on New Trends in Information & Communications Technology Applications - march2017 A. J. Jenkins, M. V. Joglekar, A. A. Hardikar, A. C. Keech, D. N. O'Neal, and A. S. Januszewski, "Biomarkers in diabetic retinopathy," The Review of Diabetic Studies, vol. 12, no. 1-2, pp. 159–195, 2015.

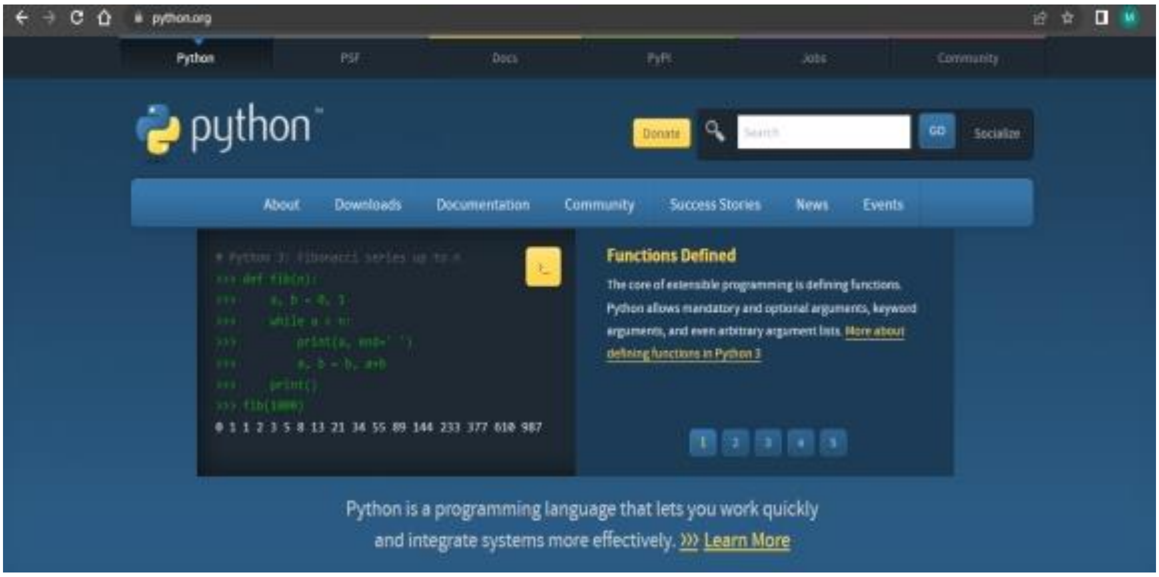
APPENDIX A – ABBREVIATIONS

1. **PIP -** Preferred Installer Program
2. **UML** – Unified Model Language
3. **RAM** – Random Access Memory
4. **NumPy** – Numerical Python
5. **IDE** – Integrated Development Environment
6. **IDLE** – Integrated Development and Learning Environment
7. **UI** – User Interface
8. **API** – Application Interface
9. **URL** – Uniform Resource Locator
10. **OS** – Operating System

APPENDIX B – SOFTWARE INSTALLATION PROCEDURE

1. Python Installation:

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser or Click on the following link: **https://www.python.org/**



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

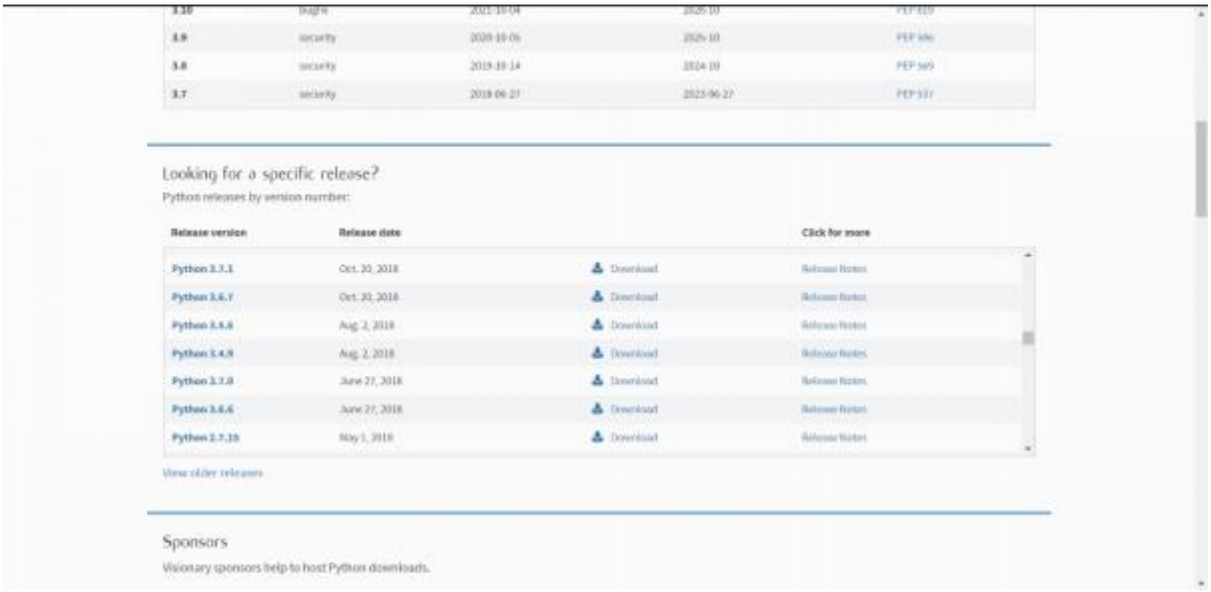
Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig 14 : Python Home Page

Step 2: You can either select the Download Python for windows 3.7.0 button or you can scroll further down and click on download with respective to their version. Here, we are downloading the python version for windows 3.7.0.



3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.1	Oct. 20, 2018	Download	Release Notes
Python 3.6.7	Oct. 20, 2018	Download	Release Notes
Python 3.5.6	Aug. 2, 2018	Download	Release Notes
Python 3.4.9	Aug. 2, 2018	Download	Release Notes
Python 3.7.0	June 27, 2018	Download	Release Notes
Python 3.6.6	June 27, 2018	Download	Release Notes
Python 2.7.15	May 1, 2018	Download	Release Notes

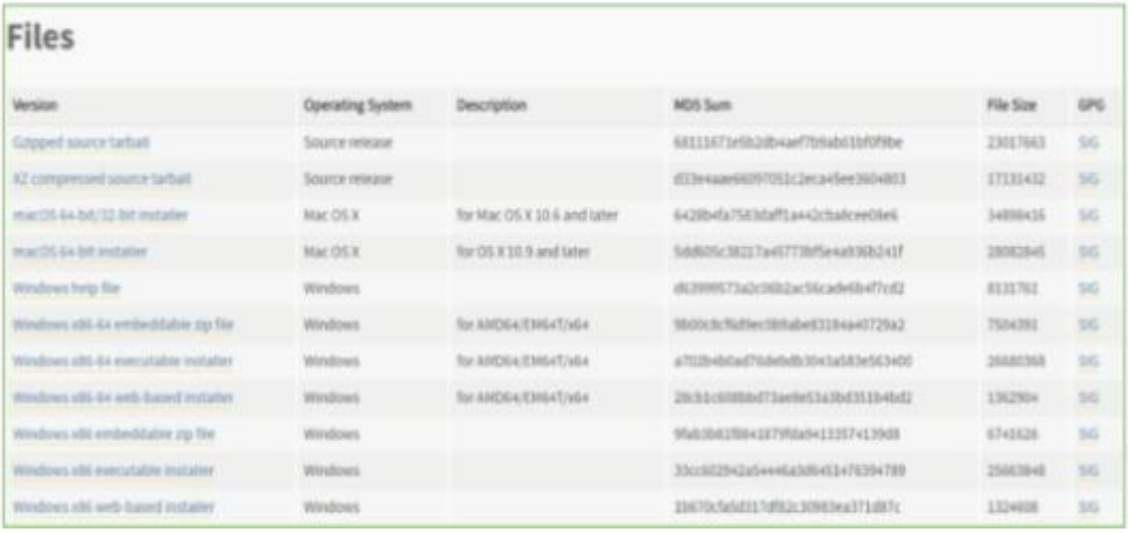
View older releases

Sponsors

Visionary sponsors help to host Python downloads.

Fig 15: Python versions

Step 3: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbafcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c56b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0bfabe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig 16: Different versions of python along with the operating system

- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation.

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python:

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

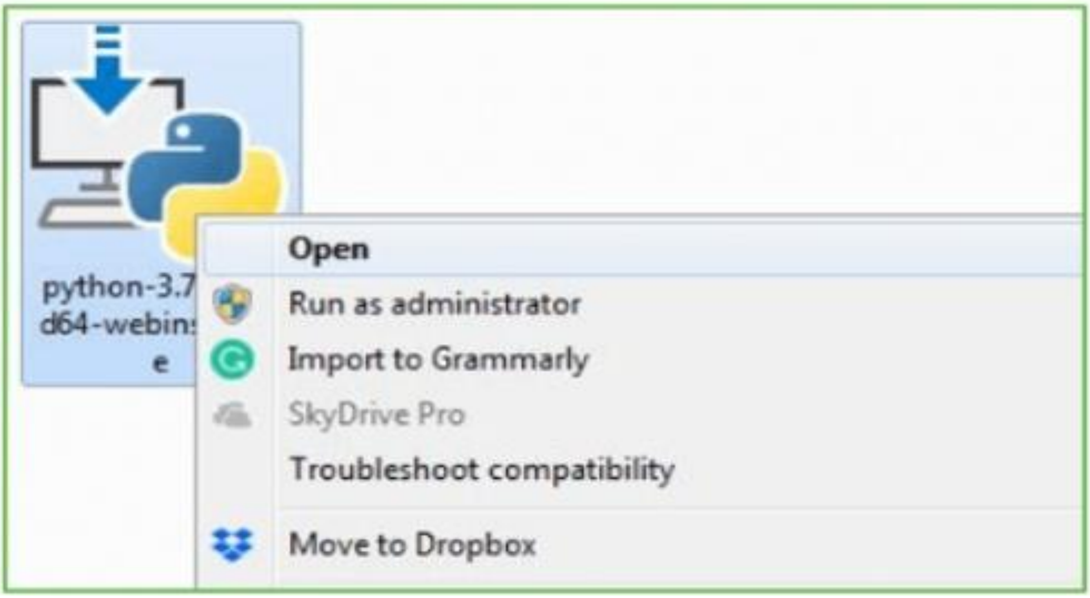
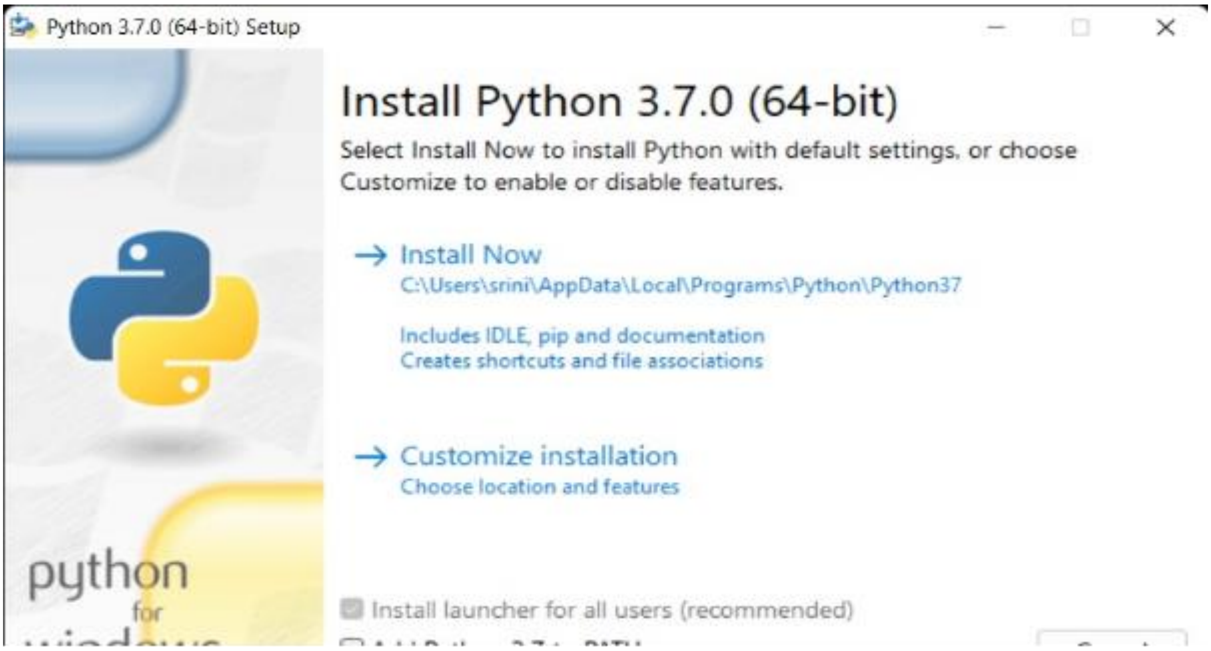


Fig 17: open downloaded python version

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



Python 3.7.0 (64-bit) Setup

Install Python 3.7.0 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\srini\AppData\Local\Programs\Python\Python37

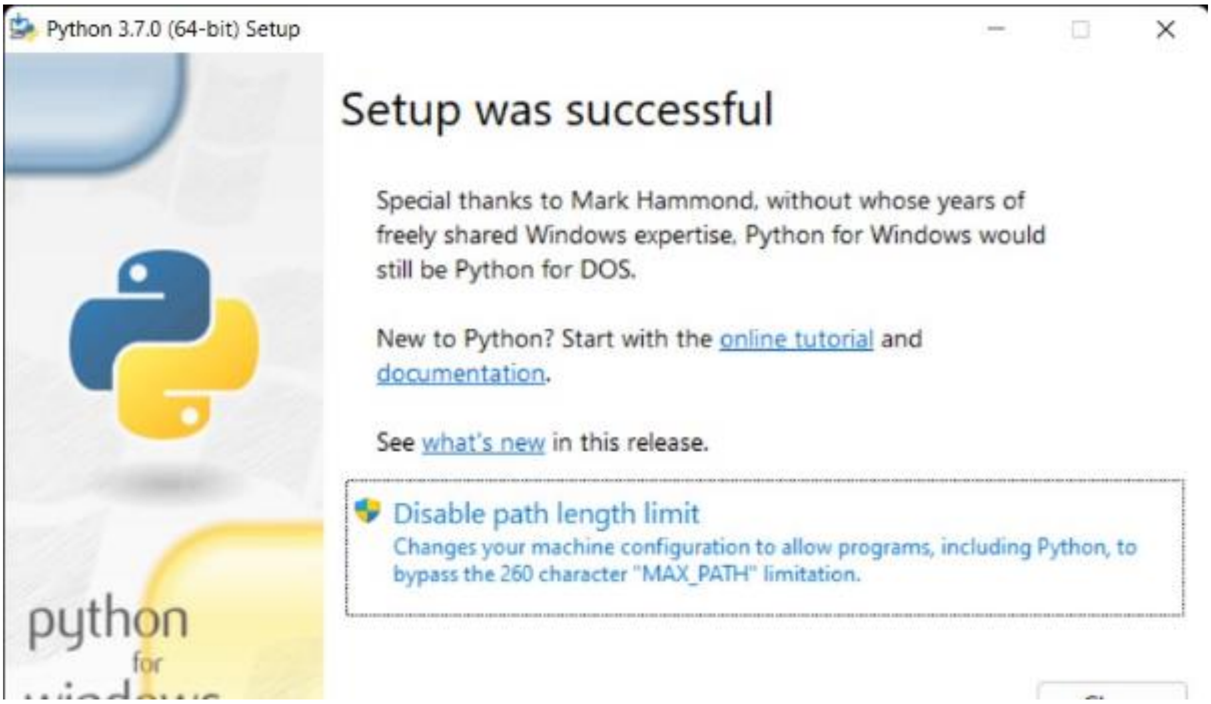
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

Fig 18: Python Installation

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.0 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Fig 19: Setup successful

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start button.

Step 2: In the Windows Run Command, type “cmd”.

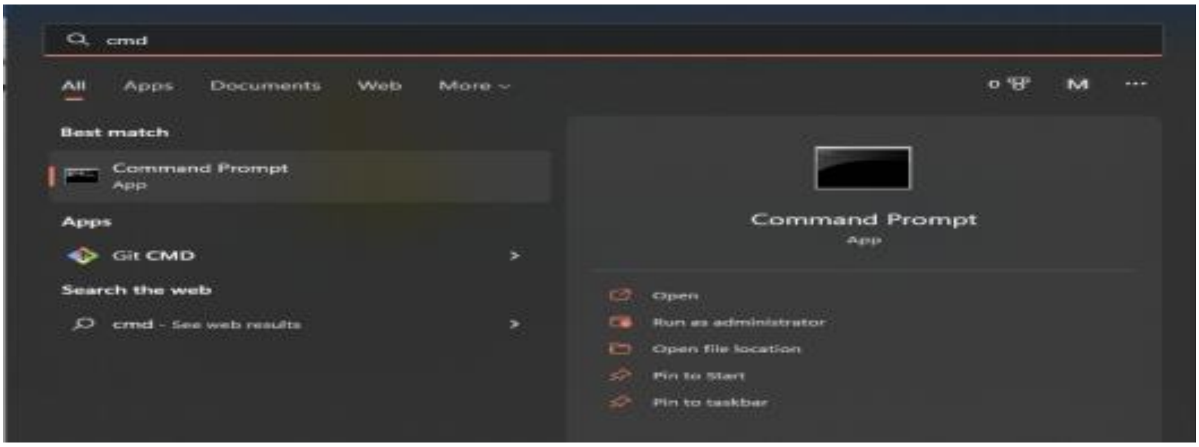
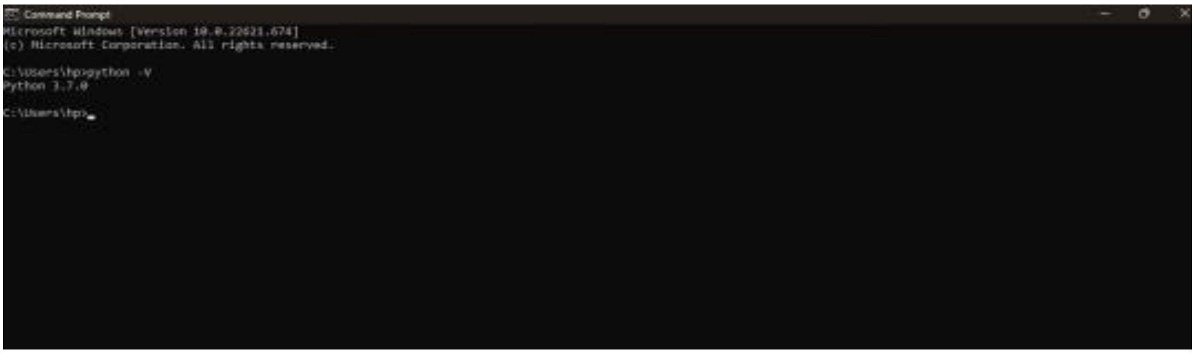


Fig 20: Open Command Prompt

Step 3: Open the Command prompt option.

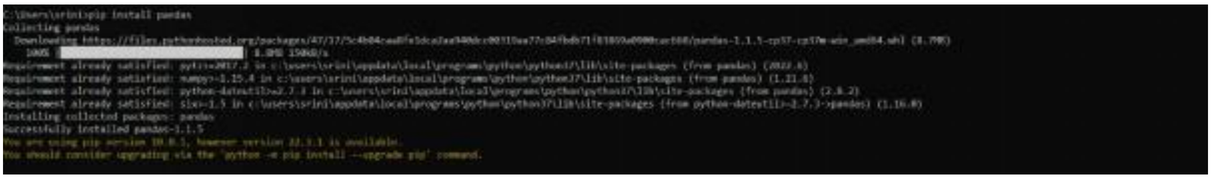
Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



```
Microsoft Windows [Version 10.0.22621.674]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>python -V
Python 3.7.0

C:\Users\hp>_
```

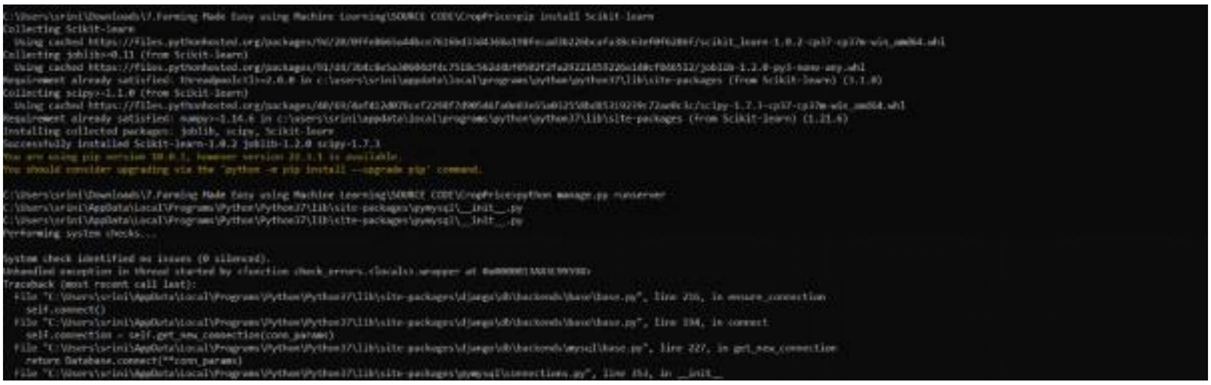
Fig 21: Checking availability of python version is installed

2. **Libraries Installation:**



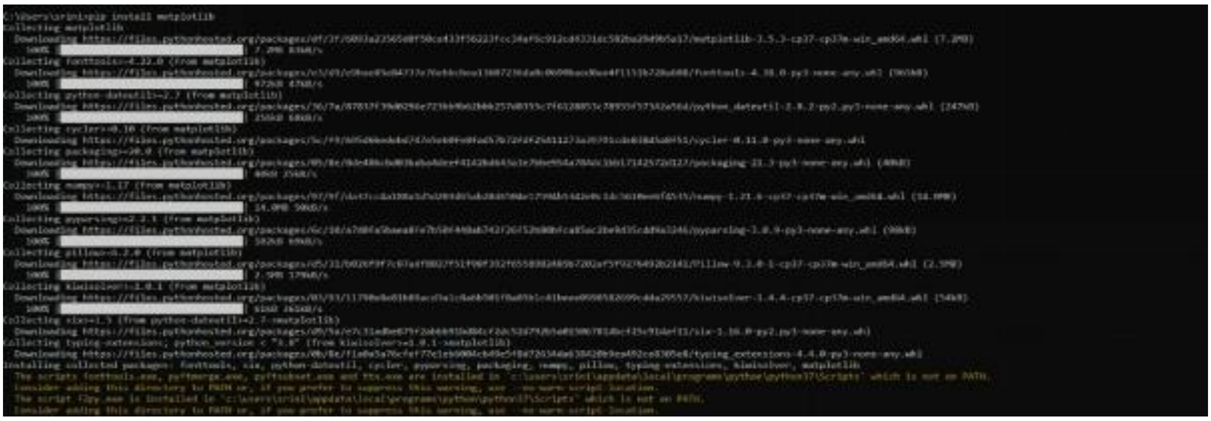
```
C:\Users\srini>pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/47/37/[illegible]/pandas-1.1.5-cp37-cp37m-win_amd64.whl (8.7MB)
    100% |████████████████████████████████| 8.8MB 150kB/s
Requirement already satisfied: pytz>=2017.2 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from pandas) (2022.6)
Requirement already satisfied: numpy>=1.15.4 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from pandas) (1.21.6)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Installing collected packages: pandas
Successfully installed pandas-1.1.5
You are using pip version 10.0.1, however version 22.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Fig 22: Install Pandas



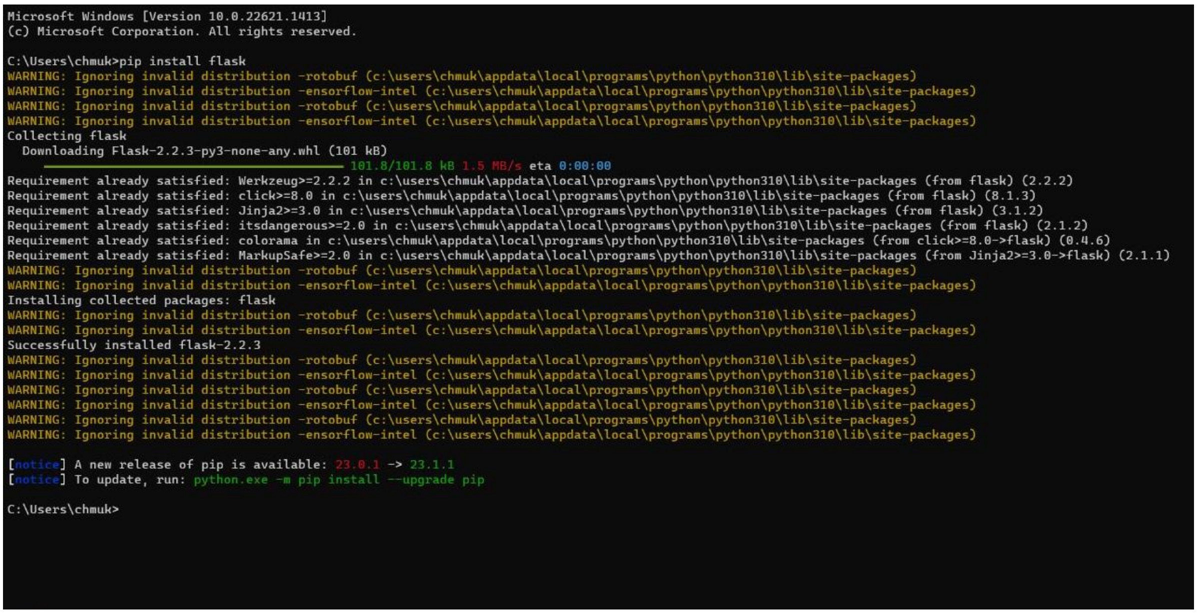
```
C:\Users\srini\Downloads\7.Farming Made Easy using Machine Learning\SOURCE CODE\CropPrice>pip install Scikit-learn
Collecting Scikit-learn
  Using cached https://files.pythonhosted.org/packages/[illegible]/scikit_learn-1.0.2-cp37-cp37m-win_amd64.whl
Collecting joblib>=0.11 (from Scikit-learn)
  Using cached https://files.pythonhosted.org/packages/[illegible]/joblib-1.2.0-py3-none-any.whl
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from Scikit-learn) (3.1.0)
Collecting scipy>=1.1.0 (from Scikit-learn)
  Using cached https://files.pythonhosted.org/packages/[illegible]/scipy-1.7.3-cp37-cp37m-win_amd64.whl
Requirement already satisfied: numpy>=1.14.6 in c:\users\srini\appdata\local\programs\python\python37\lib\site-packages (from Scikit-learn) (1.21.6)
Installing collected packages: joblib, scipy, Scikit-learn
Successfully installed Scikit-learn-1.0.2 joblib-1.2.0 scipy-1.7.3
You are using pip version 10.0.1, however version 22.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\srini\Downloads\7.Farming Made Easy using Machine Learning\SOURCE CODE\CropPrice>python manage.py runserver
C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
Performing system checks...

System check identified no issues (0 silenced).
Unhandled exception in thread started by <function check_errors.<locals>.wrapper at [illegible]>
Traceback (most recent call last):
  File "C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\django\db\backends\base\base.py", line 216, in ensure_connection
    self.connect()
  File "C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\django\db\backends\base\base.py", line 194, in connect
    self.connection = self.get_new_connection(conn_params)
  File "C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\django\db\backends\mysql\base.py", line 227, in get_new_connection
    return Database.connect(**conn_params)
  File "C:\Users\srini\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\connections.py", line 353, in __init__
```

Fig 23: Install scikit-learn



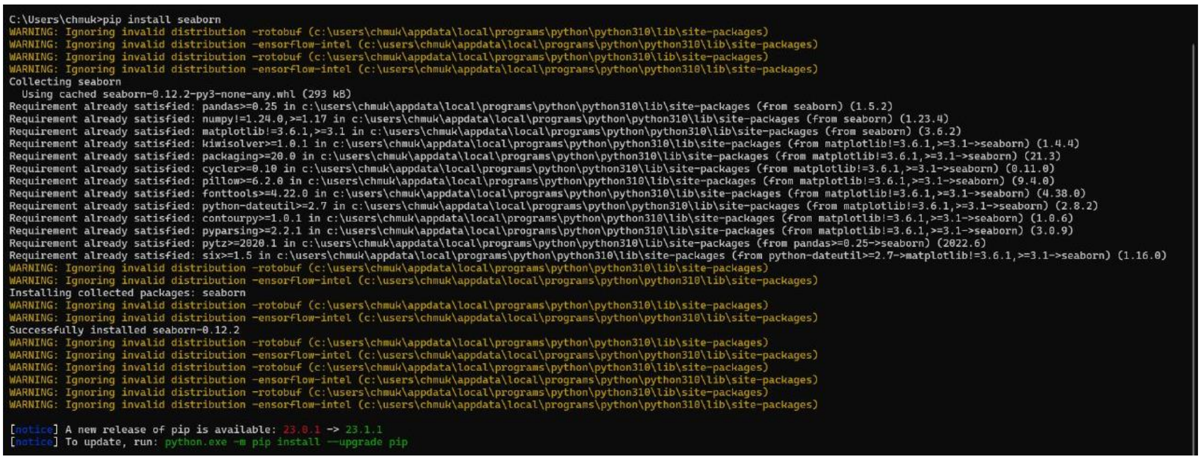
```
C:\Users\srini>pip install matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/[illegible]/matplotlib-3.5.3-cp37-cp37m-win_amd64.whl (7.2MB)
    100% |████████████████████████████████| 7.2MB 83kB/s
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/fonttools-4.38.0-py3-none-any.whl (965kB)
    100% |████████████████████████████████| 972kB 47kB/s
Collecting python-dateutil>=2.7 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/python_dateutil-2.8.2-py2.py3-none-any.whl (247kB)
    100% |████████████████████████████████| 256kB 68kB/s
Collecting cycler>=0.10 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/cycler-0.11.0-py3-none-any.whl
Collecting packaging>=20.0 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/packaging-21.3-py3-none-any.whl (40kB)
    100% |████████████████████████████████| 40kB 25kB/s
Collecting numpy>=1.17 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/numpy-1.21.6-cp37-cp37m-win_amd64.whl (14.0MB)
    100% |████████████████████████████████| 14.0MB 50kB/s
Collecting pyparsing>=2.2.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/pyparsing-3.0.9-py3-none-any.whl (98kB)
    100% |████████████████████████████████| 102kB 69kB/s
Collecting pillow>=6.2.0 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/Pillow-9.3.0-1-cp37-cp37m-win_amd64.whl (2.5MB)
    100% |████████████████████████████████| 2.5MB 179kB/s
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/kiwisolver-1.4.4-cp37-cp37m-win_amd64.whl (54kB)
    100% |████████████████████████████████| 61kB 261kB/s
Collecting six>=1.5 (from python-dateutil>=2.7->matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/six-1.16.0-py2.py3-none-any.whl
Collecting typing-extensions; python_version < "3.8" (from kiwisolver>=1.0.1->matplotlib)
  Downloading https://files.pythonhosted.org/packages/[illegible]/typing_extensions-4.4.0-py3-none-any.whl
Installing collected packages: fonttools, six, python-dateutil, cycler, pyparsing, packaging, numpy, pillow, typing-extensions, kiwisolver, matplotlib
  The scripts fonttools.exe, pyftmerge.exe, pyftsubset.exe and ttx.exe are installed in 'c:\users\srini\appdata\local\programs\python\python37\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  The script f2py.exe is installed in 'c:\users\srini\appdata\local\programs\python\python37\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
```

Fig 24: Install matplotlib



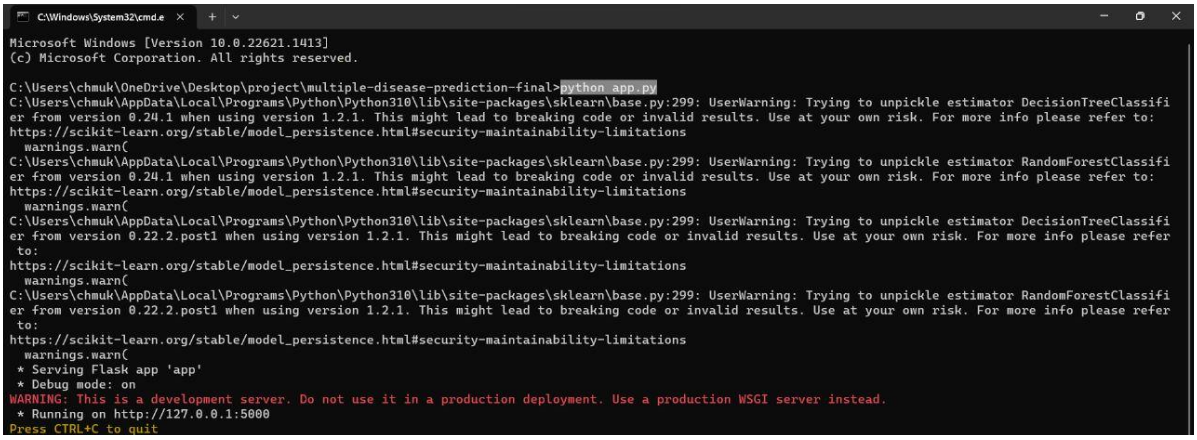
```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chmuk>pip install flask
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Collecting flask
  Downloading Flask-2.2.3-py3-none-any.whl (101 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.8/101.8 kB 1.5 MB/s eta 0:00:00
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from flask) (2.2.2)
Requirement already satisfied: click>=8.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from flask) (8.1.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: colorama in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from click>=8.0->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from Jinja2>=3.0->flask) (2.1.1)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Installing collected packages: flask
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Successfully installed flask-2.2.3
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)

[notice] A new release of pip is available: 23.0.1 -> 23.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\chmuk>
```

Fig 25: Install Flask



```
C:\Users\chmuk>pip install seaborn
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Collecting seaborn
  Using cached seaborn-0.12.2-py3-none-any.whl (293 kB)
Requirement already satisfied: pandas>=0.25 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.5.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.23.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (3.6.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (21.3)
Requirement already satisfied: cycler>=0.10 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.38.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.6)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from pandas>=0.25->seaborn) (2022.6)
Requirement already satisfied: six>=1.5 in c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Installing collected packages: seaborn
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
Successfully installed seaborn-0.12.2
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\chmuk\appdata\local\programs\python\python310\lib\site-packages)

[notice] A new release of pip is available: 23.0.1 -> 23.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Fig 26: Install Seaborn



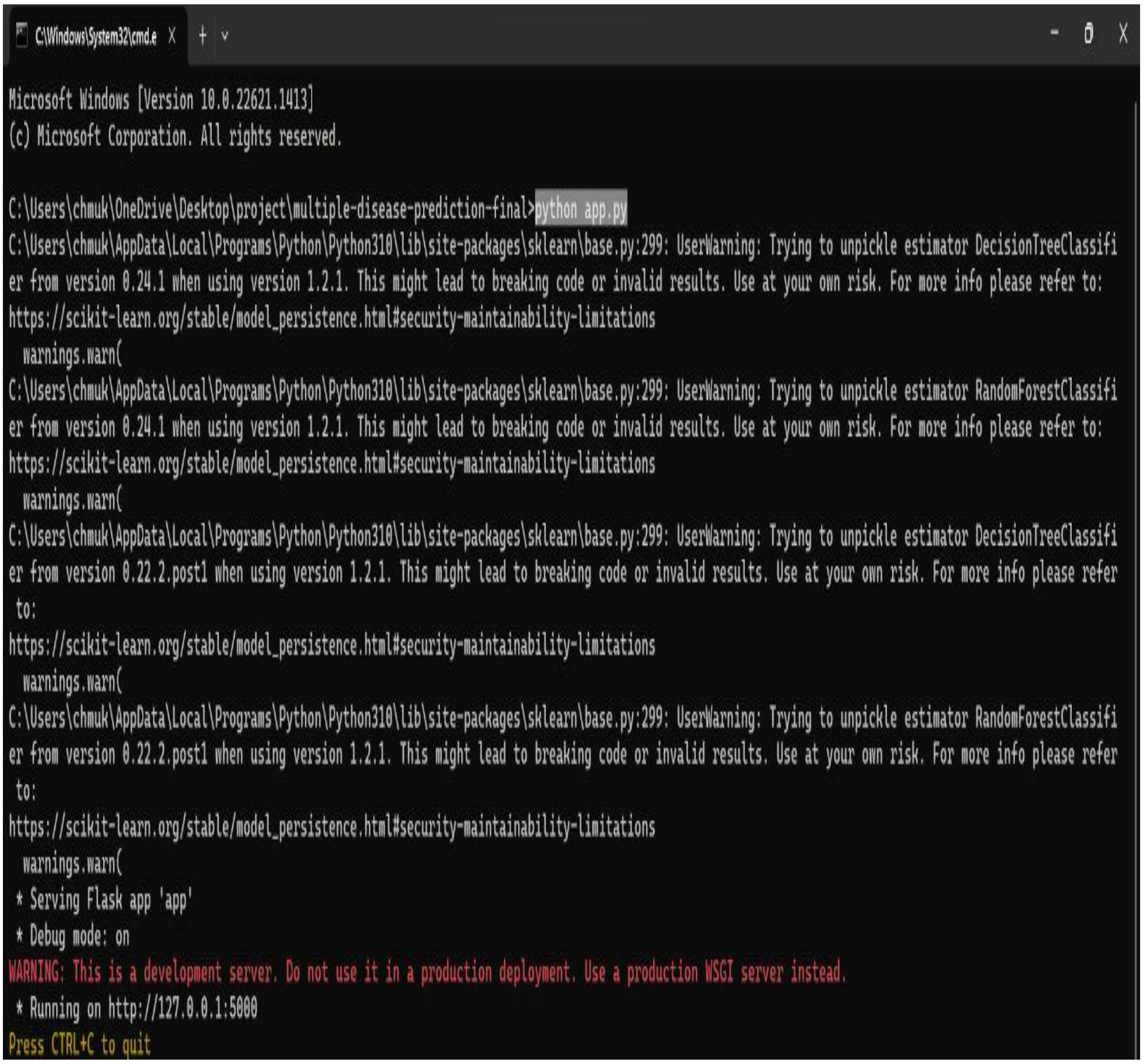
```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chmuk\OneDrive\Desktop\project\multiple-disease-prediction-final>python app.py
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator DecisionTreeClassifi
er from version 0.24.1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator RandomForestClassifi
er from version 0.24.1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator DecisionTreeClassifi
er from version 0.22.2.post1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
 to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator RandomForestClassifi
er from version 0.22.2.post1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
 to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Fig 27: Run the application

APPENDIX C – SOFTWARE USAGE PROCESS

1. Starting the Application:

For starting the execution, we use the command python app.py in command prompt.

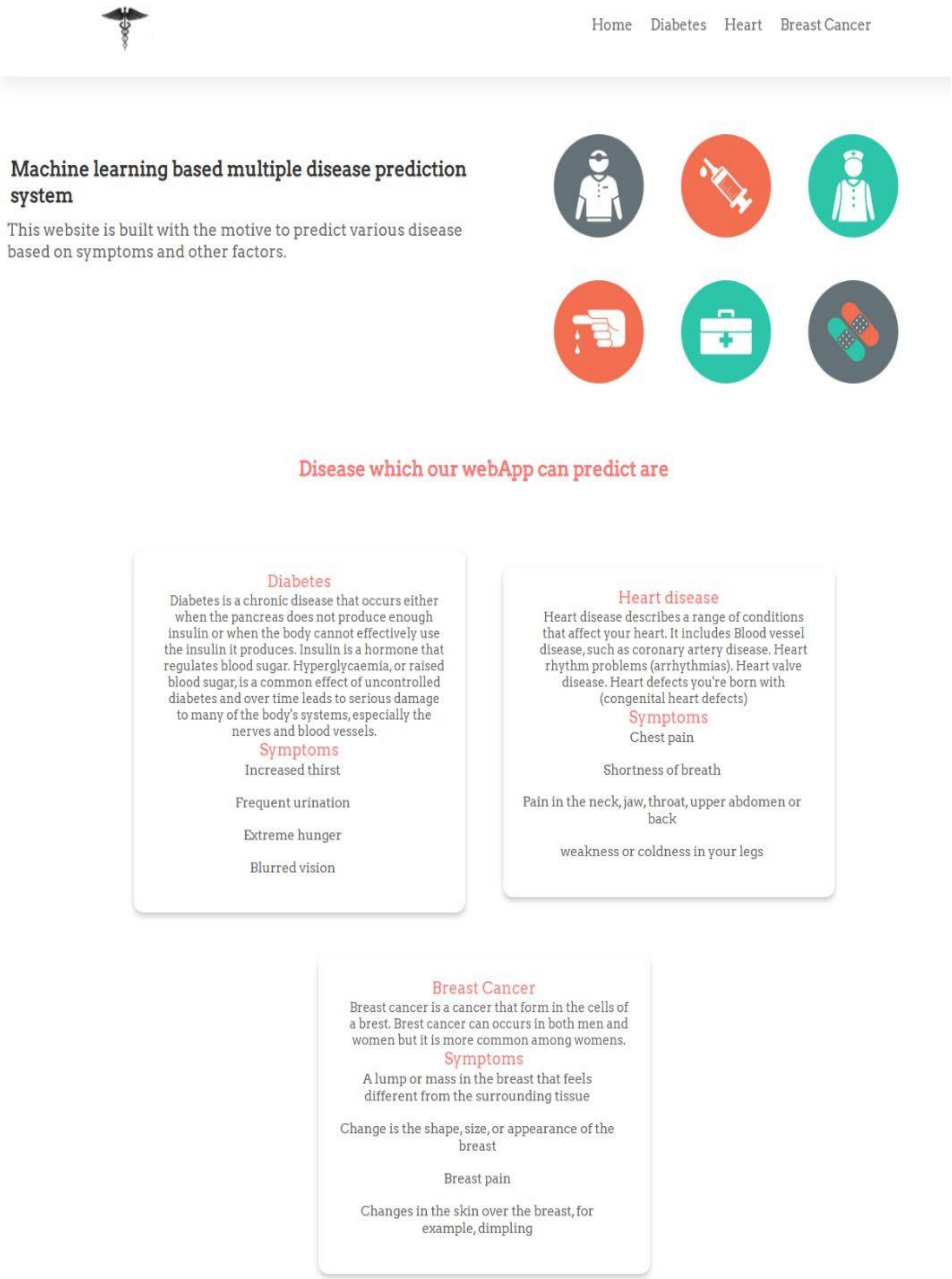


```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chmuk\OneDrive\Desktop\project\multiple-disease-prediction-final>python app.py
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator DecisionTreeClassifi
er from version 0.24.1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator RandomForestClassifi
er from version 0.24.1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator DecisionTreeClassifi
er from version 0.22.2.post1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
 to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chmuk\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator RandomForestClassifi
er from version 0.22.2.post1 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
 to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Fig 28: Deployment server estabilished

After entering the URL in the browser , it will navigate the user to the Multiplle Disease Prediction Home Page.

Here , user can see about the diseases and user can navigate to the preferred disease prediction page from the top right corner of the home page.



Machine learning based multiple disease prediction system

This website is built with the motive to predict various disease based on symptoms and other factors.

Disease which our webApp can predict are

Diabetes

Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels.

Symptoms

Increased thirst

Frequent urination

Extreme hunger

Blurred vision

Heart disease

Heart disease describes a range of conditions that affect your heart. It includes Blood vessel disease, such as coronary artery disease. Heart rhythm problems (arrhythmias). Heart valve disease. Heart defects you're born with (congenital heart defects)

Symptoms

Chest pain

Shortness of breath

Pain in the neck, jaw, throat, upper abdomen or back

weakness or coldness in your legs

Breast Cancer

Breast cancer is a cancer that form in the cells of a brest. Brest cancer can occurs in both men and women but it is more common among womens.

Symptoms

A lump or mass in the breast that feels different from the surrounding tissue

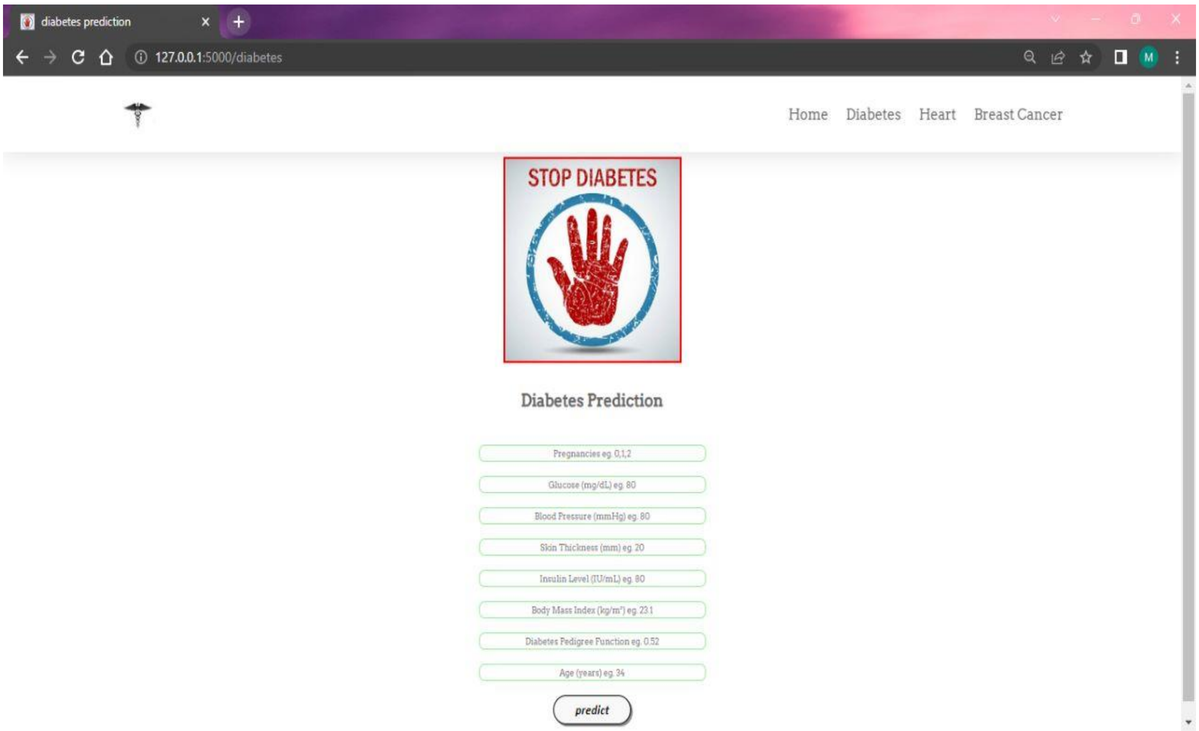
Change is the shape, size, or appearance of the breast

Breast pain

Changes in the skin over the breast, for example, dimpling

Fig 29: Home Page

The below screen is the Diabetes Disease Prediction Page.



Diabetes Prediction

Pregnancies eg. 0,1,2

Glucose (mg/dL) eg. 80

Blood Pressure (mmHg) eg. 80

Skin Thickness (mm) eg. 20

Insulin Level (IU/mL) eg. 80

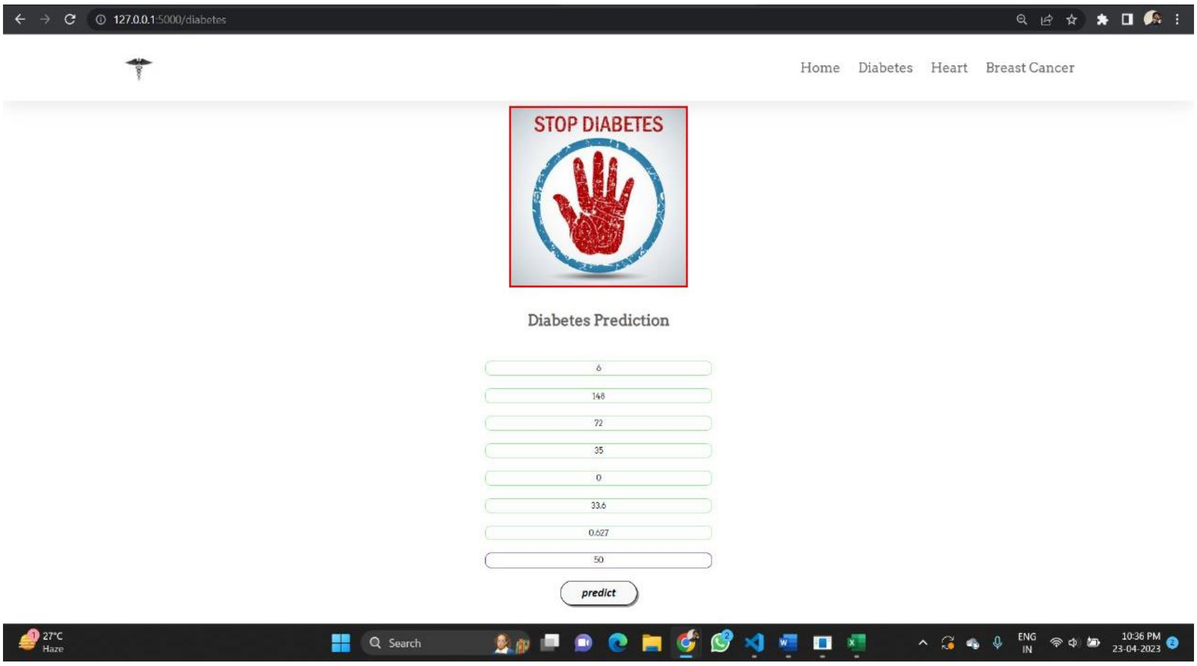
Body Mass Index (kg/m²) eg. 23.1

Diabetes Pedigree Function eg. 0.52

Age (years) eg. 34

Fig 30: Diabetes Disease Prediction Page

The below screen is the User entered values for the Diabetes Disease Prediction.



Diabetes Prediction

6

148

72

35

0

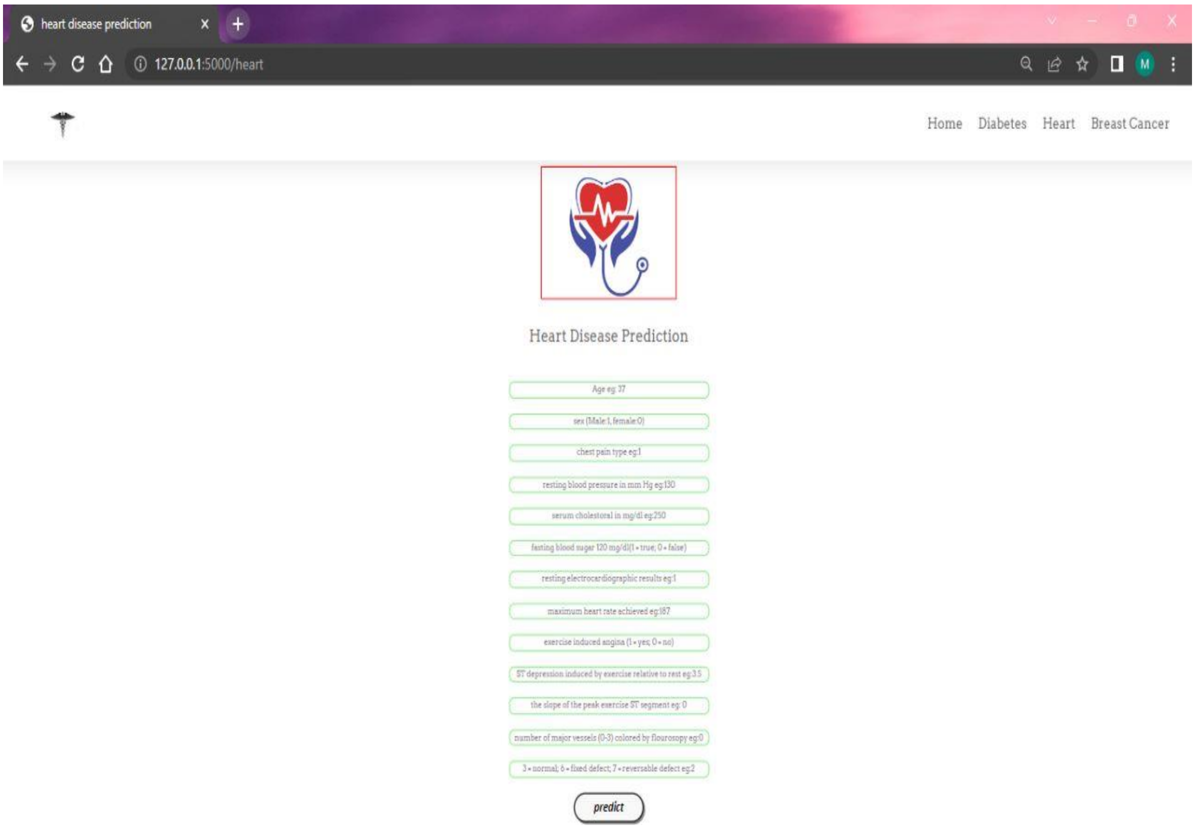
33.6

0.627

50

Fig 31: Diabetes Disease Prediction values

The below screen is the Heart Disease Prediction Page.



Heart Disease Prediction

Age eg: 37

sex (Male:1, female:0)

chest pain type eg:1

resting blood pressure in mm Hg eg:130

serum cholestoral in mg/dl eg:250

fasting blood sugar 120 mg/dl(1 = true; 0 = false)

resting electrocardiographic results eg:1

maximum heart rate achieved eg:187

exercise induced angina (1 = yes; 0 = no)

ST depression induced by exercise relative to rest eg:3.5

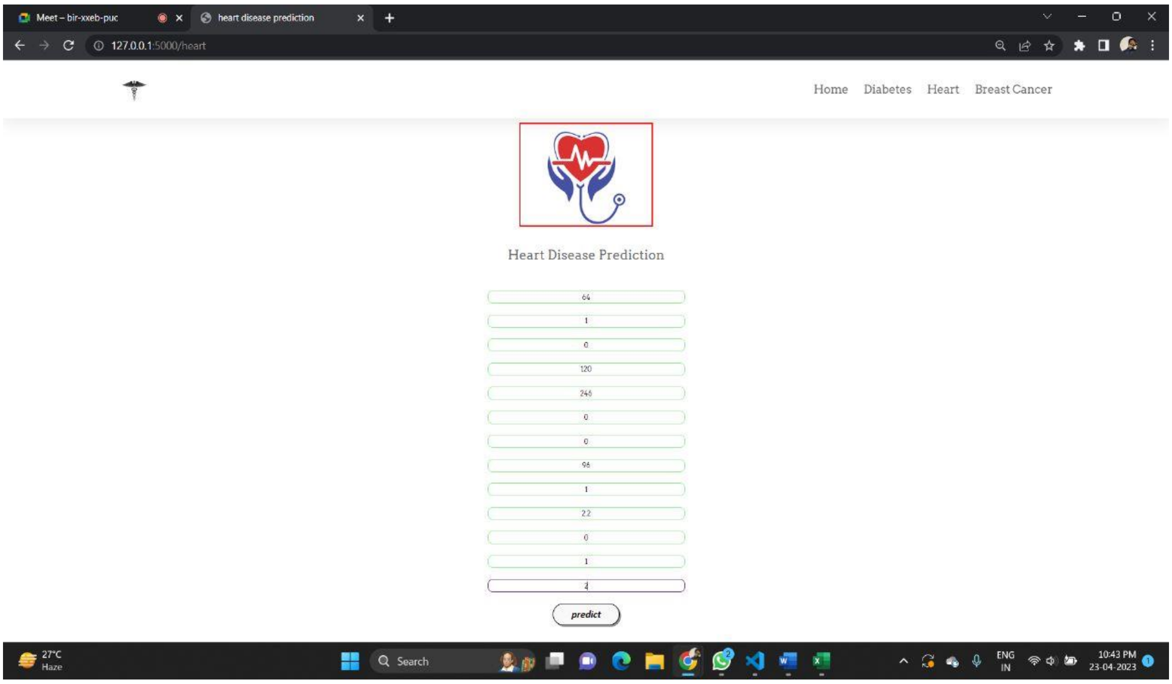
the slope of the peak exercise ST segment eg: 0

number of major vessels (0-3) colored by flourosopy eg:0

3 = normal; 6 = fixed defect; 7 = reversable defect eg:2

Fig 32: Heart Disease Prediction page

The below screen is the User entered values for the Heart Disease Prediction.



Heart Disease Prediction

64

1

0

120

246

0

0

96

1

2.2

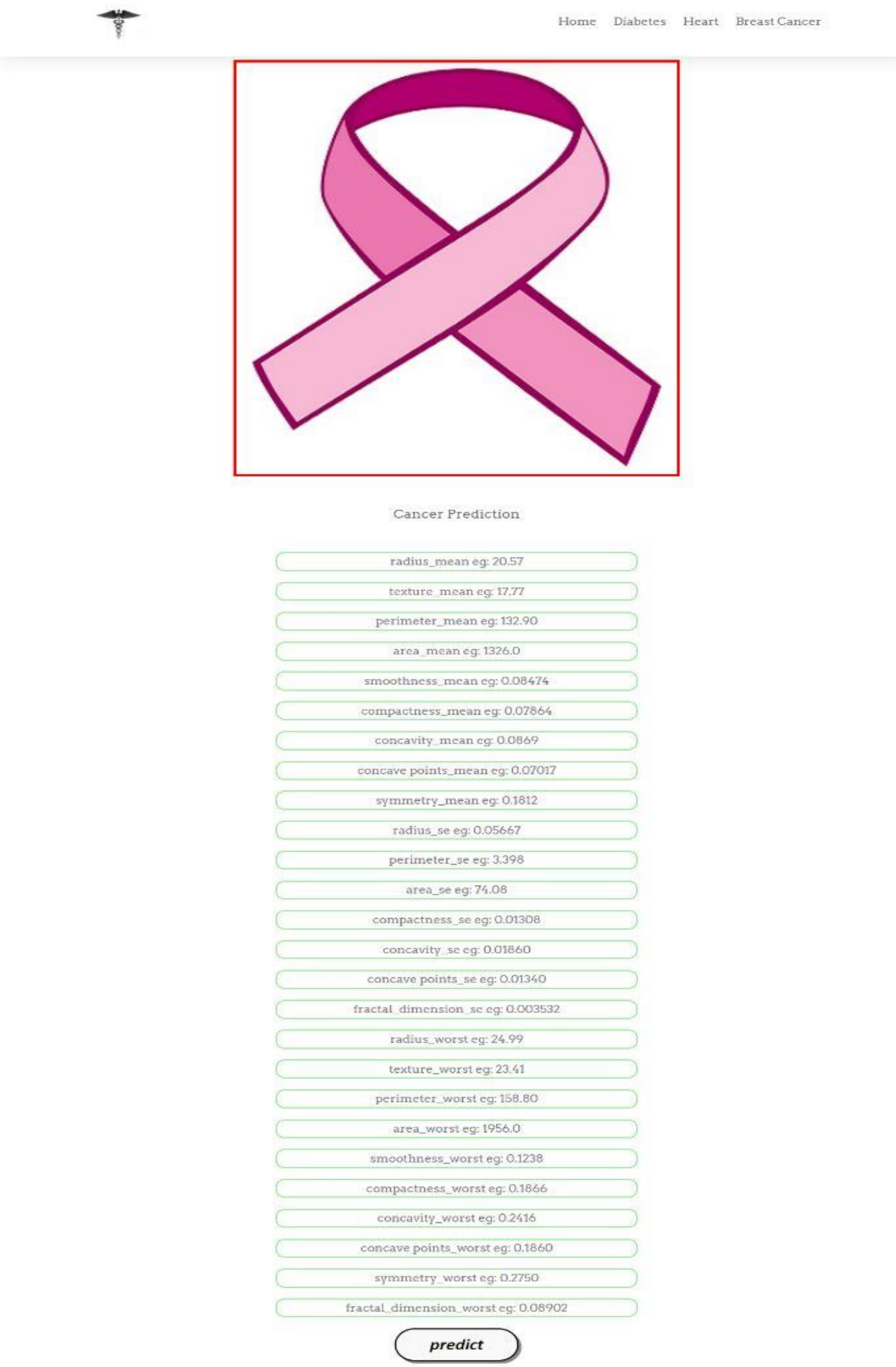
0

1

2

Fig 33: Heart Disease Prediction values

The below screen is the Breast Cancer Disease Prediction Page.



Cancer Prediction

radius_mean eg: 20.57

texture_mean eg: 17.77

perimeter_mean eg: 132.90

area_mean eg: 1326.0

smoothness_mean eg: 0.08474

compactness_mean eg: 0.07864

concavity_mean eg: 0.0869

concave points_mean eg: 0.07017

symmetry_mean eg: 0.1812

radius_se eg: 0.05667

perimeter_se eg: 3.398

area_se eg: 74.08

compactness_se eg: 0.01308

concavity_se eg: 0.01860

concave points_se eg: 0.01340

fractal_dimension_se eg: 0.003532

radius_worst eg: 24.99

texture_worst eg: 23.41

perimeter_worst eg: 158.80

area_worst eg: 1956.0

smoothness_worst eg: 0.1238

compactness_worst eg: 0.1866

concavity_worst eg: 0.2416

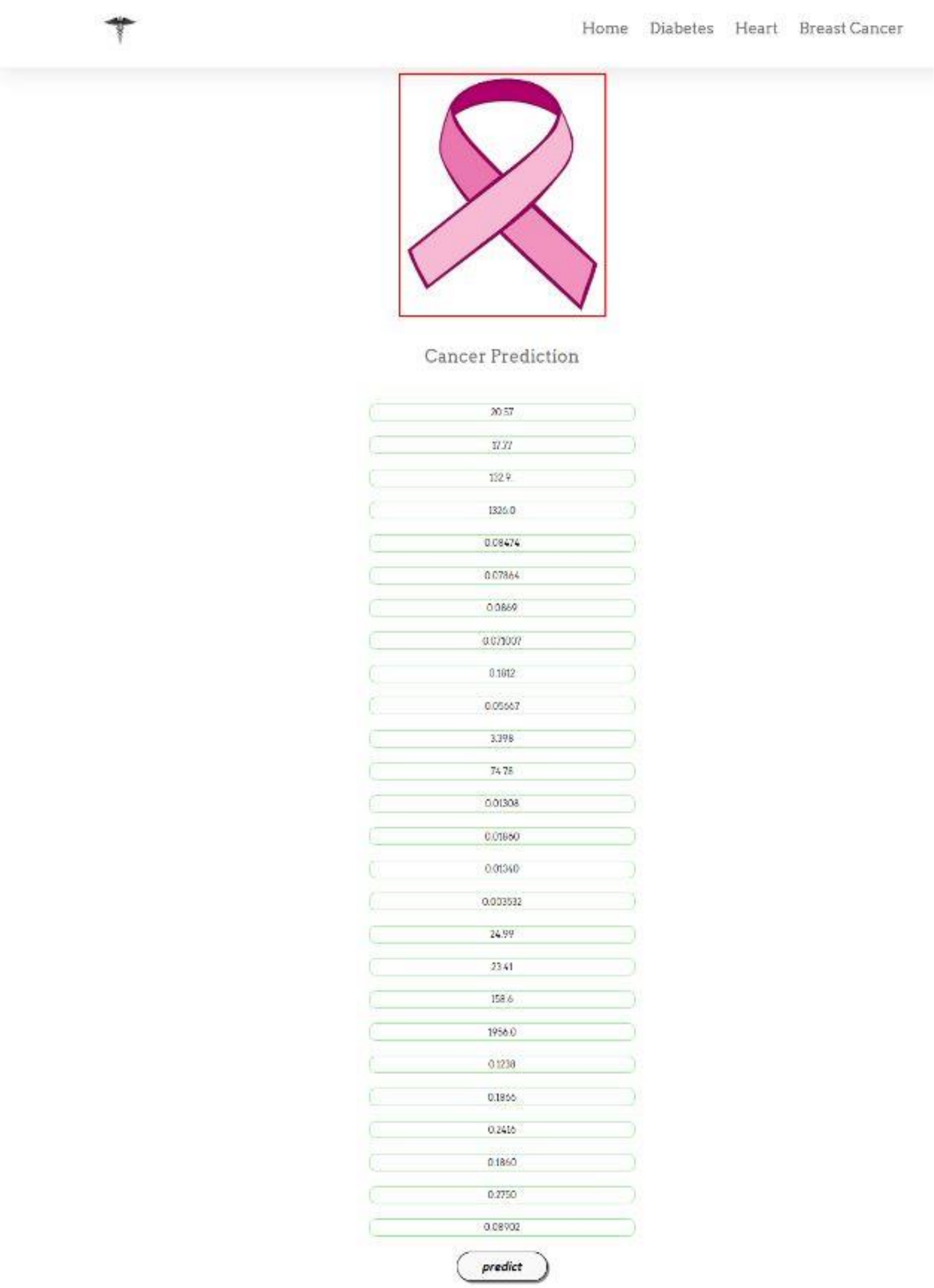
concave points_worst eg: 0.1860

symmetry_worst eg: 0.2750

fractal_dimension_worst eg: 0.08902

Fig 34: Breast Cancer Disease Prediction

The below screen is the User entered values for the Breast Cancer Disease Prediction.



Cancer Prediction

20.57

17.77

132.9

1326.0

0.08474

0.07864

0.0869

0.071007

0.1812

0.05667

3.398

74.78

0.01308

0.01860

0.01340

0.003532

24.99

23.41

158.6

1956.0

0.1238

0.1866

0.2416

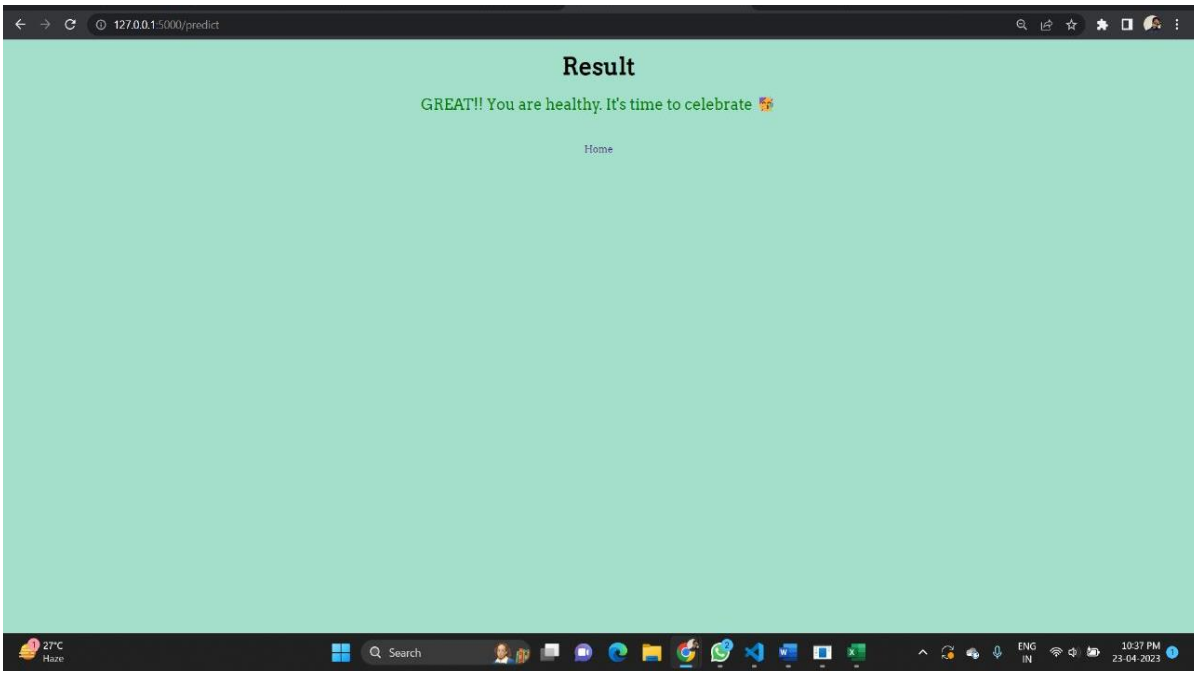
0.1860

0.2750

0.08902

Fig 35: Breast Cancer Disease Prediction values

After prediction, if user is not suffering from disease, the result is shown as below.

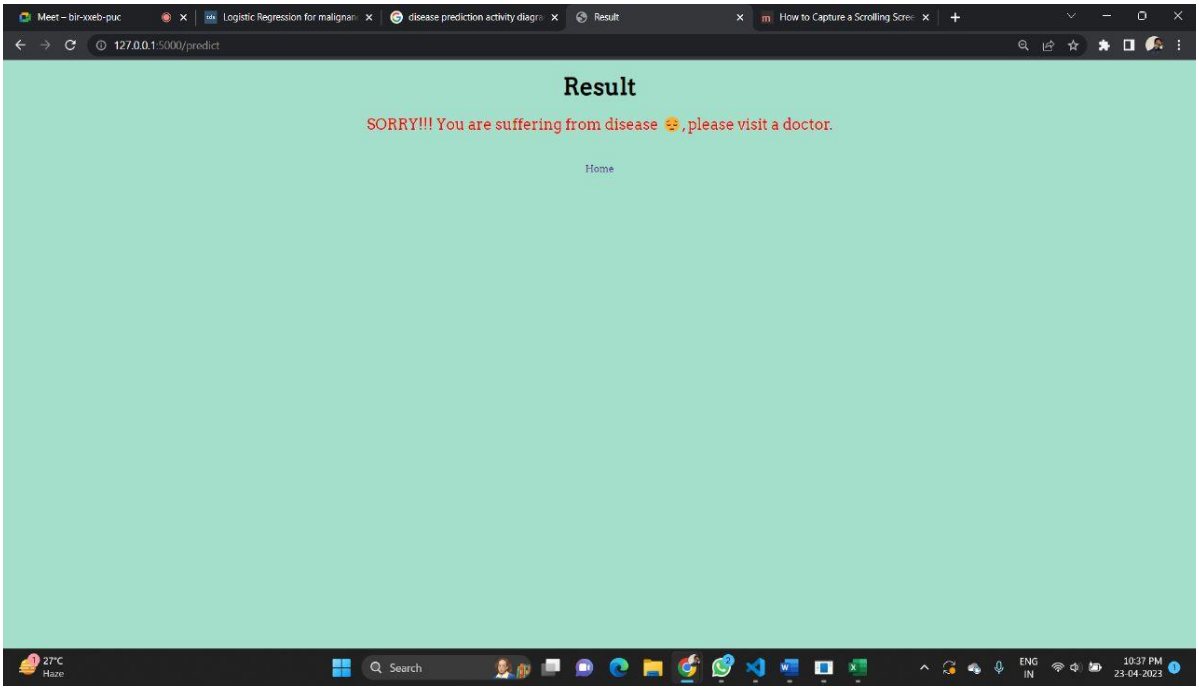


Result

GREAT!! You are healthy. It's time to celebrate 🥳

Fig 36: Result if user is not suffering from disease

After prediction, if user is suffering from disease, the result is shown as below.



Meet – bir-xxeb-puc
Logistic Regression for malignan
disease prediction activity diagra
Result
How to Capture a Scrolling Scree
127.0.0.1:5000/predict

Result

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

Home
27°C
Haze
Search
ENG
IN
10:37 PM
23-04-2023

Fig 37: Result if user is suffering from disease