CHRONIC DISEASE PREDICTION MODEL USING MACHINE LEARNING

Enrolment no- 9919102032, 9919102035, 9919102056

Name of student- Mridnal Garg, Raghav Joshi, Pranati Tiwari

Name of supervisor- DR. Kapil Dev Tyagi

May-2023

SUBMITTED IN PARTIAL FULFILLMEMT FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA (U.P.)

CERTIFICATE

This is to certify that the work titled "**Chronic Disease Prediction Model using Machine Learning**" submitted by "**Pranati Tiwari, Mridnal Garg & Raghav Joshi**" in partial fulfillment for the award of degree of B.Tech. of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Name of Supervisor: Dr. Kapil Dev Tyagi

Designation:

Date:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas and words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed

Name: Mridnal Garg

Enrolment: 9919102032

Name: Pranati Tiwari

Enrolment: 9919102052

Name: Raghav Joshi

Enrolment: 9919102035

Date:

ABSTRACT

Chronic diseases refer to long-term health conditions that persist for a prolonged period, often years or even a lifetime. This class of illnesses encompasses a broad range of conditions, including heart disease, diabetes, cancer, and arthritis, among others.

The hallmark of chronic diseases is their gradual progression and the possibility of complications over time. Furthermore, chronic diseases can significantly impact an individual's quality of life, limiting their ability to participate in daily activities and decreasing their overall well-being. Regular check-ups are crucial for maintaining good health, but many people find it time-consuming to visit a hospital.

The web application deployed for users provides easy portability, configuration, and remote access, making health management more accessible, even for individuals who cannot easily reach a doctor. The Disease Detection system is based on predictive modelling and predicts the probability of a disease based on the symptoms provided by the user as input. The system's main objective is to accurately predict the presence of a disease with few tests and attributes and also allow it’s user to talk to a doctor for online consultation.

The system considers primary attributes as the basis for tests to provide more accurate results. While more input attributes can be used, the system's goal is to predict the risk of having diseases with fewer attributes for faster efficiency. Using doctors' intuition and experience for decision-making can lead to unwanted errors and excessive medical costs, affecting the quality of service provided to patients. Therefore, using data-driven prediction tools can improve healthcare outcomes for patients and even for doctors for early detection and treatment can prevent complications and improve outcomes for those already living with these conditions.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Chronic diseases have become a significant health burden in India in recent years. These long-term health conditions persist for a prolonged period, often years or even a lifetime, and include cardiovascular disease, cancer, diabetes, and chronic respiratory diseases.

The World Health Organization (WHO) reports that chronic diseases are responsible for over 60% of deaths in India, and the figure is projected to rise in the future. Globally, three out of every five deaths are attributed to chronic diseases, including cancer, cardiovascular disease, diabetes, and chronic lung diseases. In 2001, chronic illnesses accounted for around 60% of the 56.5 million total reported deaths worldwide, and approximately 46% of the global disease burden. By 2020, this ratio had increased to 57%.

Risk factors such as tobacco use, unhealthy diets, physical inactivity, and alcohol consumption are contributing to the rise of chronic diseases in the country. Unfortunately, the burden of chronic diseases falls disproportionately on the poor, who have limited access to healthcare services and often cannot afford the high costs of treatment. Furthermore, due to the lack of awareness of chronic diseases, many people may not recognise the early signs and symptoms, leading to late diagnosis and treatment.

The Indian government has taken steps to address the growing burden of chronic diseases in the country.Despite these efforts, much more needs to be done to combat the growing burden of chronic diseases in India. The growing burden of chronic diseases in India is a major public health challenge that requires a comprehensive and coordinated approach from both the government and private sectors. It is critical to continue promoting preventive measures, raising awareness, and increasing access to healthcare services to mitigate the impact of chronic diseases and improve the health and well-being of the Indian population.

There is an increasing need for an accurate and timely prediction system to help manage this problem. One of the major challenges in managing chronic diseases is the lack of awareness and

early diagnosis, leading to delayed treatment and poor health outcomes. A chronic disease prediction system can help address this challenge by identifying individuals at risk of developing chronic diseases and enabling early intervention.

The use of predictive modelling techniques can enable the development of chronic disease prediction systems that are accurate, reliable, and can provide timely predictions. Such systems can use a range of inputs, such as demographic and clinical data, lifestyle information, and medical history, to predict the risk of developing a particular chronic disease.

The development of chronic disease prediction systems can also help to reduce the burden on the healthcare system by enabling targeted interventions and reducing the need for costly and invasive diagnostic tests. By identifying individuals at risk of developing chronic diseases, healthcare providers can offer targeted lifestyle interventions, such as dietary advice, exercise programs, and smoking cessation programs, to prevent or delay the onset of chronic diseases.

The ability to detect health changes in real-time is crucial for prioritizing care. As a result, advancements in technology that enable real-time reporting of symptoms and signs, as well as the integration of prioritization processes and care delivery, can aid healthcare providers in rapidly identifying patients who require immediate attention. This approach saves healthcare providers time by eliminating the need to search through extensive medical records for necessary information and allowing them to concentrate on providing high-quality care.

Chronic disease management is crucial for patients who require appropriate medical evaluation and treatment information. In addition, such a system can also aid individuals who need to practice self-care to improve their health status. Self-management is considered the primary care for individuals with chronic diseases and is an essential part of the treatment. Mobile applications can be used to record patient health information and serve as effective tools for enabling self-management.

CHAPTER 2

LITERATURE SURVEY

Artificial intelligence (AI) approaches have been widely used for the development of diagnosis and treatment systems in healthcare. With the emergence of the COVID-19 pandemic, the field of AI has been presented with a new challenge. Developing intelligent systems that can assist practitioners in terms of diagnosis, monitoring, prediction of patient conditions, and offering treatment measures can be extremely helpful in supporting already under pressure health systems.

AI applications have the potential to help solve the "iron triangle" problem in healthcare, which involves three interlocking factors: access, affordability, and effectiveness. AI can be used for diagnosis and treatment recommendations, patient engagement and adherence, and administration activities.

This Chapter's motive is to provide a comprehensive survey of the various AI applications that have been employed in diagnosing, monitoring, and predicting patient conditions as well as offering treatment measures. The report covers both peer-reviewed and pre-print works in order to provide a detailed picture. This chapter also includes pre-print works in order to provide a more complete picture of the current state of AI in healthcare. Additionally, it provides suggestions for future research to address the challenges more effectively.

The purpose of this chapter is to provide a comprehensive survey of the applications of ML in battling the difficulties in diagnostics and treatment. It aims to explain the most common techniques and the biggest challenges in disease detection, as well as to summarise the various results from the latest papers.

To ensure that new achievements in the field are accurately compared with the latest information, it is important to compare results with the most up-to-date methods rather than just the state-of-the-art methods. This chapter serves as a valuable resource, providing the information needed to make precise result comparisons.

Due to limited resources and the vast number of articles related to this topic, the literature survey focused on three digital libraries. However, it is evident that these libraries cover a significant amount of the related literature sources for the study.

Three different digital libraries were used to execute a research:

- IEEE Xplore
- Web of Science—WOS (previously known as Web of Knowledge)
- Research Gate

2.1 Prevalence and potential determinants of chronic disease among elderly in India: Rural-urban perspectives[1]

Chronic illnesses are a major cause of disability and untimely death among the elderly population in India. According to reports, roughly 21% of the elderly in India suffer from at least one chronic disease. Around 17% of the elderly in rural areas and 29% in urban areas are affected by chronic diseases. Among all chronic diseases, hypertension and diabetes account for approximately 68%. The likelihood of having a chronic illness is 1.15 times higher in urban areas compared to rural areas. Kerala has the highest prevalence of chronic diseases (54%), followed by Andhra Pradesh (43%), West Bengal (36%), and Goa (32%).

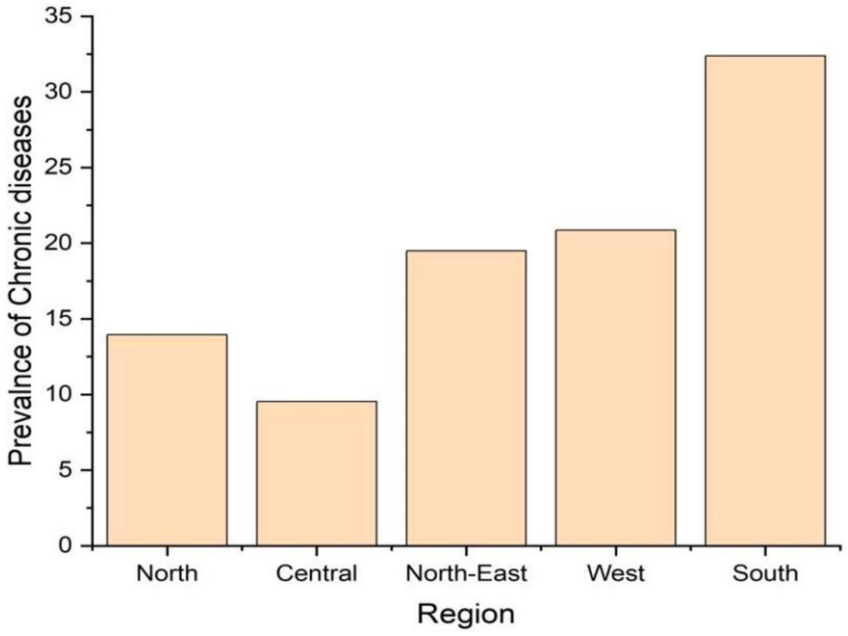

Fig 2.1 Regionwise prevalence of chronic diseases among elderly in India

2.2 Designing Disease Prediction Model Using Machine Learning Approach[2]

In today's world, people often suffer from multiple health conditions due to environmental conditions and their living habits. Early prediction of diseases is essential, but accurately predicting a disease based on symptoms alone is a challenging task for doctors. Data mining techniques can play an important role in disease prediction. With the exponential growth of medical data, data analysis has benefited from early patient care. By using disease data, data mining can reveal hidden patterns in a vast amount of medical information. In this study, we propose a general disease prediction model based on patient symptoms. This uses K-Nearest Neighbour (KNN) and Convolutional Neural Network (CNN) machine learning algorithms to predict diseases accurately.

For disease prediction, we require a dataset of disease symptoms. In this general disease prediction model, we consider the patient's living habits and checkup information to improve the accuracy of predictions. Our results show that the accuracy of general disease prediction using CNN is 84.5%, which is higher than that of KNN. Additionally, the time and memory requirements for CNN are lower than for KNN.

Today, machine learning algorithms are ubiquitous and often used unconsciously in our daily lives. CNN uses both forms of data namely, structured and unstructured data; from hospitals to classify diseases. In contrast, other machine learning algorithms work on structured data and have high computation time requirements. These algorithms are often "lazy" because they store entire data as training datasets and use complex methods for calculation.

At the outset, a dataset comprising a list of diseases and their respective symptoms was obtained from the UCI machine learning website. Subsequently, the dataset was subjected to pre-processing, which involved eliminating commas, punctuations, and white spaces to ensure cleanliness. The pre-processed dataset was then employed as a training dataset, with feature extraction and selection being carried out thereafter. Finally, classification techniques such as K-Nearest Neighbour (KNN) and Convolutional Neural Network (CNN) were utilized to classify the data and predict the disease with accuracy based on machine learning.

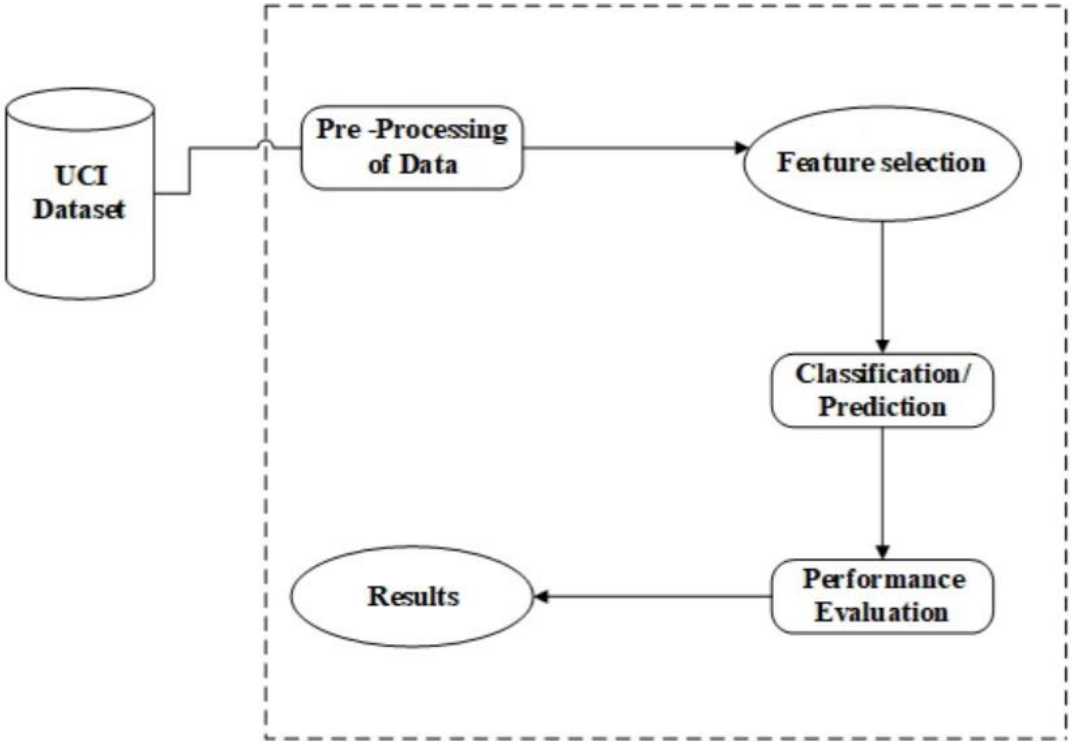

Fig.2.1 Experiment workflow with UCI dataset

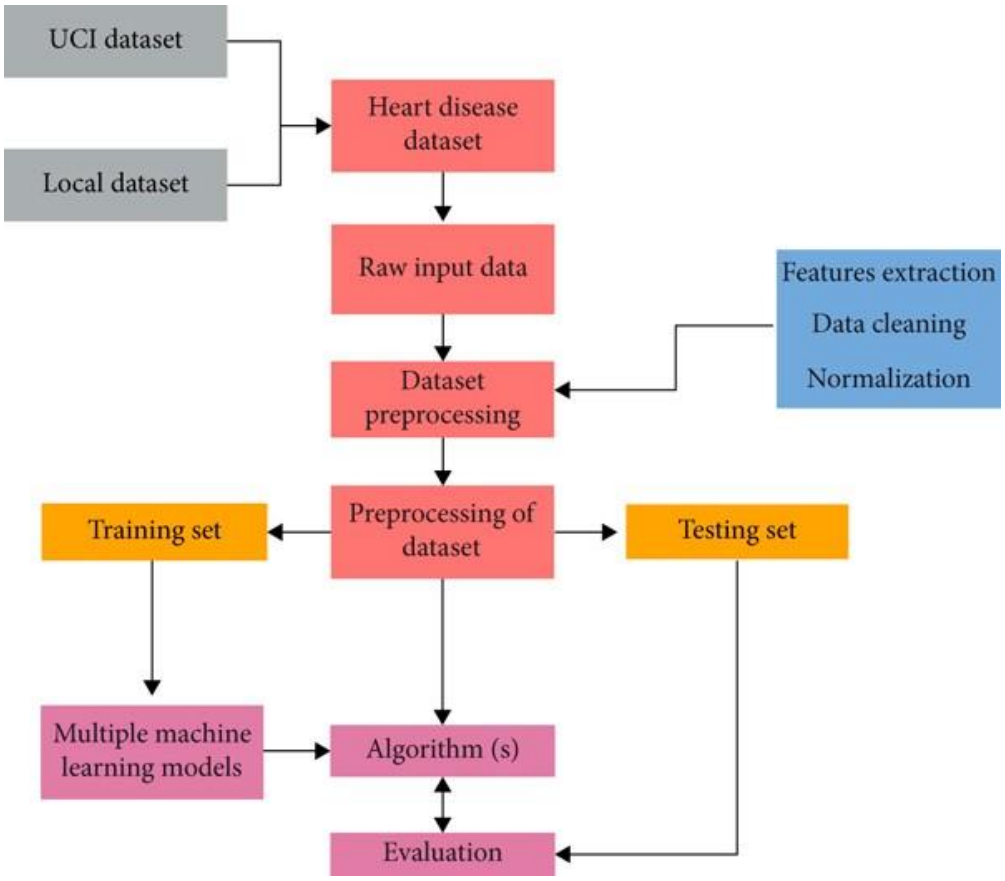

Fig. 2.2 System Architecture for ML Model

2.3 Comparing different supervised machine learning algorithms for disease prediction [3]

Supervised machine learning techniques have emerged as a popular approach in the field of data mining, particularly for predicting diseases using health data. The objective of this study is to investigate the key trends, performance, and usage of different types of supervised machine learning algorithms for disease risk prediction.

2.3.1 Supervised machine learning algorithm

One of the main features of machine learning is the use of algorithms that analyse input data and optimise their operations to make accurate predictions. These algorithms can be categorized into three types: supervised, unsupervised, and semi-supervised, based on their purpose and teaching method.

2.3.2 Logistic regression

It is a widely used supervised classification technique. It extends the concept of ordinary regression to model dichotomous variables, typically representing the occurrence or non-occurrence of an event. LR predicts the probability of a new instance belonging to a certain class, and this probability is typically assigned a threshold to differentiate between two classes. The generalized version is known as multinomial logistic regression.

2.3.3 Support vector machine

The SVM or support vector machine algorithm is capable of classifying both linear and non-linear data. It transforms each data point into an n-dimensional feature space, where n is the number of features, and then determines the hyperplane that distinguishes the data points into two classes. This is accomplished by reducing classification errors while simultaneously maximizing the marginal distance for both classes.

2.3.4 Decision tree

DT is one of the oldest and most widely used machine learning algorithms. It models decision logic and classifies data items into a tree-like structure based on tests and corresponding outcomes. The

nodes of a DT tree normally have multiple levels, with the top-most node called the root node. All internal nodes represent tests on input variables or attributes.

2.3.5 Random forest

It is a type of ensemble classifier used in machine learning that consists of multiple decision trees. It can be visualized as a forest of many trees, where each tree is a classifier used to make predictions. It combines the results of these decision trees to make more accurate predictions than a single decision tree would.

2.3.6 Naïve Bayes

NB is a classification algorithm that utilizes Bayes' theorem to determine the probability of an event based on the conditions related to that event. Unlike other classification methods, NB assumes that the features in a class are independent of each other, although there may be some level of dependence among them. This independence assumption makes the algorithm computationally efficient and suitable for large datasets with high dimensionality.

2.4 Feature selection and classification systems for chronic disease prediction: A review[4]

Chronic diseases are a significant concern in clinical practice, and early prediction and diagnosis of such diseases is crucial. This review focuses on the classification and prediction of chronic diseases using specific techniques. Accurate selection of features can improve the classification process, and size reduction can enhance the overall performance of machine learning algorithms. The classification method for disease datasets has shown promising results in improving the adaptability, technology, and intelligence of long-term disease diagnosis. Additionally, even distribution can accelerate the process and improve the effectiveness of the results.

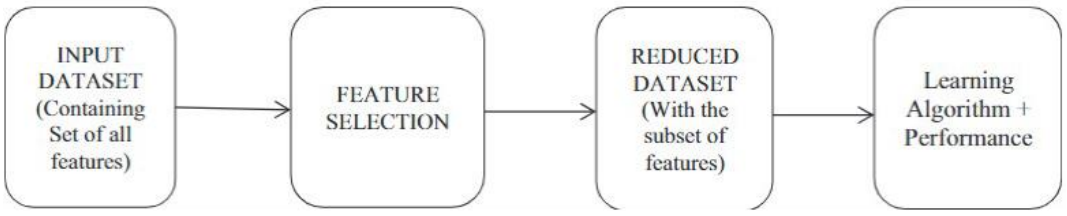

Fig 2.4 Feature selection process.

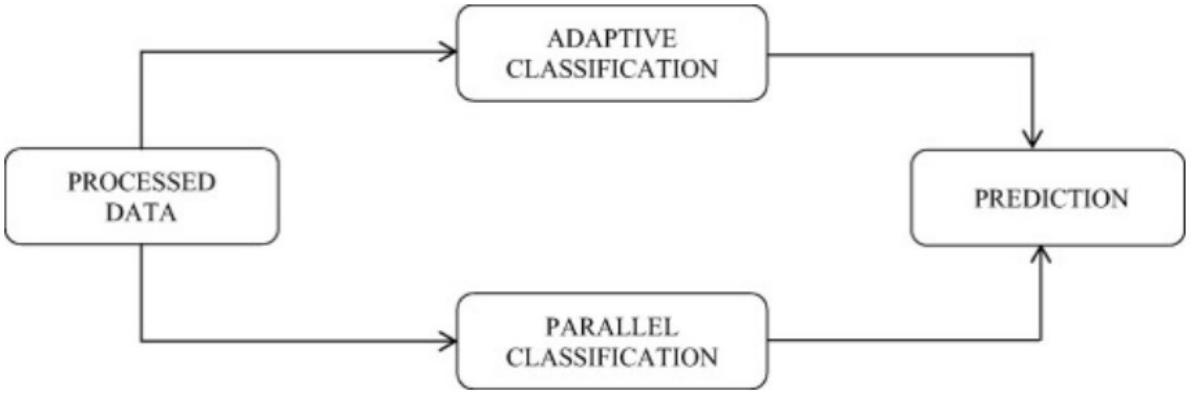

Fig 2.5 how processed data can be utilized for predictive results using both adaptive and parallel classification processes.

This study offers a thorough examination of different feature selection techniques and their respective advantages and disadvantages. Additionally, we evaluate adaptive and parallel classification systems in the context of predicting chronic diseases.

2.5 Disease Prediction by Machine Learning Over Big Data From Healthcare Communities [5]

Accurate analysis of medical data is of utmost importance for early disease detection, patient care, and community services in the healthcare and biomedical communities. Incomplete medical data, however, can reduce the accuracy of the analysis, and unique regional disease characteristics in different regions can weaken disease outbreak predictions. In this research, the focus is on predicting chronic disease outbreaks in disease-frequent communities using machine learning algorithms. Hospital data collected from central China between 2013-2015 is used, and to address incomplete data, a latent factor model is used to reconstruct missing data. Cerebral infarction, a regional chronic disease, is the focus of the study. A novel convolutional neural network (CNN)-based multimodal disease risk prediction algorithm is proposed, using structured and unstructured hospital data. The proposed algorithm achieves a prediction accuracy of 94.8% with a faster convergence speed when compared to existing prediction algorithms. This study is unique as it is the first to consider both structured and unstructured data types in medical big data analytics for disease prediction.

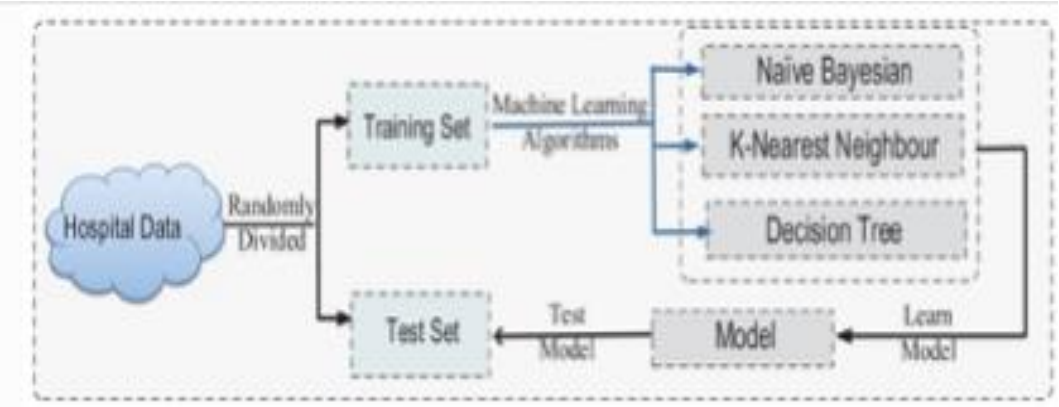

Fig 2.6 The three machine learning algorithms used in our disease prediction experiments.

The authors of this study introduce a novel algorithm for predicting the risk of chronic diseases, called the Convolutional Neural Network-based Multimodal Disease Risk Prediction (CNN-MDRP) algorithm. They use both structured and unstructured data obtained from hospitals, which is a unique approach in the field of medical big data analytics. To the best of their knowledge, no previous studies have focused on utilizing both data types. Their proposed algorithm achieves a prediction accuracy of 94.8%, outperforming several typical prediction algorithms, and has a faster convergence speed than the CNN-based Unimodal Disease Risk Prediction (CNN-UDRP) algorithm.

2.6 Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis [6]

The aim of this research is to investigate the potential of machine learning (ML) predictive models in the diagnosis of chronic diseases (CDs), which are a major contributor to global healthcare expenses. The use of predictive models has gained increasing attention for the diagnosis and prediction of CDs. This review examines the latest ML approaches utilized for primary diagnosis of CDs. The study found that there are no standardized methods for selecting the optimal approach in real-time clinical practice, as each method has its own advantages and disadvantages. Support vector machines (SVM), logistic regression (LR), and clustering were among the most commonly used methods and are highly applicable in the classification and diagnosis of CDs, with the potential to become increasingly important in medical practice.

However, it is important to choose the appropriate methods and models, as malicious data can compromise some ML models, leading to life-threatening consequences. Incorrect diagnosis may also lead to skepticism in the use of ML predictive models. Reviews on predictive models can provide evidence for selecting the best methods for CD diagnosis. In the future, AI techniques such as ML, cognitive computing, and deep learning may play a critical role in the interpretation of chronic diseases. With increasing access to electronic data, these models can enhance decision support and improve patient care quality while reducing healthcare costs.

2.7 GDPS - General Disease Prediction System [7]

The use of data mining has proven to be successful in many industries such as e-business, commerce, and trade, leading to its adoption in the healthcare sector. In the medical field, there is a need for robust analytical tools to uncover hidden relationships and trends within the data. Medical data mining techniques, such as association rule mining, classification, and clustering, are utilized to analyze various healthcare issues. Classification is a significant problem in data mining, and decision trees are commonly used to construct class models. To classify data, the ID3 Decision Tree algorithm is often used, and the accuracy is evaluated using cross-validation and partitioning techniques based on entropy to compare the results.

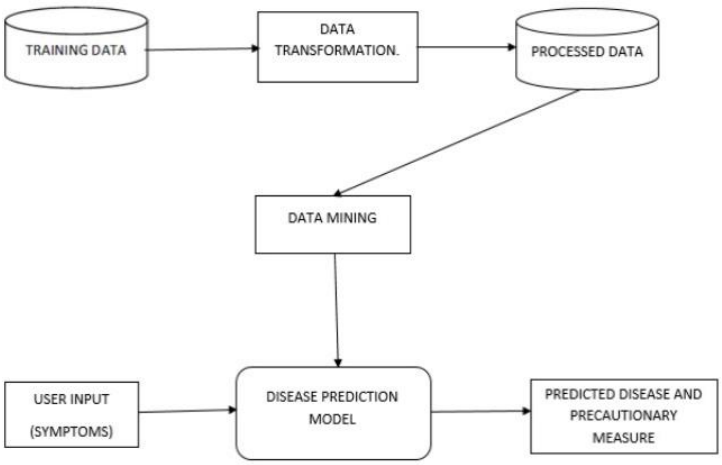

Fig 2.7 Block Diagram for General disease prediction system

Currently, the system has demonstrated an accuracy rate of 86.67% based on the analysis of 120 patient data, which is limited to general or common diseases. However, the plan is to extend the system's scope to cover diseases with higher mortality rates, such as various types of cancer, to

facilitate early prediction and treatment. This expansion has the potential to decrease the mortality rate of such illnesses, leading to long-term economic benefits.

2.8 A Survey on technique for prediction of disease in Medical Data [8]

Data mining has become an essential tool in the field of medicine for predicting diseases. With the increasing research on disease prediction systems, it has become crucial to unveil hidden patterns and relationships from medical databases. Traditional clinical diagnosis requires numerous tests, which can complicate disease prediction. Therefore, data mining techniques can assist medical experts in making decisions about the disease using computer-aided decision support systems. This paper presents a comprehensive review of various data mining techniques used for disease prediction.

Table 2.1. The compara0ve study of breast cancer

Authors	**Year**	**Accuracy**
M. Yaghoobi et al.	(2014)	More than 90%
Burke B.H et al.	(1999)	5yrs-0.909, 10yrs-0.086 &for 15yrs0.883
G.Walker et al.	(2005)	93.6%
E. Gauven et al.	(2006)	87%
R. Ceylan et al.	(2013)	98.05%
Behnam H et al.	(2005)	ROC-0.96
S.Nahavandi et al.	(2015)	97.40%
Sulong et al.	(2012)	Test at 45 years
P. Lim et al.	(2014)	98.84%
D. Chen et al.	(2001)	96.67%
Choi Chul Sang et al.	(2001)	

The fundamental intention of this paper is to examine the decision parameters, attributes, and features utilized in disease prediction. It also highlights the significance of diverse classification methods for predicting diseases in medical datasets. Various data mining techniques have been

utilized as classifiers to develop predictive models for diagnosing diseases. The datasets used in these techniques primarily pertain to heart and breast cancer. The aim of these studies is to extract necessary information from medical data to support informed diagnoses and decisions. This comprehensive survey highlights the importance of utilizing data mining techniques in developing cost-effective predictive models for disease diagnosis.

2.9 Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis[9]

Heart disease/syndrome is a global leading cause of mortality, with an estimated 23.6 million deaths predicted by 2030 if current trends continue. Despite the abundance of heart disease data collected by the healthcare industry, proper analysis of the data is often lacking to uncover hidden information that could enhance decision-making. This study examines the utilization of principal component analysis (PCA) to determine the minimum number of attributes required to enhance the accuracy of various supervised machine learning algorithms in predicting heart disease. Categorization and preprocessing techniques in data mining are also explored.

Furthermore, this research focuses on predicting diabetes, a life-threatening disease that is prevalent in urbanized and emerging countries. The paper discusses various algorithmic approaches utilized in data mining to predict diabetes.

Fig 2.8 Applications of Data Mining

2.9.1 Types of Machine Learning

The three main types of machine learning approaches are:

- Supervised Learning: In this approach, the algorithm is trained using labeled data, where the data is already categorized or classified. The algorithm learns to predict or classify new data based on the patterns it has learned from the labeled data. image recognition, spam filtering, and prediction of housing prices can be the examples for SL.

- Unsupervised Learning: In this approach, the algorithm is trained on unlabeled data, where the data is not categorized or classified. The algorithm learns to identify patterns and similarities in the data without any prior knowledge of the categories or classes. clustering, dimensionality reduction, and anomaly detection are the examples for the same.

- Reinforcement Learning: an algorithm interacts with an environment and receives feedback in the form of rewards or penalties. The algorithm learns to make decisions based on the feedback it receives, with the goal of maximizing its rewards over time. Reinforcement learning has a wide range of applications, including game playing, robotics, and autonomous driving.

In this particular research, the SVM, Naive Bayes, and Decision Tree algorithms were employed with and without the principal component analysis (PCA) technique on a dataset to predict heart disease. The PCA technique was utilized to minimize the number of attributes, and it was discovered that the SVM algorithm outperformed the Naive Bayes and Decision Tree algorithms after minimizing the dataset size.

The research outcomes can be utilized to create a graphical user interface (GUI) desktop application that can predict the probability of cardiovascular disease in a patient. As for predicting diabetes data, the primary objective was to predict the disease using the WEKA data mining tool. The implemented algorithms were evaluated based on their accuracy, correctly classified instances, time needed to build the model, mean absolute error, and ROC Area. The algorithm with the maximum ROC Area is regarded as having excellent prediction performance compared to the other algorithms.

2.10 Symptoms Based Disease Prediction Using Machine Learning Techniques[10]

Computer-aided diagnosis (CAD) is a rapidly evolving field in medicine that aims to develop computer-aided diagnostic applications that can improve the accuracy of medical diagnoses. Machine learning (ML) is a key player in CAD as it enables pattern recognition and the analysis of complex biomedical data with multiple modalities and high dimensions. ML is a powerful approach to developing automated algorithms for disease detection and decision making. This study presents a comprehensive review of various ML techniques used in the diagnosis of diseases such as heart disease and diabetes.

Statistical prediction models have limitations in producing high-quality outcomes, and ML offers potential solutions for various implementations, i.e., image recognition, natural language processing, data mining, and disease diagnosis. The paper highlights the advantages and disadvantages of ML algorithms, such as Support Vector Machines (SVM) for identifying heart disease and Naive Bayes for diagnosing diabetes, which have demonstrated excellent results in prior research. The survey also describes various tools available in the AI community that can assist in analysis and improve the decision-making process. Overall, the research emphasizes the significance of ML in CAD for enhanced disease detection and decision-making.

2.11 Smart Self-Checkup for Early Disease Prediction[11]

The paper introduces a mobile application that offers early disease prediction to the public. The app is designed following the Process Model for Healthcare (PMH) methodology. The proposed app aims to provide a self-checkup process for individuals to diagnose their own health conditions. By analyzing a person's symptoms, the app can identify potential conditions or diseases related to the symptoms, thus enabling early disease detection.

2.12 Web based disease prediction and recommender system[12]

The proposed system is a web-based platform that aims to improve disease diagnosis by storing patients' medical history and predicting possible diseases based on their current symptoms. The system utilizes machine learning classification techniques on a dataset to ensure accurate and faster diagnosis. Early disease prediction can aid patients in determining the severity of their condition and taking swift action. By providing a central platform to store and analyze medical history, the

system can assist healthcare professionals in making informed decisions regarding patient care. The use of machine learning in the system can improve the accuracy of disease prediction and provide a valuable tool in disease diagnosis.

Conclusion

This chapter presents a comprehensive review of intelligent data analysis tools used in the medical field. It includes examples of algorithms, trends, techniques, and application areas. The pros and cons of each technique are also discussed to help determine the most suitable one for real-life situations. In the past, the detection of interesting patterns in databases for medical analysis was performed through data mining. However, this process is challenging, and Artificial Intelligence techniques were developed, with Machine Learning being a method for providing intelligent data analysis tools.

CHAPTER 3

METHODOLOGY

3.1 Methodology to be used

In the project, the "Waterfall" methodology was utilized. A methodology refers to a set of practices, procedures, techniques, and rules used in a particular discipline. The Waterfall model is suitable for projects with well-defined requirements and a stable product definition, as well as a clear understanding of the technology involved. It is also more suitable for projects without ambiguous requirements, with adequate resources available and sufficient expertise. Additionally, it is considered more appropriate for relatively shorter projects.

This model comprises several sequential phases that need to be finished one after the other, and moving to the next phase is only possible when its preceding phase is entirely done.

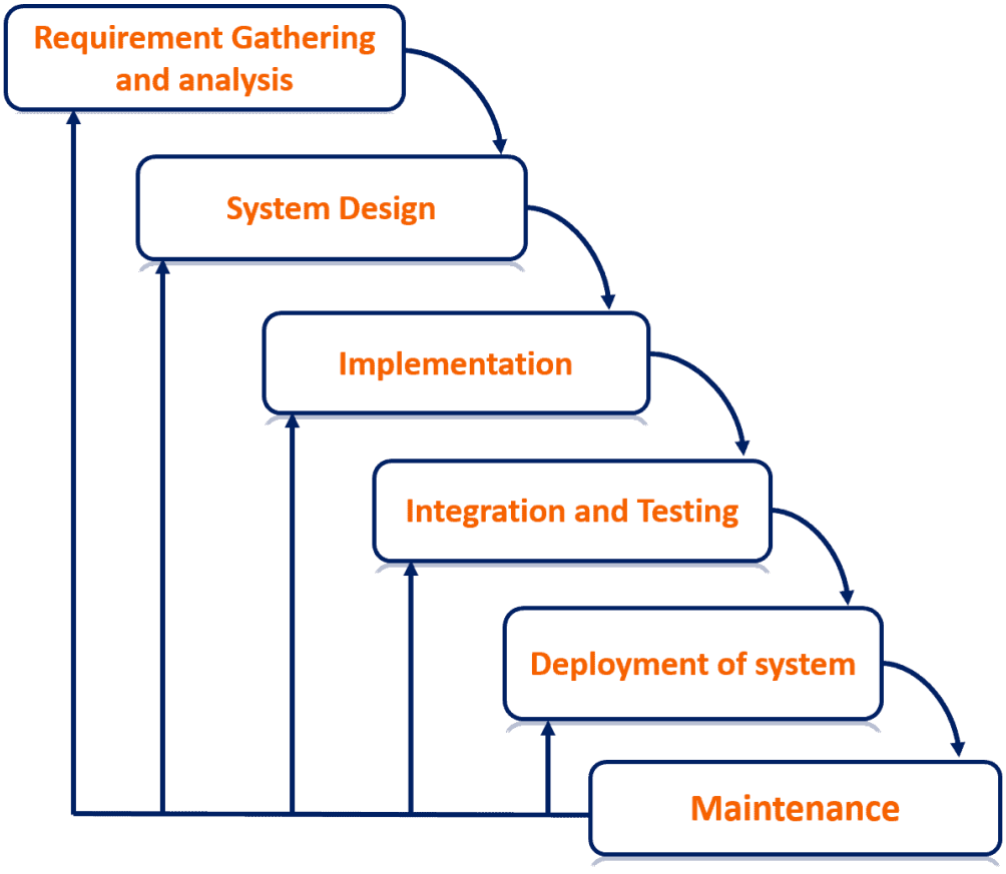

Fig 3.1 steps involved in waterfall methodology

3.2 Data Training process

The system is trained on data obtained from various sources, including datasets available on the internet. The collected data undergoes pre-processing to remove any noise or inconsistencies in the raw data. The model has been developed using different classification algorithms to accurately predict the presence of a disease based on the given symptoms. A block diagram of the system is depicted in Figure 3.1.

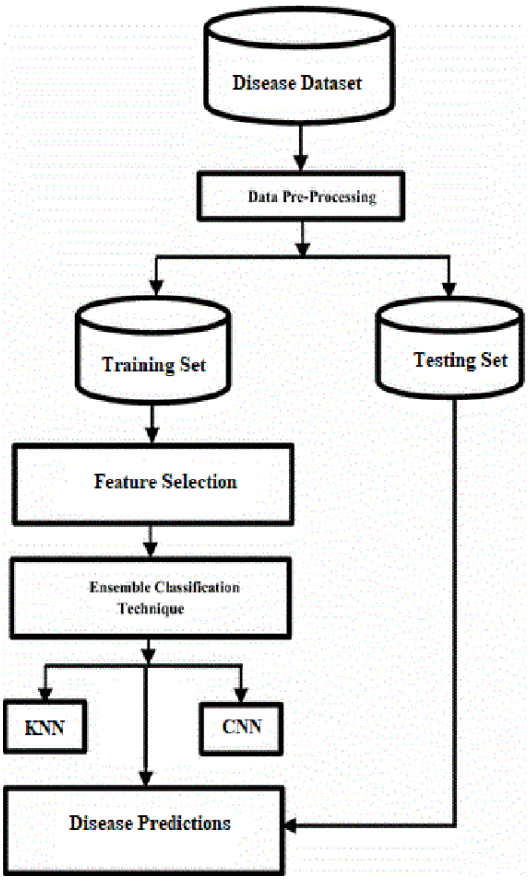

Fig. 3.2 Approach diagram for disease prediction model using machine learning

Data preprocessing is an essential step in machine learning that involves transforming raw data into a format that can be easily understood and processed by a machine learning model. It is a crucial step that needs to be performed before creating a machine learning model as it helps to improve the accuracy and effectiveness of the model. In most cases, the raw data is unstructured and requires cleaning, formatting, and transforming to make it suitable for use. techniques can be used to remove duplicates, handle missing data, normalize the data, and deal with outliers, among other things. This ensures that the data is in a usable format for machine learning algorithms.

3.2.1 Why is Data Preprocessing essential?

In practical scenarios, data can be corrupted with noise, incompleteness, or may have incompatible structures that cannot be processed by machine learning models. To overcome these issues, data preprocessing is a crucial step that involves cleaning, formatting, and transforming the data to make it usable for machine learning models. By performing data preprocessing, the accuracy and efficiency of machine learning models can be significantly improved.

It involves the below steps -

- Getting the dataset - To develop a machine learning model, the foremost requirement is a dataset, as machine learning models operate entirely on data. The data that is collected for a specific problem in a suitable format is referred to as the dataset. Different datasets are required for different purposes. Hence, each dataset varies from another dataset. Typically, we store the dataset in a CSV file format to use it in our code.The CSV file format is an abbreviation for "Comma-Separated Values" files. This file format is commonly used to store tabular data, such as spreadsheets. CSV files are useful for handling large datasets and can be easily used in various programming languages and applications.

- Importing libraries - To perform data preprocessing in Python, we need to utilize certain predefined libraries. These libraries are specifically designed to carry out various tasks related to data cleaning, transformation, and manipulation. By importing these libraries, we can easily perform data preprocessing on our datasets. These libraries offer a wide range of functions and methods to handle various data preprocessing tasks such as missing value imputation, outlier detection, feature scaling, and data visualization. With the help of these libraries, we can prepare our data for machine learning models and improve their accuracy.

  There are three specific libraries that we will use for data preprocessing, which are:

  Numpy: NumPy is a Python library used for scientific computing and data analysis. It is primarily used to perform mathematical operations and supports the creation and manipulation of large, multi-dimensional arrays and matrices. It is an essential package for scientific calculation in Python and is commonly used in machine learning, data science, and other scientific computing applications. Its array-based approach makes it an excellent choice for mathematical and

scientific applications in Python.

Matplotlib: It is a popular 2D plotting library in Python, which provides a sub-library called pyplot. This sub-library is used for creating various types of plots and charts in Python.

Pandas: It is a popular Python library used for data manipulation and analysis, allowing users to import and manage datasets. It is an open-source library that provides easy-to-use data structures and data analysis tools for handling structured data.

- Importing datasets - To import a dataset into a machine learning project using Python, the first step is to set the current directory as the working directory, where the dataset is stored. This can be done by saving the Python file in the same directory as the dataset and selecting the required directory in the file explorer option in the IDE. Then, the file can be executed by clicking on the F5 button or the run option. To import the dataset, the Pandas library's read_csv() function can be used. This function is specifically designed to read csv files and perform various operations on them.

- Finding Missing Data - After collecting data for a machine learning model, one of the critical steps is to preprocess the data to make it ready for the model. Handling missing data is a crucial aspect of data preprocessing. Missing data can negatively impact the performance of a machine learning model, so it is important to handle it properly to avoid any bias or error in the model.

  There are two main ways to handle missing data. The first approach involves deleting the row or column containing the missing data, but this may lead to a loss of valuable information and may not result in accurate output. The second approach involves calculating the mean of the column or row containing the missing data and replacing it with the mean value.

  To handle missing data, Scikit-learn library can be used, which offers various libraries for building machine learning models.

- Encoding Categorical Data - Categorical data can create issues in machine learning models since they operate on mathematical operations and numbers. It is crucial to encode these categorical variables into numbers to avoid any correlation assumption that can produce incorrect output. One

approach to encoding categorical data is dummy encoding, which involves using the OneHotEncoder class from the preprocessing library. This class allows for the creation of dummy variables for categorical variables, effectively removing any correlation assumptions.

- Splitting dataset into training and test set - This is a crucial step in data preprocessing for a machine learning model. This step is necessary to evaluate the performance of the model on new, unseen data and avoid overfitting. By using a portion of the data for training and a separate portion for testing, we can assess the model's accuracy and adjust it if necessary. In Python, we can split the dataset into training and test sets using the train_test_split() function from the scikit-learn library.

  When developing a ML prediction model, it is crucial to set a seal on that it can be generalized well to new, unseen data. One way to achieve this is by splitting our dataset into a training set and a test set during data preprocessing. The training set is used to train the model, while the test set is used to evaluate its performance on new data. If we were to test the model on a completely different dataset, it may struggle to understand the relationships between the features in the new data, and even if it performs well on the training set, it may not perform well on new data.

  Therefore, it is important to ensure that our model performs well on both the training set and the test set. The training set is a subset of the dataset used to train the model, where we already know the output. The test set is a subset of the dataset used to evaluate the model, where the model predicts the output based on the input features.

- Feature scaling - It is a crucial step in data preprocessing for machine learning models. It involves scaling or standardizing the values of independent variables in the dataset to a specific range. The purpose is to bring all variables to the same scale so that no variable dominates over the other. This is important as many ML algorithms rely on the Euclidean distance between data points, and if variables are not scaled properly, it may cause inaccuracies in the model's predictions.

The user system is divided into two parts, the patient part and the doctor part. The admin is responsible for training the system to create the disease prediction model. Once the model is

created, the user can log in and enter their symptoms, which are then analysed by the system to generate a prediction of the likely disease. The system also recommends necessary precautionary measures to manage the predicted disease.

Overall, this system has the potential to improve public health by identifying general diseases at an early stage, which in turn enables timely medical intervention and management of the disease.

3.3 Algorithms and Techniques Used

Classification is a type of machine learning technique used to categorize data into a specific number of classes, whether the data is structured or unstructured. The primary objective of a classification problem is to identify the appropriate category or class to which a new data point belongs. Few of the terminologies encountered in machine learning classifications are stated below -

- Classifier: an algorithm that takes input data and maps it into a specific category or class.
- Classification model: a machine learning model that draws conclusions from input values given during training and predicts the class labels/categories for new data.
- Feature: It is an individual measurable property or characteristic of a phenomenon being observed.
- Binary Classification: a classification task with two possible outcomes or classes, such as gender classification (male/female).
- Multi-class classification: It involves more than two classes, where each sample is assigned to one and only one target label. For example, an animal can be classified as a cat or a dog but not both at the same time.
- Multi-label classification: a classification task where each sample is assigned to a set of target labels or classes, which means that there can be more than one label assigned to a single sample. An example of multi-label classification could be a news article that discusses sports, a particular person, and a location simultaneously.

The following are the steps involved in building a classification model:

- Initializing the classifier: Before training the classifier, we need to initialize it with appropriate hyperparameters.

- Training the classifier: All classifiers in scikit-learn use the fit(X, y) method to train the model on the given training data X and training labels y.
- Predicting the target: Once the model is trained, we can use it to predict the labels for new, unlabeled observations X. The predict(X) method returns the predicted label y for each observation in X.
- Evaluating the classifier model: To evaluate the performance of the classifier, we can use various evaluation metrics such as accuracy, precision, recall, F1 score, and confusion matrix. These metrics help us to understand how well the classifier is performing and if any improvements can be made.

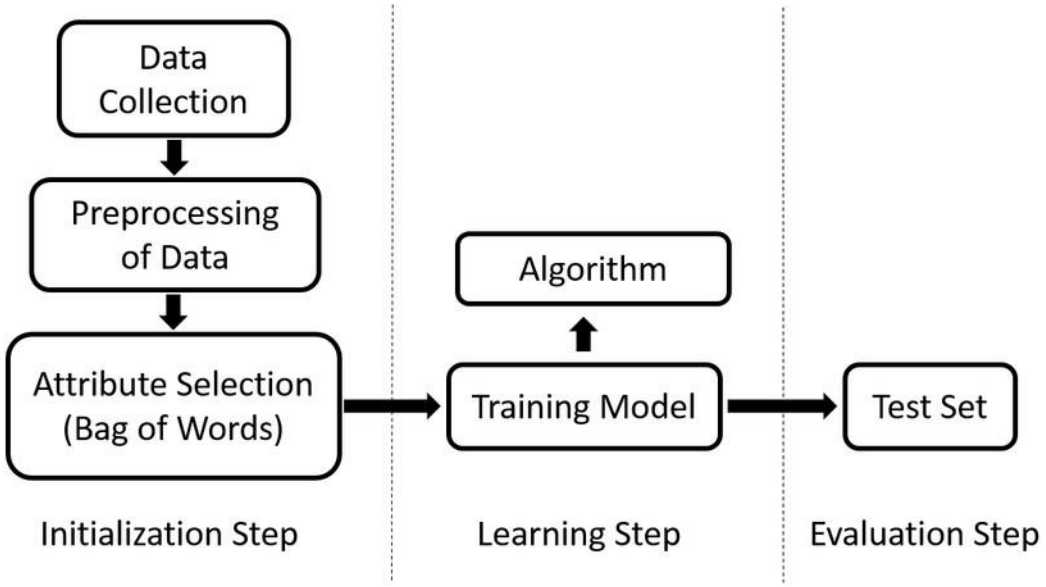

Fig 3.3 steps for training a data classifier

3.3.1 Logistic Regression

Logistic Regression is a classification algorithm used to estimate discrete values, rather than continuous values like regression. It is used to predict the probability of an event occurring based on a given set of independent variables. The output is typically limited to a value between 0 and 1 since it represents a probability. Logistic regression finds applications in various fields such as machine learning, social sciences, and medicine. In medical research, logistic regression has been used in the development of scales such as the Trauma and Injury Severity Score (TRISS) to predict mortality in injured patients.

Medical practitioners also utilise logistic regression to assess the severity of a patient's condition based on various observed characteristics, and to predict the presumption of developing specific diseases, such as diabetes or coronary heart disease, using patient data such as age, sex, and blood test results.

The sigmoid function, also known as the logistic function, is a mathematical function used in binary classification to predict the probability of a certain outcome. It is called sigmoid because its curve has an S-shape. The function takes any input value and transforms it to a number between 0 and 1, which represents the probability of the outcome. In machine learning, the sigmoid function is often used as an activation function in neural networks to introduce nonlinearity to the model and to produce a probability output for binary classification problems. The formula of Logistic Function is:

$$F(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

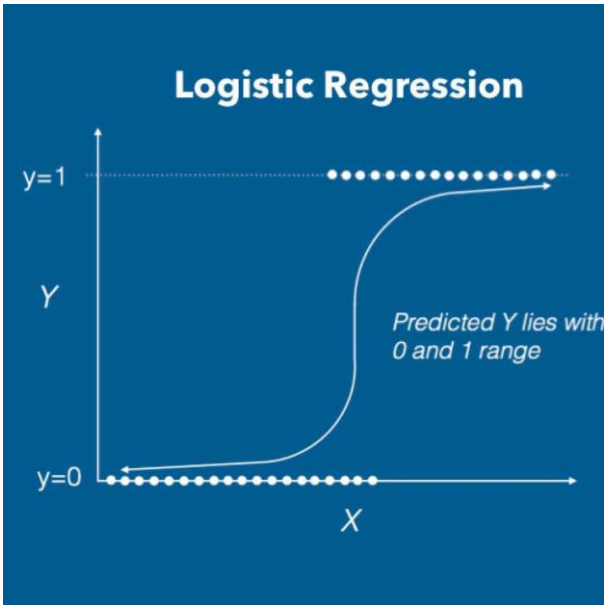

Fig 3.4 Logistic regression for predicting disease

3.3.2 Random Forest

Random forest is a popular and versatile machine learning algorithm that is easy to use and can achieve great results even without extensive hyperparameter tuning. It is widely used for both classification and regression tasks due to its simplicity and effectiveness. Random forest is a

supervised learning algorithm that constructs an ensemble of decision trees, typically using the "bagging" method. This method involves training multiple learning models and combining their predictions to improve overall accuracy.

Random forest is a versatile machine learning algorithm that can be used for both classification and regression tasks. This is a significant advantage, as it allows the algorithm to be applied to a wide range of real-world problems. When it comes to classification, random forest is a popular choice due to its effectiveness in producing accurate results. The algorithm works by creating an ensemble of decision trees, using the "bagging" method to improve overall performance. Figure 3.5 illustrates an example of how a random forest would look with classifier trees.

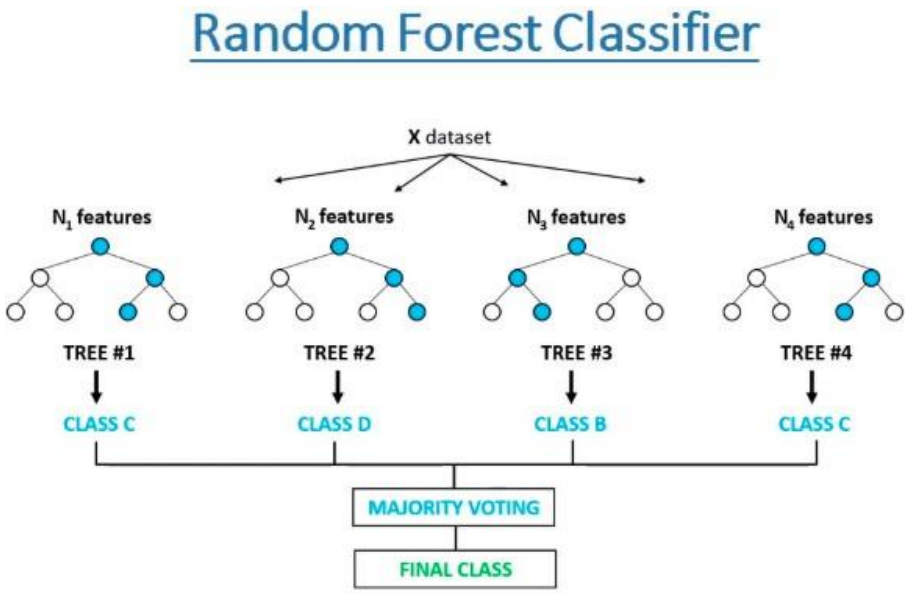

Fig 3.5 How does random forest technique works

Random forest is a type of ensemble machine learning algorithm that introduces additional randomness into the model while growing the trees. Unlike traditional decision trees, where the algorithm searches for the most important feature when splitting a node, random forest searches for the best feature among a random subset of features. By doing so, the model becomes more diverse, resulting in better overall performance. This feature is particularly useful when dealing with high-dimensional datasets where certain features may dominate the decision-making process.

Random forests are a versatile classification method that provides estimates for variable importance, similar to neural nets. One of their major strengths is their ability to effectively handle missing data by substituting missing values with the most common variable in a given node. In fact, random forests are known for their superior accuracy in comparison to other classification methods.

Moreover, random forests are well-suited for big data sets with a high number of variables, even running into thousands. They can automatically balance data sets when a certain class is under represented, and they can process variables quickly, which makes them a suitable choice for complex tasks.

The random forest algorithm finds its applications in various fields such as banking, e-commerce, medicine, and the stock market. In finance, it is used to identify customers who are more likely to repay their debts on time or those who would use a bank's services frequently. It can also be used to detect fraudsters trying to scam the bank. For trading, the algorithm is employed to forecast the future behaviour of stocks. In healthcare, it is used to determine the appropriate combination of components in medicine and to analyse a patient's medical history to diagnose diseases.

3.3.3 Decision Tree

A decision tree is a model represented by a hierarchical tree structure, where internal nodes correspond to features or attributes, branches denote decision rules based on attribute values, and leaf nodes represent decisions or outcomes. The initial partitioning of the data is represented by the root node. Decision trees are widely used in classification and regression tasks, and their main benefit is interpretability and explainability. As they can be easily visualized as a flowchart-like structure, they emulate human-level thinking.

A Decision Tree is a transparent machine learning algorithm, also known as a white box model, because it reveals its internal decision-making logic. In contrast, black box models such as Neural Networks do not provide this level of transparency. The time complexity of Decision Trees is proportional to the number of records and attributes in the dataset, which can become computationally expensive for large datasets.

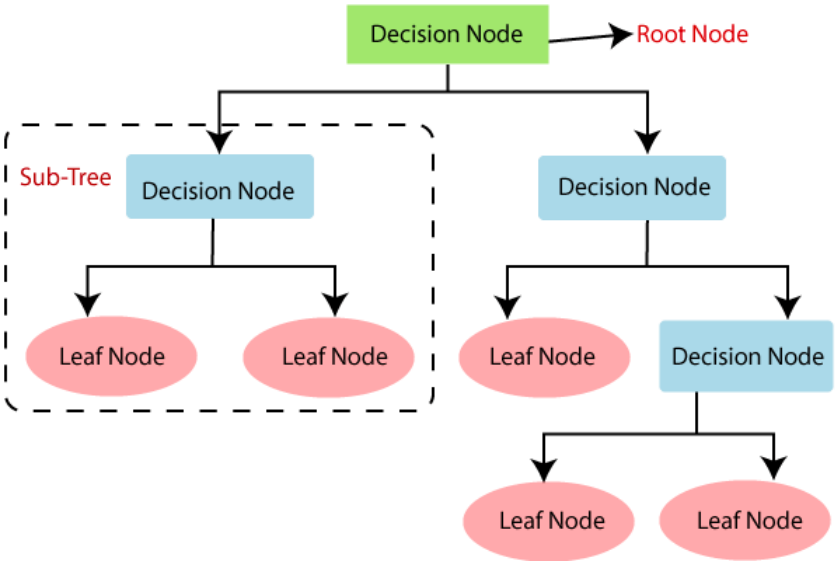

Fig 3.6 Decision Tree Classifier

Decision tree is a non-parametric, distribution-free algorithm which doesn't rely on probability distribution assumptions. It can efficiently handle high-dimensional data with great accuracy, and involves less data wrangling compared to other machine learning algorithms. Moreover, the training time for decision tree models is generally faster compared to neural network algorithms.

There are different algorithms for building decision trees, such as ID3, C4.5, and CART. These algorithms use different criteria for selecting the best feature to split the data, such as information gain or Gini impurity.

The decision tree algorithm works as follows:

i. Start at the root node S, which consists of the entire dataset.
ii. Determine the most suitable attribute in the dataset using an attribute selection measure (ASM).
iii. Divide S into subsets based on the values of the best attribute.
iv. Create decision nodes in the decision tree based on the best attribute.
v. Recursively repeat steps 2-4 on the subsets until reaching a stage where no further classification is possible, resulting in a leaf node.

3.3.4 Naïve Bayes

It is a classification algorithm that can be utilised to predict diseases on the basis of the probability of a particular disease given the presence of certain symptoms. This algorithm works by utilising Bayes' theorem to calculate the likelihood of having a specific disease given evidence like the presence of certain symptoms. Before applying Naive Bayes for disease prediction, the algorithm should be trained using a dataset of known cases and their corresponding symptoms and diagnoses. The algorithm uses this training data to calculate the conditional probability of each symptom given the presence or absence of the disease.

When a new patient's symptoms are presented, the algorithm calculates the probability of the patient having the disease based on these symptom probabilities. The algorithm makes an assumption that each symptom is independent of all others, which is why it is referred to as "naive".

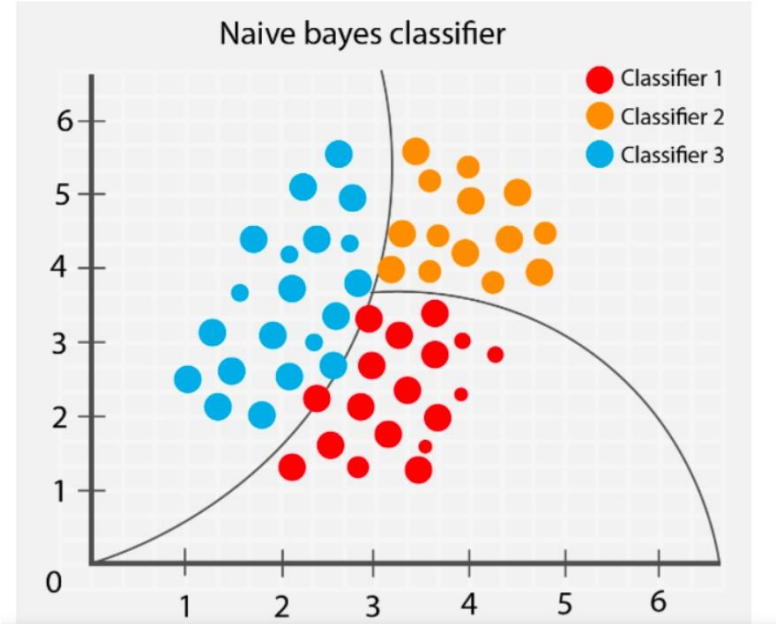

Fig 3.7 Naive bayes classifier

Naive Bayes is a popular classification algorithm that is commonly used for text classification and spam filtering. One of the advantages of Naive Bayes is its simplicity and ease of implementation, making it a popular choice for beginners in machine learning. Another advantage is its ability to handle both numerical and categorical data. Naive Bayes performs well even with missing data or noisy input, and it can be used for both binary and multi-class classification. Additionally, it is a

powerful algorithm that can be beneficial for disease prediction, particularly when working with a large number of symptoms, and it can provide accurate predictions even with incomplete or missing data.

Steps to implement:

i. Data pre-processing step involves cleaning and transforming the raw data to make it suitable for analysis. This may include tasks such as handling missing values, encoding categorical variables, and scaling numerical variables.
ii. Fitting Naive Bayes to the training set involves training the model on the prepared data. In this step, the algorithm learns from the data and creates a probability model that can be used to make predictions.
iii. Predicting the test result involves using the trained model to make predictions on new data that was not used in the training phase.
iv. Test accuracy of the result is measured by creating a confusion matrix, which compares the predicted values with the actual values. From the confusion matrix, different evaluation metrics such as accuracy, precision, recall, and F1 score can be calculated.
v. Visualising the test set result can help in understanding the model's performance and identifying any areas where it might be making errors. This can be done using various visualisation techniques such as scatter plots, confusion matrices, and ROC curves.

3.3.5 Bagging

Bagging, or bootstrap aggregation, is a technique used in machine learning to improve the accuracy and stability of ensemble models. Ensemble models combine the predictions of multiple models to improve their performance, and bagging is one of the most popular techniques for creating ensembles.

Bagging works by creating multiple copies of the training data set, each copy being slightly different due to random sampling with replacement. A different model is trained on each copy of the data, using the same learning algorithm, but with different random subsets of the data.

Once all the models are trained, they are combined by taking the average (for regression problems) or majority vote (for classification problems) of their predictions. This averaging or voting helps to reduce the variance of the model and improve its accuracy.

Bagging is particularly useful when the underlying learning algorithm is unstable or sensitive to the specific training data set. By creating multiple models with different training sets, bagging helps to reduce the impact of outliers or noisy data, and improves the stability and accuracy of the ensemble.

3.3.6 Boosting

The "boosting" ensemble modeling approach is designed to create a powerful classifier by combining multiple weak classifiers. The approach includes constructing a sequence of weak models, where each new model aims to rectify the mistakes of the previous one. The initial model is trained on the complete training dataset, and subsequent models are trained on the data that was incorrectly classified by the preceding model. The process continues until a specified number of models have been trained, or until the training data has been correctly classified.

3.3.7 Gradient Boosting

A well-liked machine learning method for classification and regression issues is gradient boosting. This particular ensemble approach combines a number of weak models to produce a stronger model. The principle of gradient boosting is to train several models in succession, with each model fixing the flaws of the one before it. In other words, each new model concentrates on the data points that the preceding model failed to accurately anticipate.

The "weak learners" models utilised in gradient boosting are often decision trees. Gradient boosting decision trees, in contrast to conventional decision tree models, employ a modified learning technique to enhance the model's performance. "Gradient descent" is the name of this algorithm.

Gradient boosting involves training a model by minimising a loss function that gauges the discrepancy between expected and actual values. The direction and size of the steepest fall in the loss function are determined using the gradient descent technique, and this information shows how the model parameters should be modified to lessen the loss.

Gradient boosting's main tenet is that by integrating a number of imperfect models, each of which corrects the flaws in the preceding model, the final model becomes more more powerful and corpulent accurate. The weighted average of each weak model's predictions is used to achieve this.

3.4 Technologies Used

3.4.1 Hardware Requirements

• GPU

• RAM : Minimum of 4GB

• Processor : Minimum Dual Core Processor

3.4.2 Software Requirements

- VISUAL STUDIO Code
- Google Collab

3.4.3 Frontend Technologies

- HTML
- Bootstrap
- Javascript
- Jquery

3.4.4 Database

PostgreSQL - PostgreSQL is a powerful open-source relational database management system that provides various advanced features for efficient data management. It is advantageous for its high performance, scalability, and reliability. Additionally, PostgreSQL provides various functionalities for data processing, including support for various programming languages, indexing, and replication.

One of its most useful features is the ability to extend its functionality using user-defined functions, which can be developed using different programming languages. These functions can be used to implement custom business logic or to integrate with external systems. With its advanced

capabilities and open-source nature, PostgreSQL is a popular choice for many organizations to manage their data effectively.

3.5 Language and Framework used

3.5.1 Python

Python can be used for developing predictive models for chronic disease prediction. It is a popular programming language used in the field of machine learning and predictive analytics, offering a vast selection of libraries and frameworks such as TensorFlow, Keras, PyTorch, scikit-learn, and numerous others.

In chronic disease prediction, Python can be used to preprocess the data, perform feature selection and engineering, and train various machine learning models. The data can be collected from electronic health records, surveys, or wearable devices. Once the data is collected, Python can be used to preprocess the data by handling missing values, scaling, and normalising the data.

Python can also be used to perform feature selection and engineering to identify the most relevant features for the prediction model. Various machine learning models can be trained and evaluated using Python. The models can be trained using different algorithms, such as logistic regression, decision trees, random forests, and support vector machines.

Python also provides various tools and libraries for model evaluation, such as confusion matrix, accuracy, precision, recall, and F1-score. The models can be tuned and optimised using hyper parameter tuning and cross-validation techniques.

Overall, Python is a powerful tool for chronic disease prediction, providing a wide range of libraries and frameworks for machine learning and predictive analytics.

3.5.2 Django

Django is a web framework that is developed using Python, and it follows the model-template-view (MTV) pattern. Its primary objective is to make web development easier for developers by

providing a strong framework with a variety of tools and libraries to construct secure, scalable, and maintainable web applications.

One of the advantages of using Django for web application development is that it comes with a built-in administration panel, which makes it easy to manage and interact with data in a database. The ORM system of Django is considered as one of the most powerful features of the framework. It enables developers to work with databases using Python objects and methods, resulting in a more efficient and maintainable codebase.

Django, a popular deep learning framework, offers numerous built-in features such as authentication and authorization, form handling, URL routing, and template rendering. Its active and sizeable community provides extensive support and various third-party packages to facilitate development.

The framework also has a large community and extensive documentation, making it easy for to troubleshoot issues and implement new features.

CHAPTER 4
IMPLEMENTATION

The project aims to address the issue of general diseases that often go unnoticed and untreated due to the busy lifestyle of individuals. It is based on machine learning and has been developed using Python. The project includes an interface built using the Flask library. It predicts the likelihood of a patient suffering from a particular disease based on their input symptoms and suggests necessary precautionary measures for treatment as per the doctor's consultation.

The system works by predicting the likelihood of particular disease based on the symptoms presented by the user patient. The system is initially fed with data from various sources, including publicly available datasets, which are then pre-processed to remove noise and ensure data quality. The system uses different classification algorithms to build the disease prediction model.

The prediction model has been implemented and deployed as a python-based web application using Django for user offering them several advantages; such as accessibility, cross-platform compatibility, easy updates; making is more user-friendly and versatile.

After discussing the basic concepts of prediction models, this section outlines the experiments that were performed to test hypotheses regarding their performance. It should be noted that any improvements made can only be measured if an appropriate evaluation function is established. When dealing with disease prediction, accuracy is a crucial metric that must be taken into consideration.

Despite significant progress in the field of machine learning, there is still a significant reliance on computational power when making changes to a model. As a result, when designing this thesis, it was quickly determined that conducting experiments that involve adjusting the weights in the model would be impractical if one aims to significantly improve the performance of the model using an average computer.

4.1 Dataset Used

▾ Correlation matrix & Matrix Visualisation

[27] df.corr()

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893	-0.175905	-0.160650	0.202850	-0.086906	-0.059893	...
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	-0.029324	0.171134	0.161784	-0.094786	-0.065324	...
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	-0.022444	...
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608981	0.446238	-0.087351	-0.047254	-0.047254	-0.032566	...
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000	0.295332	-0.060200	-0.032566	-0.032566	-0.022444	...
...	...	...	...	...	...	...	...	...	...	...	...
inflammatory_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	0.359845	-0.033480	-0.033480	-0.023073	...
blister	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	...
red_sore_around_nose	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	...
yellow_crust_ooze	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	...
prognosis	-0.351936	0.079612	-0.253230	-0.113211	-0.240568	-0.053683	0.192448	-0.312820	-0.245787	-0.227907	...

133 rows × 133 columns

[26] df.describe()

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	...
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.162195	0.139024	0.045122	0.045122	0.021951	...
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.368667	0.346007	0.207593	0.207593	0.146539	...
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...

8 rows × 133 columns

Fig 4.1 Dataset description and correlation for prediction

4.2 Code snippets for prediction techniques

```
def DecisionTree():

    from sklearn import tree

    clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
    clf3 = clf3.fit(X,y)

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf3.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        # print (k,)
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = clf3.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break


    if (h=='yes'):
        t1.delete("1.0", END)
        t1.insert(END, disease[a])
    else:
        t1.delete("1.0", END)
        t1.insert(END, "Not Found")
```

Fig 4.2 Code for decision tree algorithm

```
def NaiveBayes():
    from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb=gnb.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=gnb.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
    for k in range(0,len(l1)):
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = gnb.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t3.delete("1.0", END)
        t3.insert(END, disease[a])
    else:
        t3.delete("1.0", END)
        t3.insert(END, "Not Found")
```

Fig 4.3 Code for Naive Bayes algorithm

```
def randomforest():
    from sklearn.ensemble import RandomForestClassifier
    clf4 = RandomForestClassifier()
    clf4 = clf4.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf4.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        for z in psymptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = clf4.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(disease)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t2.delete("1.0", END)
        t2.insert(END, disease[a])
    else:
        t2.delete("1.0", END)
        t2.insert(END, "Not Found")
```

Fig 4.4 Code for Random Forest

4.3 Output for chronic disease prediction model

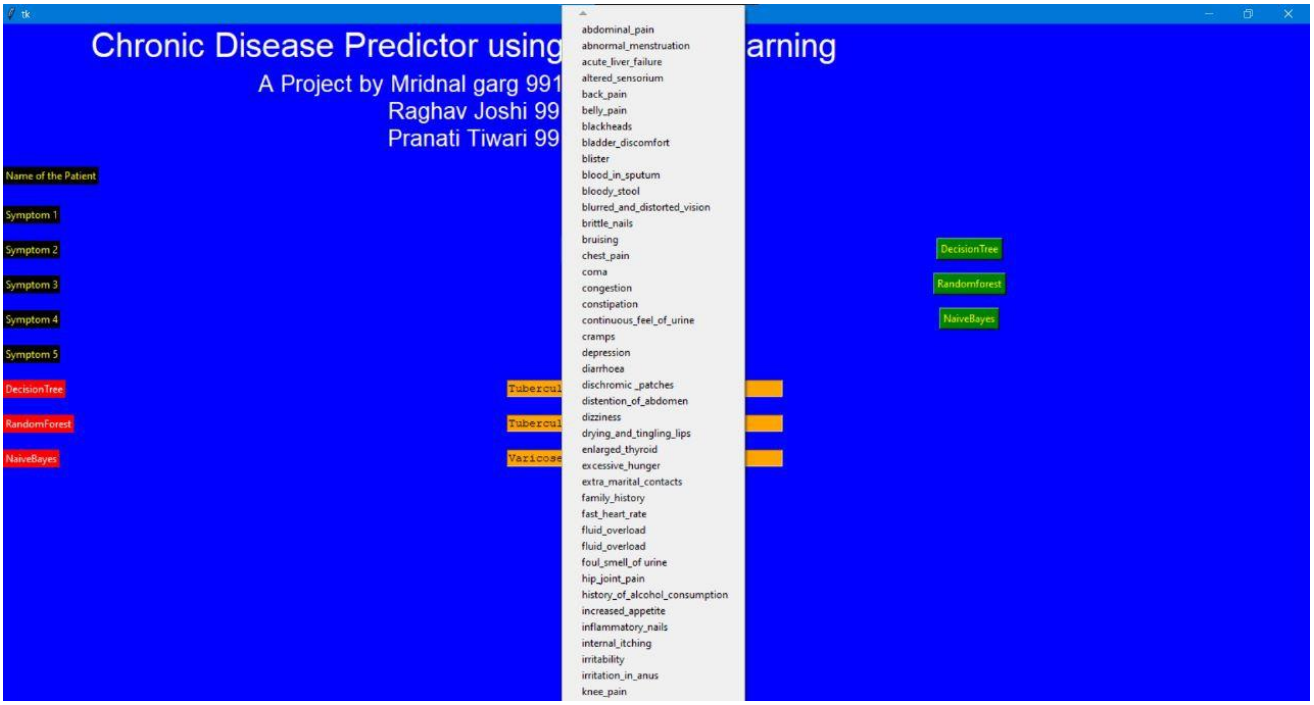

Fig 4.5 List of all symptoms

Fig 4.5 shows the list of all possible symptoms available for users to choose from based on the dataset trained. The user can select multiple symptoms from the list and further the prediction model will analyse and process it to generate the possible disease user is suffering from using 3 different algorithm, i.e., Decision Tree(DT), Random Forest(RF) & Naive Bayes(NB) and can also consult the doctor (one of the functionality offered in web app).

CHAPTER 5

SIMULATION RESULTS

This chapter shows the final outcome after integrating the backend model for chronic disease prediction model with the user interface by combining the possible methodologies with proposed solution and design which can be easily accessible by the use once the web app is made available widely.

5.1 Home Page

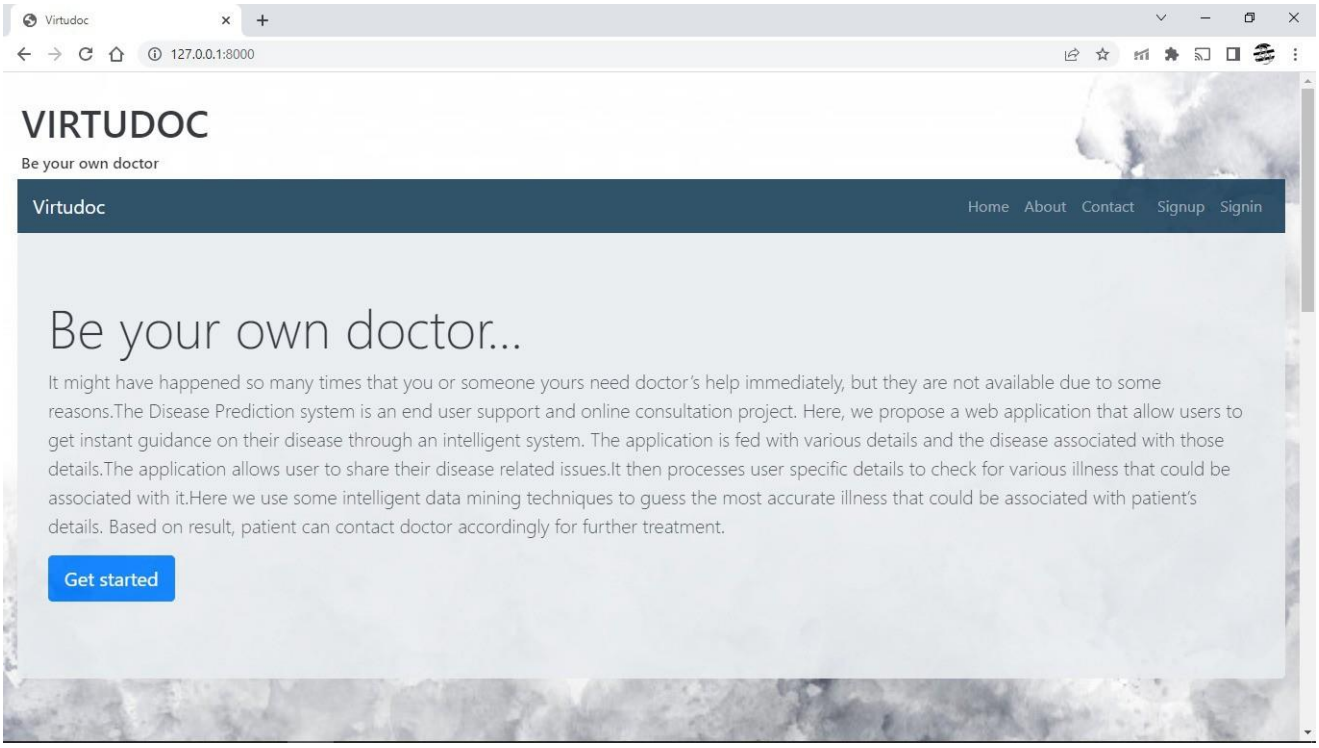

Fig 5.1 Home page of the web app

Fig 5.1 shows the landing/home page of the web app name "VIRTUDOC". This page give a brief description of what is the main motive of this web application. The application is fed with various dataset containing symptoms of various chronic diseases and also also the user to contact the doctor to consult and get the required prescription. The user may click on "Get started" to move ahead.

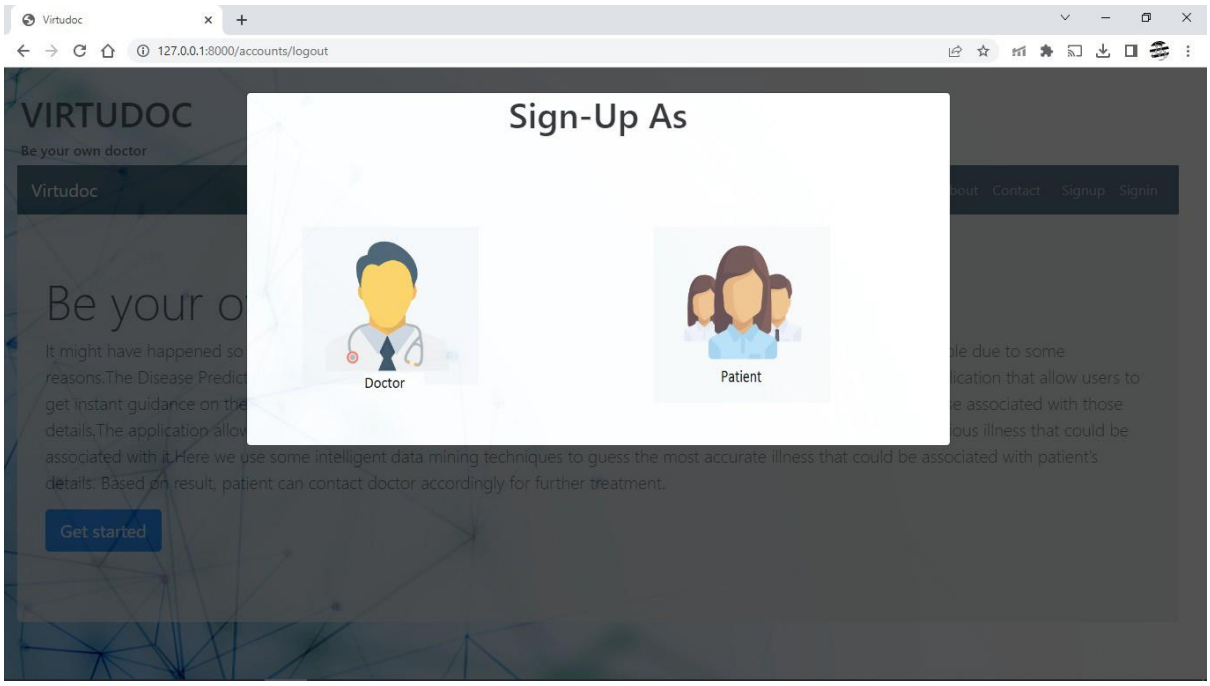

Fig 5.2 Sign up options available

Fig 5.2 show the interface displayed to user when he/she clicks on “Get Started”. The web app consists of user system which is further classified into two parts, the patient part and the doctor part.

5.2 Signup Form

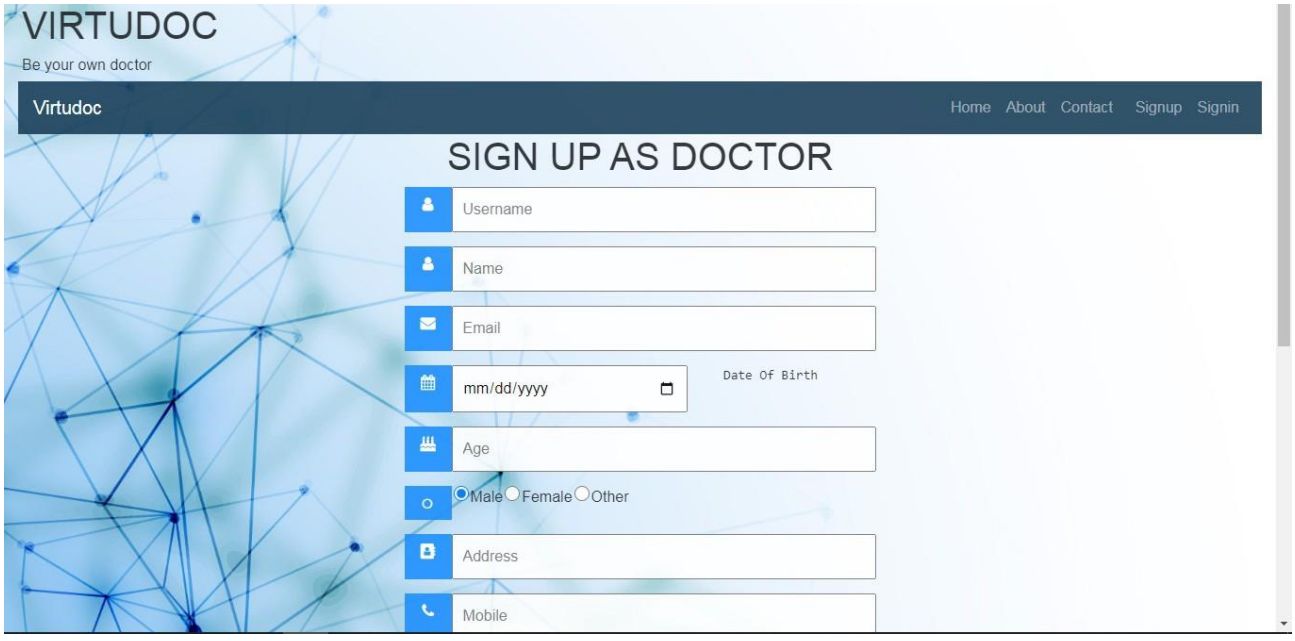

Fig 5.3 Signup form as Doctor

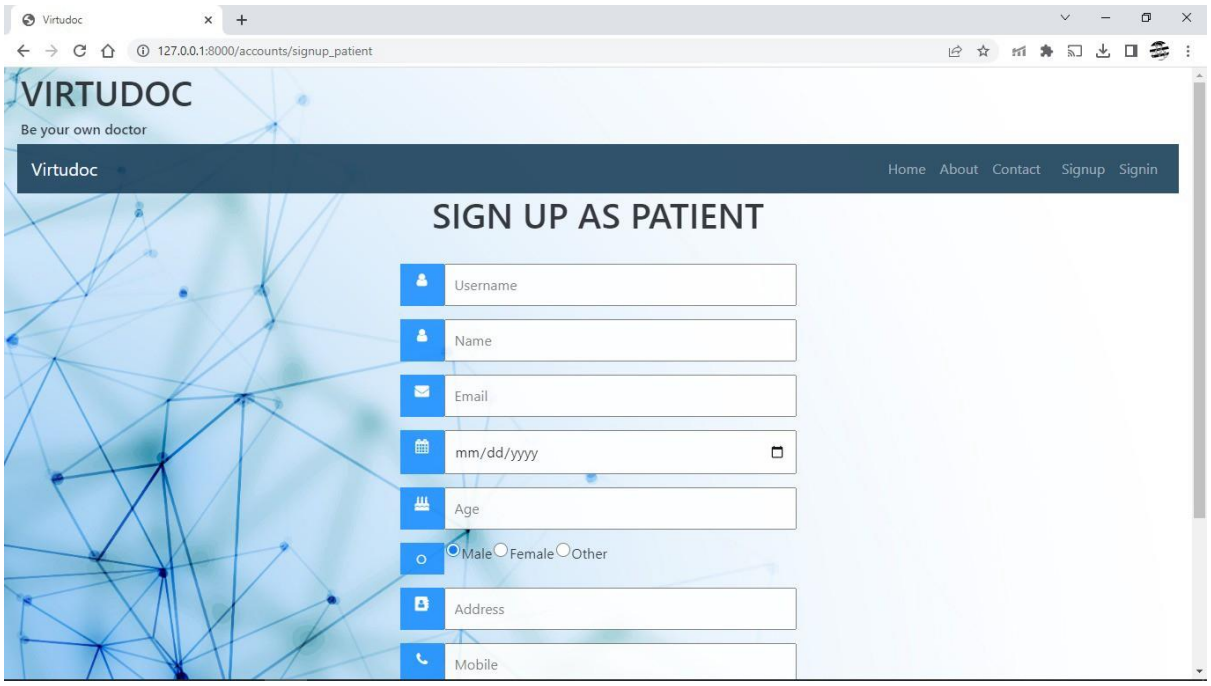

Fig 5.4 Signup form as Patient

The signup for is created using HTML and CSS where the user needs to fill up all the details asked in the form asked in respective form. A user is allowed to access the features and functionalities of the web app only if he is an authorised user.

5.3 SignIn/Login Form

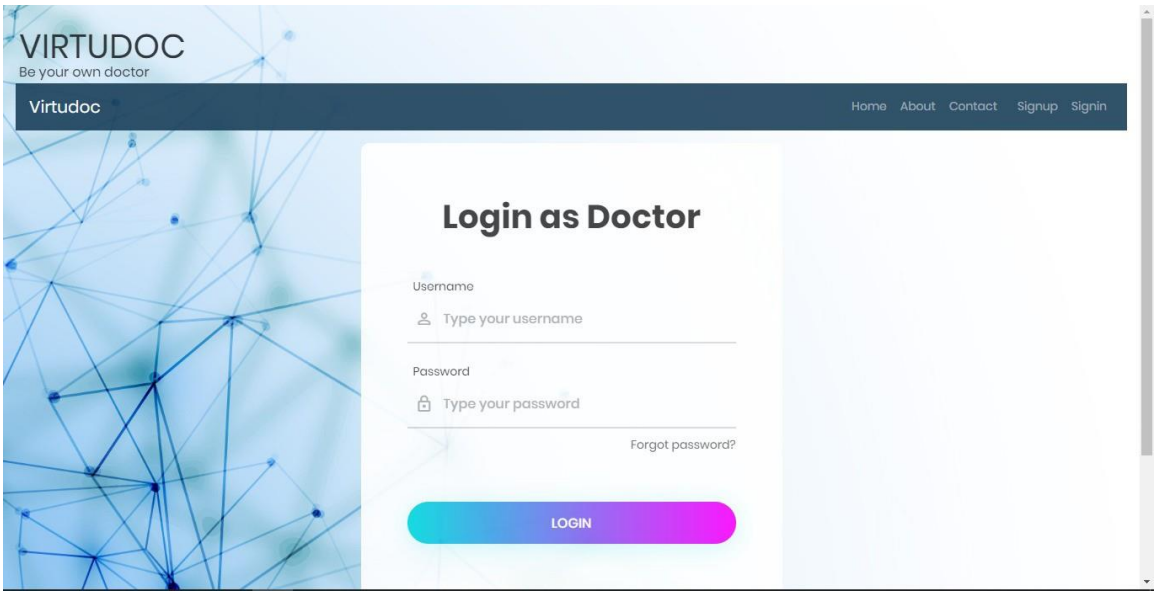

Fig 5.5 Login Page as Doctor

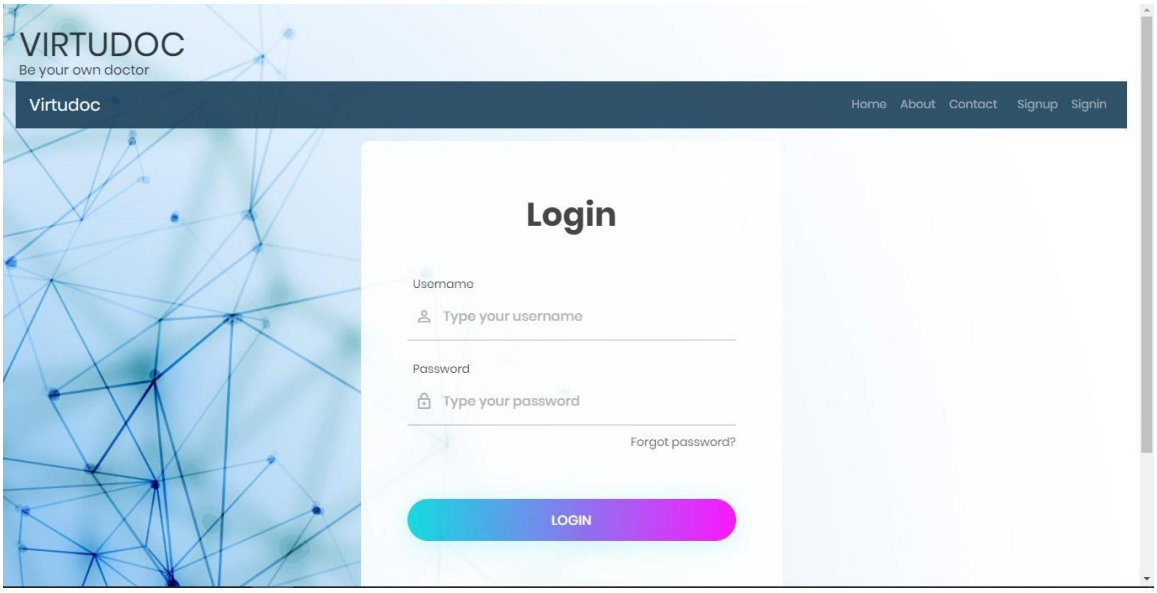

Fig 5.6 Login page for Patient

Login Page allows the already signup user to sign in the app. when a user enters login credentials, these are matched with the entries in database and a user is denied access if there is no match in the database.

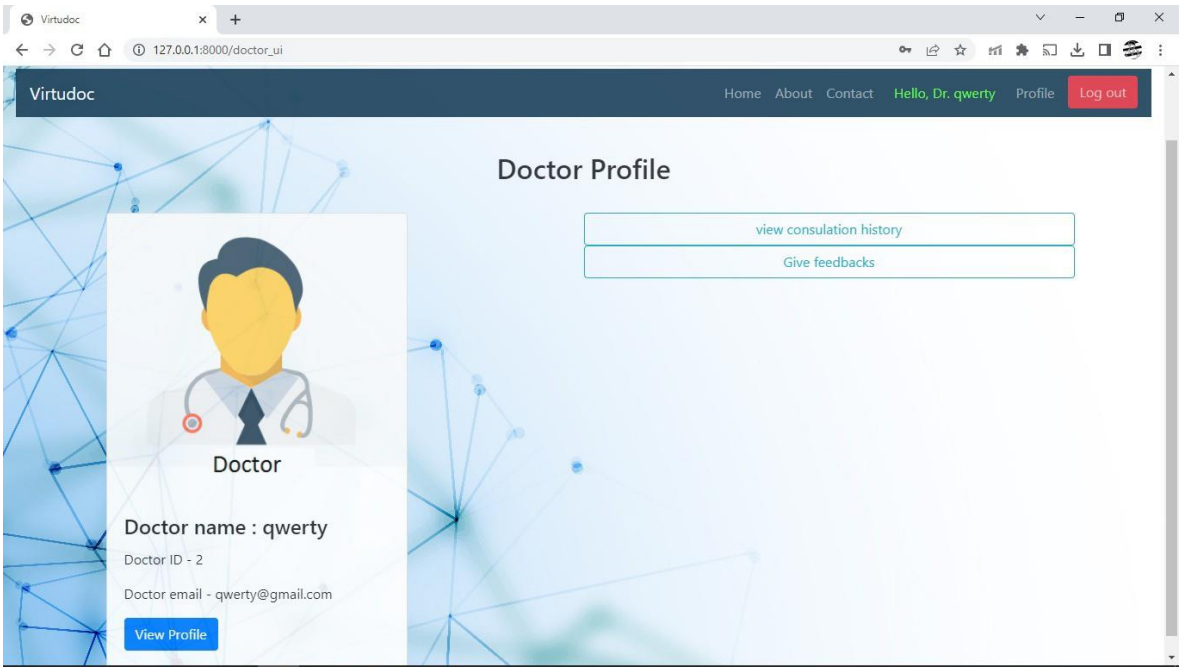

Fig 5.7 Profile Page for Doctor

Fig 5.7 shows the Profile section for doctors where he/she can see/edit his personal details and also see the past consultation history and chat with his patients

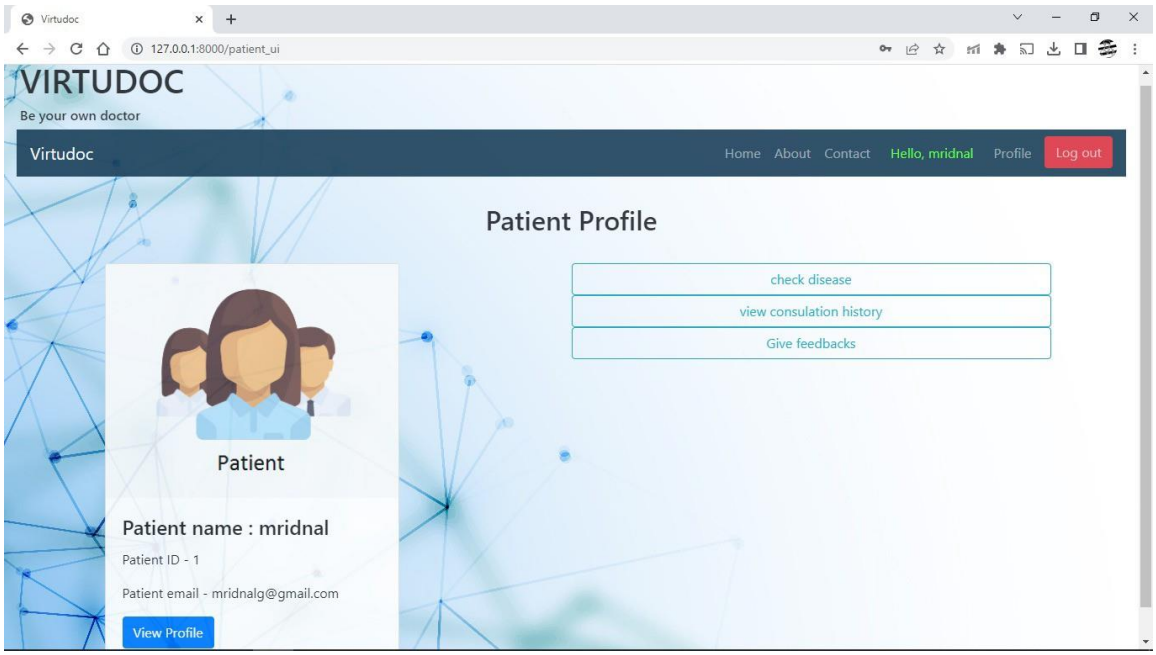

Fig 5.8 Profile Section for Patient

Fig 5.8 shows the Profile section for Patients allows the user to see/edit his personal details and also see the past consultation history and chat with his doctor along with the past diseases he has been suffering from. Patient can also give feedbacks based on the chat he/she had with the doctor

5.4 Disease Prediction

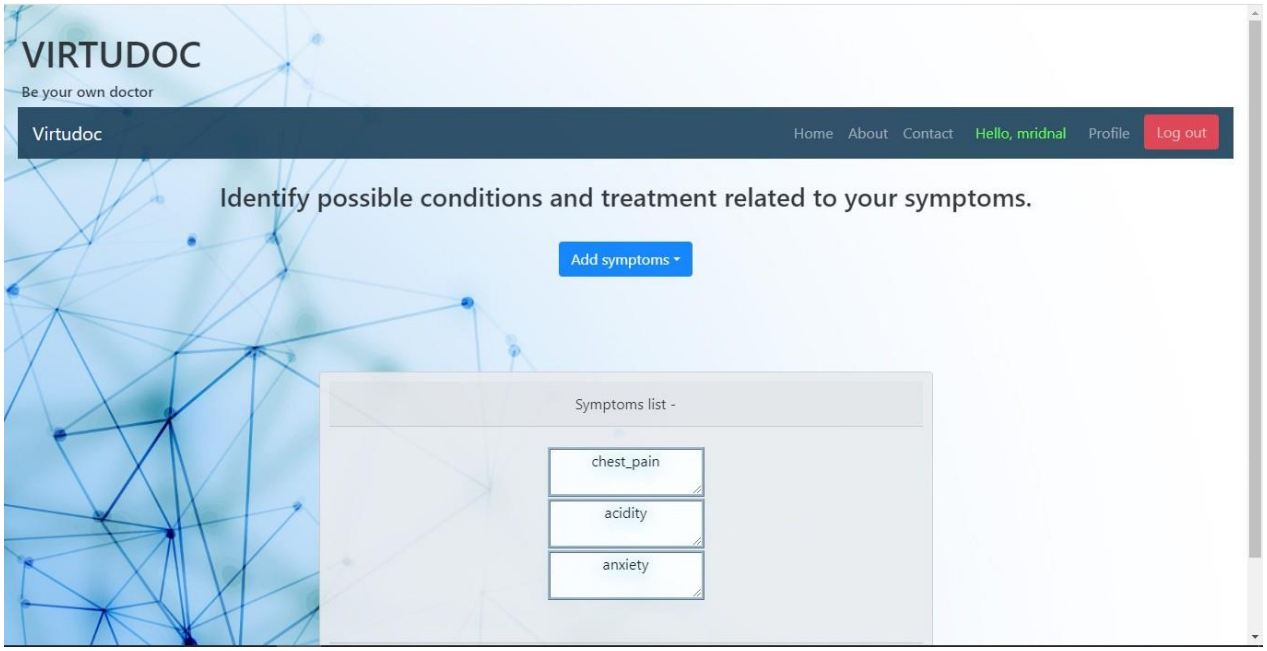

Fig 5.9 Disease Prediction System on WebApp

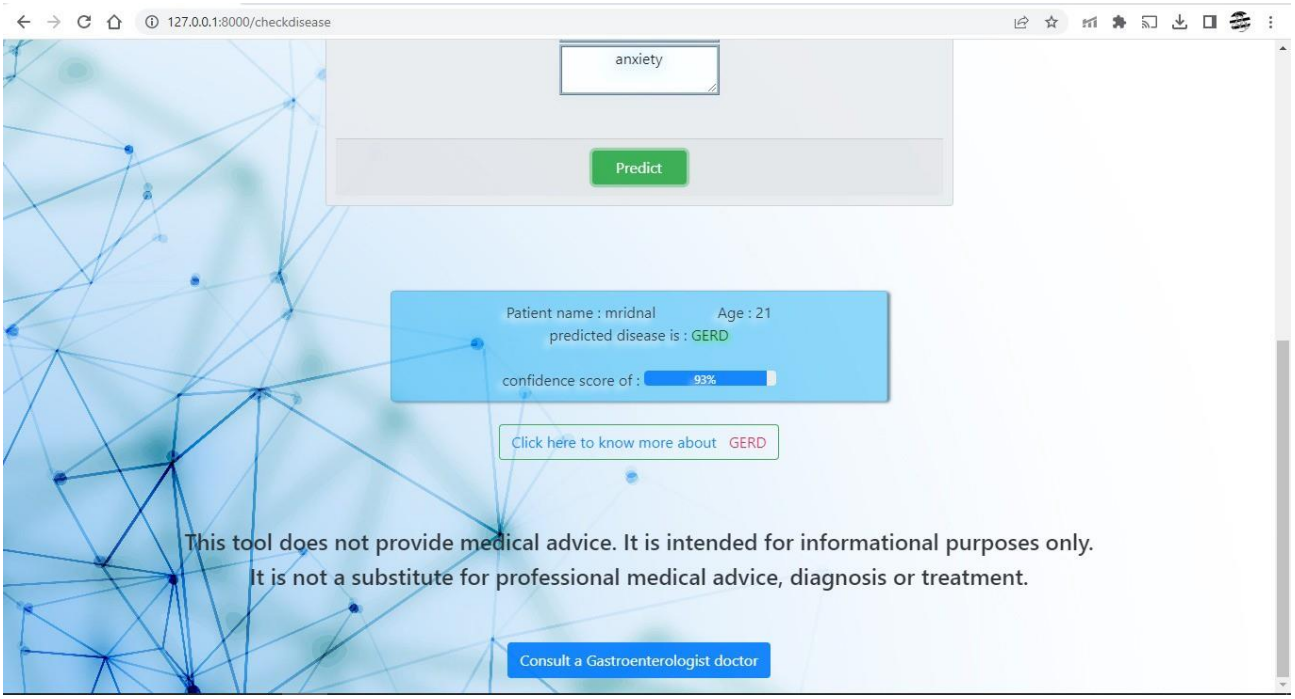

Fig 5.10 Result based on symptoms entered

Fig 5.9 & 5.10 shows how a user can choose the symptoms he's suffering from the list shown and the result predicted based on the ML algorithm integrated. Below that a clickable button is also shown to consult a specialist doctor.

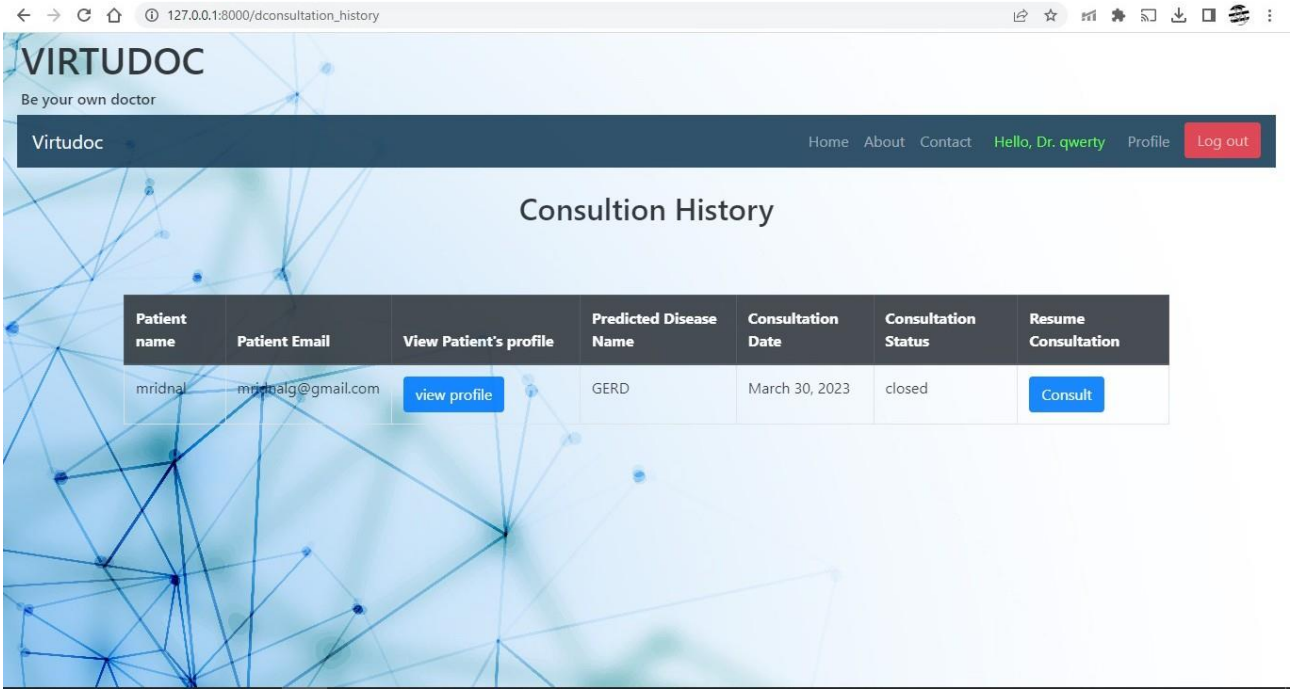

Fig 5.11 Consultation History shown to Doctor

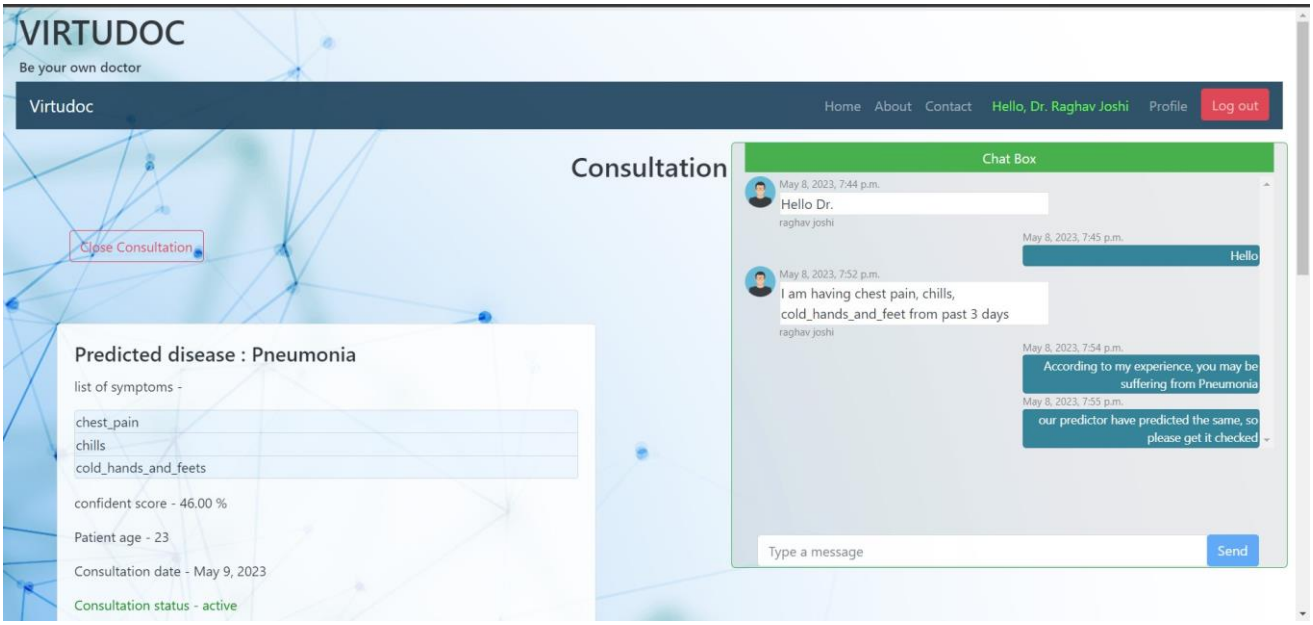

Fig 5.12 Chat window of Doctor's consultation

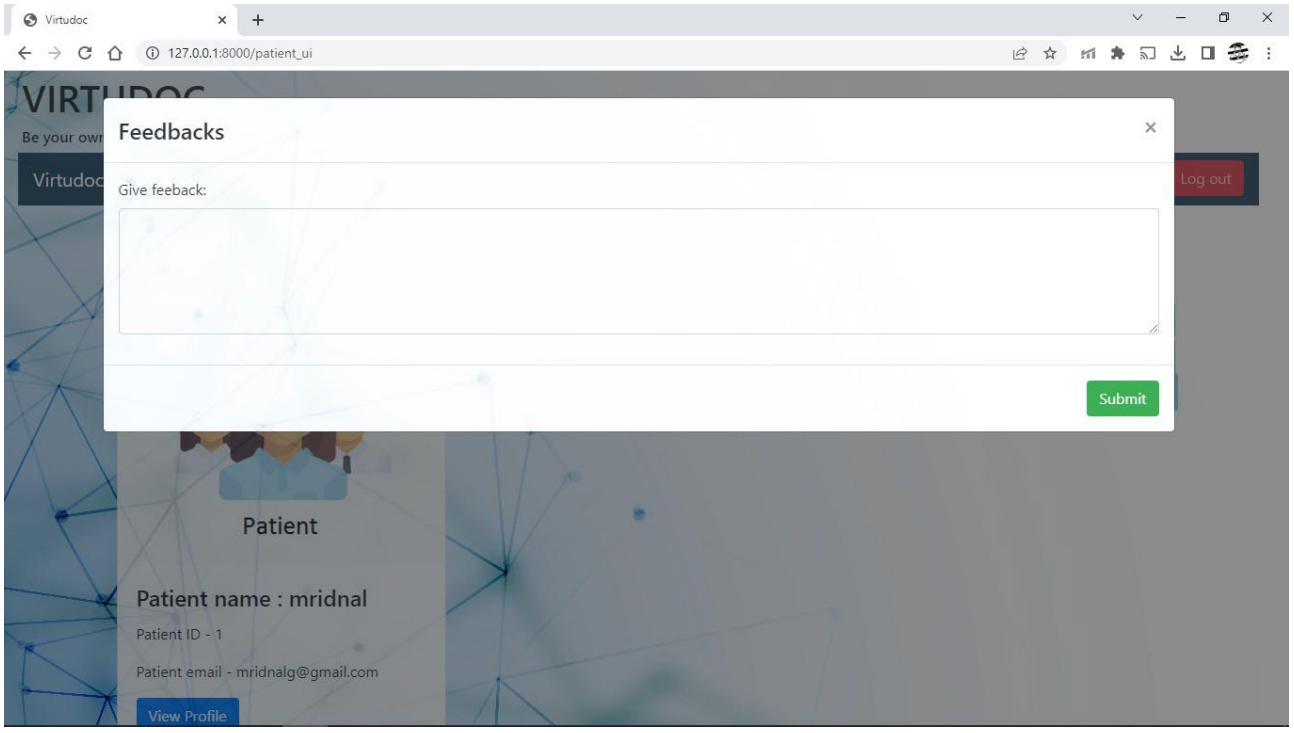

Fig 5.13 Feedback form at chat end

Fig 5.13 shows a feedback form to both doctor and patient to rate the later as per the chat experience. These feedback affects the rating of the respective.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

This project has offered us a valuable opportunity to learn and gain experience. We were able to learn how to perform data analysis and cleaning of complex disease datasets, which was more challenging than anticipated. Furthermore, we obtained practical knowledge in utilizing Django, which is considered one of the most extensively used deep learning frameworks in the field.

Additionally, the success of this project heavily relies on the understanding of mathematical concepts and software architecture. While the web version of the model was completed ahead of schedule, a significant amount of time was dedicated to developing the model itself.

Thirdly, collaborating as a team allowed us to exchange and improve upon our initial ideas. We were able to identify and address potential challenges that may have been overlooked if we were working individually.

5.1 Conclusion

The application of machine learning in the healthcare sector can enhance the precision of risk models. Due to the growing population and shortage of trained healthcare professionals, the doctor-to-patient ratio in India is often far higher than the recommended ratio of 1 in every 100 patients set by the World Health Organization. With the added strain caused by the COVID-19 pandemic, there is an even greater need for efficient and effective healthcare solutions. However, increasing the number of healthcare workers is not always feasible. This is the point where the application of machine learning and artificial intelligence can play a significant part in augmenting the efficiency and precision of the existing healthcare systems, while also minimizing expenses.

Ensuring proper medical attention and treatment information is essential for the effective management of chronic diseases in individuals. Additionally, this system can benefit those seeking to improve their health condition through self-care, as self-management is an essential component of chronic disease treatment. Mobile applications can be used to record patients' health information, making them a more effective tool for enabling self-management.

In recent times, the healthcare industry has witnessed significant transformation owing to the incorporation of information technology (IT) into its operations. The primary objective of integrating IT into healthcare is to enhance the affordability and convenience of healthcare services for individuals, similar to how smartphones have simplified various aspects of daily lifestyle.

The predominant intent of the system proposed is to employ a machine learning approach to recognize and anticipate chronic diseases in an individual. These diseases tend to persist for an extended period and have a higher mortality rate, thereby making their diagnosis of utmost significance in the healthcare sector. Predicting the occurrence of these diseases can enable one to take precautionary measures and evade its impact. Moreover, early detection of chronic diseases can facilitate better treatment.

5.2 Future Scope

Undoubtedly, this is just the initial implementation of the proposed system, and there are numerous opportunities to enhance this health app, including the addition of more features. The obtained high accuracy should be verified since it could be due to overfitting. Therefore, it is recommended to perform testing with new data in the near future to verify the robustness of the model.

The potential of deep learning in the domain of anomaly prediction is significant and has the ability to make significant progress in the challenging field of anomaly detection. However, this progress can only be made possible if a large amount of data is made available to the research community, which could be obtained from hospitals and medical practitioners. Researchers are currently exploring more optimization techniques, feature selection algorithms, and classification algorithms to further enhance the predictive system's performance for the diagnosis of diseases.

The system we are developing will have a website that will offer various other features such as a health tracker and availability information of hospitals. Electronic health records are a comprehensive collection of medical and health data in one system to ensure data availability and accessibility. Our models can be trained using datasets from one electronic health record and can be applied to predict outcomes from other systems.

A chronic disease prediction web application has significant potential for future development and enhancement. One potential area for improvement is in the accuracy of the predictions. As more data becomes available and machine learning algorithms are further refined, the accuracy of the predictive model can be improved.

Another domain for development is in the user interface and user experience. The web application can be made more user-friendly and intuitive, with better visualization of the data and prediction results. Additionally, the application can be made more personalized to the user's specific health profile, incorporating information such as lifestyle habits, medical history, and genetic factors.

Integration with other health technologies, such as wearable devices and electronic health records, is also a possibility. This could allow for real-time monitoring of a user's health status and provide more accurate predictions.

Furthermore, collaboration with healthcare providers can provide more robust and accurate predictions, as well as better treatment recommendations. Overall, the future scope of a chronic disease prediction web application is vast and offers many opportunities for innovation and advancement in the field of healthcare.

REFERENCES

[1] Arup Jana, Aparajita Chattopadhyay, “Prevalence and potential determinants of chronic disease among elderly in India: Rural-urban perspectives”, PLOS ONE 17(3): e0264937.

[2] D. Dahiwade, G. Patle and E. Meshram, "Designing Disease Prediction Model Using Machine Learning Approach," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2019, pp. 1211-1215, doi: 10.1109/ICCMC.2019.8819782.

[3] Uddin, S., Khan, A., Hossain, M. *et al.* Comparing different supervised machine learning algorithms for disease prediction. *BMC Med Inform Decis Mak* **19**, 281 (2019). https://doi.org/10.1186/s12911-019-1004-8

[4] Divya Jain, Vijendra Singh, “Feature selection and classification systems for chronic disease prediction: A review”, Egyptian Informatics Journal, Volume 19, Issue 3, 2018, Pages 179-189, ISSN 1110-8665. https://doi.org/10.1016/j.eij.2018.03.002.

[5] M. Chen, Y. Hao, K. Hwang, L. Wang and L. Wang, "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities," in IEEE Access, vol. 5, pp. 8869-8879, 2017, doi: 10.1109/ACCESS.2017.2694446.

[6] Battineni G, Sagaro GG, Chinatalapudi N, Amenta F., “Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis”, J Pers Med. 2020 Mar 31;10(2):21. doi: 10.3390/jpm10020021. PMID: 32244292; PMCID: PMC7354442.

[7] Shratik J. Mishra, Albar M. Vasi, Vinay S. Menon, Prof. K. Jayamalini, “GDPS - General Disease Prediction System”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03 | Mar-2018 |e-ISSN: 2395-0056.

[8] A. Tikotikar and M. Kodabagi, "A survey on technique for prediction of disease in medical data," 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), Bengaluru, India, 2017, pp. 550-555, doi: 10.1109/SmartTechCon.2017.8358432.

[9] B. D. Kanchan and M. M. Kishor, "Study of machine learning algorithms for special disease prediction using principal of component analysis," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 2016, pp. 5-10, doi: 10.1109/ICGTSPICC.2016.7955260.

[10] P. Hamsagayathri and S. Vigneshwaran, "Symptoms Based Disease Prediction Using Machine Learning Techniques," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 747-752, doi: 10.1109/ICICV50876.2021.9388603.

[11] N. A. Afiqah Mohd Johari, N. Mohamad and N. Isa, "Smart Self-Checkup for Early Disease Prediction," 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 2020, pp. 33-38, doi: 10.1109/I2CACIS49202.2020.9140205.

[12] Rajora, Harish & Punn, Narinder & Sonbhadra, Sanjay & Agarwal, Sonali. (2021), “Web based disease prediction and recommender system”, Indian Institute of Information Technology Allahabad, India.

chronic disease prediction using ML

ORIGINALITY REPORT

21%	17%	13%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github-wiki-see.page Internet Source	1%
2	bmcmedinformdecismak.biomedcentral.com Internet Source	1%
3	ebin.pub Internet Source	1%
4	www.mdpi.com Internet Source	1%
5	Submitted to Middle East College of Information Technology Student Paper	1%
6	pdfcoffee.com Internet Source	1%
7	www.researchgate.net Internet Source	1%
8	oaijse.com Internet Source	1%
9	Ahelam Tikotikar, Mallikarjun Kodabagi. "A survey on technique for prediction of disease	1%

	in medical data", 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), 2017 Publication	
10	www.coursehero.com Internet Source	<1 %
11	Mohammad-H. Tayarani-N.. "Applications of Artificial Intelligence in Battling Against Covid-19: A Literature Review", Chaos, Solitons & Fractals, 2020 Publication	<1 %
12	Submitted to University of Computer Studies Student Paper	<1 %
13	www.irjmets.com Internet Source	<1 %
14	www.eurchembull.com Internet Source	<1 %
15	"ICDSMLA 2020", Springer Science and Business Media LLC, 2022 Publication	<1 %
16	analyticsindiamag.com Internet Source	<1 %
17	www.ncbi.nlm.nih.gov Internet Source	<1 %
18	Submitted to Green University Of Bangladesh Student Paper	<1 %

19	Submitted to Visvesvaraya National Institute of Technology Student Paper	<1 %
20	uis.brage.unit.no Internet Source	<1 %
21	dzone.com Internet Source	<1 %
22	hdl.handle.net Internet Source	<1 %
23	ijariie.com Internet Source	<1 %
24	dokumen.pub Internet Source	<1 %
25	thedeveloperblog.com Internet Source	<1 %
26	bbrc.in Internet Source	<1 %
27	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
28	opac.elte.hu Internet Source	<1 %
29	Submitted to CSU Northridge Student Paper	<1 %

30	Nur Aliyah Afiqah Mohd Johari, Norizan Mohamad, Norulhidayah Isa. "Smart Self-Checkup for Early Disease Prediction", 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2020 Publication	<1 %
31	Submitted to Savitribai Phule Pune University Student Paper	<1 %
32	blog.4geeks.io Internet Source	<1 %
33	ezeichen.com Internet Source	<1 %
34	Melike Colak, Talya Tumer Sivri, Nergis Pervan Akman, Ali Berkol, Yahya Ekici. "A Study of Disease Prediction on Weighted Symptom Data Using Deep Learning and Machine Learning Algorithms", 2022 International Conference on Theoretical and Applied Computer Science and Engineering (ICTASCE), 2022 Publication	<1 %
35	docu.tips Internet Source	<1 %
36	"International Conference on Artificial Intelligence and Sustainable Engineering",	<1 %

	Springer Science and Business Media LLC, 2022 Publication	
37	Submitted to National University of Ireland, Maynooth Student Paper	<1 %
38	Pothana Hema, Arunarkavalli Darbha, N. Sunny, Raavi Venkata Naganjani. "Disease Prediction Using Symptoms based on Machine Learning Algorithms and Natural Language Processing", 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), 2023 Publication	<1 %
39	Submitted to University of Bradford Student Paper	<1 %
40	Submitted to CSU, San Jose State University Student Paper	<1 %
41	www.clariba.com Internet Source	<1 %
42	www.mordorintelligence.com Internet Source	<1 %
43	Sandeepkumar Hegde, Rajalaxmi Hegde. "Symmetry Based Feature Selection with Multi layer Perceptron for the prediction of Chronic	<1 %

	Disease", International Journal of Recent Technology and Engineering (IJRTE), 2019 Publication	
44	Submitted to University of Glasgow Student Paper	<1 %
45	Submitted to Vel Tech University Student Paper	<1 %
46	esource.dbs.ie Internet Source	<1 %
47	www.ijraset.com Internet Source	<1 %
48	ijarcs.info Internet Source	<1 %
49	Submitted to Bocconi University Student Paper	<1 %
50	Submitted to Coventry University Student Paper	<1 %
51	Hadda Ben Elhadj, Farag Sallabi, Amira Henaien, Lamia Chaari, Khaled Shuaib, Maryam Al Thawadi. "Do-Care: A dynamic ontology reasoning based healthcare monitoring system", Future Generation Computer Systems, 2021 Publication	<1 %
52	Submitted to Otto-von-Guericke-Universität Magdeburg	<1 %

	Student Paper	
53	Yin Zhang, Haiyang Wang, Min Chen, Jiafu Wan, Iztok Humar. "IEEE Access Special Section Editorial: Healthcare Big Data", IEEE Access, 2018 Publication	<1 %
54	towardsdatascience.com Internet Source	<1 %
55	vsip.info Internet Source	<1 %
56	Submitted to Kwame Nkrumah University of Science and Technology Student Paper	<1 %
57	Submitted to Nanyang Technological University Student Paper	<1 %
58	Submitted to University of Brighton Student Paper	<1 %
59	journals.plos.org Internet Source	<1 %
60	www.science.gov Internet Source	<1 %
61	Submitted to Bannari Amman Institute of Technology Student Paper	<1 %
	Submitted to Belgium Campus iTversity NPC	

62	Student Paper	<1 %
63	Submitted to SP Jain School of Global Management Student Paper	<1 %
64	Submitted to Unicaf University Student Paper	<1 %
65	Submitted to University of London External System Student Paper	<1 %
66	doaj.org Internet Source	<1 %
67	Submitted to Carnegie Mellon University Student Paper	<1 %
68	Submitted to Intercollege Student Paper	<1 %
69	Submitted to Kennesaw State University Student Paper	<1 %
70	Submitted to University of North Texas Student Paper	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		