

DISEASE PREDICTION BY USING MACHINE LEARNING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Information Technology

By

VISHWA B (Reg.No - 40120109)

ROHITH P (Reg.No - 40120075)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A++" by NAAC | 12B Status by UGC |

Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

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

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VISHWA B (40120109)** and **ROHITH P (40120075)** who carried out the Project entitled **"DISEASE PREDICTION BY USING MACHINE LEARNING"** under my supervision from January 2024 to April 2024.

Internal Guide

Dr. P. JEYANTHI M.E., Ph.D.

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on-----

Internal Examiner

External Examiner



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

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

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VISHWA B (40120109)** and **ROHITH P (40120075)** who carried out the Project entitled "**DISEASE PREDICTION BY USING MACHINE LEARNING**" under my supervision from January 2024 to April 2024.

Internal Guide

Dr. P. JEYANTHI M.E., Ph.D.

Head of the Department
Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 18/04/2024

Internal Examiner

External Examiner

DECLARATION

I **VISHWA B (40120109)**, hereby declare that the Project Phase-2 Report entitled “**DISEASE PREDCITION BY USING MACHINE LEARNING**” done by me under the guidance of **Dr. P. JEYANTHI,M.E., Ph.D.** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Information Technology**.

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr.R. Subhashini M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. Jeyanthi M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

The integration of machine learning techniques in healthcare has revolutionized disease prediction and diagnosis. This paper presents a comprehensive review of recent advancements in disease prediction utilizing machine learning algorithms. The primary focus is on the application of various machine learning models, including but not limited to decision trees, support vector machines, neural networks, and ensemble methods, in predicting diseases across different medical domains such as cardiology, oncology, neurology, and infectious diseases. The review highlights the diverse datasets employed for training and validation, ranging from electronic health records to genomic data, wearable sensor data, and medical imaging. Furthermore, the paper discusses the challenges associated with disease prediction using machine learning, including data privacy concerns, model interpretability, and generalization across diverse populations. Additionally, it explores promising future directions, such as the integration of multimodal data sources, transfer learning techniques, and federated learning approaches to enhance disease prediction accuracy and robustness. Overall, this review provides valuable insights into the current state-of-the-art in disease prediction using machine learning and offers guidance for future research endeavours in this burgeoning field. With the advent of machine learning algorithms, disease prediction has seen a significant shift towards more accurate and personalized approaches. This paper delves into the landscape of disease prediction utilizing machine learning techniques, encompassing a wide array of diseases and predictive models. Through an extensive review of literature, we explore the methodologies employed in disease prediction, including feature selection, model optimization, and evaluation metrics. We discuss the importance of high-quality datasets and the challenges associated with data pre-processing and imbalanced class distributions. Moreover, we analyse the interpretability of machine learning models in the context of disease prediction and the implications for clinical decision-making.

TABLE OF CONTENT

Chapter No	TITLE	Page No
	ABSTRACT	iv
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 BACKGROUND AND MOTIVATION	1
	1.2 PROBLEM STATEMENT	2
	1.3 OBJECTIVE AND SCOPE	2
	1.3.1 SCOPE	3
	1.4 APPLICATIONS OF DISEASE PREDICTION	3
	1.5 ORGANIZATION OF THESIS	4
2	LITERATURE SURVEY	5
	2.1 INTERFACES FROM LITERATURE SURVEY	8
	2.2 EXISTING SYSTEM AND PROPOSED SYSTEM	8
	2.2.1 EXISTING SYSTEM	8
	2.2.2 PROPOSED SYSTEM	9
	2.3 OPEN PROBLEMS FOR EXISTING PROBLEM	9
3	METHODOLOGY	11
	3.1 FLOW CHART	11
	3.2 DATA COLLECTION AND PREPROCESSING	12
	3.2.1 DATA COLLECTION	12
	3.2.2 DATA PREPROCESSING	12
	3.3 SELECTED METHODOLOGY	12
	3.4 ARCHITECTURE OF PROPOSED SYSTEM	14

3.5	ALGORITHM	15
3.5.1	GRADIENT BOOSTING MACHINES	15
3.5.2	CONVOLUTIONAL NEURAL NETWORK	15
3.5.3	K- NEAREST NEIGHBORS	16
3.7	DESCRIPTION OF S/W FOR IMPLEMENTATION	17
3.8	FINANCIAL REPORT COSTING	18
3.8.1	DEVELOPMENT COSTS	18
3.8.2	OPERATIONAL COSTS	18
3.8.3	MARKETING AND SALES	19
3.8.4	REVENUE PROJECTIONS	19
3.8.5	FINANCIAL PROJECTIONS	19
3.8.6	RISK MANAGEMENT	19
4	RESULTS AND DISCUSSION	20
4.1	DEVELOPMENT AND DEPLOYMENT SETUP	20
4.1.1	DEVELOPMENT ENVIRONMENT SETUP	20
4.1.2	DATA ACQUISITION AND PRE-PROCESSING	20
4.1.3	MODEL DEVELOPMENT AND EVALUATION	21
4.1.4	INTEGRATION AND TESTING	21
4.1.5	DEPLOYMENT SETUP	22
4.1.6	MONITORING AND MAINTENANCE	22
4.2	TESTING	23
4.2.1	CLINICAL IMPACTS	23
4.2.2	ETHICAL CONSIDERATIONS	24

	4.2.3 FUTURE DIRECTIONS	24
4.3	TRADEOFF IN THE PROJECT	24
	4.3.1 ACCURACY VS. INTERPRETABILITY	24
	4.3.2 DATA PRIVACY VS. UTILITY	25
	4.3.3 GENERALIZATION VS. SPECIALIZATION	25
	4.3.4 INNOVATION VS. REGULATION	25
	4.3.5 COST VS. BENEFIT	25
	4.3.6 SHORT-TERM VS. LONG-TERM OUTCOMES	25
5	CONCLUSION	27
	5.1 CONCLUSION	27
	5.2 FUTURE WORKS	28
	REFERENCES	29
	APPENDIX	30
	SOURCE CODE	30
	SCEENSHOT	44

LIST OF FIGURES

Figure No	Figure Name	Page No
3.1	Decision Tree	11
3.2	Block Diagram	14
3.3	Gradient Boosting Machines	15
3.4	Convolutional Neural Network	16
3.5	K-Nearest Neighbors	16
4.1	Collecting Datasets	20
4.2	Training Model	21
4.3	Homepage	22

CHAPTER 1

INTRODUCTION

Disease prediction using patient treatment history and health data by applying data mining and machine learning techniques is ongoing struggle for the past decades. Many works have been applied data mining techniques to pathological data or medical profiles for prediction of specific diseases. These approaches tried to predict the reoccurrence of disease. Also, some approaches try to do prediction on control and progression of disease. The recent success of deep learning in disparate areas of machine learning has driven a shift towards machine learning models that can learn rich, hierarchical representations of raw data with little preprocessing and produce more accurate results. With the development of big data technology, more attention has been paid to disease prediction from the perspective of big data analysis; various researches have been conducted by selecting the characteristics automatically from a large number of data to improve the accuracy of risk classification rather than the previously selected characteristics. The main focus is on to use machine learning in healthcare to supplement patient care for better results. Machine learning has made easier to identify different diseases and diagnosis correctly. Predictive analysis with the help of efficient multiple machine learning algorithms helps to predict the disease more correctly and help treat patients.

The ultimate goal is to create a robust and interpretable predictive tool that can be integrated into clinical workflows. Such a tool holds the potential to assist healthcare professionals in identifying individuals at high risk of developing specific diseases, enabling timely intervention and personalized treatment plans. This research contributes to the ongoing paradigm shift in healthcare towards proactive and preventive strategies, harnessing the power of machine learning to transform disease prediction into a valuable clinical asset.

1.1 Background and Motivation

In recent years, the field of healthcare has witnessed a surge in interest and investment in utilizing machine learning (ML) techniques for disease prediction. This momentum is fueled by several factors. These rich datasets provide an opportunity

to develop robust predictive models for various diseases. The rapid advancement of ML algorithms, coupled with increased computational power, has enabled the development of more sophisticated and accurate disease prediction models. Techniques such as deep learning, ensemble methods, and transfer learning have shown promise in extracting complex patterns from healthcare data. There is a growing recognition of the importance of personalized medicine, which tailors treatment and preventive strategies to individual characteristics. ML-based disease prediction facilitates the customization of healthcare interventions by identifying high-risk individuals and targeting interventions accordingly.

1.2 Problem Statement

Despite the promising advancements in machine learning (ML) for disease prediction, several challenges hinder the widespread adoption and effectiveness of these techniques in clinical practice. The primary issues include. Many healthcare datasets suffer from issues such as missing values, noise, and inconsistencies, which can compromise the performance of ML models. Moreover, access to high-quality and diverse datasets representative of different populations and healthcare settings remains limited. ML models often operate as "black boxes," making it challenging for clinicians to interpret their predictions and understand the underlying factors contributing to disease risk. Lack of interpretability can hinder trust and acceptance of ML-based predictions in clinical decision-making. ML models trained on biased or imbalanced datasets may exhibit poor generalization performance, leading to inaccurate predictions, particularly for underrepresented or marginalized populations. Addressing bias and ensuring fairness in disease prediction models is crucial for equitable healthcare delivery.

1.3 Objective and Scope

The objective of this study is to conduct a comprehensive review of disease prediction methodologies utilizing machine learning (ML) techniques, with a focus on recent advancements, challenges, and future directions. To examine a variety of ML models and algorithms employed in disease prediction, including decision trees, support vector machines, neural networks, methods, and deep learning

architectures, among other. To investigate disease prediction applications across a range of medical specialties, such as cardiology, oncology, neurology, infectious diseases, and others, to understand the breadth of ML applications in healthcare.

1.3.1 Scope

The scope for disease prediction using machine learning is quite vast and holds significant potential for improving healthcare outcomes. Here are some aspects to consider within this scope. Gathering relevant data is crucial. This may include medical records, patient demographics, genetic information, environmental factors, and more. Preprocessing involves cleaning, normalizing, and formatting the data to make it suitable for analysis. Identifying the most relevant features (variables) for predicting diseases is essential. Feature engineering may involve transforming or combining features to enhance predictive performance. Choosing appropriate machine learning algorithms (such as decision trees, support vector machines, neural networks, etc.) based on the nature of the data and the problem at hand. Training these models using historical data to learn patterns and relationships between variables. Validating the models using techniques such as cross-validation to ensure their generalizability. Evaluating model performance using metrics like accuracy, sensitivity, specificity, and area under the ROC curve.

1.4 Applications for Disease Prediction

The applications for disease prediction using machine learning are diverse and can be applied across various domains within healthcare and biomedical research. Some notable applications include.

Early Disease Detection: Machine learning models can analyze patient data to identify early signs and symptoms of diseases such as cancer, diabetes, cardiovascular diseases, and neurological disorders. By analyzing genetic data, medical history, and other patient-specific factors, machine learning algorithms can predict individual responses to treatments and medications. This enables the development of personalized treatment plans tailored to each patient's unique characteristics.

Public Health Surveillance: Machine learning techniques can analyze large-scale population health data, including electronic health records, medical claims data, and social media feeds, to identify disease outbreaks, track disease trends, and inform public health interventions. Machine learning models can stratify patients into different risk categories based on their likelihood of developing certain diseases or experiencing adverse health outcomes. This information can guide preventive measures and targeted interventions for high-risk individuals.

Drug Discovery and Development: Machine learning techniques can analyze biological data, such as genomic, proteomic, and metabolomics data, to identify potential drug targets, predict drug efficacy, and optimize drug design. This accelerates the drug discovery and development process. These applications demonstrate the versatility and potential impact of using machine learning for disease prediction across various stages of healthcare delivery, from prevention and diagnosis to treatment and management.

1.5 Organization for thesis

Explain disease prediction's significance in healthcare and state research objectives. Review previous studies on disease prediction and identify gaps. Describe data sources, Pre-processing steps, and machine learning techniques. Discuss relevant features and their engineering processes. Present machine learning models and their parameters. Display performance metrics and compare model outcomes. Interpret results, address limitations, and propose future research directions. Discuss data privacy and biases, ensuring fairness in predictions.

CHAPTER 2

LITERATURE SURVEY

- [1] D. Dahiwade, G. Patle, and E. Meshram, “Designing disease prediction model using machine learning approach,” Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019, no. Iccmc, pp. 1211–1215, 2019.**

Dahiwade et al. [9] proposed a ML based system that predicts common diseases. The symptoms dataset was imported from the UCI ML depository, where it contained symptoms of many common diseases. The system used CNN and KNN as classification techniques to achieve multiple diseases prediction. Moreover, the proposed solution was supplemented with more information that concerned the living habits of the tested patient, which proved to be helpful in understanding the level of risk attached to the predicted disease.

- [2] H. L. Chen, C. C. Huang, X. G. Yu, X. Xu, X. Sun, G. Wang, and S. J. Wang, “An efficient diagnosis system for detection of Parkinson’s disease using fuzzy k-nearest neighbor approach,” Expert Systems with Applications, vol. 40, no. 1, pp. 263–271, 2013.**

In light of this study, the findings of Chen et al. [10] also agreed that CNN outperformed typical supervised algorithms such as KNN, NB, and DT. The authors concluded that the proposed model scored higher in terms of accuracy, which is explained by the capability of the model to detect complex nonlinear relationships in the feature space. Moreover, CNN detects features with high importance that renders better description of the disease, which enables it to accurately predict diseases with high complexity [9], [10]. In addition, the analysis of the performances is only evaluated in terms of accuracy, which debunks the validity of the presented findings [9]. Moreover, the authors did not take into consideration the bias problem that is faced by the tested algorithms [9], [10]. In illustration, the incorporation of more feature variables could immensely ameliorate the performance metrics of underperformed algorithms [11].

- [3] Y. Amirgaliyev, S. Shamiluulu, and A. Serek, “Analysis of chronic kidney disease dataset by applying machine learning methods,” in 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), 2018.**

Serek et al. [12] planned a comparative study of classifiers performance for Chronic Kidney disease (CKD) detection using The Kidney Function Test (KFT) dataset. In this study, the classifiers used are KNN, NB, and RF classifier; their performance is examined in terms of F-measure, precision, and accuracy. In consideration of this study, Vijayarani [13] aimed to detect kidney diseases using SVM and NB. The classifiers were used to identify four types of kidney diseases namely Acute Nephritic Syndrome, Acute Renal Failure, Chronic Glomerulonephritis, and CKD. Additionally, the research was focused on determining the better performing classification algorithm based on the accuracy and execution time. From the results, SVM considerably achieved higher accuracy than NB, which make it the better performing algorithm. However, NB classified data with minimum execution time. Other several empirical studies also focused on locating CKD.

- [4] Charleonnann, T. Fufaung, T. Niyomwong, W. Chokchueypattanakit, S. Suwannawach, and N. Ninchawee, “Predictive analytics for chronic kidney disease using machine learning techniques,” 2016 Management and Innovation Technology International Conference, MITiCON 2016, pp. MIT80–MIT83, 2017.**

Charleonnann et al. [14] and Kotturu et al. [15] concluded that the SVM classifier is the most adequate for kidney diseases because it deals well with semi-structured and unstructured data. Such flexibility allowed SVM to handle larger features spaces, which resulted in acquiring high accuracy when detecting complex kidney diseases. Although supported by findings, the conclusion is weakened by prior suggestion that different hyper-parameters were not experimented when evaluating the performances of ML algorithms. According to Uddin [3] the exploration of the hyper-parameter space can generate different accuracy results and render better performances for ML algorithms.

- [5] M. Marimuthu, M. Abinaya, K. S., K. Madhankumar, and V. Pavithra, “A Review on Heart Disease Prediction using Machine Learning and Data Analytics Approach,” *International Journal of Computer Applications*, vol. 181, no. 18, pp. 20–25, 2018.**

Marimuthu et al. [16] aimed to predict heart diseases using supervised ML techniques. The authors structured the attributes of data as gender, age, chest pain, gender, target and slope [16]. The applied ML algorithms that were deployed are DT, KNN, LR and NB. As per analysis, the LR algorithm gave a high accuracy of 86.89%, which deemed to be the most effective compared to the other mentioned algorithms.

- [6] K. Dwivedi, “Performance evaluation of different machine learning techniques for prediction of heart disease,” *Neural Computing and Applications*, vol. 29, no. 10, pp. 685–693, 2018.**

In 2018, Dwivedi [17] attempted to add more precision to the prediction of heart diseases by accounting for additional parameters such as Resting blood pressure, Serum Cholesterol in mg/dl, and Maximum Heart Rate achieved. The used dataset was imported from the UCI ML laboratory; it was comprised with 120 samples that were heart disease positive, and 150 samples that were heart disease negative. Dwivedi attempted to evaluate the performance of Artificial Neural Networks (ANN), SVM, KNN, NB, LR and Classification Tree. This conclusion is strengthened by the findings of Polaraju [18] and Vahid et al. [19], where the Logistic Regression outperformed other techniques such as ANN, SVM, and Adaboost. The studies excelled in conducting an extensive analysis on the ML models. For instance, various hyper-parameters were tested at each ML algorithm to converge to the best possible accuracy and precision values. Despite that advantage, the small size of the imported datasets constraints the learning models from targeting diseases with higher accuracy and precision.

- [7] P. P. Sengar, M. J. Gaikwad, and A. S. Nagdive, “Comparative study of machine learning algorithms for breast cancer prediction,” *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, pp. 796–801, 2020.

Shubair [20] attempted for the detection of breast cancer using ML algorithms, namely RF, Bayesian Networks and SVM. The researchers obtained the Wisconsin original breast cancer dataset from the UCI Repository and utilized it for comparing the learning models in terms of key parameters such as accuracy, recall, precision, and area of ROC graph. The classifiers were tested using K-fold validation method, where the chosen value of K is equal to 10 [20].

2.1 Inferences from Literature Survey

Based on the literature survey on disease prediction using machine learning, several Interfaces emerge that facilitate the interaction between machine learning models And Health care systems. Many studies leverage EHR systems to access patient data, including demographics.

Training and prediction. Application Programming Interfaces (APIs) is provide standardized interfaces for accessing machine learning models deployed within healthcare systems. These interfaces allow other software applications, such as clinical decision support systems or mobile health apps, to interact with the predictive models and receive predictions. Web-based interfaces offer user-friendly dashboards for clinicians and healthcare providers to interact with predictive models. These interfaces provide visualizations of model outputs, such as risk scores or probability estimates, along with explanations of predictions and recommendations for patient care.

2.2 Existing System and Proposed System

2.2.1 Existing System

Prediction using traditional disease risk model usually involves a machine learning and supervised learning algorithm which uses training data with the labels for the training of the model. High-risk and Low-risk patient classification is done in groups test sets. But these models are only valuable in clinical situations and are widely

studied. A system for sustainable health monitoring using smart clothing by Chen et.al. He thoroughly studied heterogeneous systems and was able to achieve the best results for cost minimization on the tree and simple path cases for heterogeneous systems. The information of patient's statistics, test results, and disease history is recorded in EHR which enables to identify potential data-centric solutions which reduce the cost of medical case studies. Bates et al. propose six applications of big data in the healthcare field. Existing systems can predict the diseases but not the subtype of diseases. It fails to predict the condition of people.

2.2.2 Proposed System

In this paper, we have combined the structure and unstructured data in healthcare fields that let us assess the risk of disease. The approach of the latent factor model for reconstructing the missing data in medical records which are collected from the hospital. And by using statistical knowledge, we could determine the major chronic diseases in a particular region and in particular community. To handle structured data, we consult hospital experts to know useful features.

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

2.3 Open Problems for Existing Problems

Obtaining high-quality and diverse datasets for training ML models is a significant challenge. Often, medical data are incomplete, noisy, and biased, which can affect the performance and generalization of predictive models. Many ML and AI models used in disease prediction are considered "black-box" models, meaning they lack transparency in how they arrive at their predictions. Interpretable and explainable AI techniques are crucial for building trust among clinicians and patients and for understanding the reasoning behind the predictions. ML models trained on data

from one population or healthcare system may not generalize well to other populations or settings due to differences in demographics, healthcare practices, and disease prevalence. Developing models that can transfer knowledge across different populations while maintaining performance is essential. In healthcare datasets, the number of patients with a particular disease often vastly outweighs the number of healthy individuals or patients without that disease, leading to class imbalance. This can bias the model towards the majority class and result in poor performance in predicting minority classes. Disease prediction can benefit from the integration of various types of data sources, including medical images, electronic health records (EHRs), genetic information, lifestyle data, and patient-reported outcomes. Developing ML models capable of effectively integrating and utilizing multi-modal data is challenging but holds great potential for improving prediction accuracy.

CHAPTER 3

METHODOLOGY

This proposed system aims to address the open problems in disease prediction using ML and AI while ensuring operational feasibility, scalability, privacy, and clinical relevance. Through interdisciplinary collaboration and rigorous validation, IDPS has the potential to improve healthcare outcomes, enhance patient care, and advance medical research and innovation.

3.1 Flow Chart

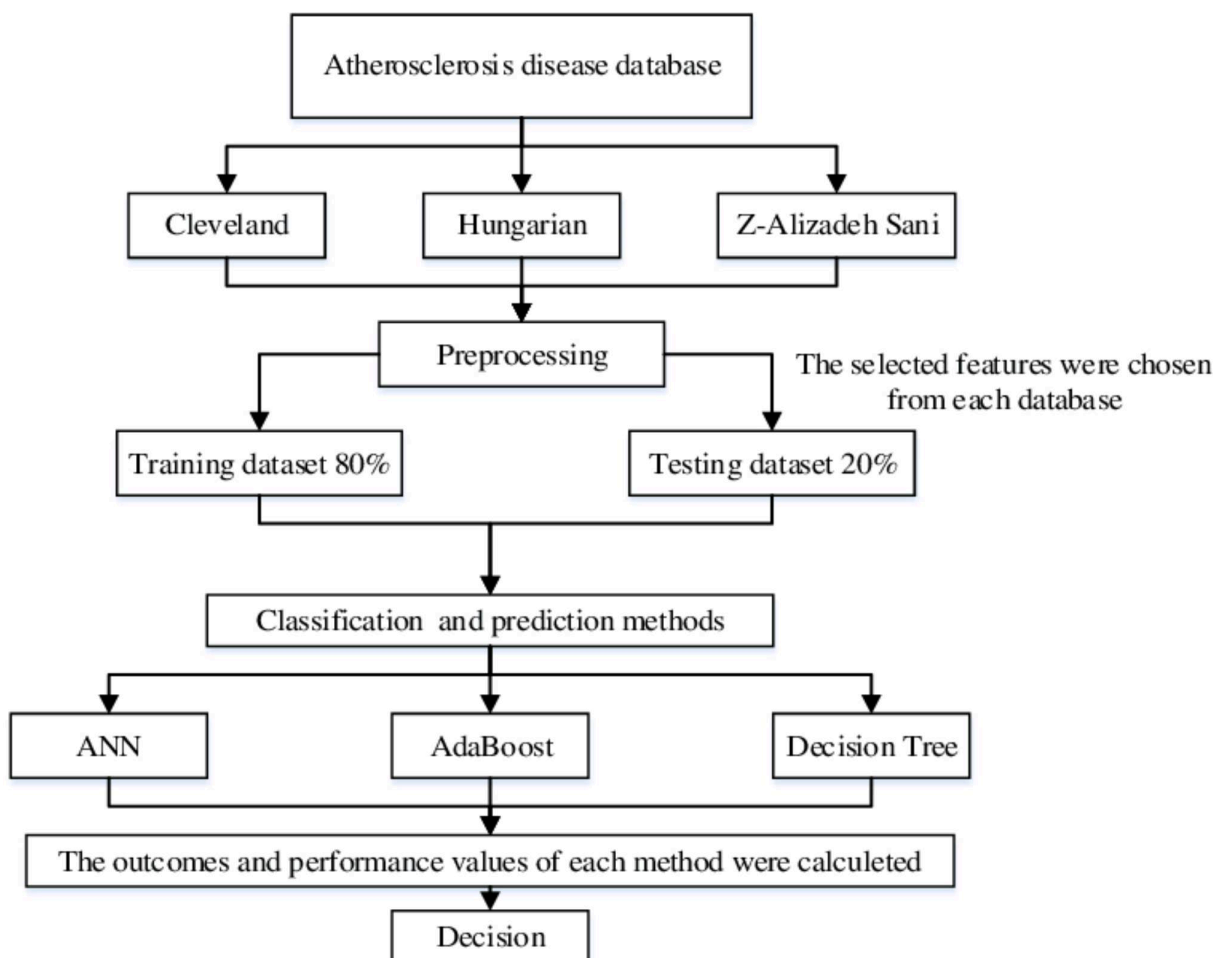


Fig 3.1: Decision Tree

3.2 Data Collection and Preprocessing

Data collection and preprocessing are crucial steps in developing a disease prediction system using machine learning. Here's a detailed explanation of each step.

3.2.1 Data Collection

Determine where to obtain the necessary data. This could include electronic health records (EHRs), medical databases, research studies, or publicly available datasets. Gather comprehensive datasets containing patient information, including demographics (age, gender, ethnicity), medical history (previous diagnoses, treatments), lifestyle factors (diet, exercise), clinical measurements (blood pressure, cholesterol levels), and any other relevant features for disease prediction. Verify the integrity and quality of the collected data by checking for completeness, consistency, and accuracy. Address any issues such as missing values or outliers.

3.2.2 Data Preprocessing

Decide on strategies for dealing with missing data, such as imputation (replacing missing values with estimated values based on other observations), deletion of missing values, or using algorithms that can handle missing values directly. Normalize numerical features to a similar scale to prevent certain features from dominating the model's learning process. Common scaling techniques include min-max scaling or standardization. Convert categorical variables into numerical representations that machine learning algorithms can process. This could involve techniques such as one-hot encoding or label encoding. Detect and address outliers in the data that could skew the model's predictions. Identify relevant features that have a significant impact on disease prediction. This can involve statistical methods like correlation analysis, feature importance scores from ensemble models, or domain knowledge.

3.3 Selected Methodology

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised

learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model. Before we dive deep, let's get familiar with some of the terminologies:

- **Instances:** Refer to the vector of features or attributes that define the input space
- **Attribute:** A quantity describing an instance
- **Concept:** The function that maps input to output
- **Target Concept:** The function that we are trying to find, i.e., the actual answer
- **Hypothesis Class:** Set of all the possible functions
- **Sample:** A set of inputs paired with a label, which is the correct output (also known as the Training Set)
- **Candidate Concept:** A concept which we think is the target concept
- **Testing Set:** Similar to the training set and is used to test the candidate concept and determine its performance.

3.4 Architecture of Proposed System

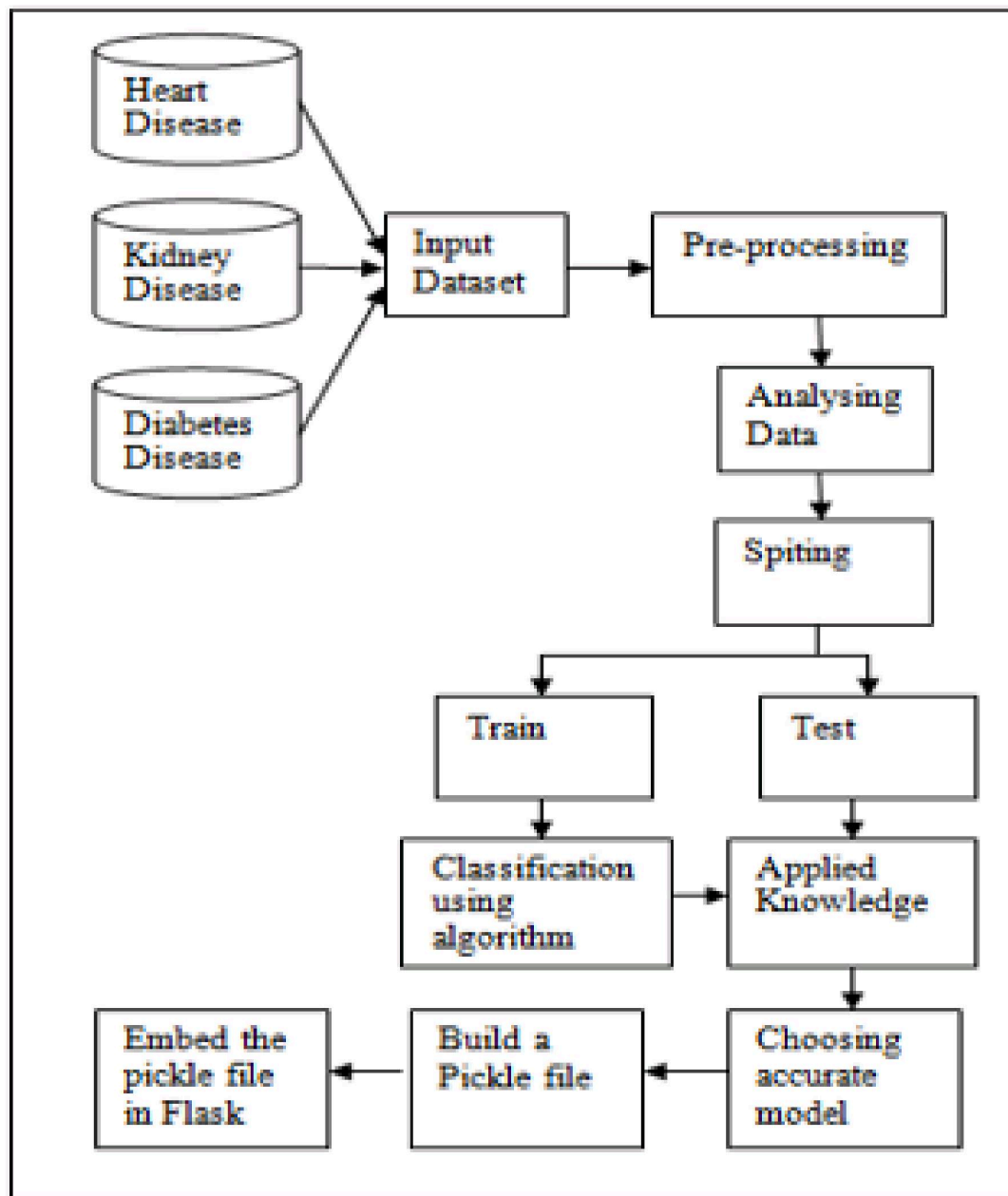


Fig 3.2: Block Diagram

3.5 Algorithms

In disease prediction using machine learning, several algorithms can be employed depending on the nature of the data, the complexity of the problem, and the specific disease being predicted. Here are some commonly used algorithms.

3.5.1 Gradient Boosting Machines (GBM)

Similar to Random Forests, GBM is an ensemble learning technique that builds multiple decision trees sequentially, each focusing on correcting the errors of its predecessor. Can achieve high predictive accuracy and is robust to outliers and noisy data. Requires careful tuning of hyper parameters and may be computationally intensive. Simple to implement and understand, but may be sensitive to the choice of distance metric and the value of k .

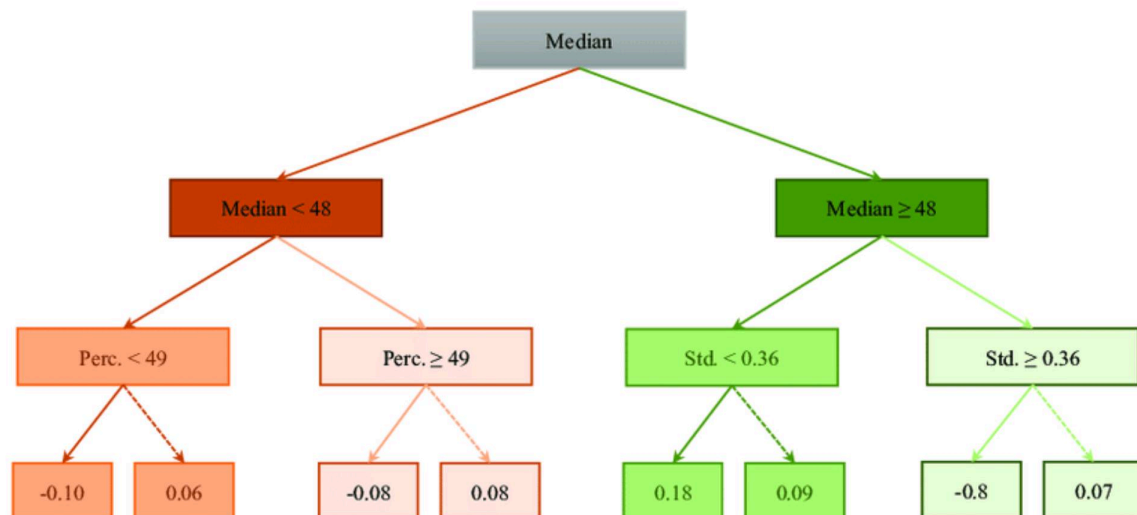


Fig 3.3: Gradient Boosting Machines (GBM)

3.5.2 Convolutional Neural Network (CNN)

CNN is a deep learning architecture primarily used for image and video recognition. CNNs use a series of convolutional and pooling layers to extract and learn features from input images, followed by fully connected layers for classification. The convolutional layer applies a filter or kernel to the input image, sliding it across the image and computing a dot product at each position.

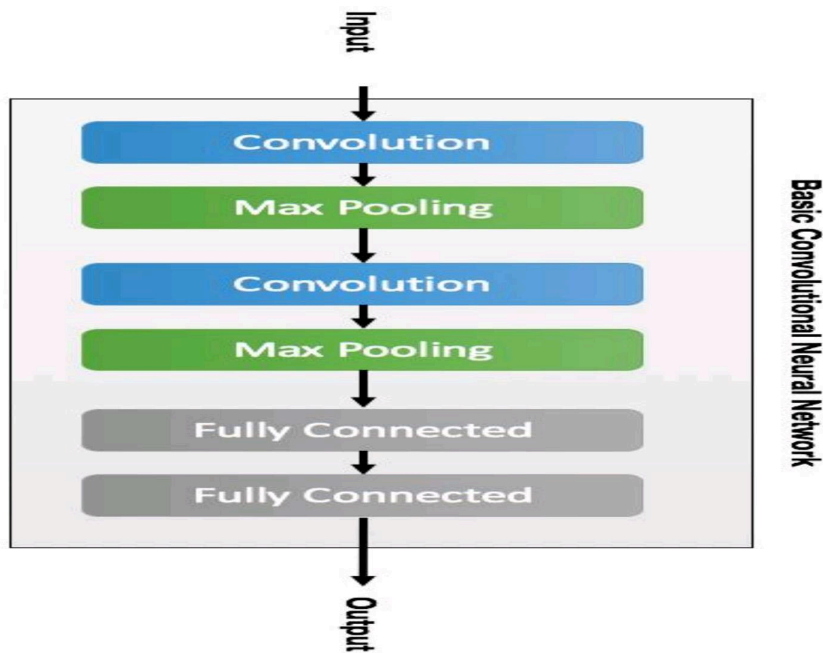


Fig 3.4: Convolutional Neural Network (CNN)

3.5.3 K- Nearest Neighbors (KNN)

Imported necessary libraries from scikit-learn for KNN, data splitting, and evaluation. Split your data into training and testing sets using `train_test_split`. Defined your feature vectors X and corresponding class labels y . Initialized a KNN classifier with the desired number of neighbors (here, 5 neighbors). Trained the classifier on the training data using `fit`. Made predictions on the testing data using `predict`. Evaluated the accuracy of the model using `accuracy score` from scikit-learn.

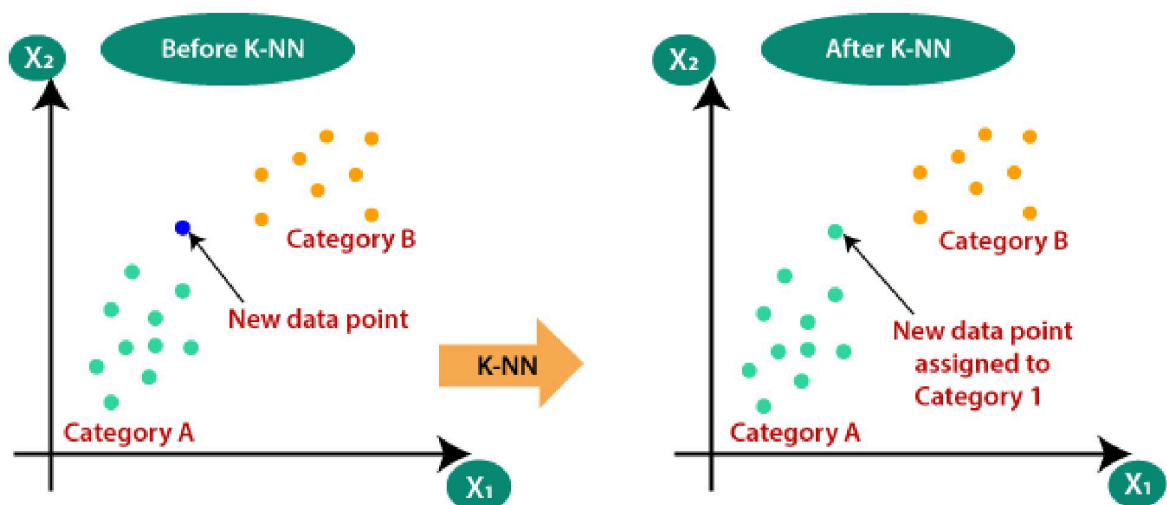


Fig 3.5: K- Nearest Neighbors (KNN)

3.7 Description of Software for Implementation

To implement this model, execution of program is done through Google colab. Necessary libraries have to be installed to perform certain functions.

Description of Dataset

In the present study, groundwater samples are collected following random sampling technique and the sampling stations are chosen in a near-grid pattern. Samples were collected at 44 locations in the study area representing pre monsoon and post-monsoon periods in order to evaluate the variations in the physic-chemical parameters of groundwater due to seasonal impact. The dataset consists of Latitude, Longitude points and the data points of 8 heavy metal in groundwater namely Zinc, Lead, Manganese, Nickel, Cobalt, Iron, Copper and Chromium each of 44 values.

Open Your Project in VS Code:

Open VS Code and navigate to your Flask project directory using the File menu or by dragging the folder onto the VS Code window.

Create a Virtual Environment (Optional but Recommended):

It's a good practice to use a virtual environment for your Flask project. Open the integrated terminal in VS Code (`Ctrl+``) and create a virtual environment:

```
python -m venv venv
```

Activate the virtual environment:

On Windows:

```
.\venv\Scripts\activate
```

On macOS/Linux:

```
bash
```

```
source venv/bin/activate
```

Run Your Flask Application:

In the terminal within VS Code, run your Flask application:

```
python app.py
```

You should see output indicating that your Flask application is running.

Access Your Flask App:

Open a web browser and navigate to `http://localhost:5000` (or `http://127.0.0.1:5000`). You should see the message "Hello, World!" displayed in the browser

You can set breakpoints in your Flask application code and debug it using VS Code's built-in debugger. Simply set breakpoints by clicking in the margin next to the line numbers, then launch the debugger (F5) and interact with your Flask app as usual. That's it! You're now using Flask in Visual Studio Code. You can continue to expand your Flask project.

3.8 Financial Report on Estimated Costing

Incorporating a software financial plan into the development of a disease prediction system involves estimating the costs associated with various aspects of the project and projecting potential revenue streams. Here's a breakdown of the key components of the financial plan.

3.8.1 Development Costs

Estimate the salaries or hourly rates of developers, data scientists, domain experts, and project managers involved in the development process. Include costs for purchasing or licensing software tools, libraries, and frameworks necessary for data preprocessing, modeling, and deployment. Estimate costs for computing resources, including servers, cloud services, and storage, required for data processing, model training, and deployment. Allocate funds for training team members on relevant technologies, machine learning algorithms, and healthcare domain knowledge. Set aside a portion of the budget for unexpected expenses or changes in project scope.

3.8.2 Operational Costs

Estimate ongoing costs for maintaining and supporting the software system, including bug fixes, updates, and user support. Include recurring expenses for cloud services, server maintenance, and data storage. Allocate funds for implementing security measures to protect patient data and comply with healthcare regulations.

3.8.3 Marketing and Sales

Allocate funds for market research to identify potential customers, competitors, and market trends. Budget for marketing activities such as advertising, content creation, and promotional events to raise awareness of the software solution. Estimate costs for hiring sales representatives or contracting with sales agencies to promote the software and secure partnerships with healthcare providers. Allocate funds for activities aimed at acquiring new customers, such as lead generation, sales demos, and customer outreach.

3.8.4 Revenue Projections

If the disease prediction system is offered as a subscription-based service, estimate potential revenue based on subscription fees charged to healthcare providers or individual users. If the software is sold under a licensing model, project revenue based on one-time licensing fees charged to healthcare organizations or institutions. Consider offering consulting services or customization options to generate additional revenue streams. Explore opportunities for forming partnerships with healthcare providers, research institutions, or pharmaceutical companies to license the software or collaborate on research projects.

3.8.5 Financial Projections

Develop a revenue forecast based on projected sales, subscription renewals, and other revenue sources over a specified period (e.g., one to five years). Estimate total development and operational costs, including salaries, infrastructure, marketing, and sales expenses, over the same period. Calculate the projected profitability of the software solution by subtracting total costs from total revenue and analyzing profitability metrics such as net profit margin and return on investment (ROI).

3.8.6 Risk Management

Identify potential risks and uncertainties that could impact the financial viability of the project, such as regulatory changes, market competition, or technological disruptions. Develop contingency plans and mitigation strategies to address these risks and ensure the long-term sustainability of the software solution.

RESULTS AND DISCUSSION

4.1 Development and Deployment Setup

4.1.1 Development Environment Setup

Select appropriate tools and frameworks for software development, such as IDEs (Integrated Development Environments) like PyCharm, Visual Studio Code, or Jupyter Notebooks for Python-based development. Utilize version control systems like Git to manage code changes, collaborate with team members, and track project history. Set up development environments with necessary dependencies and libraries for machine learning, such as Python packages (scikit-learn, TensorFlow, PyTorch), database systems (e.g., PostgreSQL, MySQL), and other supporting libraries.

4.1.2 Data Acquisition and Pre-processing

Implement scripts or modules to collect data from various sources, such as APIs, databases, or file systems.

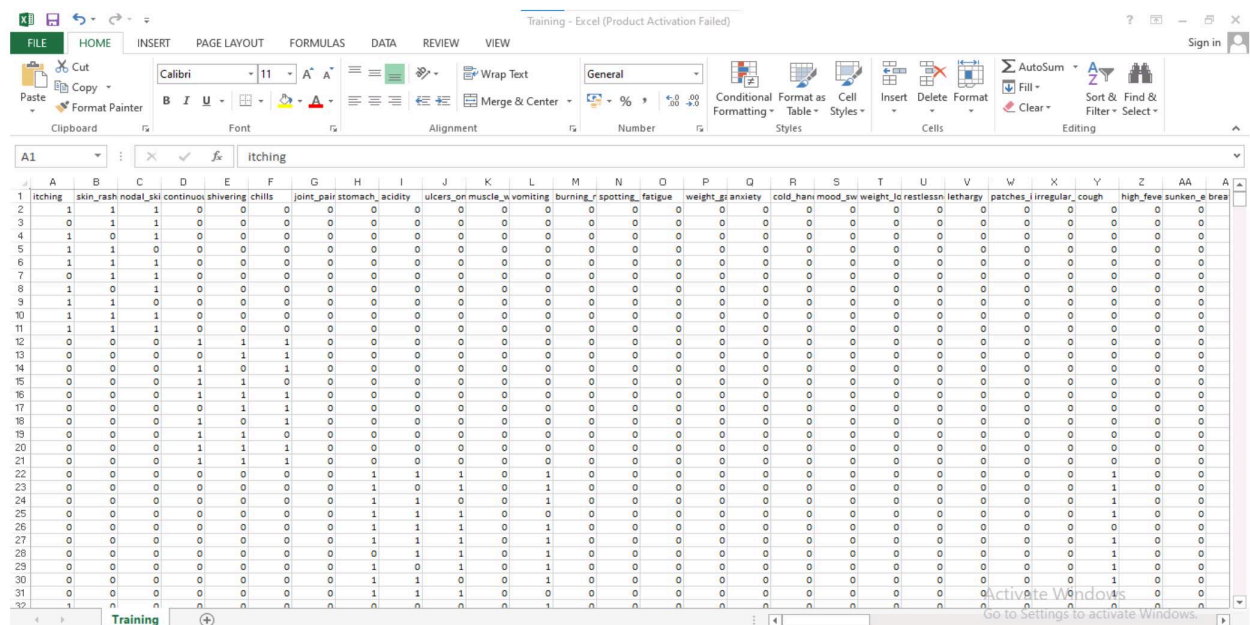


Fig 4.1: Collecting Dataset

Develop pipelines for automated data pre-processing, including handling missing values, feature scaling, encoding categorical variables, and other data cleaning

tasks. Implement mechanisms to validate data integrity, consistency, and quality throughout the pre-processing pipeline.

4.1.3 Model Development and Evaluation

Research and select appropriate machine learning algorithms based on the nature of the data and the specific disease prediction task. Develop scripts or workflows to train machine learning models using label datasets, optimizing hyper parameters and model architectures as necessary.

Fig 4.2: Training Dataset Model

Implement evaluation metrics and techniques to assess the performance of trained models, such as cross-validation, confusion matrices, ROC curves, and precision-recall curves.

4.1.4 Integration and Testing

Integrate trained machine learning models into the backend infrastructure of the software application, ensuring compatibility and efficient data processing. Develop user interfaces (UIs) and visualization components to interact with the prediction models, providing input features and displaying prediction results.

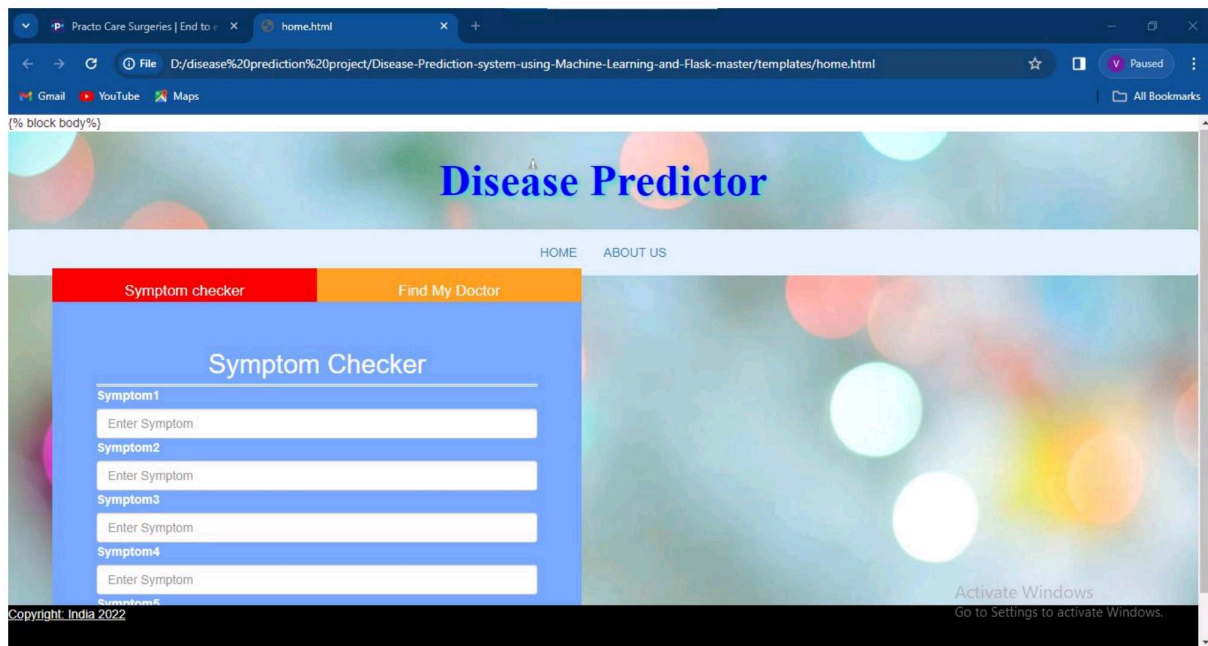


Fig 4.3: Home Page

Write and execute unit tests to verify the functionality of individual components and modules, ensuring code correctness and reliability. Perform end-to-end testing to validate the interaction between different system components, including data input/output, model inference, and user interactions.

4.1.5 Deployment Setup

Choose a cloud service provider (e.g., AWS, Google Cloud, Microsoft Azure) for deploying the software application, considering factors like scalability, reliability, and compliance requirements. Containerize the application using tools like Docker to package dependencies and ensure consistent deployment across different environments. Utilize container orchestration platforms like Kubernetes to manage and scale containerized applications efficiently. Implement CI/CD pipelines to automate the build, testing, and deployment processes, ensuring rapid and reliable delivery of updates to production environments.

4.1.6 Monitoring and Maintenance:

Set up logging and monitoring solutions to track system performance, resource utilization, and user interactions, enabling proactive identification and resolution of issues. Establish procedures for routine maintenance tasks, including software updates, database migrations, and infrastructure scaling.

4.2 Testing

In the context of implementing Gaussian Naive Bayes classifier in Python, testing typically involves assessing the performance of the trained model on unseen data to evaluate its generalization ability. Here's how testing fits into the implementation: After splitting the dataset into training and testing sets using the `train_test_split` function, the testing phase begins. The testing set, which was not used during training, serves as a benchmark to assess the model's performance on unseen data. The Gaussian Naive Bayes classifier, initialized as naïve Bayes, is then trained on the training data using the `fit` method. This step involves learning the underlying patterns and distributions of the features in the training set.

The application of machine learning (ML) in disease prediction has revolutionized the field of healthcare, enabling early detection, personalized treatment plans, and improved patient outcomes. The outcomes and discussions around the use of machine learning for disease prediction can be broad and multifaceted, encompassing technical achievements, clinical impacts, ethical considerations, and future directions. ML models have achieved high accuracy in predicting various diseases, including cancer, diabetes, heart diseases, and neurodegenerative disorders. By analyzing vast datasets, these models can identify patterns and correlations that may not be immediately evident to human researchers or practitioners.

Machine learning algorithms have been successful in integrating different types of data (clinical records, imaging, genetic information, etc.), providing a holistic view of patient health and enhancing predictive capabilities.

4.2.1 Clinical Impacts

ML facilitates personalized medicine by predicting disease risk and treatment outcomes based on individual patient data. This tailored approach improves treatment effectiveness and minimizes side effects. By accurately predicting diseases, healthcare systems can better allocate resources, prioritize high-risk patients, and reduce unnecessary treatments or tests. Wearable's and mobile health technologies integrated with ML algorithms enable continuous monitoring of patients, predicting. Exacerbations or complications in chronic diseases.

4.2.2 Ethical Considerations

The use of sensitive health data raises significant privacy concerns. Ensuring the security of patient data and adhering to data protection regulations is paramount. ML models can perpetuate or even exacerbate biases present in the training data, leading to disparities in disease prediction across different populations. Efforts are needed to develop unbiased algorithms and validate them across diverse cohorts. The "black box" nature of some ML models can hinder their acceptance among healthcare providers. Increasing the transparency and interpretability of these models is crucial for their adoption in clinical practice.

4.2.3 Future Directions

Seamless integration of ML models with EHR systems can enhance real-time decision-making and patient care. The development of more sophisticated ML algorithms, including deep learning and reinforcement learning, promises to further improve disease prediction and management. Adapting and implementing ML-based disease prediction tools in low-resource settings can address global health disparities, provided challenges related to data availability and infrastructure are overcome. The future of disease prediction using ML will benefit from closer collaboration between computer scientists, biostatisticians, clinicians, and epidemiologists. The integration of machine learning (ML) in disease prediction, despite its potential, faces several constraints. These challenges span technical, ethical, social, and practical domains.

4.3 Tradeoff in the Project

Incorporating machine learning (ML) into disease prediction involves navigating a series of trade-offs. These trade-offs reflect the balance between technological capabilities and the limitations, ethical considerations, and practical constraints of deploying ML in healthcare settings. Understanding these trade-offs is crucial for developers, healthcare professionals, and policymakers to make informed decisions about the design, implementation, and use of ML tools in disease prediction.

4.3.1 Accuracy vs. Interpretability

One of the fundamental trade-offs in ML is between model accuracy and interpretability. Highly complex models, such as deep neural networks, often

achieve superior performance in terms of accuracy but are usually considered "black boxes" due to their complexity, making it hard to understand how they arrived at a particular prediction. On the other hand, simpler models like decision trees or linear regression are more interpretable but might not capture the nuances in the data as well, potentially leading to lower accuracy.

4.3.2 Data Privacy vs. Utility

The use of extensive healthcare data can significantly improve the performance of ML models in disease prediction. This raises significant privacy concerns, as sensitive health information must be protected. The trade-off here involves balancing the utility of the data for improving health outcomes with the need to safeguard patient privacy and comply with data protection regulations.

4.3.3 Generalization vs. Specialization

ML models trained on vast, diverse datasets may excel at generalizing across various contexts but might not perform as well in specific subpopulations or rare conditions due to underrepresentation in the training data. Conversely, models trained on highly specialized or niche datasets can become very effective in those specific areas but lose their applicability to the broader population. Balancing generalization and specialization is crucial for developing ML applications that are both widely applicable and effective in particular scenarios.

4.3.4 Innovation vs. Regulation

The rapid pace of innovation in ML and healthcare technology often outstrips the slower, deliberative process of regulation. While innovation drives advancements and improvements in disease prediction, regulation is necessary to ensure safety, efficacy, and ethical considerations are adequately addressed. The trade-off involves fostering an environment that encourages innovation while ensuring robust regulatory frameworks are in place to protect patients and healthcare systems.

4.3.5 Cost vs. Benefit

Implementing ML solutions in healthcare can be expensive, requiring investments in data infrastructure, technology, training, and ongoing maintenance. The trade-off between the initial and ongoing costs against the potential benefits, such as improved patient outcomes, efficiency, and reduced healthcare costs over time, is a

significant consideration for healthcare providers and systems.

4.3.6 Short-term vs. Long-term Outcomes

Decisions around the implementation of ML in disease prediction also involve weighing short-term benefits, such as immediate improvements in diagnostic accuracy or efficiency, against potential long-term outcomes, including the impact on patient care practices, healthcare system adaptation, and possible unforeseen consequences of widespread ML adoption. Navigating these trade-offs requires careful consideration and collaboration across disciplines, including healthcare, technology, ethics, and policy.

CHAPTER 5

CONCLUSION

5.1 Conclusion

The integration of machine learning (ML) into disease prediction represents a significant advancement in healthcare, offering the potential to transform patient outcomes through personalized treatment, early detection, and efficient resource allocation. However, this integration is not without its challenges. It necessitates navigating complex trade-offs between accuracy and interpretability, data privacy and utility, innovation and regulation, among others. These trade-offs underscore the importance of a balanced approach that considers not just the technological capabilities but also the ethical, legal, and social implications of deploying ML in healthcare settings.

To harness the full potential of ML in disease prediction while mitigating its risks, a multidisciplinary effort is essential. Collaboration across fields—combining the expertise of healthcare professionals, computer scientists, ethicists, and policymakers—is crucial for developing ML solutions that are not only technically advanced but also ethically sound, equitable, and aligned with patient care values. Such collaboration can also foster the development of standards and guidelines that address data quality, model interpretability, privacy concerns, and bias mitigation. In the future, the model can be used in various sectors and can enhance efficiency by considering more symptoms to predict disease. The model can be used for providing an enhanced, more accurate framework that would lead to a better human disease prediction model. Explore and implement more advanced machine learning models and algorithms for disease prediction, such as deep learning models (e.g., convolutional neural networks, recurrent neural networks) or ensemble methods. Experiment with feature engineering techniques and hyper parameter optimization to improve model performance.

5.2 Future Works

For advancing disease prediction in sentences, several future directions can be explored to enhance accuracy, efficiency, and applicability. Here are some potential areas for further research and development:

Explore advanced natural language processing (NLP) techniques like contextual embeddings (e.g., ELMo, GPT, BERT) to capture richer semantic meanings and contextual nuances within sentences. Integrate additional sources of information such as images, electronic health records (EHRs), genomic data, and patient demographics to enrich the predictive power of the model. Move beyond binary or coarse-grained disease classification and develop models capable of predicting specific subtypes or stages of diseases based on sentence-level information. Investigate how disease prediction models can account for temporal dynamics by considering the progression of symptoms over time and incorporating time-series data from longitudinal studies or patient records. Develop techniques to enhance the interpretability of disease prediction models, allowing clinicians to understand the underlying factors contributing to predictions and providing transparent reasoning for model decisions. Explore transfer learning approaches to adapt pre-trained models to the domain of medical text, leveraging large-scale datasets from related tasks to improve performance with limited labeled data. Investigate strategies to intelligently select informative samples for annotation, reducing the annotation burden and improving model performance over time through active learning and semi-supervised learning approaches.

REFERENCES

- [1] Gavhane, G. Kokkula, I. Pandya, and K. Devadkar, "Prediction of heart disease using machine learning," in 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1275–1278.
- [2] Y. Hasija, N. Garg, and S. Sourav, "Automated detection of dermatological disorders through image-processing and machine learning," in 2017 International Conference on Intelligent Sustainable Systems (ICISS), 2017, pp. 1047–1051.
- [3] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, pp. 1–16, 2019.
- [4] R. Katarya and P. Srinivas, "Predicting heart disease at early stages using machine learning: A survey," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 302–305.
- [5] P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1–4.
- [6] M. Patil, V. B. Lobo, P. Puranik, A. Pawaskar, A. Pai, and R. Mishra, "A proposed model for lifestyle disease prediction using support vector machine," in 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018, pp. 1–6.
- [7] F. Q. Yuan, "Critical issues of applying machine learning to condition monitoring for failure diagnosis," in 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2016, pp. 1903–1907.
- [8] S. Ismaeel, A. Miri, and D. Chourishi, "Using the extreme learning machine (elm) technique for heart disease diagnosis," in 2015 IEEE Canada International Humanitarian Technology Conference (IHTC2015), 2015, pp. 1–3.

APPENDIX

A . SOURCE CODE

```
import csv
from flask import Flask, render_template,request,redirect,url_for
import diseaseprediction
#import MySQLdb
app = Flask(__name__)
#conn =
MySQLdb.connect(host='localhost',user='root',password='',db='disease_database')

with open('templates/Testing.csv', newline='') as f:
    reader = csv.reader(f)
    symptoms = next(reader)
    symptoms = symptoms[:len(symptoms)-1]
@app.route('/', methods=['GET'])
def dropdown():
    return render_template('includes/default.html', symptoms=symptoms)

@app.route('/disease_predict', methods=['POST'])
def disease_predict():
    selected_symptoms = []
    if(request.form['Symptom1']!="") and (request.form['Symptom1'] not in
selected_symptoms):
        selected_symptoms.append(request.form['Symptom1'])
    if(request.form['Symptom2']!="") and (request.form['Symptom2'] not in
selected_symptoms):
        selected_symptoms.append(request.form['Symptom2'])
    if(request.form['Symptom3']!="") and (request.form['Symptom3'] not in
selected_symptoms):
        selected_symptoms.append(request.form['Symptom3'])
    if(request.form['Symptom4']!="") and (request.form['Symptom4'] not in
selected_symptoms):
        selected_symptoms.append(request.form['Symptom4'])
    if(request.form['Symptom5']!="") and (request.form['Symptom5'] not in
selected_symptoms):
        selected_symptoms.append(request.form['Symptom5'])

    # disease_list = []
    # for i in range(7):
    #     disease = diseaseprediction.dosomething(selected_symptoms)
    #     disease_list.append(disease)
    # return
    render_template('disease_predict.html',disease_list=disease_list)
    disease = diseaseprediction.dosomething(selected_symptoms)
    return
    render_template('disease_predict.html',disease=disease,symptoms=symptoms)
```



```

# @app.route('/default')
# def default():
#     return render_template('includes/default.html')

@app.route('/find_doctor', methods=['POST'])
def get_location():
    location = request.form['doctor']
    return
render_template('find_doctor.html',location=location,symptoms=symptoms)

@app.route('/drug', methods=['POST'])
def drugs():
    medicine = request.form['medicine']
    return
render_template('homepage.html',medicine=medicine,symptoms=symptoms)

if __name__ == '__main__':
    app.run(debug=True)

```

diseaseprediction.py

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import csv,numpy as np,pandas as pd
import os

data = pd.read_csv(os.path.join("templates", "Training.csv"))
df = pd.DataFrame(data)
cols = df.columns
cols = cols[:-1]
x = df[cols]
y = df['prognosis']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
random_state=42)

print ("DecisionTree")
dt = DecisionTreeClassifier()
clf_dt=dt.fit(x_train,y_train)
#print ("Accuracy: ", clf_dt.score(x_test,y_test))

# with open('templates/Testing.csv', newline='') as f:
#     reader = csv.reader(f)
#     symptoms = next(reader)
#     symptoms = symptoms[:len(symptoms)-1]

indices = [i for i in range(132)]
symptoms = df.columns.values[:-1]

dictionary = dict(zip(symptoms,indices))

```

```

def dosomething(symptom):
    user_input_symptoms = symptom
    user_input_label = [0 for i in range(132)]
    for i in user_input_symptoms:
        idx = dictionary[i]
        user_input_label[idx] = 1

    user_input_label = np.array(user_input_label)
    user_input_label = user_input_label.reshape((-1,1)).transpose()
    return(dt.predict(user_input_label))

#
print(dosomething(['headache','muscle_weakness','puffy_face_and_eyes','mild_
fever','skin_rash']))
# prediction = []
# for i in range(7):
#     pred = dosomething(['headache'])
#     prediction.append(pred)
# print(prediction)

```

Default.html

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiIiSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

    <!-- Optional theme -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp"
crossorigin="anonymous">

    <!-- Latest compiled and minified JavaScript -->
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnPcJA712mCWNIpG9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>

    <style type="text/css">

```

```

    body, html {
      height: 100%;
    }
    .bg {
      /* The image used */
      background-image: url("https://img.freepik.com/free-photo/female-doctors-checking-results-digital-tablet_329181-9236.jpg?w=1060");

      /* Full height */
      height: 150%;

      /* Center and scale the image nicely */
      background-position: center;
      background-repeat: no-repeat;
      background-size: cover;
    }

    .head {
      color: white;
      font-family: sans-serif;
      font-size: 30px;
      border-bottom: double;
      text-align: center;
    }

    .nav.navbar-nav {
      display: flex;
      justify-content: center;
      flex-wrap: wrap;
    }
    @media (min-width: 768px) {
      .navbar-nav {
        float: none;
      }
    }

    .tablink {
      background-color: #FEA127;
      color: white;
      float: left;
      border: none;
      outline: none;
      cursor: pointer;
      padding: 14px 16px;
      font-size: 17px;
      width: 25%;
    }

    .tablink:hover {

```

```

        background-color: green;
    }

    /* Style the tab content */
    .tabcontent {
        color: white;
        display: none;
        padding: 50px;
        width: 855px;
        position: relative;
        height: 350px;
    }

    #footer{background-color: black;
text-decoration: underline;
text-align: center;
font-family: sans-serif;
text-decoration-color: white;
color: white;
align-self: bottom;
position:relative;
bottom: 0px;
left: 0px;
width: 100%;
height: 50px;}
#Symptom {background-color:#78A9FD; width: 600px; top:45px ;left:
50px;height:450px;}
#Doctor {background-color: #78A9FD; width: 600px;top:45px; left: 50px}
#Drug {background-color: #78A9FD; width: 600px;top:45px ;left: 50px}

</style>

</head>
<body>
    {% block body%}
    <div class="bg">
        <br>
        <div>
            <center>
            <span style="font-size:50px; text-shadow: 3px 3px 6px #40e0d0;font-
weight: bolder;font-family:'Garamound';color:blue"> HEALTH
CHECKUP</span></center>
        </div>
        <br>
        <nav class="navbar navbar-light" style="background-color: #e3f2fd;">

```



```

        <div class="container-fluid">
            <ul class="nav navbar-nav navbar-center">
                <li class=""><a href="/">HOSPITAL</a></li>
                <!-- <li class=""><a href="#">ABOUT US</a></li> -->
            </ul>
        </div>
    </nav>
    <div>
        <button class="tablink" onclick="openTab('Symptom', this, 'red')"
id="defaultOpen" style="width: 300px; position: absolute; top: 175px;
left:50px; ">Symptom checker</button>
        <button class="tablink" onclick="openTab('Doctor', this, 'red')"
id="doctortab" style="width: 300px; position: absolute; top: 175px;
left:350px; ">Find My Doctor</button>
        <!-- <button class="tablink" onclick="openTab('Drug', this, 'red')"
id="drugdetail" style="width: 200px; position: absolute; top: 175px;
left:450px; ">Know My Drug</button> -->
    </div>

<div id="Symptom" class="tabcontent">
    <form name="SymptomForm" id="SymForm" method="POST"
action=/disease_predict>
        <div class="head">
            Disease Checker
        </div>
        <div class="form-group">
            <!-- <select multiple data-role="tagsinput" class="form-control"
size=1>
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </select> -->
            <label for="Symptom1" align="left">Symptom1</label>
            <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom1">
            <datalist id="symp">
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </datalist>

            <label for="Symptom2" align="left">Symptom2</label>
            <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom2">
            <datalist id="symp">
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </datalist>

```

```

        <label for="Symptom3" align="left">Symptom3</label>
        <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom3">
        <datalist id="symp">
            {% for symp in symptoms %}
            <option value= "{{symp}}">{{symp}}</option>
            {% endfor %}
        </datalist>

        <label for="Symptom4" align="left">Symptom4</label>
        <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom4">
        <datalist id="symp">
            {% for symp in symptoms %}
            <option value= "{{symp}}">{{symp}}</option>
            {% endfor %}
        </datalist>

        <label for="Symptom5" align="left">Symptom5</label>
        <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom5">
        <datalist id="symp">
            {% for symp in symptoms %}
            <option value= "{{symp}}">{{symp}}</option>
            {% endfor %}
        </datalist>

    </div>

    <button type="submit" class="btn btn-primary" style="padding: 10px;"
name=form>Check</button>
</form>
</div>

<div id="Doctor" class="tabcontent">
    <div class="head">
        Find Doctor Nearby
    </div>
    <form name="Doctor_search" id="DoctorForm" method="POST"
action=/find_doctor>
    <input type="text" class="form-control" name="doctor"><br>
    <button type="submit" class="btn btn-primary" style="padding: 10px;"
name="search_doctor">Search</button>
<H3>DOCTOR APPOINTMENT :</H3>
<ol>
    <li><button onclick="location.href='https://www.practo.com/'" type="button"
class="btn btn-primary" style="padding: 10px;">
        PRACTO</button></li>

```

```

        &nbsp;
<li><button onclick="location.href='https://www.lybrate.com/'" type="button"
class="btn btn-primary" style="padding: 10px;">
    LYBRATE</button></li>
</ol>
</form>
</div>

<!-- <div id="Drug" class="tabcontent">
    <form name="MedicineForm" id="medForm" method="POST">
        <div class="head">
            Know Your Drug
        </div>
        <div class="form-group">
            <label for="medicine" align="left">Name</label>
            <input type="text" class="form-control" id="medicine"
placeholder="medicine name" name="medicine">
            <br>
            <button class="btn btn-primary" type="submit" method='POST'>Know about
it</button>
        </div>
    </form>
<script type="text/javascript">
    function showResult(argument) {
        // body...
        document.getElementById(argument).style.color = "white";
    }
</script>
</div> -->

</div>
<div id="footer" >
    <p>Copyright: India 2022</p>
</div>
</div>
<script>

function openTab(tabName,elmnt,color) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablink");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].style.backgroundColor = "";
    }
}

```



```

        document.getElementById(tabName).style.display = "block";
        elmnt.style.backgroundColor = color;
    }

    // Get the element with id="defaultOpen" and click on it
    document.getElementById("defaultOpen").click();

</script>

    {% endblock %}
</body>
</html>

```

Home.html

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiISIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

    <!-- Optional theme -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp"
crossorigin="anonymous">

    <!-- Latest compiled and minified JavaScript -->
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpPD7Txa"
crossorigin="anonymous"></script>

    <style type="text/css">
      body, html {
        height: 100%;
      }
      .bg {
        /* The image used */
        background-image: url("https://images.pexels.com/photos/255379/pexels-
photo-255379.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260");

```

```

    /* Full height */
    height: 100%;

    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
}

.head {
    color: white;
    font-family: sans-serif;
    font-size: 30px;
    border-bottom: double;
    text-align: center;
    padding-top: -200px;
}

.nav.navbar-nav {
    display: flex;
    justify-content: center;
    flex-wrap: wrap;
}

@media (min-width: 768px) {
    .navbar-nav {
        float: none;
    }
}

.tablink {
    background-color: #FEA127;
    color: white;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    font-size: 17px;
    width: 25%;
}

.tablink:hover {
    background-color: green;
}

/* Style the tab content */
.tabcontent {
    color: white;

```

```

display: none;
padding: 50px;
width: 855px;
position: relative;
height: 350px;
}

#footer{background-color: black;
text-decoration: underline;
text-align: center;
font-family: sans-serif;
text-decoration-color: white;
color: white;
align-self: bottom;
display: flex;
flex-wrap: nowrap;
}
#Symptom {background-color:#78A9FD; width: 600px; top:10px ;left:
50px;height:450px; }
#Doctor {background-color: #78A9FD; width: 600px;top:45px; left: 50px}
#Drug {background-color: #78A9FD; width: 600px;top:45px ;left: 50px}

</style>

</head>
<body>
{% block body%}
<div class="bg">
<br>
<div>
<center>
<span style="font-size:50px; text-shadow: 3px 3px 6px #40e0d0;font-
weight: bolder;font-family:'Garamound';color:blue">Disease
Predictor</span></center>
</div>
<br>
<nav class="navbar navbar-light" style="background-color: #e3f2fd;">
<div class="container-fluid">
<ul class="nav navbar-nav navbar-center">
<li class=""><a href="/">HOME</a></li>
<li class=""><a href="#">ABOUT US</a></li>
</ul>
</div>
</nav>
</div>

```

```

        <button class="tablink" onclick="openTab('Symptom', this, 'red')"
id="defaultOpen" style="width: 300px; position: absolute; top: 175px;
left:50px; ">Symptom checker</button>
        <button class="tablink" onclick="openTab('Doctor', this, 'red')"
id="doctortab" style="width: 300px; position: absolute; top: 175px;
left:350px; ">Find My Doctor</button>
        <!-- <button class="tablink" onclick="openTab('Drug', this, 'red')"
style="width: 200px; position: absolute; top: 175px; left:450px; ">Know My
Drug</button> -->
</div>

<div id="Symptom" class="tabcontent">
    <form name="SymptomForm" id="SymForm" method="POST"
action=/disease_predict>
        <div class="head">
            Symptom Checker
        </div>
        <div class="form-group">
            <!-- <select multiple data-role="tagsinput" class="form-control"
size=1>
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </select> -->
            <label for="Symptom1" align="left">Symptom1</label>
            <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom1" placeholder="Enter Symptom">
            <datalist id="symp">
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </datalist>

            <label for="Symptom2" align="left">Symptom2</label>
            <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom2" placeholder="Enter Symptom">
            <datalist id="symp">
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}
            </datalist>

            <label for="Symptom3" align="left">Symptom3</label>
            <input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom3" placeholder="Enter Symptom">
            <datalist id="symp">
                {% for symp in symptoms %}
                <option value= "{{symp}}">{{symp}}</option>
                {% endfor %}

```



```

</datalist>

<label for="Symptom4" align="left">Symptom4</label>
<input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom4" placeholder="Enter Symptom">
<datalist id="symp">
    {% for symp in symptoms %}
        <option value= "{{symp}}">{{symp}}</option>
    {% endfor %}
</datalist>

<label for="Symptom5" align="left">Symptom5</label>
<input type="text" list="symp" multiple="multiple" class="form-
control" name="Symptom5" placeholder="Enter Symptom">
<datalist id="symp">
    {% for symp in symptoms %}
        <option value= "{{symp}}">{{symp}}</option>
    {% endfor %}
</datalist>

</div>

<button type="submit" class="btn btn-primary" style="padding: 10px;"
name=form>Check</button>
</form>
</div>

<div id="Doctor" class="tabcontent">
    <div class="head">
        Find Doctor<br>
    </div>
    <form name="Doctor_search" id="DoctorForm" method="POST"
action=/find_doctor>
    <input type="text" class="form-control" name="doctor"><br>
    <button type="submit" class="btn btn-primary" style="padding: 10px;"
name="search_doctor">Search</button>
    </form>
</div>

<!-- <div id="Drug" class="tabcontent">
    <form name="MedicineForm" id="medForm" method="POST">
        <div class="head">
            Know Your Drug
        </div>
        <div class="form-group">
            <label for="medicine" align="left">Name</label>
            <input type="text" class="form-control" id="medicine"
placeholder="medicine name" name="medicine">

```

```

        <br>
        <button class="btn btn-primary" type="submit" method='POST'>Know
about it</button>
    </div>
</form>
<script type="text/javascript">
    function showResult(argument) {
        // body...
        document.getElementById(argument).style.color = "white";
    }
</script>
</div> -->

</div>
<div id="footer" style="position: absolute; bottom: 0px; left: 0px; width:
100%; height: 50px;">
    <footer><p >Copyright: India 2022</p></footer>
</div>
</div>
<script>

function openTab(tabName,elmnt,color) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablink");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].style.backgroundColor = "";
    }
    document.getElementById(tabName).style.display = "block";
    elmnt.style.backgroundColor = color;
}

// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();

</script>

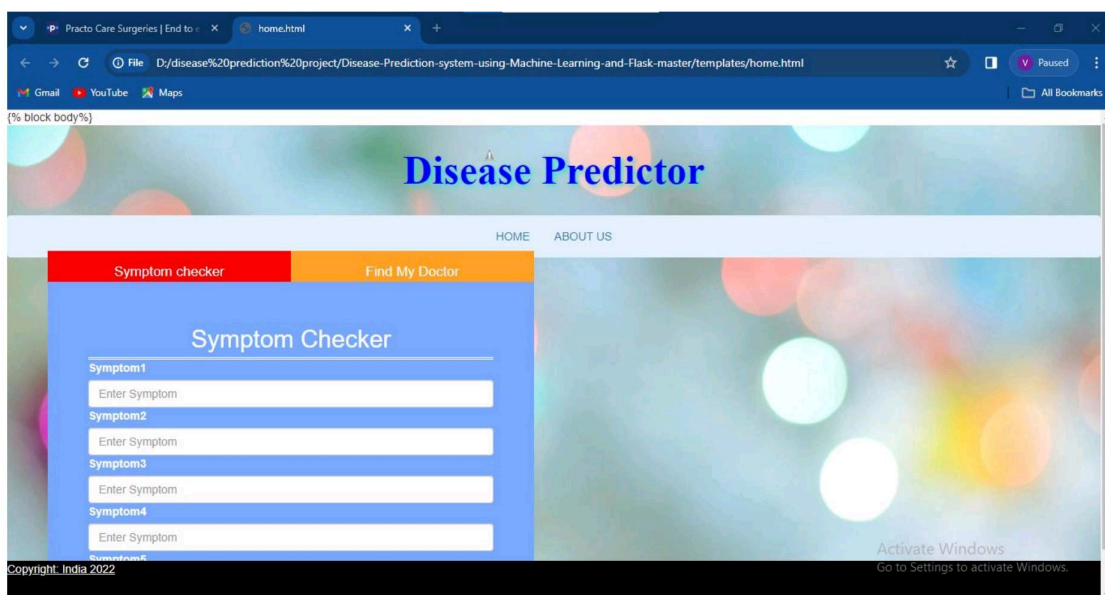
    {% endblock %}
</body>
</html>

```

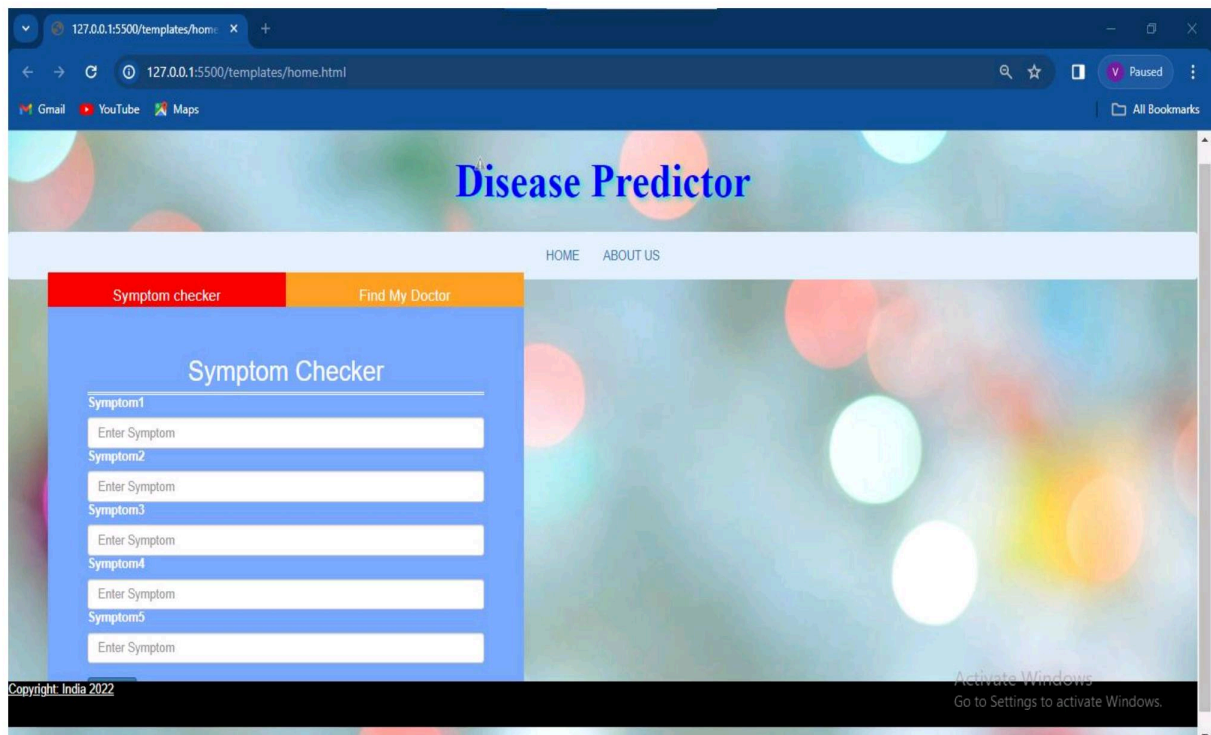

B. SCREENSHOT

```
D:\disease prediction project > Disease-Prediction-system-using-Machine-Learning-and-Flask-master > app.py > ...
1 import csv
2 from flask import Flask, render_template, request, redirect, url_for
3 import diseaseprediction
4 #import MySQLdb
5
6 app = Flask(__name__)
7 #conn = MySQLdb.connect(host='localhost',user='root',password='',db='disease_database')
8
9 with open('templates/Testing.csv', newline='') as f:
10     reader = csv.reader(f)
11     symptoms = next(reader)
12     symptoms = symptoms[:len(symptoms)-1]
13 @app.route('/', methods=['GET'])
14 def dropdown():
15     return render_template('includes/default.html', symptoms=symptoms)
16
17 @app.route('/disease_predict', methods=['POST'])
18 def disease_predict():
19     selected_symptoms = []
20     if(request.form['Symptom1'] != "") and (request.form['Symptom1'] not in selected_symptoms):
21         selected_symptoms.append(request.form['Symptom1'])
22     if(request.form['Symptom2'] != "") and (request.form['Symptom2'] not in selected_symptoms):
23         selected_symptoms.append(request.form['Symptom2'])
24     if(request.form['Symptom3'] != "") and (request.form['Symptom3'] not in selected_symptoms):
25         selected_symptoms.append(request.form['Symptom3'])
26     if(request.form['Symptom4'] != "") and (request.form['Symptom4'] not in selected_symptoms):
27         selected_symptoms.append(request.form['Symptom4'])
28     if(request.form['Symptom5'] != "") and (request.form['Symptom5'] not in selected_symptoms):
29         selected_symptoms.append(request.form['Symptom5'])
30
31     # disease_list = []
32     # for i in range(7):
33     #     disease = diseaseprediction.dosomething(selected_symptoms)
34     #     disease_list.append(disease)
35     # return render_template('disease_predict.html',disease_list=disease_list)
36     disease = diseaseprediction.dosomething(selected_symptoms)
37     return render_template('disease_predict.html',disease=disease,symptoms=symptoms)
38
39 # @app.route('/default')
40 # def default():
41 #     return render_template('includes/default.html')
42
43 @app.route('/find_doctor', methods=['POST'])
44 def get_location():
45     location = request.form['doctor']
46     return render_template('find_doctor.html',location=location,symptoms=symptoms)
47
48 @app.route('/drug', methods=['POST'])
49 def drugs():
50     medicine = request.form['medicine']
51     return render_template('homepage.html',medicine=medicine,symptoms=symptoms)
52
53 if __name__ == '__main__':
54     app.run(debug=True)
```

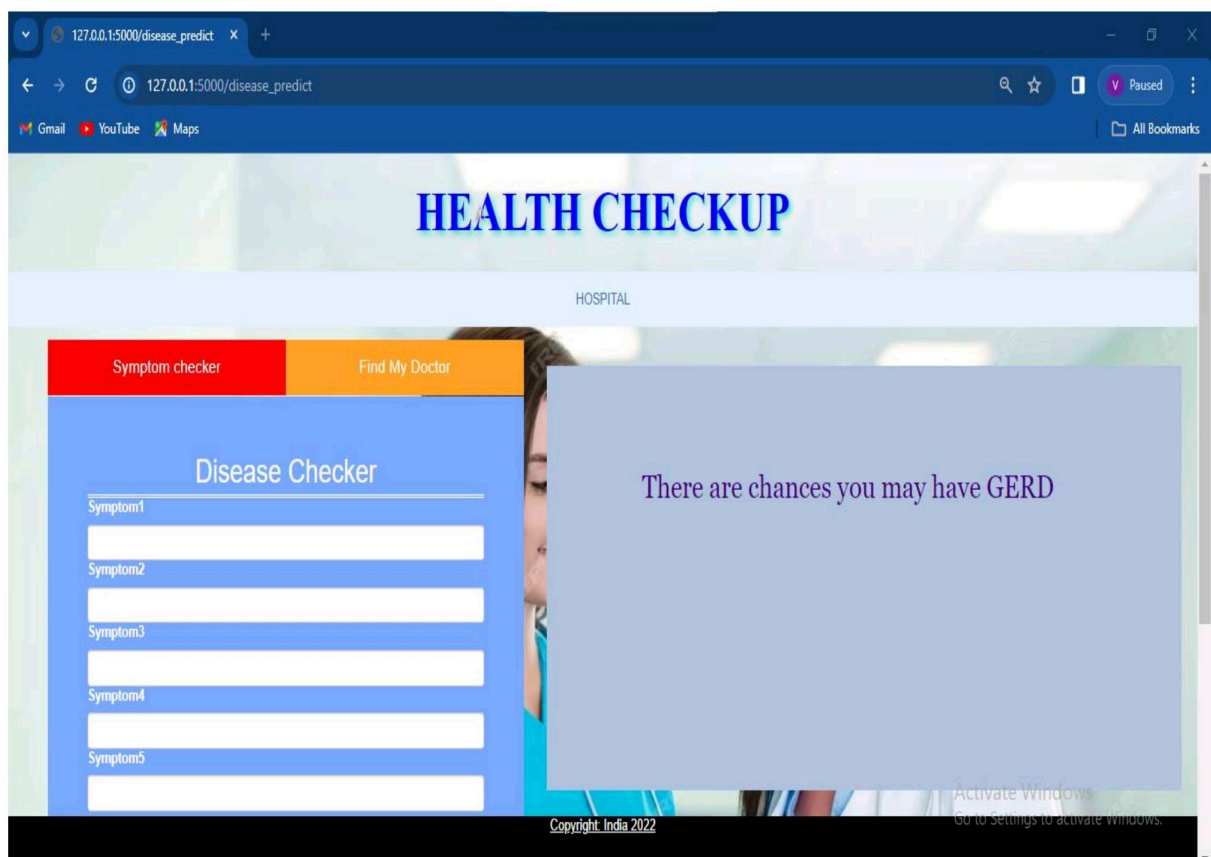
Importing required libraries



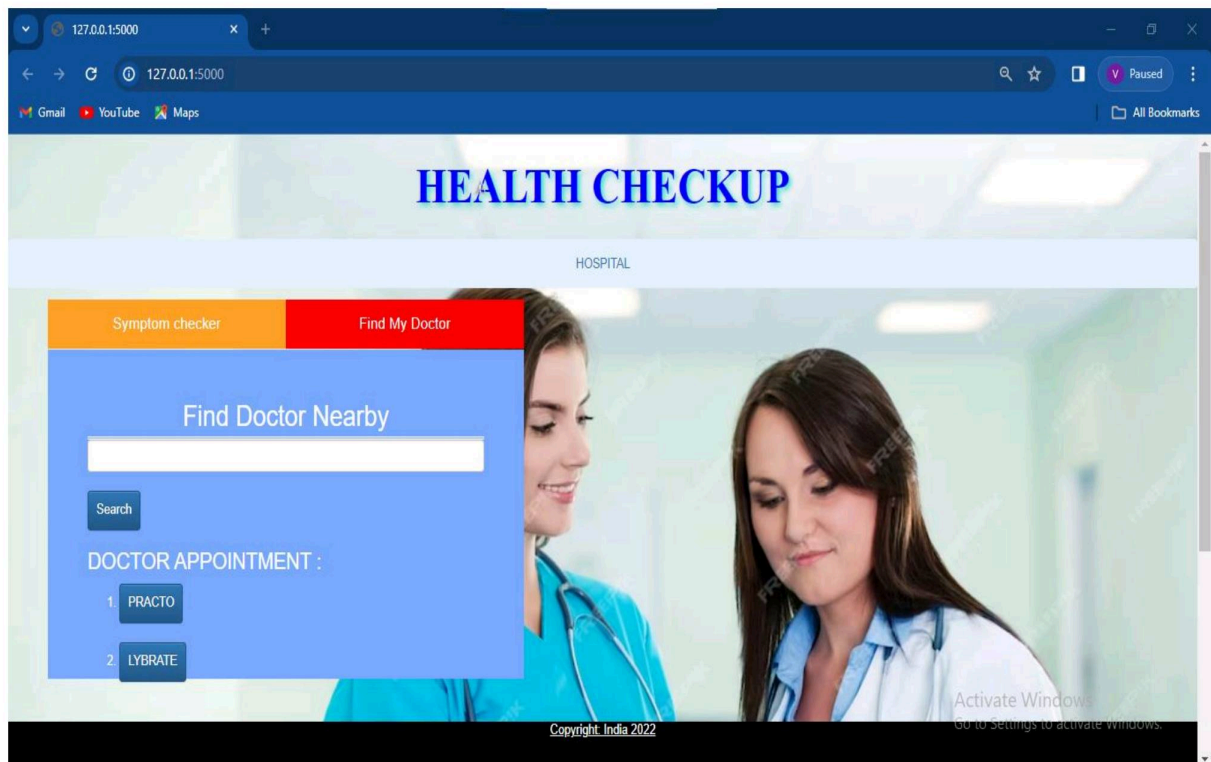
Home page



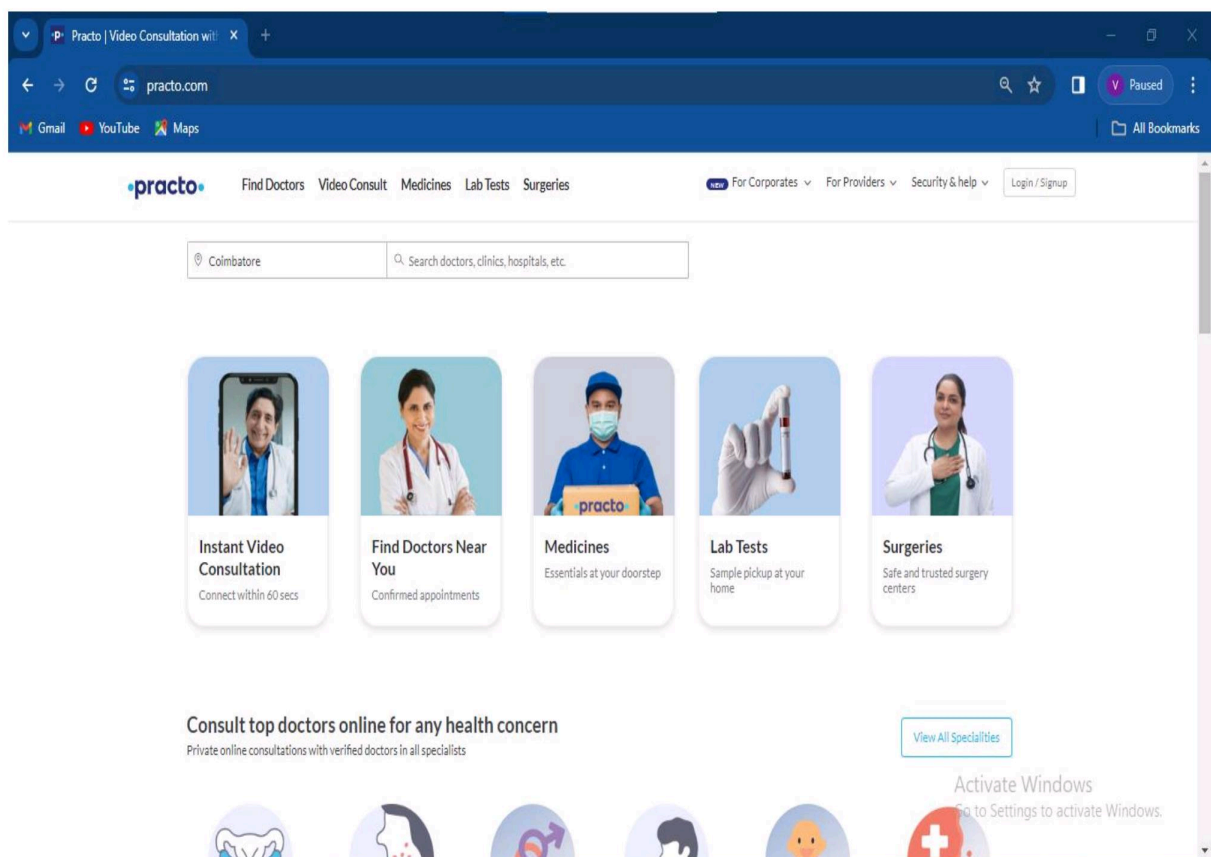
Symptoms Checker



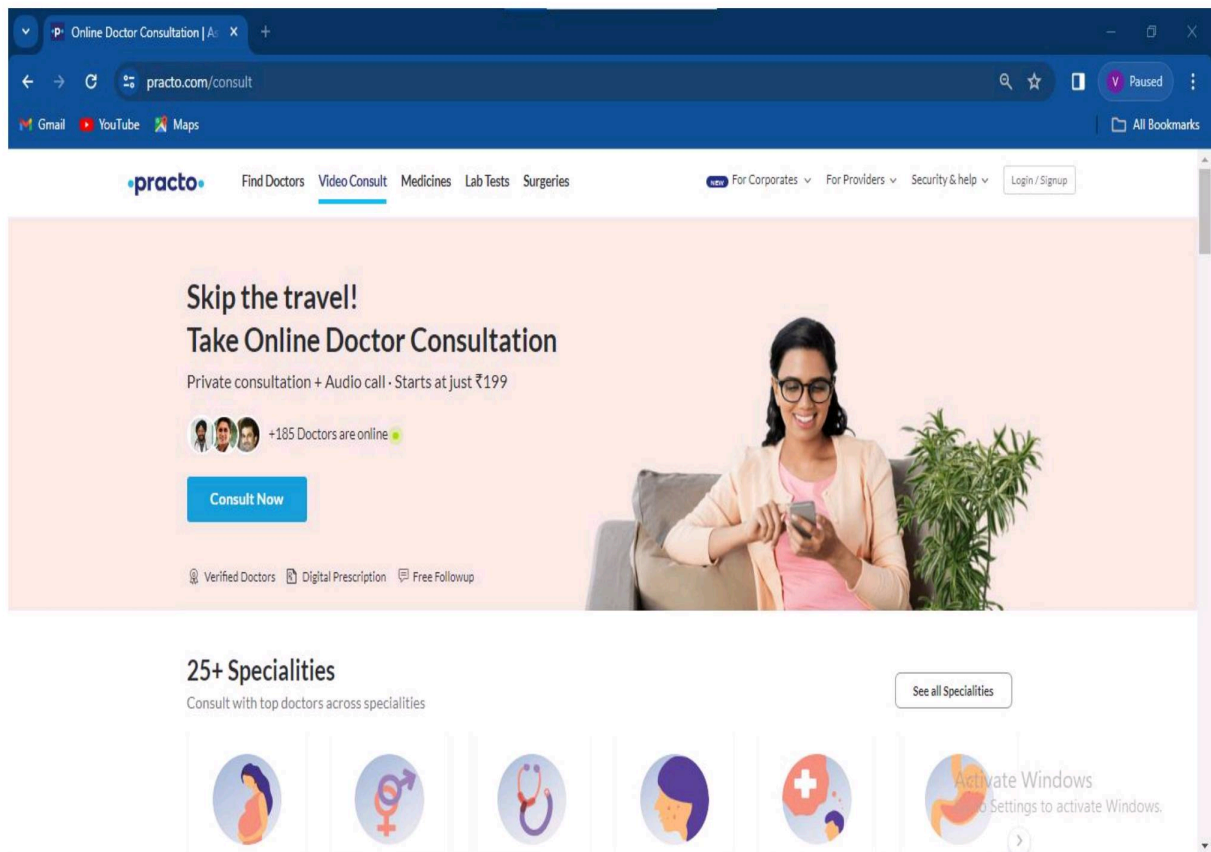
Result for Symptoms



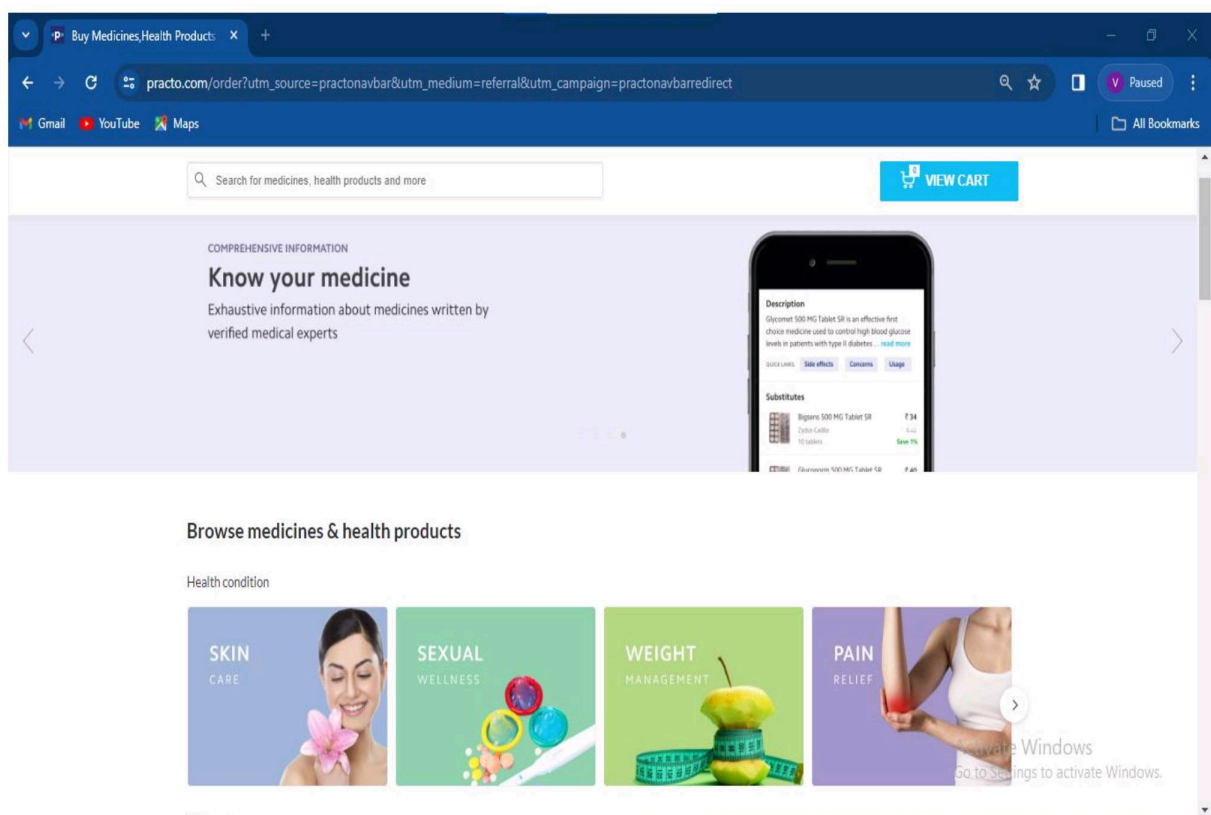
Finding Doctors



Medicines and Doctors Platform



Video Consulting



Medicines and Health Products

Practo Care Surgeries | End to e x +

practo.com/care

Gmail YouTube Maps

practo Find Doctors Video Consult Medicines Lab Tests Surgeries For Corporates For Providers Security & help Login / Signup

End to end care from Top Surgeons at our Practo Care Clinics

Talk to our care expert today to book your consultation.
All insurance accepted & No-cost EMI available.

Reach out to us on | 08045685554

We are experts in Surgical solutions for 50+ ailments.

Popular Surgeries

- Piles
- Hernia Treatment
- Kidney Stone
- Cataract
- Circumcision

Book your consultation today

Bangalore

Select Ailment*

Consulting Doctors for Surgeries

Training - Excel (Product Activation Failed)

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW

Calibri 11 A A

B I U

Wrap Text

General

Conditional Formatting

Format as Table

Cell Styles

Insert

Delete

Format

AutoSum

Fill

Sort & Filter

Find & Select

Clipboard Font Alignment Number Styles Cells Editing

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	itching	skin_rash_nodal_skin	continuous_shivering	chills	joint_pain	stomach_acidity	ulcers_on_muscle	vomiting	burning_spotting	fatigue	weight_gain	anxiety	cold_han	mood_sw	weight_loss	restless	lethargy	patches	irregular_cough	high_fever	sunken_eye	breast						
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
23	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
24	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
25	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
26	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
28	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
30	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
31	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
32	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Training

Training Data

Testing - Excel (Product Activation Failed)

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW

Clipboard Font Alignment Number Styles Cells Editing

itching

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	itching	skin_rash	nodal_skin	continuous	shivering	chills	joint_pain	stomach	acidity	ulcers_on	muscle_w	vomiting	burning_r	spotting	fatigue	weight_g	anxiety	cold_hanc	mood_sw	weight_lo	restle
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
17	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
19	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
20	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
21	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Testing

Testing Data