SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE LEARNING

A project report submitted in partial fulfillment of the requirement for the Award of the Degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

P. JAYA PRAKASH RAO (160720733061)

Y. ADITHYA VARDHAN REDDY (160720733065)

G. SHIVA SANJANA (160720733076)

Under the Guidance of

Mrs. B. VASAVI SRAVANTHI,
Assistant Professor, Dept. of CSE

Department of Computer Science and Engineering

Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001.

2023-2024

Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001,

Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATES

We, **P. JAYA PRAKASH RAO (160720733061), Y. ADITHYA VARDHAN REDDY (160720733065)** and **G. SHIVA SANJANA (160720733076)** students of Methodist College of Engineering and Technology, pursuing Bachelor's degree in Computer Science and Engineering, hereby declare that this Project report entitled **"SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE LEARNING",** carried out under the guidance of **Mrs. B. VASAVI SRAVANTHI** submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science. This is a record work carried out by us and the results embodied in this report have not been reproduced/copied from any source.

P. JAYA PRAKASH RAO (160720733061)
Y. ADITHYA VARDHAN REDDY (160720733065)
G. SHIVA SANJANA (160720733076)

Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001.

Department of Computer Science and Engineering

CERTIFICATE BY THE SUPERVISOR

This is to certify that this Dissertation on project work entitled "**SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE LEARNING**" *by* **P. JAYA PRAKASH RAO (160720733061), Y. ADITHYA VARDHAN REDDY (160720733065) AND G. SHIVA SANJANA (160720733076)** submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2023-2024, is a bonafide record of work carried out by them.

Mrs. B. VASAVI SRAVANTHI,
Assistant Professor,
Dept. of CSE.

Date:

Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001.

Department of Computer Science and Engineering

CERTIFICATE BY HEAD OF THE DEPARTMENT

This is to certify that this Dissertation on project work entitled "**SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE LEARNING**" *by* **P. JAYA PRAKASH RAO (160720733061), Y. ADITHYA VARDHAN REDDY (160720733065) AND G. SHIVA SANJANA (160720733076)** submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2023-2024, is a bonafide record of work carried out by them.

Dr. P. LAVANYA,
Professor &
Head of the Department.

Date:

Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001.

Department of Computer Science and Engineering

PROJECT APPROVAL CERTIFICATE

This is to certify that this Dissertation on project work entitled "**SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE LEARNING**" *by* **P. JAYA PRAKASH RAO (160720733061), Y. ADITHYA VARDHAN REDDY (160720733065) AND G. SHIVA SANJANA (160720733076),** submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2023-2024, is a bonafide record of work carried out by them.

INTERNAL **EXTERNAL** **HOD**

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to my Project guide **Mrs. B. VASAVI SRAVANTHI, Assistant Professor, CSE,** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without her innovative ideas and her relentless support and encouragement.

We would like to thank our project coordinator **Dr. M. SHARADA VARALAKSHMI, Professor, CSE,** who helped us by being an example of high vision and pushing towards greater limits of achievement.

Our sincere thanks to **Dr. P. Lavanya, Professor** and **Head of the Department of Computer Science and Engineering,** for her valuable guidance and encouragement which has played a major role in the completion of the project and for helping us by being an example of high vision and pushing towards greater limits of achievement.

We would like to express a deep sense of gratitude towards the **Dr. Prabhu G. Benakop, Principal, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We would like to express a deep sense of gratitude towards the **Dr. Lakshmipathi Rao, Director, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We are indebted to the Department of Computer Science & Engineering and Methodist College of Engineering and Technology for providing us with all the required facility to carry our work in a congenial environment. We extend our gratitude to the CSE Department staff for providing us to the needful time to time whenever requested.

We would like to thank our parents for allowing us to realize our potential, all the support they have provided us over the years was the greatest gift anyone has ever given us and also for teaching us the value of hard work and education. Our parents have offered us with tremendous support and encouragement, thanks to our parents for all the moral support and the amazing opportunities they have given us over the years.

METHODIST COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, Affiliated to Osmania University, - College Code – 1607

Department of Computer Science & Engineering
Vision & Mission

VISION

To become a leader in providing Computer Science & Engineering education with emphasis on knowledge and innovation.

MISSION

M1: To offer flexible programs of study with collaborations to suit industry needs
M2: To provide quality education and training through novel pedagogical practices
M3: **To** Expedite high performance of excellence in teaching, research and innovations.
M4: To impart moral, ethical valued education with social responsibility.

Program Educational Objectives

Graduates of Compute Science and Engineering at Methodist College of Engineering and Technology will be able to:

PEO1: Apply technical concepts, Analyze, synthesize data to Design and create novel products and solutions for the real-life problems.

PEO2: Apply the knowledge of Computer Science Engineering to pursue higher education with due consideration to environment and society.

PEO3: Promote collaborative learning and spirit of team work through multidisciplinary projects

PEO4: Engage in life-long learning and develop entrepreneurial skills.

Program Specific Outcomes

At the end of 4 years, Compute Science and Engineering graduates at MCET will be able to:

PSO1: Apply the knowledge of Computer Science and Engineering in various domains like networking and data mining to manage projects in multidisciplinary environments.

PSO2: Develop software applications with open-ended programming environments**.**

PSO3: Design and develop solutions by following standard software engineering principles and implement by using suitable programming languages and platforms

PROGRAM OUTCOMES

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

ABSTRACT

Nowadays, people face various diseases due to environmental conditions and their living habits. So, the prediction of disease at an earlier stage becomes an important task. The correct prediction of disease is the most challenging task. To overcome this problem Machine Learning plays an important role in predicting disease. Medical science has a large amount of data growth per year. Due to the increasing amount of data growth in the medical and healthcare field the accurate analysis of medical data has been benefits from early patient care. With the help of disease data, machine learning finds hidden pattern information in a huge amount of medical data. With the help of disease symptoms data set disease prediction is done by using Machine learning algorithms like Random Forest and Support Vector Machine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.INTRODUCTION

Disease diagnosis is a critical and decisive aspect of healthcare. Accurate and timely diagnosis can help in effective treatment and management of a patient's health problems. The conventional diagnostic methods have been based on the physician's clinical experience and expertise. Healthcare systems worldwide are facing increasing pressure due to rising patient numbers, limited resources, and the need for accurate and timely diagnoses.

Traditional approach to medical diagnosis relies on a doctor's expertise in interpreting a patient's symptoms, medical history, and physical examination. However, this process can be time-consuming, and human judgment is inherently prone to errors. This is where "Symptoms-Based Disease Diagnosis Using Machine Learning" emerges as a promising field with the potential to revolutionize healthcare.

1.1 Present Healthcare Landscape

The current healthcare landscape is undergoing a significant transformation, driven by advancements in technology and a growing emphasis on preventative and personalized medicine. Traditional methods of disease diagnosis often rely on a combination of patient history, physical examination, and laboratory tests. While these methods remain crucial, they can be time-consuming, expensive, and sometimes limited in their accuracy, particularly in the early stages of disease.

1.1.1 Traditional Diagnosis

Strengths:

- Established infrastructure: Hospitals, clinics, and a network of healthcare professionals provide essential medical services.
- Expertise of medical professionals: Doctors and nurses possess a wealth of knowledge and experience in diagnosing and treating diseases.
- Standardized procedures: Established protocols ensure consistent and reliable care.

Limitations:

- Limited accessibility: Costs, geographic location, and lack of resources can restrict access to healthcare for some populations.
- Delays in diagnosis: Traditional diagnostic methods can be slow, leading to delayed treatment and potentially worse outcomes.
- Subjectivity: Diagnosis can sometimes be subjective, relying on the experience and interpretation of the healthcare professional.
- Time Constraints: Doctors often have limited time for each patient, hindering detailed symptom evaluation and increasing the risk of misdiagnosis.
- Data Overload: The vast amount of medical information available can be overwhelming for even the most experienced medical professionals.
- Limited Resources: Access to specialized tests and equipment might be limited, especially in remote areas.

- Human Error: Subjectivity and human fatigue can lead to diagnostic errors.

1.1.2 Need for Innovation

These challenges highlight the need for innovative approaches to improve the efficiency and accuracy of disease diagnosis. Machine learning (ML) offers a promising solution by leveraging data analysis and pattern recognition to automate some aspects of diagnosis.

1.2 Revolutionizing Diagnosis in Healthcare

Machine learning is making significant strides in the realm of medical diagnosis. ML algorithms can analyse vast amounts of medical data, including patient history, lab results, and imaging scans, to identify patterns associated with specific diseases. This allows for earlier detection of illnesses, facilitating timely intervention and potentially improving patient outcomes.

One prominent application is in medical imaging analysis. ML algorithms can be trained to detect abnormalities in X-rays, mammograms, CT scans, and MRIs with high accuracy, surpassing even the most skilled radiologists in certain cases. This not only reduces the workload for healthcare professionals but also aids in early diagnosis, leading to better treatment options and prognoses.

Another area of impact is in risk prediction. ML models can analyse patient data to identify individuals at high risk of developing certain diseases like heart disease, diabetes, or cancer. This enables preventive measures and early interventions, potentially preventing the onset of the disease or mitigating its severity.

1.3 Benefits of Machine Learning in Healthcare

The incorporation of Machine Learning into the healthcare field offers numerous advantages for patients, healthcare providers, and the overall healthcare system. Some of the primary advantages encompass the following:

- **Improved Diagnostic Accuracy:** ML algorithms can analyse complex medical data, leading to more accurate and early detection of diseases. This allows for prompt interventions, leading to improved patient outcomes and potentially saving lives.
- **Personalized Treatment:** ML algorithms have the capability to examine patient data and create personalized treatment strategies taking into account individual attributes. This method enhances the efficacy of treatments and mitigates the chances of unfavourable responses, ultimately leading to increased patient contentment and improved overall health.
- **Enhanced Efficiency and Cost Savings:** ML algorithms can automate various healthcare processes, such as administrative tasks, patient triaging, and resource allocation. This streamlines operation reduces manual errors, and optimizes resource utilization, resulting in improved efficiency and cost savings.

- **Predictive Analytics:** ML algorithms can predict patient outcomes, disease progression, and potential complications. This enables healthcare providers to take proactive measures, such as early interventions or preventive treatments, to improve patient outcomes and reduce healthcare costs.

Support for Healthcare Professionals:

- Decision Support: ML systems can provide real-time decision support to healthcare professionals by suggesting potential diagnoses and treatment options based on data-driven insights.
- Continuous Learning: ML models can continuously learn from new data, improving their accuracy and relevance over time, thus providing up-to-date support for medical professionals.

Expanding Access to Care:

- Remote Diagnostics: ML-powered tools can facilitate remote diagnostics, providing access to medical expertise in underserved or remote areas, thus bridging the gap in healthcare accessibility.
- Telemedicine Integration: Integration with telemedicine platforms allows ML algorithms to assist in virtual consultations, enhancing the quality of care provided remotely.

Enhanced Research and Development:

- Drug Discovery: ML can accelerate drug discovery and development by identifying potential drug candidates and predicting their efficacy and safety, thus reducing the time and cost associated with bringing new treatments to market.
- Clinical Trials: ML can optimize clinical trial design and participant selection, ensuring more effective and efficient trials.
- By harnessing the power of Machine Learning, the healthcare industry can address many of the challenges faced by traditional diagnostic methods, leading to a more efficient, accurate, and patient-centred approach to disease diagnosis and treatment.

The integration of Machine Learning into the healthcare sector represents a significant advancement in medical diagnostics and patient care. By leveraging the vast amounts of data generated in healthcare settings, ML algorithms can provide precise, personalized, and timely diagnoses that enhance patient outcomes and streamline healthcare delivery. As technology continues to evolve, the potential for Machine Learning to transform the healthcare landscape will only grow, making it an indispensable tool in the quest for better health and well-being.

2.LITERATURE REVIEW

2.1 LITERATURE SURVEY

S.no	Title	Author/s	Year	Findings	Advantage/s	Disadvantage/s
1	Healthcare prediction using Machine Learning	Naveenkumar S, Kirubakaran R, Sobhana M, Sangetha K, Jeeva G.	2022	This paper is based on predictive modelling, predicts the disease of users on the basis of the symptoms that the user provides symptoms as an input to the system. Smart health predictions are made by the implementation of the Naive Bayes Classifier. The Naive Bayes Classifier measures the disease percentage probability by considering all its features and provides the likelihood of the disease.	The algorithms and database management techniques and extract new patterns from data. The datasets will be compared with the incoming queries and an Association Rule Mining Report will be generated.	Only few diseases with high accuracy, where all the diseases are not predicted in an accurate way
2	Health Care Application using Machine Learning	Ajinkya Padule, Aman Patel, Arsalan Patel, Aman Shaikh, Jyoti Gavhane.	2022	This paper, present an intelligent healthcare application that can predict and provide information on various diseases. Datasets for Diabetes and Heart diseases were collected in Comma Separated Values (CSV) format. Decision tree and Logistic Regression Algorithms were used to predict the disease	The 1567 training data and 525 testing data for malaria disease and 5216 training data and 624 testing data for pneumonia disease Heart and Diabetes diseases Were predicted by Decision tree and Logistic Regression With 80% accuracy	Logistic Regression and Decision Tree were able to predict the Liver diseases with low accuracy of 65% and 60%

3	A Smart Health Care System Using Machine Learning.	Anargha P, Justin Joseph, Sam Sunil, Sanjai S, Divya Mohan.	2022	People face various diseases due to environment and living habits, the prediction of disease at an earlier stage becomes an important task. To predict the disease taking symptoms as input Disease dataset was used. For accurate disease prediction various ML algorithms where used.	The LR has the highest accuracy of 86.89%. The accuracy and processing time of KNN were 82.5% and 11.1 seconds, respectively.	The accuracy can be decreased when it comes to high dimensional Data.
4.	Heart disease prediction system	Mrs.Jayashree L K, Sushmita Makapur, T D Vyshnavi, T D Prathyusha, V K Kavya	2019	Heart disease is a widespread problem affecting a significant portion of the global population (nearly 60%). There's a need for accessible tools to aid in early prediction and prevention of heart disease.	The application leverages machine learning (KNN) to analyze user-provided data and provide a probability of heart disease. The application offers preventive measures by suggesting nearby doctors, hospitals, and their locations.	The application shouldn't be a substitute for professional medical diagnosis. It should encourage users to seek medical consultation for confirmation and proper treatment. The document mentions only KNN algorithm. It would be beneficial to explore other machine learning approaches for potentially higher accuracy.
5	Machine Learning in Healthcare	Hafsa Habehh and Suril Gohel	2021	Machine learning can improve accuracy, prediction, and quality of care in healthcare. Machine learning can help physicians identify healthcare trends and develop disease prediction models.	Increased efficiency in organizing electronic health records. Improved identification of irregularities in blood samples, using medical	Not mentioned in the excerpt but the text suggests there are ethical and logistical risks and challenges to consider.

				Machine learning can be used to analyse electronic health records, medical images, and monitor patients.	imaging.	
6	Health Care Digitalization and Machine Learning	Ajinkya Yelne, Archana Raut	2022	Digitalization of healthcare has led to a surge in data related to patients, hospitals, and diseases. This data offers a valuable resource for improving healthcare delivery through machine learning on digital platforms.	Large-scale digitalization of healthcare services in India creates a rich data pool for tackling new challenges.	The text doesn't explicitly mention disadvantages, but some potential concerns surrounding healthcare data and machine learning
7	Medicine Recommendation System	Varun A. Goyal, Dilip J. Parmar, Namaskar I. Joshi, Prof. Komal Champanerkar	2020	Doctors prescribing based on limited experience contributes significantly to medication errors. Data mining and recommender systems show promise in reducing these errors by leveraging historical diagnosis data.	Improved accuracy in prescribing medication by suggesting options based on past diagnoses. Potential reduction in medication errors, particularly for new doctors.	The report highlights the need for further research to choose the best algorithm for optimal performance. The system relies on the quality and completeness of the diagnosis data used for training.
8	Smart Health Prediction Using Machine Learning	Naveen kumar S, Kirubhakaran R, Jeeva G, Shobana M, Sangeetha K	2021	The Naive Bayes Classifier is a suitable algorithm for analyzing symptoms and providing disease likelihood. Analyzing historical data with machine learning can reveal new patterns in disease occurrence.	Improved forecast accuracy with machine learning algorithms. Clearer understanding of potential diseases for patients/users. Potential for faster consultations with doctors through a chat window.	The system relies on user-reported symptoms, which may be inaccurate or incomplete. The Naïve Bayes Classifier may not capture complex relationships between symptoms and diseases.

9	Exploring the possibilitys of using Machine learning in health care	Dheeraj B. Shukla, Ajahar I. Pathan, Rijawan A. Shaik, Vasim I. Memon	2021	Machine learning is being explored as a potential solution for problems across various domains, including healthcare. This study demonstrates the feasibility of integrating machine learning into healthcare processes.	Machine learning can potentially improve healthcare by: Identifying patterns in medical data for accurate diagnosis. Supporting personalized treatment plans.	The study acknowledges challenges associated with integrating machine learning into clinical settings, such as: Ensuring data security and patient privacy. Potential biases in algorithms trained on medical data.
10	An Intelligent Medicine Recommender System Framework.	Bao, Y. and Jiang, X	2016	Medication errors are a significant problem, causing hundreds of thousands of deaths annually in China and the USA. Doctors prescribing based on limited experience contributes to a large portion of these errors. Data mining and recommender systems offer potential to decrease medication errors.	The system incorporates multiple algorithms (SVM, decision tree) and selects the best performing one (SVM) for optimal accuracy, efficiency, and scalability. A mistake-check mechanism is included to ensure diagnosis accuracy and service quality. The study demonstrates excellent efficiency, accuracy, and scalability of the proposed system.	The proposed system utilizes data mining to recommend medications based on diagnosis history, potentially leading to fewer errors.
11	GalenOWL: Ontology-based drug-recommendations discovery.	Doulaverakis, C., Nikolaidis, G., Kleontas, A., and Kompatsiaris, I.	2012	Identifying drug interactions is becoming increasingly difficult due to the vast number of drugs and ongoing research.	GalenOWL, a semantic web-based online service, offers real-time drug-drug and drug-	It's unclear how the system addresses potential biases in the underlying knowledge base.

				Existing medical data standards aid information sharing, but staying updated on interactions remains challenging.	disease interaction discovery. Medical information is translated into structured ontologies, enabling efficient knowledge retrieval. The system leverages existing standards for data compatibility, highlighting its potential benefits.	The complexity of building
12	DiaTrack. Web-based application for assisted decision-making in treatment	Medvedeva, O., Knox, T., and Paul, J	2007	There is a potential benefit to using a web-based system for sharing knowledge and experiences among doctors to improve type 2 diabetes treatment.	Doctors can access a broader and richer pool of patient data and treatment approaches. Shared knowledge among doctors could lead to better treatment recommendations. The system might learn and adapt over time based on which factors doctors find most important in similar cases.	This research is still in progress with a prototype under development. The effectiveness of the system in improving treatment outcomes is not yet established. The abstract doesn't mention how the system addresses potential biases in the data or ensures the quality of information contributed by doctors. User interface design and ease of use for doctors haven't been fully explored yet.

13	Medicine Recommender Systems	Benjamin Stark, Constanze Knahl ,Mert Aydin ,Karim Elish	2019	Medicine recommender systems can be helpful for doctors in choosing the right medication for patients. These systems are developed using various algorithms that leverage advanced technologies.	More concise decisions: Recommenders can analyze vast amounts of data to suggest appropriate medications, potentially leading to quicker and more informed decisions by doctors.	The report doesn't explicitly list disadvantages, but it highlights the need for further research, suggesting there might be limitations in current systems.
14	Medicine Recommendation System	Namaskar I. Joshi, Prof. Komal .	2019	Doctors prescribing based on limited experience contributes significantly to medication errors. Data mining and recommender systems show promise in reducing these errors by leveraging historical diagnosis data.	Improved accuracy in prescribing medication by suggesting options based on past diagnoses. Potential reduction in medication errors, particularly for new doctors.	The report highlights the need for further research to choose the best algorithm for optimal performance. The system relies on the quality and completeness of the diagnosis data used for training.

Table 2.1: *Literature Survey*

2.2 GAP ANALYSIS

- Existing systems were ill-equipped to handle the evaluation of datasets that incorporate multiple modes of information. Advancements in deep learning and ensemble methods provide more powerful tools for extracting complex patterns from data.

- The current models have encountered difficulties in accurately analyzing implicit aspects.

- Earlier systems might have had basic interfaces. Current systems likely offer more user-friendly interfaces with features like symptom selection tools and interactive visualizations of results.

2.3 PROBLEM STATEMENT

To tackle the prevalent issue of diagnosing health conditions accurately from symptoms. By allowing users to input their symptoms and promptly receiving potential diagnoses, the system offers a convenient solution. This helps individuals dealing with overlapping symptoms to distinguish between different diseases efficiently.

2.4 OBJECTIVES

- **Personalized Model:** To develop a ML model that analyze the patient symptoms to identify potential health and risks to provide timely diagnosis.

- **Comprehensive Information:** In addition to predictions, the app offers comprehensive information about the predicted disease, including descriptions, precautions, medications, diet recommendations, and workout tips.

- **Accurate Predictions:** Leveraging machine learning, our model provides accurate disease predictions based on the symptoms provided by the user.

- **Increase Accessibility:** Provide a user-friendly platform for individuals to explore health concerns, particularly for those lacking immediate access to healthcare professionals.

3. DESGIN ANALYSIS

3.1 SYSTEM ARCHITECTURE

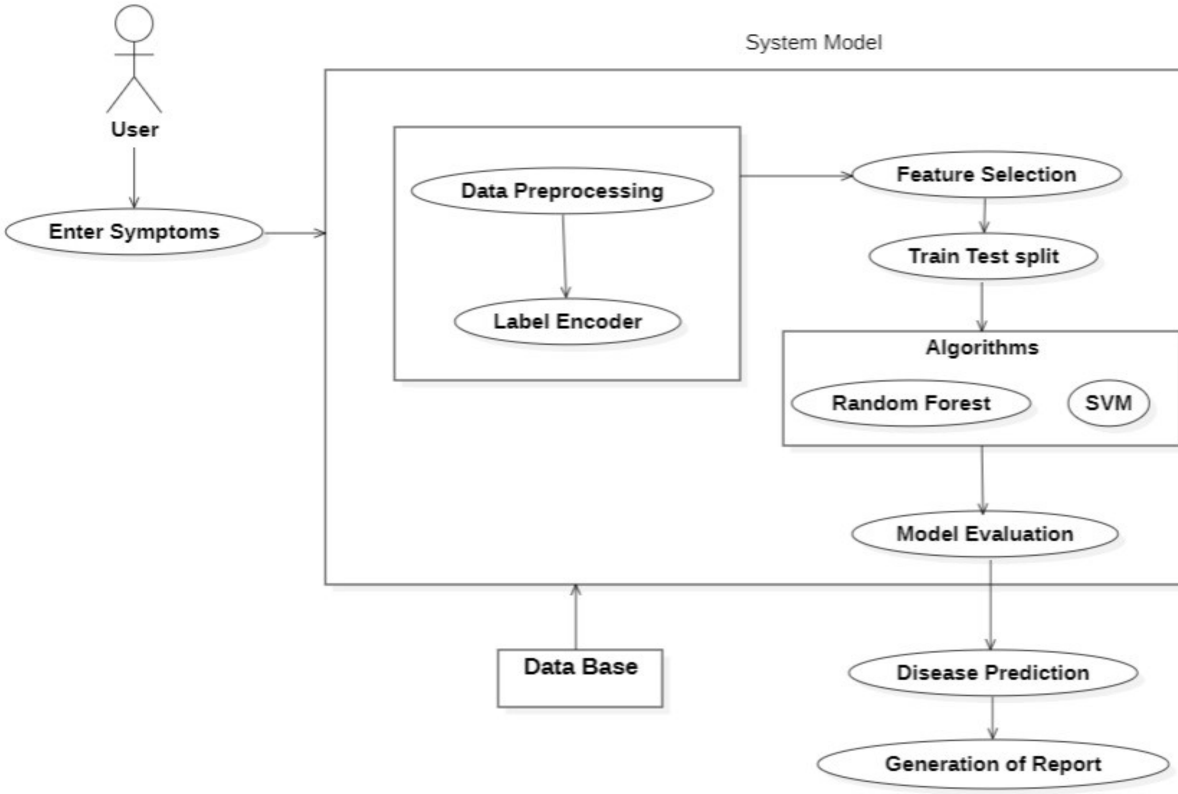


Fig 3.1: *System Architecture*

3.2 UML Diagrams

UML stands for unified modeling language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standardized is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. UML is comprised of two major components: a Mta-model and a notation. In the future, some form of method or process may also be added to; or association with UML. It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The set of diagrams need not completely cover the model and deleting a diagram does not change the model. The model may also contain documentation that drives the model elements and diagrams. In 1997 UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005 UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then it has been periodically revised to cover the latest revision of UML.

UML is not a development method by itself; however, it was designed to be compatible with the leading object-oriented software development methods of its time, for example OMT, Booch method, Objectory and especially RUP that it was originally intended to be used with when work began at Rational Software (Unified Modeling Language) is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology. The major perspectives of a UML are Design, Implementation, Process and Deployment. The center is the Use Case view which connects all these four. A Use case represents the functionality of the system.

3.2.1 Use Case Diagrams

A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you will use a set of specialized symbols and connectors

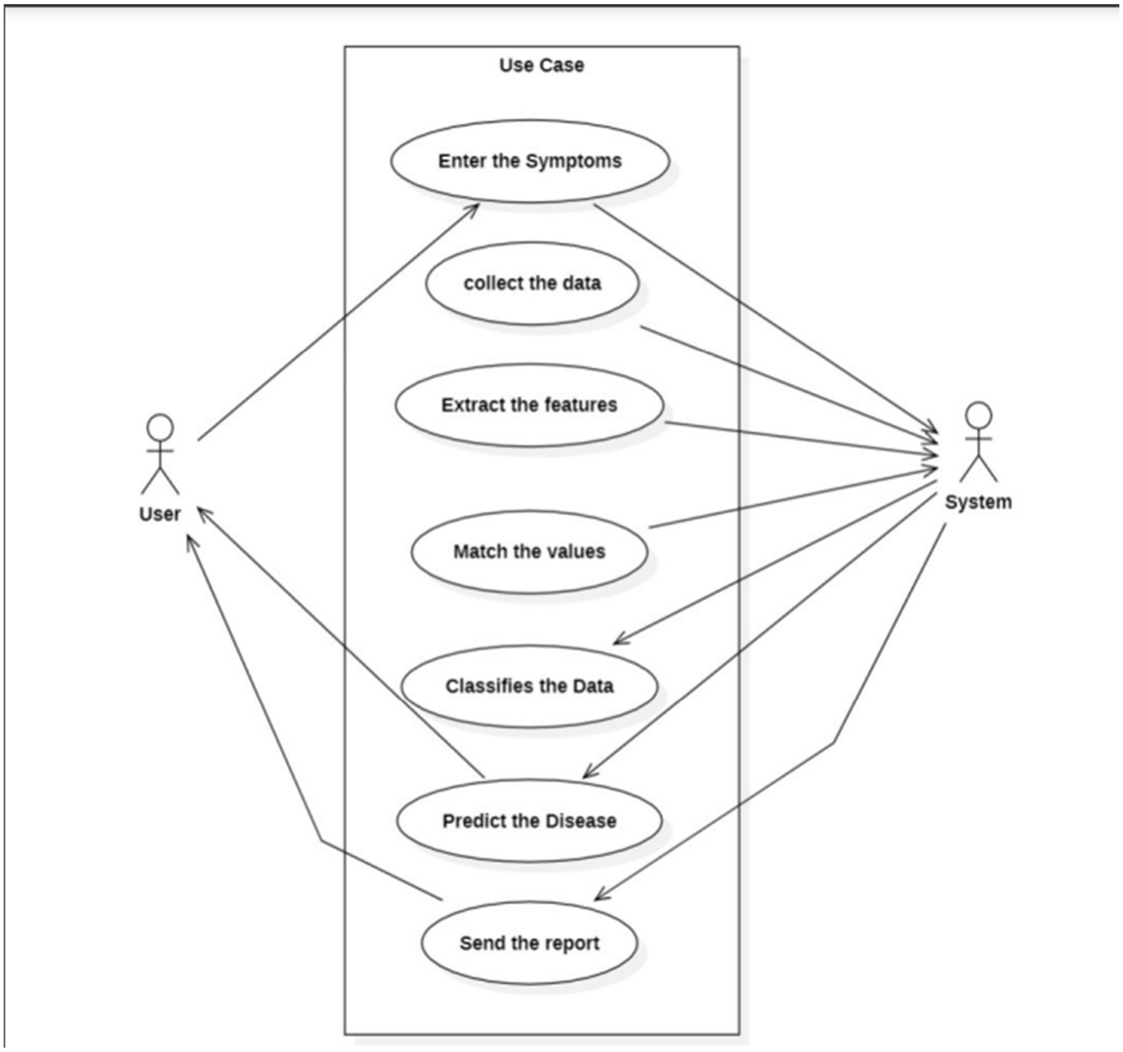


Fig 3.2.1: Use Case Diagram

- **Enter Symptoms:** The user initiates the interaction by entering their symptoms into the system. This could involve text input, selecting options from a list, or even voice input.
- **Collect Data:** The system gathers the user-entered information about their symptoms.
- **Extract Features:** The system might extract relevant features from the collected data. Feature extraction is a process that transforms raw data into a more usable format. In this case, it might involve identifying keywords or patterns related to specific diseases.
- **Match Values:** The system compares the extracted features to a database of known diseases.
- **Classify Data:** Based on the matching results, the system classifies the data and suggests potential illnesses.
- **Predict Disease:** The system might use the analysis to predict the most likely disease based on the entered symptoms.

- **Send Report:** The system generates a report for the user that summarizes the analysis and presents possible diagnoses.

3.2.2 Data Flow Diagrams

- A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects.
- A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

3.2.2.1 DFD Level-0

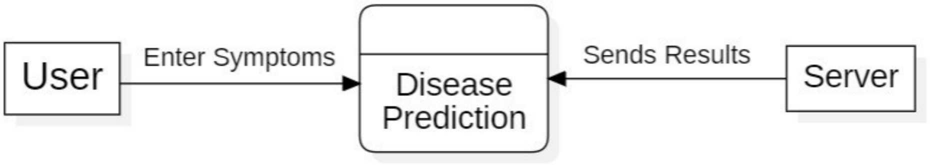


Fig 3.2.2.1: *Data Flow Diagram level-0*

- User Enters Symptoms: The user initiates the process by entering their symptoms into the system. This could involve text input, selecting options from a list, or even voice input.

- Sends Results: The user’s entered symptoms are then sent to the server for analysis.

- Server Processes Data: The server receives the data from the user and performs the necessary computations to arrive at a disease prediction. This might involve:

- Extracting features from the user-entered symptoms.
    - Matching the extracted features to a database of known diseases.
    - Running the data through a machine learning model to predict the most likely disease.
    - Disease Prediction: The server generates a disease prediction based on the analysis of the user-entered symptoms.

- Server Sends Results: The server sends the disease prediction results back to the user.

3.2.2.2 DFD Level-1

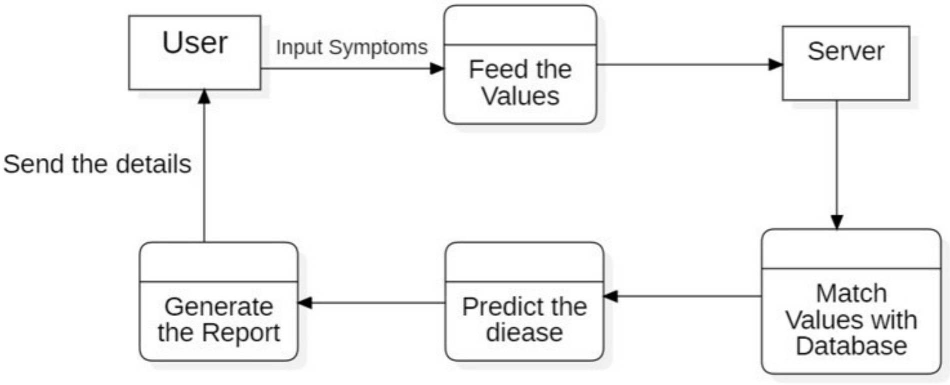


Fig 3.2.2.2: *Data Flow Diagram level-1*

- User Inputs Symptoms: The user initiates the process by entering their symptoms into the system. This could involve text input, selecting options from a list, or even voice input.
- Feeds the Values: The user's entered symptoms are then sent to the server, which is depicted by the arrow labelled "Feed the Values".
- Server Generates Report: The server receives the data from the user and performs the necessary computations to arrive at a disease prediction. This might involve:
  - Extracting features from the user-entered symptoms.
  - Matching the extracted features to a database of known diseases.
  - Running the data through a machine learning model to predict the most likely disease.
- Predicts the Disease: The server analyses the data and predicts the disease based on the user-entered symptoms.
- Matches Values with Database: The predicted disease is then matched against a database of known diseases to identify the most likely cause of the symptoms.
- Sends the Report: The server sends the disease prediction results back to the user. The report might include the predicted disease and other relevant information, such as the confidence level of the prediction.

3.2.3 Activity Diagram:

- Activity Diagram is similar to a business work flow diagram or simply a flowchart with much richer semantics.
- It describes the system activities, or the person who does the activity, and the sequential flow of these activities.
- The activity diagram is one of the UML diagrams associated with object-oriented approach, through it can be used in any other software development paradigm
- Activity Diagram Notations:
    1. Initial
    2. Final
    3. Flow Final Node.

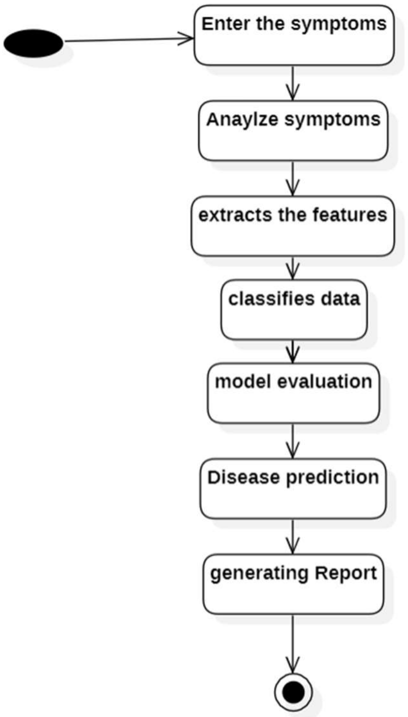


***Fig 3.2.3**: Activity Diagram*

3.2.4 Class Diagram

- Class Diagram is a type of static structure diagram that describes the structure of a system.
- It shows the system's classes, their attributes, operations (or methods), and the relationships among objects.

- A class notation consists of three parts:

  Class Name:

  The name of the class appears in the first partition.

  Class Attributes:

  Attributes are shown in the second partition.

  The attribute type is shown after the colon.

  Class Operations:

  Operations are shown in the third partition.

  The services provided by the class

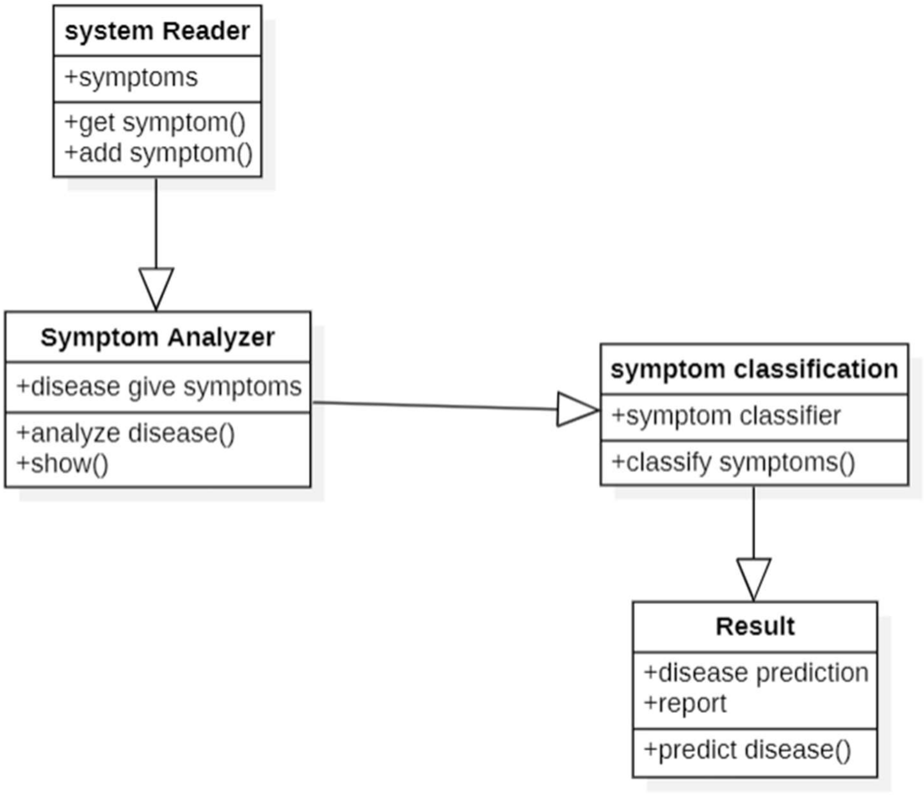


Fig 3.2.4: *Class Diagram*

4. MODULES AND IMPLEMENTATION

4.1 User

The user is the person who interacts with the system to get a disease prediction. The system assumes the user has some knowledge of their own symptoms and can enter them into the system. There is no mention of user authentication in the diagram, so it assumes anyone can enter information and receive a prediction.

4.2 Data Base

The data base stores the training data for the machine learning model. This data is likely a large collection of examples consisting of patient symptoms and their corresponding diagnoses. The quality of the data heavily influences the accuracy of the model's predictions.

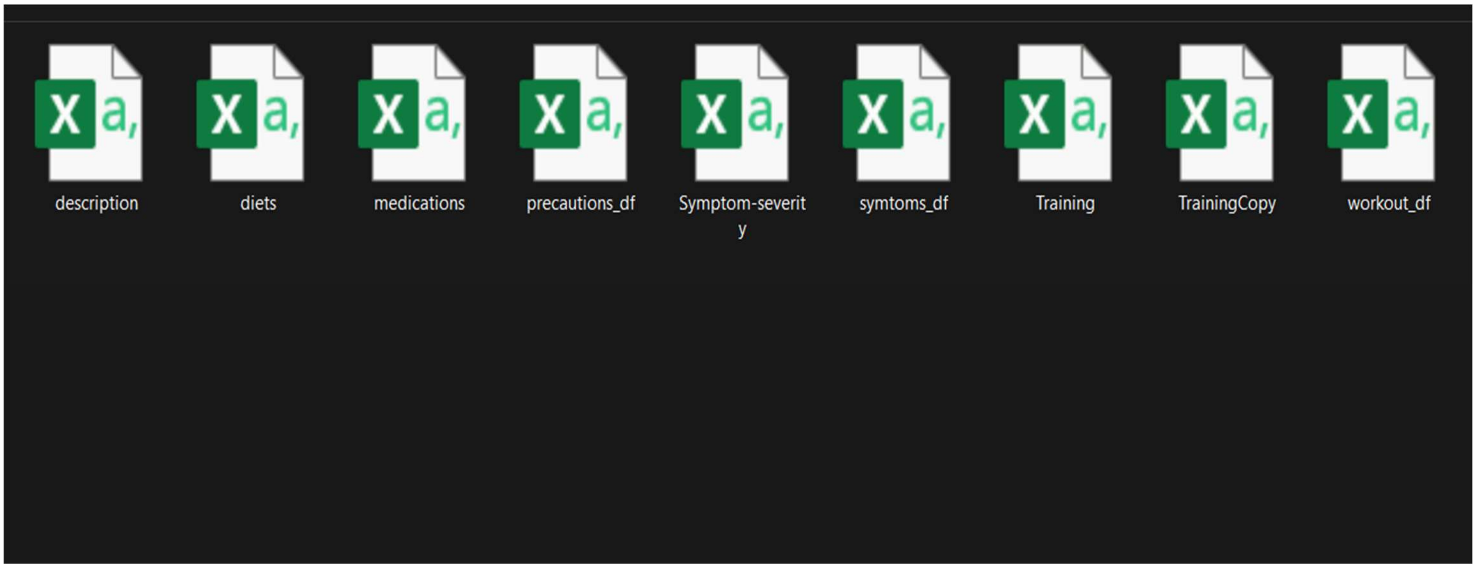
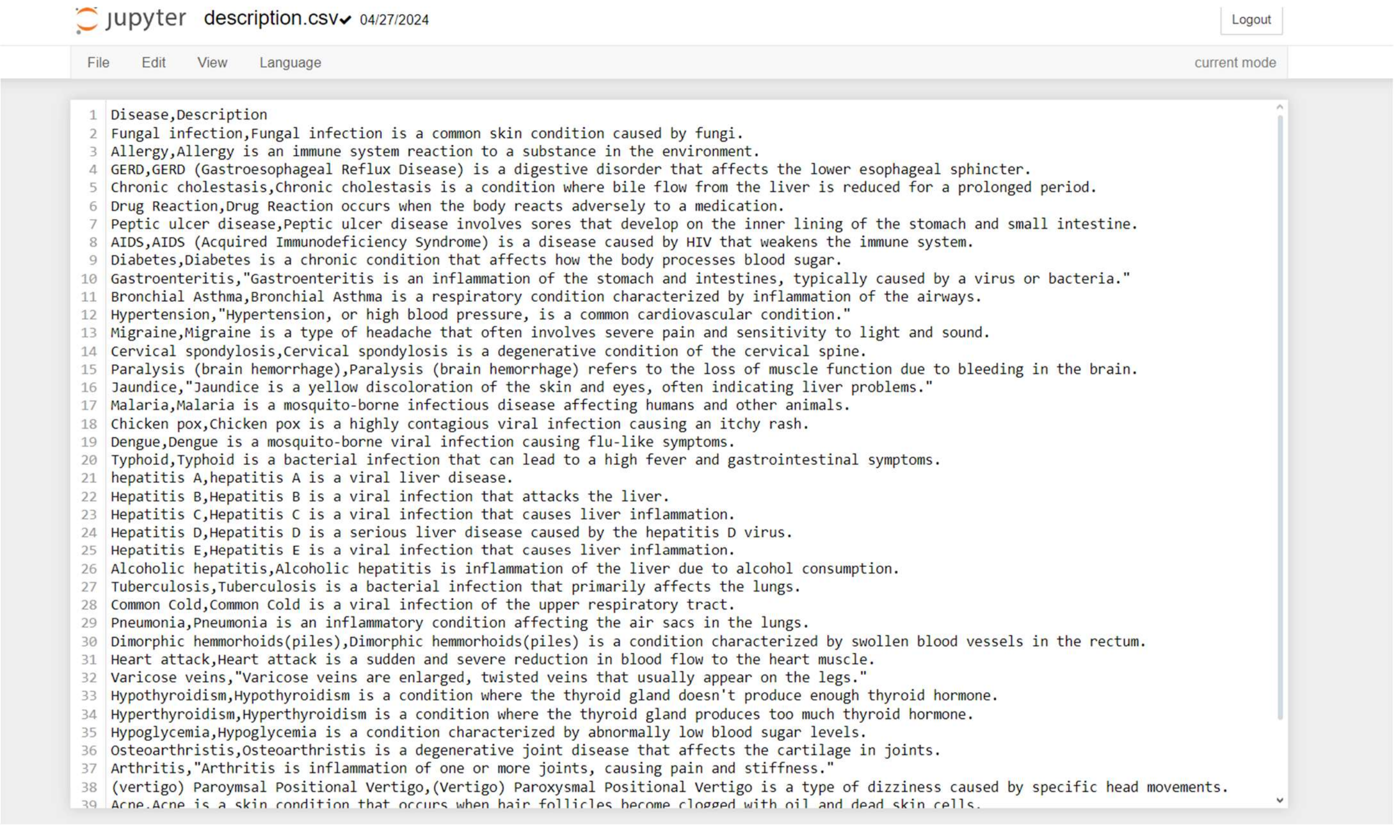


Fig 4.2.1: *Data Set*

- **Data Labelling:** For the machine learning model to learn how to predict diseases from symptoms, the training data needs to be labelled. This means that each data point needs to have a corresponding label that indicates the correct disease diagnosis. In the context of the image, this labelling process is likely done by medical professionals who can accurately identify diseases based on the listed symptoms.



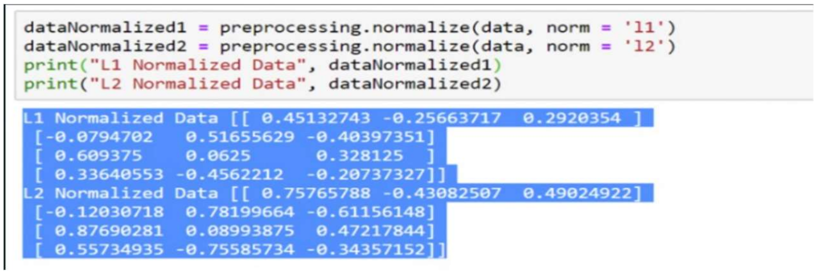
```
Disease,Description
Fungal infection,Fungal infection is a common skin condition caused by fungi.
Allergy,Allergy is an immune system reaction to a substance in the environment.
GERD,GERD (Gastroesophageal Reflux Disease) is a digestive disorder that affects the lower esophageal sphincter.
Chronic cholestasis,Chronic cholestasis is a condition where bile flow from the liver is reduced for a prolonged period.
Drug Reaction,Drug Reaction occurs when the body reacts adversely to a medication.
Peptic ulcer disease,Peptic ulcer disease involves sores that develop on the inner lining of the stomach and small intestine.
AIDS,AIDS (Acquired Immunodeficiency Syndrome) is a disease caused by HIV that weakens the immune system.
Diabetes,Diabetes is a chronic condition that affects how the body processes blood sugar.
Gastroenteritis,"Gastroenteritis is an inflammation of the stomach and intestines, typically caused by a virus or bacteria."
Bronchial Asthma,Bronchial Asthma is a respiratory condition characterized by inflammation of the airways.
Hypertension,"Hypertension, or high blood pressure, is a common cardiovascular condition."
Migraine,Migraine is a type of headache that often involves severe pain and sensitivity to light and sound.
Cervical spondylosis,Cervical spondylosis is a degenerative condition of the cervical spine.
Paralysis (brain hemorrhage),Paralysis (brain hemorrhage) refers to the loss of muscle function due to bleeding in the brain.
Jaundice,"Jaundice is a yellow discoloration of the skin and eyes, often indicating liver problems."
Malaria,Malaria is a mosquito-borne infectious disease affecting humans and other animals.
Chicken pox,Chicken pox is a highly contagious viral infection causing an itchy rash.
Dengue,Dengue is a mosquito-borne viral infection causing flu-like symptoms.
Typhoid,Typhoid is a bacterial infection that can lead to a high fever and gastrointestinal symptoms.
hepatitis A,hepatitis A is a viral liver disease.
Hepatitis B,Hepatitis B is a viral infection that attacks the liver.
Hepatitis C,Hepatitis C is a viral infection that causes liver inflammation.
Hepatitis D,Hepatitis D is a serious liver disease caused by the hepatitis D virus.
Hepatitis E,Hepatitis E is a viral infection that causes liver inflammation.
Alcoholic hepatitis,Alcoholic hepatitis is inflammation of the liver due to alcohol consumption.
Tuberculosis,Tuberculosis is a bacterial infection that primarily affects the lungs.
Common Cold,Common Cold is a viral infection of the upper respiratory tract.
Pneumonia,Pneumonia is an inflammatory condition affecting the air sacs in the lungs.
Dimorphic hemmorhoids(piles),Dimorphic hemmorhoids(piles) is a condition characterized by swollen blood vessels in the rectum.
Heart attack,Heart attack is a sudden and severe reduction in blood flow to the heart muscle.
Varicose veins,"Varicose veins are enlarged, twisted veins that usually appear on the legs."
Hypothyroidism,Hypothyroidism is a condition where the thyroid gland doesn't produce enough thyroid hormone.
Hyperthyroidism,Hyperthyroidism is a condition where the thyroid gland produces too much thyroid hormone.
Hypoglycemia,Hypoglycemia is a condition characterized by abnormally low blood sugar levels.
Osteoarthristis,Osteoarthristis is a degenerative joint disease that affects the cartilage in joints.
Arthritis,"Arthritis is inflammation of one or more joints, causing pain and stiffness."
(vertigo) Paroymsal Positional Vertigo,(Vertigo) Paroxysmal Positional Vertigo is a type of dizziness caused by specific head movements.
Acne,Acne is a skin condition that occurs when hair follicles become clogged with oil and dead skin cells.
```

Fig 4.2.2: *Data Labelling*

4.3 Data Preprocessing

This step involves cleaning the data, handling missing values, and formatting the data for use by machine learning algorithms. Here's a breakdown of these activities

4.3.1 Data Cleaning: This involves removing irrelevant or erroneous data entries from the dataset. Here are some common data cleaning techniques:



```
dataNormalized1 = preprocessing.normalize(data, norm = 'l1')
dataNormalized2 = preprocessing.normalize(data, norm = 'l2')
print("L1 Normalized Data", dataNormalized1)
print("L2 Normalized Data", dataNormalized2)
```

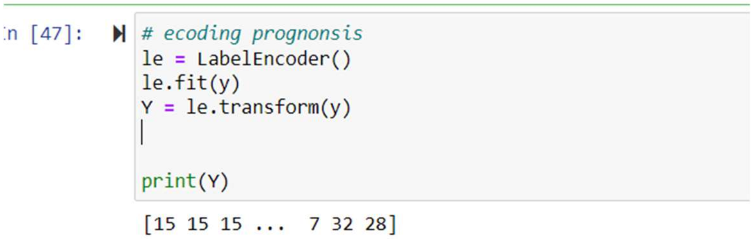
```
L1 Normalized Data [[ 0.45132743 -0.25663717  0.2920354 ]
 [-0.0794702   0.51655629 -0.40397351]
 [ 0.609375    0.0625      0.328125  ]
 [ 0.33640553 -0.4562212  -0.20737327]]
L2 Normalized Data [[ 0.75765788 -0.43082507  0.49024922]
 [-0.12030718  0.78199664 -0.61156148]
 [ 0.87690281  0.08993875  0.47217844]
 [ 0.55734935 -0.75585734 -0.34357152]]
```

Fig 4.3.1: *Data Cleaning*

- **Identifying missing values:** You may find data points missing values for certain attributes. You can choose to exclude these entries from your dataset, or impute the missing values with a strategy like using the mean or median of the existing data for that specific attribute.
- **Inconsistent formats:** Data may be entered inconsistently. For instance, dates may be formatted differently. You may need to convert all dates to a consistent format.
- **Outliers:** Outliers are data points that fall far outside the expected range for a particular attribute. Outliers can skew your machine learning model's results, so you may need to decide how to handle them. You can choose to remove them, minorize them (cap them at a certain value), or transform them.

4.3.2 Data Formatting: Machine learning algorithms typically require numerical data. You may need to format your data accordingly. Here are some common formatting techniques:

- **Label Encoding:** This technique converts categorical data (e.g., text) into numerical values. For instance, you may convert the values of a "gender" attribute from "male" and "female" to 1 and 2, respectively.



```
:n [47]:    # ecoding prognonsis
            le = LabelEncoder()
            le.fit(y)
            Y = le.transform(y)


            print(Y)

            [15 15 15 ...  7 32 28]
```

Fig 4.3.2: *Label Encoding*

4.4 Feature Selection

This step involves selecting a subset of features from the data that are most likely to be relevant to the prediction task. Here are some reasons to perform feature selection:

- **Improved Model Performance:** By selecting only relevant features, you can potentially improve the accuracy and efficiency of your machine learning model. Irrelevant features can add noise to the data and make it harder for the model to learn the underlying patterns.
- **Reduced Training Time:** Training a machine learning model with fewer features can take less time and computational resources.
- **Reduced Overfitting:** Overfitting is a condition where a machine learning model performs well on the training data but poorly on unseen data. Feature selection can help to reduce overfitting by removing irrelevant features that the model may latch onto and use to make inaccurate predictions on new data.

There are several techniques for feature selection, including:

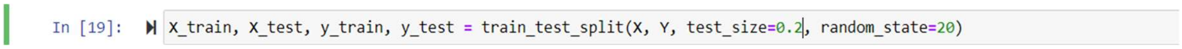
- **Filter Methods:** These methods rank features based on a statistical measure like correlation with the target variable. Features that score low according to the ranking criteria are excluded.

- **Wrapper Methods:** These methods involve training multiple models, each with a different subset of features. The model with the best performance is selected, and the features it uses are considered the most relevant.
- **Embedded Methods:** These methods are embedded within the machine learning model itself and perform feature selection as part of the model training process.

4.5 Train Test Split

The train-test split is a technique for evaluating the performance of a machine learning model. The data is split into two sets: a training set and a test set. The training set is used to train the machine learning model, and the test set is used to evaluate the performance of the model on unseen data. The size of the training and test sets is typically 80/20, but this can vary depending on the size of the data set and the complexity of the machine learning model.

- **Training set:** The training set is used to train the machine learning model. The model learns the patterns in the data and builds a model that can be used to make predictions on new data.
- **Test set:** The test set is used to evaluate the performance of the machine learning model on unseen data. The model is applied to the test set, and its predictions are compared to the actual values. This helps to assess how well the model is likely to perform on real-world data.



```
In [19]:  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=20)
```

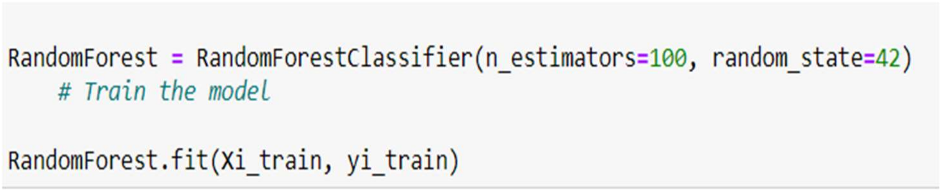
Fig 4.5: *Train Test Split*

4.6 Train models

In this stage, machine learning models are trained on the training data. The image shows two algorithms that can be used: Random Forest and SVM.

- **Random Forest:** Random forest is a machine learning algorithm that is based on the idea of ensemble learning. Ensemble learning involves training multiple machine learning models on

the data and then combining the predictions of these models. Random forest trains multiple decision tree models on random subsets of the training data.

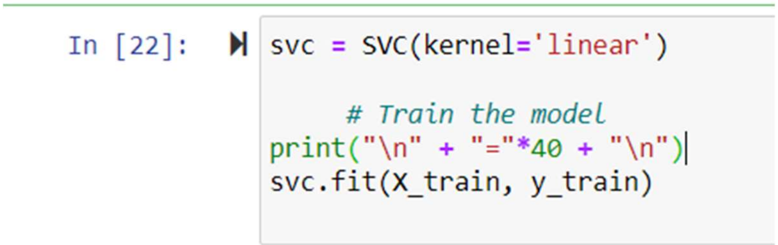


```
RandomForest = RandomForestClassifier(n_estimators=100, random_state=42)
    # Train the model

RandomForest.fit(Xi_train, yi_train)
```

Fig 4.6.1: *Random Forest*

- **SVM (Support Vector Machine):** A support vector machine (SVM) is a machine learning algorithm that can be used for classification tasks. SVMs work by finding a hyperplane that separates the data points of one class from the data points of another class. The hyperplane is a decision boundary that the SVM can use to classify new data points. SVMs are effective for classification tasks with high-dimensional data.



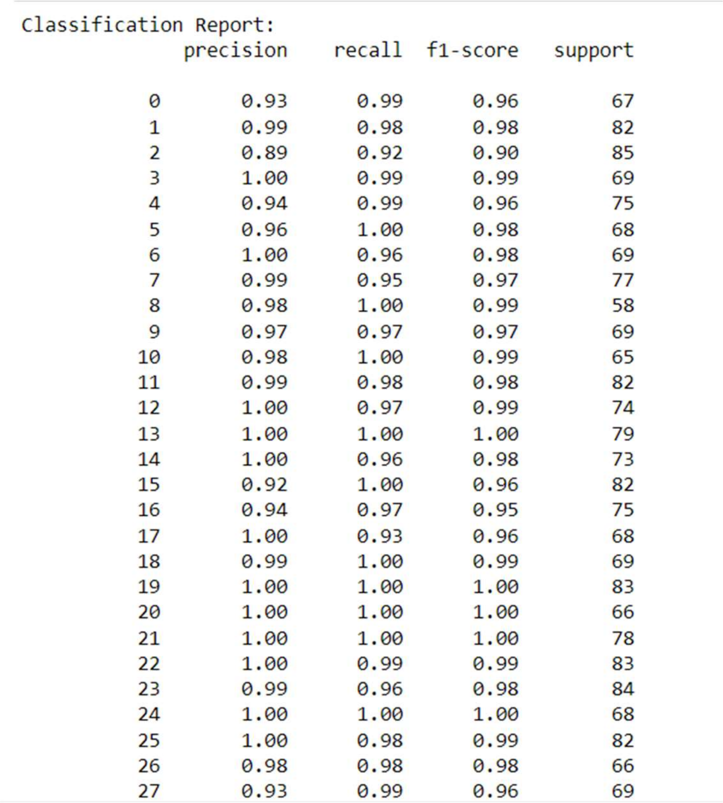
```
In [22]:    svc = SVC(kernel='linear')

                 # Train the model
            print("\n" + "="*40 + "\n")
            svc.fit(X_train, y_train)
```

__Fig 4.6.2:__ Support Vector Machine

4.7 Model Evaluation

Once the models are trained, they are evaluated on the test data. The model evaluation involves comparing the predictions of the model to the actual values in the test set. There are a number of different metrics that can be used to evaluate the performance of a machine learning model.



```
Classification Report:
               precision    recall  f1-score   support

           0       0.93      0.99      0.96        67
           1       0.99      0.98      0.98        82
           2       0.89      0.92      0.90        85
           3       1.00      0.99      0.99        69
           4       0.94      0.99      0.96        75
           5       0.96      1.00      0.98        68
           6       1.00      0.96      0.98        69
           7       0.99      0.95      0.97        77
           8       0.98      1.00      0.99        58
           9       0.97      0.97      0.97        69
          10       0.98      1.00      0.99        65
          11       0.99      0.98      0.98        82
          12       1.00      0.97      0.99        74
          13       1.00      1.00      1.00        79
          14       1.00      0.96      0.98        73
          15       0.92      1.00      0.96        82
          16       0.94      0.97      0.95        75
          17       1.00      0.93      0.96        68
          18       0.99      1.00      0.99        69
          19       1.00      1.00      1.00        83
          20       1.00      1.00      1.00        66
          21       1.00      1.00      1.00        78
          22       1.00      0.99      0.99        83
          23       0.99      0.96      0.98        84
          24       1.00      1.00      1.00        68
          25       1.00      0.98      0.99        82
          26       0.98      0.98      0.98        66
          27       0.93      0.99      0.96        69
```

__Fig 4.7:__ Model Evaluation

- **Accuracy:** Accuracy is the most common metric used to evaluate the performance of a classification model. It is the proportion of predictions that the model gets correct. While accuracy is a good starting point, it can be misleading in some cases. For instance, if a disease is very rare, a model that always predicts "no disease" might have a high accuracy, but it wouldn't be very useful for disease prediction.

- **Precision:** Precision is the proportion of positive predictions that are actually correct. In disease prediction, this would be the percentage of times the model predicts a disease and the patient actually has the disease.
- **Recall:** Recall is the proportion of actual positive cases that are correctly identified by the model. In disease prediction, this would be the percentage of patients with the disease that the model correctly identifies.
- **F1-Score:** The F1-score is a harmonic mean of precision and recall. It takes both precision and recall into account and is a good measure of a model's overall performance when dealing with imbalanced class distributions

4.8 Disease Prediction

This is the core functionality of the system. After the user enters their symptoms, the system uses the machine learning model to predict the most likely disease the user has. The machine learning model is a complex mathematical model that has been trained on the data in the database.

4.9 Report Generation

After the disease prediction is made, the system generates a report that summarizes the system's findings. This report likely includes:

- Predicted Disease: The disease that the model predicts the user is most likely to have.
- Confidence Level: This is a numerical value that indicates the model's certainty in its prediction. A higher confidence level indicates that the model is more certain about its prediction.

5 TESTING

5.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the.

Principles of Testing: -

- All the tests should meet the customer requirements.
- To make our software testing should be performed by a third party.
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the tests to be conducted should be planned before implementing it
- It follows the Pareto rule (80/20 rule) which states that 80% of errors come from20% of program components.
- Start testing with small parts and extend it to large parts

5.2 TYPES OF TESTING

5.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

5.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

- Functional testing is centered on the following items:
- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedure: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.2.5 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases have passed successfully. No defects encountered.

5.2.6 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.2.7 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works

6.RESULT

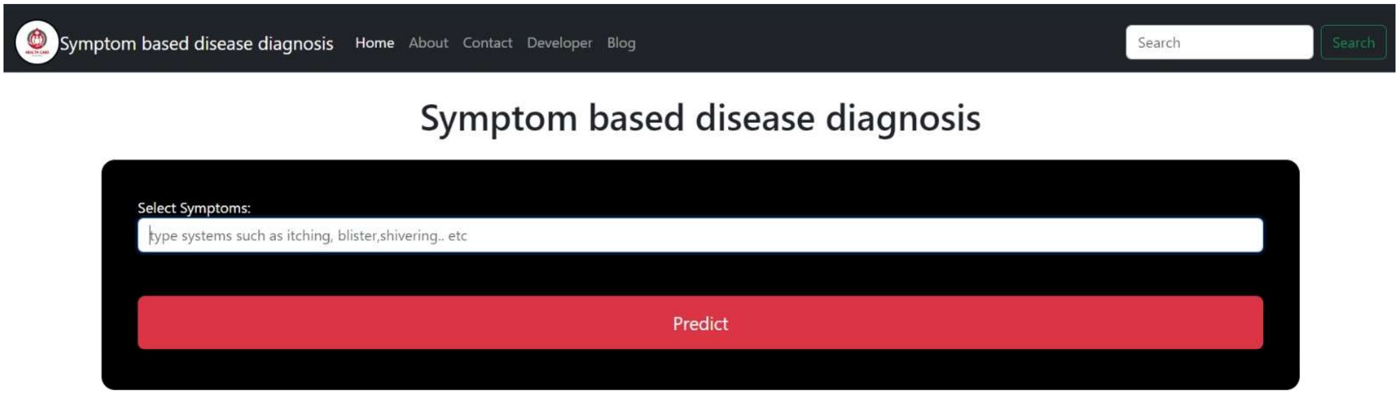

Symptom based disease diagnosis
Home
About
Contact
Developer
Blog
Search
Search
Symptom based disease diagnosis
Select Symptoms:
type systems such as itching, blister,shivering.. etc
Predict

Fig 6.1: Home page of user interface

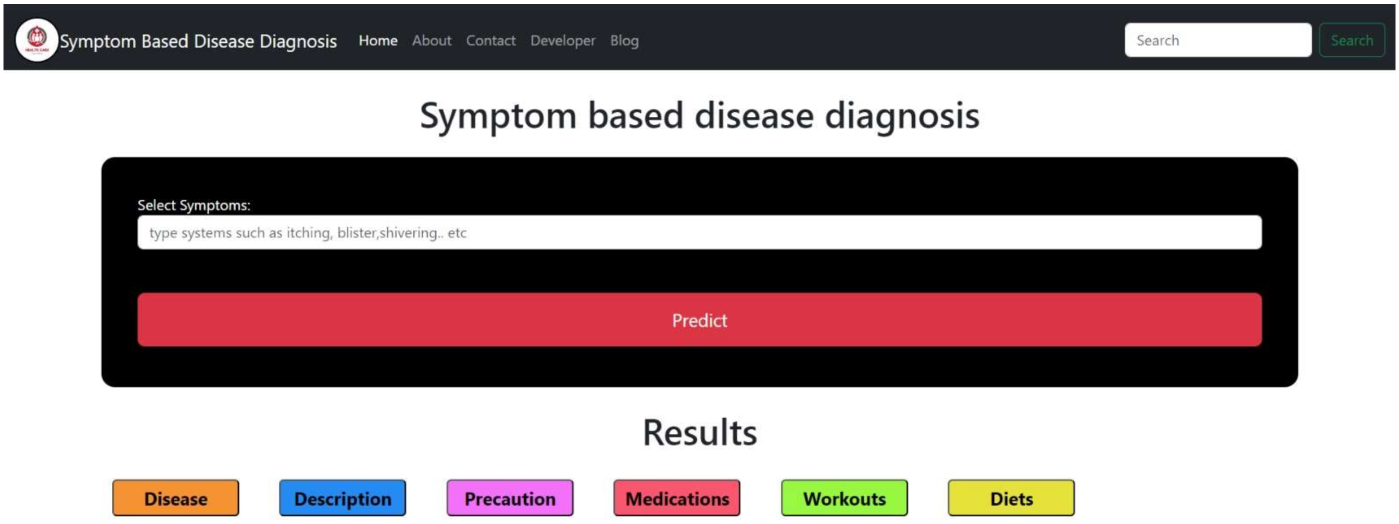
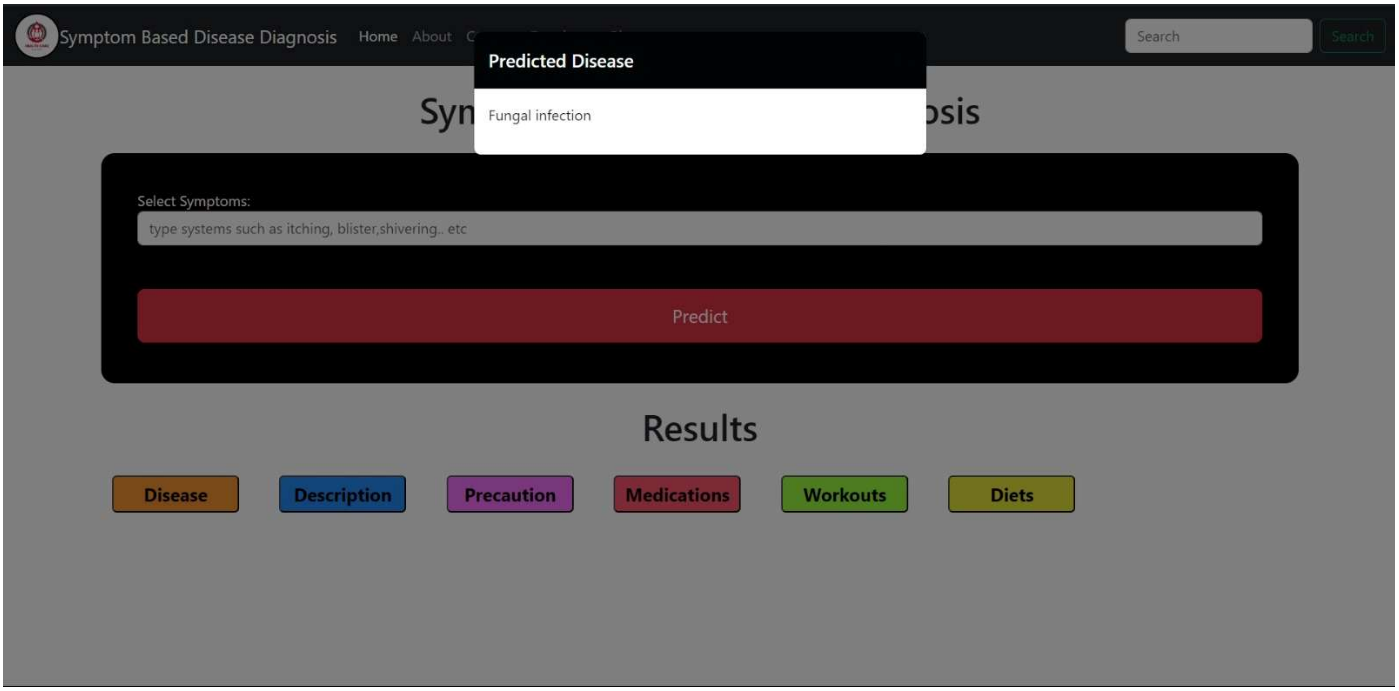


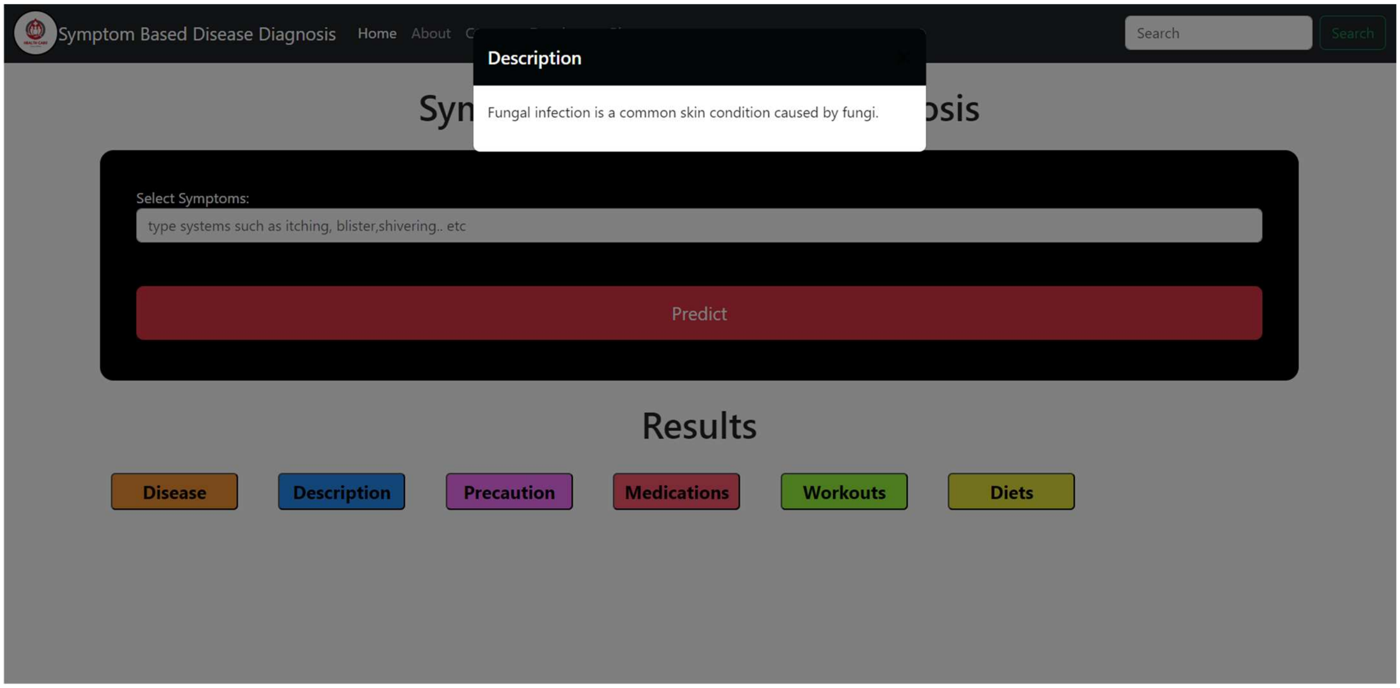
Fig 6.2: Result User interface



Predicted Disease

Fungal infection

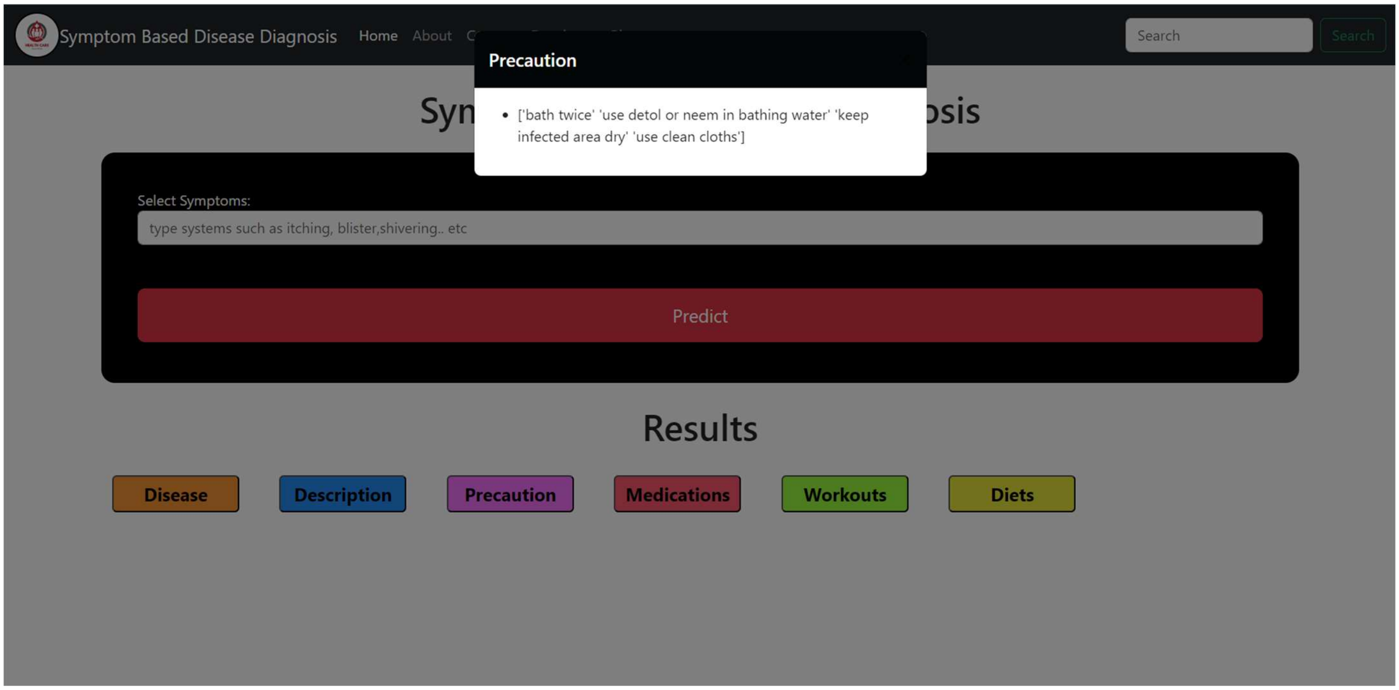
Fig 6.3: *Result Disease Name*



Description

Fungal infection is a common skin condition caused by fungi.

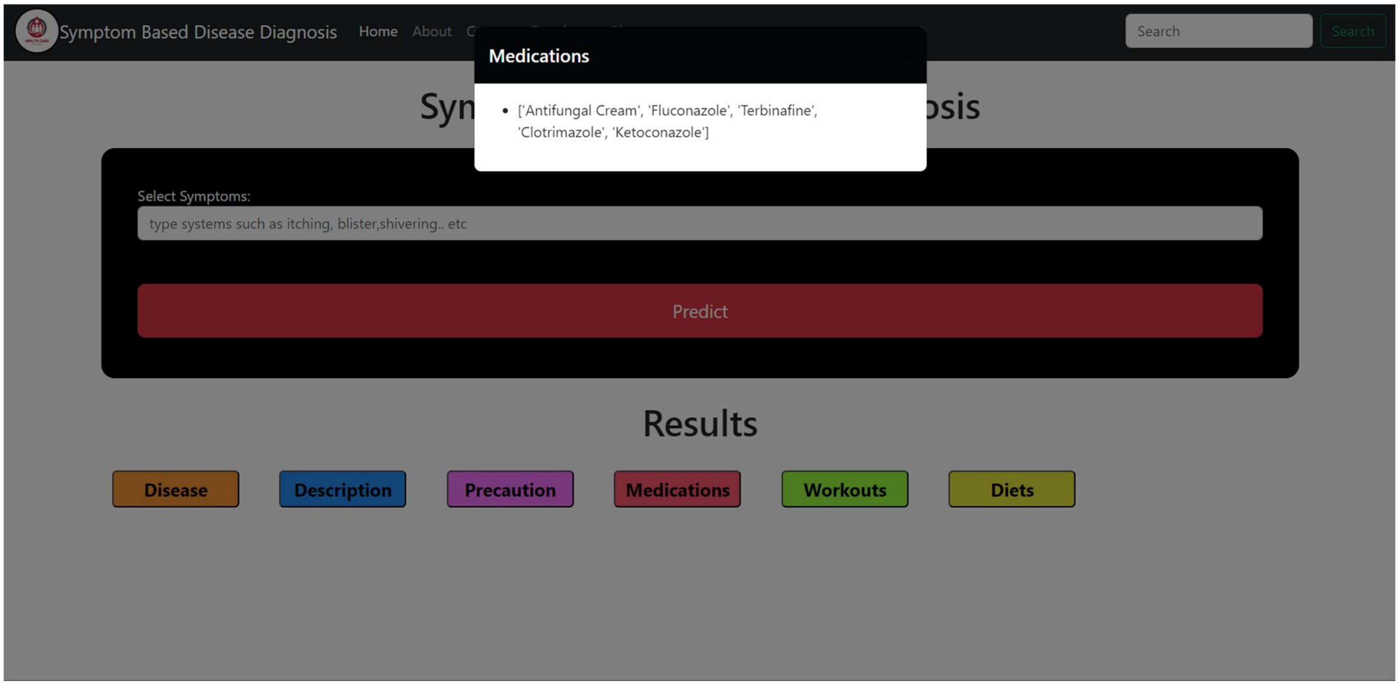
Fig 6.4: *Result Description*



Precaution

- ['bath twice' 'use detol or neem in bathing water' 'keep infected area dry' 'use clean cloths']

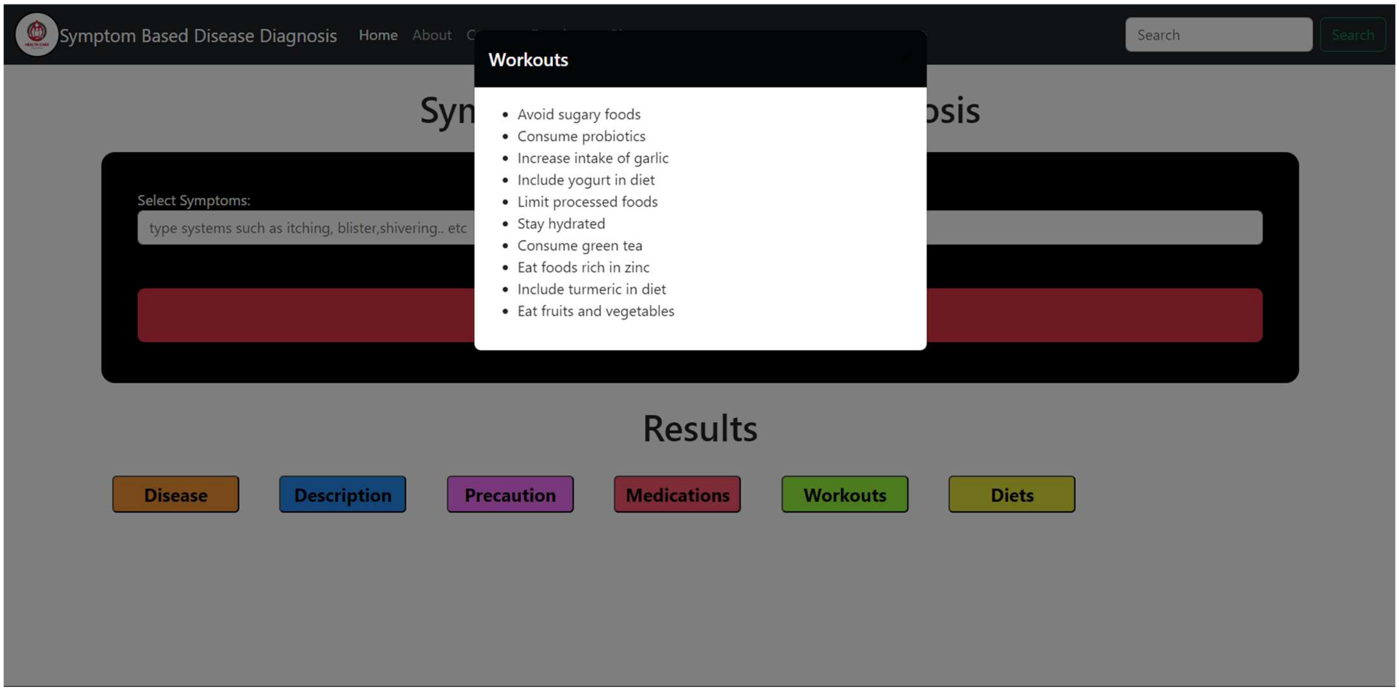
Fig 6.5: *Result Precautions*



Medications

- ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']

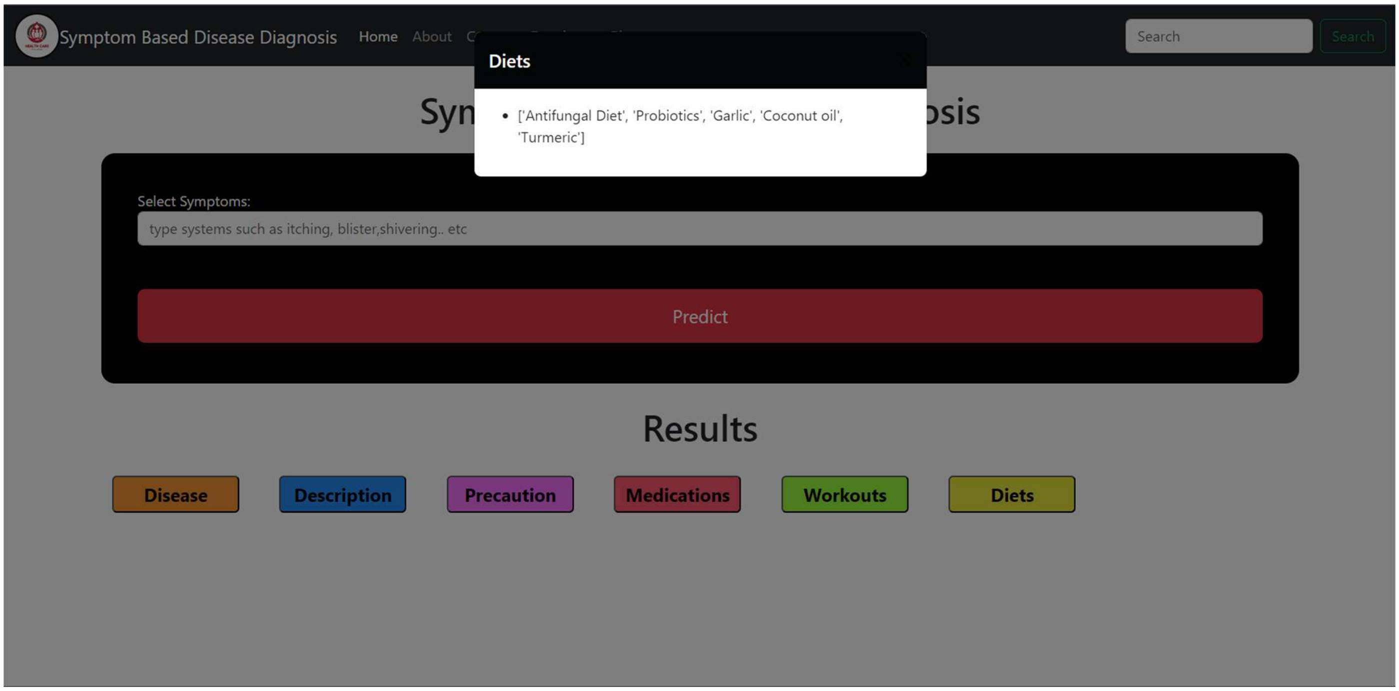
Fig 6.6: *Result Medications*



Workouts

- Avoid sugary foods
- Consume probiotics
- Increase intake of garlic
- Include yogurt in diet
- Limit processed foods
- Stay hydrated
- Consume green tea
- Eat foods rich in zinc
- Include turmeric in diet
- Eat fruits and vegetables

Fig 6.7: *Result Workouts*



Diets

- ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']

Fig 6.8: *Results Diets*

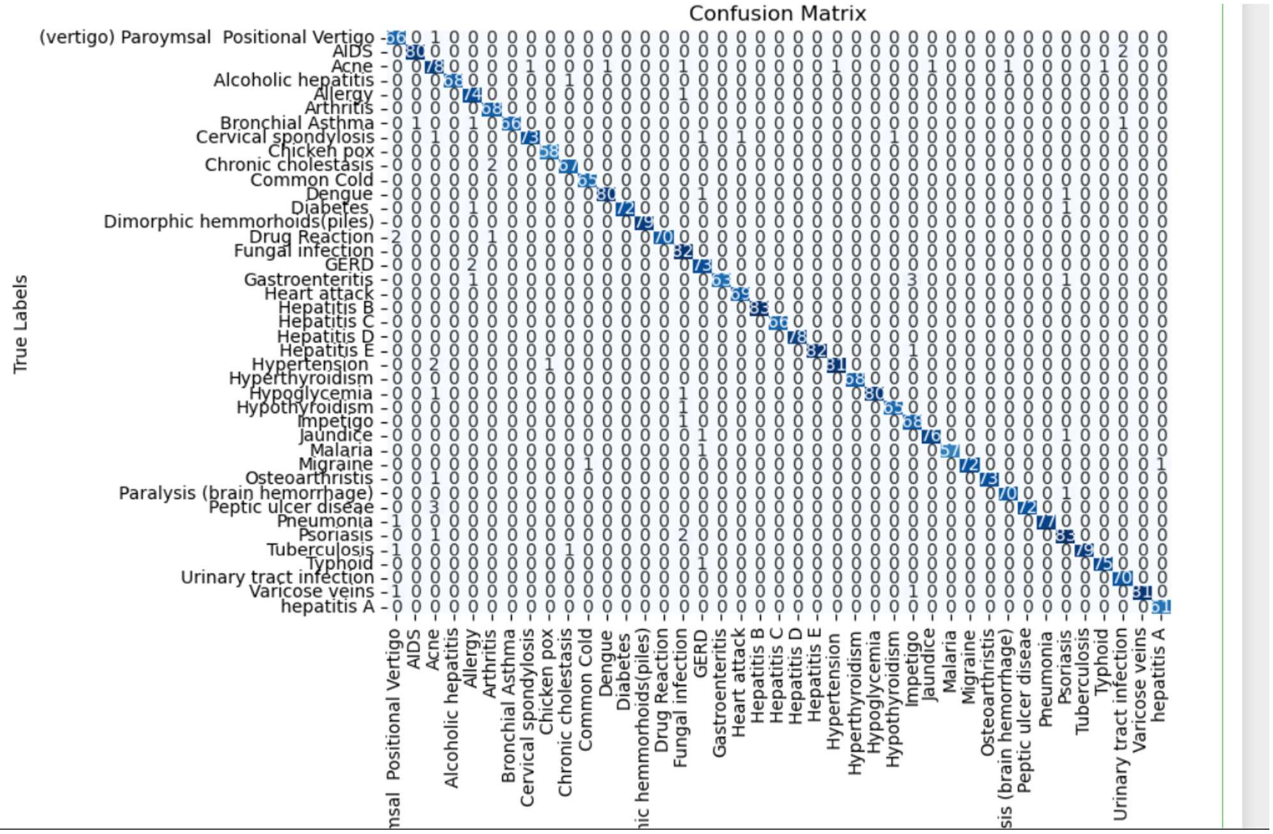


__Fig 6.9:__ Confusion Matrix

7.CONCLUSION

Machine learning offers a promising approach for symptom-based disease prediction. By analysing vast amounts of medical data, these systems can identify patterns and relationships between symptoms and specific diseases. This can lead to:

- **Earlier disease detection:** Machine learning can highlight potential diseases based on initial symptoms, prompting users to seek professional medical advice sooner.
- **Improved healthcare efficiency:** By suggesting potential diagnoses, machine learning can streamline the diagnostic process for doctors.
- **Personalized guidance:** The system can provide users with initial information about the predicted disease, including descriptions, precautions, medications (if appropriate), recommended workouts or dietary adjustments to support healing (with disclaimers to consult a doctor for confirmation and treatment plans).

In conclusion, symptom-based disease prediction using machine learning holds significant potential to improve early detection, enhance healthcare efficiency, and empower users with initial information for informed decisions. However, it is crucial to use this technology as a complementary tool alongside professional medical evaluation and treatment plans.

8.REFERENCES

1. **Medicine Recommendation System:** Varun A. Goyal, Dilip J. Parmar, Namaskar I. Joshi, Prof. Komal Champanerkar

2. **Health Care Application using Machine Learning and Deep Learning:** Ajinkya Padule, Aman Patel, Arsalan Patel, Aman Shaikh, Jyoti Gavhane

3. **Smart Health Prediction Using Machine Learning:** Naveenkumar S, Kirubhakaran R, Jeeva G, Shobana M, Sangeetha K

4. **Digital Wellness: A Smart Health Care System Using Machine Learning** Anargha P A, Justin Joseph, Sam Sunil, Sanjai S, Divya Mohan

5. **EXPLORING THE POSSIBILITIES OF USING MACHINE LEARNING IN HEALTH CARE:** Dheeraj B. Shukla, Ajahar I. Pathan, Rijawan A. Shaik, Vasim I. Memon

6. **Health Care Digitalization and Machine Learning** Ajinkya Yelne and Archana Raut

7. **Machine Learning in Healthcare:** Hafsa Habehh and Suril Gohel

8. **HEART DISEASE PREDICTION SYSTEM:** Mrs. Jayashree L K, Sushmita Makapur, T D Vyshnavi, T D Prathyusha, V K Kavya.

9. **Medicine Recommendation System:** Varun A. Goyal, Dilip J. Parmar, Namaskar I. Joshi, Prof. Komal Champanerkar

10. **An Intelligent Medicine Recommender System Framework** Bao, Y. and Jiang, X. IEEE 11th Conference on Industrial Electronics and Applications (ICIEA).

11. **The recommendation of medicines based on multiple criteria decision making and domain ontology** — An example of anti-diabetic medicines, 27–32 Chen, R.-C., Chiu, J. Y., and Batj, C. T.

12. **Ontology-based drug recommendations discovery:** Doulaverakis, C., Nikolaidis, G., Kleontas, A., and Kompatsiaris, I. GalenOWL. Journal of Biomedical Semantics.

13. **Web-based application for assisted decision-making in treatment of diabetes:** Medvedeva, O., Knox, T., and Paul, J. 2007. DiaTrack. s. Journal of Computing Sciences in Colleges.

14. **Medicine Recommender Systems (IJACSA):** Benjamin Stark, Constanze Knahl, Mert Aydin, Karim Elish. International Journal of Advanced Computer Science and Applications

APPENDIX

A.SAMPLE CODE

I. FRONT-END CODE

(i) developers.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"><html><head><META http-equiv="Content-Type" content="text/html;
charset=utf-8"><style></style></head><body><u></u>

<div>

<div class="m_container m_my-4 m_mt-4">

<h6>    About P.Jaya Prakash Rao, G.Shiva Sanjana Reddy ,Y.Adithya Vardhan Reddy

</h6>

<h5>Our Expertise

</h5>

<p>With a strong academic background in computer science and a deep fascination for AI and
ML, we embarked on a journey that has taken me across the globe, working on diverse and
innovative projects. Over the years, we&#39;ve had the privilege of collaborating on
international projects that have pushed the boundaries of what&#39;s possible in AI and ML.

</p>

<h5>A Commitment to Excellence

</h5>

<p>Our work is driven by a commitment to excellence and a belief that technology should be
accessible and beneficial to everyone. Whether it&#39;s developing predictive models,
creating intelligent algorithms, or designing user-friendly interfaces, I strive for solutions that
are both cutting-edge and user-centric.

</p>

<h5>Passion for Health Tech

</h5>

<p>The development of SYMPTOMS BASED DISEASE DIAGNOSIS USING MACHINE
LEARNING represents a convergence of my passion for AI and our dedication to improving
healthcare accessibility. we believe that AI has the potential to transform the way we approach
healthcare, making it more personalized and informative. This platform is a testament to that
vision.
```

```
</p>

<h5>Join Us on this Journey

</h5>

<p> We invite you to explore SYMPTOMS BASED DISEASE DIAGNOSIS USING
MACHINE LEARNING and experience firsthand the fusion of AI and healthcare. Together,
we can empower individuals with knowledge, promote well-being, and contribute to a brighter
and healthier future.

</p>

</div>

</div>

</body></html>
```

(ii) index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"><html><head><META http-equiv="Content-Type" content="text/html;
charset=utf-8"><style></style></head><body><u></u>

<div>

<u></u>

<div class="m_container-fluid">

 <div class="m_logo">

<img class="m_myimg" alt="">

</div>

 <a class="m_navbar-brand" href="#m__" rel="noreferrer">Symptom Based Disease
Diagnosis </a>

<button class="m_navbar-toggler" type="button" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">

 <span class="m_navbar-toggler-icon"></span>

 </button>

 <div class="m_collapse m_navbar-collapse" id="m_navbarSupportedContent">

 <ul class="m_navbar-nav m_me-auto m_mb-2 m_mb-lg-0">
 <li class="m_nav-item">

 <a class="m_nav-link m_active" href="#m__" rel="noreferrer">Home</a>

                </li>

                <li class="m_nav-item">

                    <a class="m_nav-link"
href="https://www.google.com/url?q=http:///about&amp;source=gmail-
html&amp;ust=1716727317633000&amp;usg=AOvVaw1lbCE6lqfkTrFxXMLa5RlJ"
target="_blank" rel="noreferrer">About</a>

                </li>

                <li class="m_nav-item">

                    <a class="m_nav-link"
href="https://www.google.com/url?q=http:///contact&amp;source=gmail-
html&amp;ust=1716727317633000&amp;usg=AOvVaw0yNwrN6vudd4mWcUquYD0x"
target="_blank" rel="noreferrer">Contact</a>

                </li>

                <li class="m_nav-item">

                    <a class="m_nav-link"
href="https://www.google.com/url?q=http:///developer&amp;source=gmail-
html&amp;ust=1716727317633000&amp;usg=AOvVaw1vaOlihydfEnIq96MLeJ1R"
target="_blank" rel="noreferrer">Developer</a>

                </li>

                <li class="m_nav-item">

                    <a class="m_nav-link"
href="https://www.google.com/url?q=http:///blog&amp;source=gmail-
html&amp;ust=1716727317633000&amp;usg=AOvVaw3ASik0R9OkRGD9LDLXUmWd"
target="_blank" rel="noreferrer">Blog</a>

                </li>

            </ul>

            <form class="m_d-flex" role="search" target="_blank">

                <input class="m_form-control m_me-2" type="search" aria-label="Search">

                <button class="m_btn m_btn-outline-success" type="submit">Search</button>

            </form>

        </div>
        </div>

      <u></u>

    <h1 class="m_mt-4 m_my-4 m_text-center m_text-green">Symptom based disease
diagnosis</h1>

    <div class="m_container m_my-4 m_mt-4" style="background:black;color:white;border-
radius:15px;padding:40px">

        <form action="http:///predict" method="post" target="_blank">

            <div class="m_form-group">

                <label for="m_symptoms">Select Symptoms:</label>

                <input type="text" class="m_form-control" id="m_symptoms" name="symptoms">



            </div>

            <br>

<div name="mysysms" id="m_transcription"></div>

            {% if message %}

            <p>{{ message }}</p>

             {% endif %}

            <br>

            <button type="submit" class="m_btn m_btn-danger m_btn-lg"
style="width:100%;padding:14px;margin-bottom:5px">Predict</button>

        </form>

    </div>

      {% if predicted_disease %}

     <h1 class="m_text-center m_my-4 m_mt-4">Results</h1>

      <div class="m_container">

          <div class="m_result-container">

<button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-
size:20px;font-weight:bold;width:140px;border-
radius:5px;background:#f39334;color:black">Disease</button>
```

```
            <button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-size:20px;font-weight:bold;width:140px;border-radius:5px;background:#268af3;color:black">Description</button>

            <button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-size:20px;font-weight:bold;width:140px;border-radius:5px;background:#f371f9;color:black">Precaution</button>

            <button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-size:20px;font-weight:bold;width:140px;border-radius:5px;background:#f8576f;color:black">Medications</button>

            <button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-size:20px;font-weight:bold;width:140px;border-radius:5px;background:#99f741;color:black">Workouts</button>

            <button class="m_toggle-button" style="padding:4px;margin:5px 40px 5px 0;font-size:20px;font-weight:bold;width:140px;border-radius:5px;background:#e5e23d;color:black">Diets</button>

        </div>

    </div>

{% endif %}

<div class="m_modal m_fade" id="m_diseaseModal" aria-labelledby="diseaseModalLabel" aria-hidden="true">

        <div class="m_modal-dialog">

            <div class="m_modal-content">

                <div class="m_modal-header" style="background-color:#020606;color:white">

                    <h5 class="m_modal-title" id="m_diseaseModalLabel">Predicted Disease</h5>

                    <button type="button" class="m_btn-close" aria-label="Close"></button>

                </div>

                <div class="m_modal-body" style="background-color:#modal-body-color">

                    <p>{{ predicted_disease }}</p>

                </div>

            </div>

        </div>

    </div>
```

```
<div class="m_modal m_fade" id="m_descriptionModal" aria-labelledby="descriptionModalLabel" aria-hidden="true">

    <div class="m_modal-dialog">

      <div class="m_modal-content">

        <div class="m_modal-header" style="background-color:#020606;color:white">

          <h5 class="m_modal-title" id="m_descriptionModalLabel">Description</h5>

          <button type="button" class="m_btn-close" aria-label="Close"></button>

        </div>

        <div class="m_modal-body">

          <p>{{ dis_des }}</p>

        </div>

      </div>

    </div>

  </div>



  <div class="m_modal m_fade" id="m_precautionModal" aria-labelledby="precautionModalLabel" aria-hidden="true">

    <div class="m_modal-dialog">

      <div class="m_modal-content">

        <div class="m_modal-header" style="background-color:#020606;color:white">

          <h5 class="m_modal-title" id="m_precautionModalLabel">Precaution</h5>

          <button type="button" class="m_btn-close" aria-label="Close"></button>

        </div>

        <div class="m_modal-body">

          <ul>

            {% for i in dis_pre %}

              <li>{{ i }}</li>

            {% endfor %}
```

```
                    </ul>
                </div>
            </div>
        </div>
    </div>


    <div class="m_modal m_fade" id="m_medicationsModal" aria-labelledby="medicationsModalLabel" aria-hidden="true">
        <div class="m_modal-dialog">
            <div class="m_modal-content">
                <div class="m_modal-header" style="background-color:#020606;color:white">
                    <h5 class="m_modal-title" id="m_medicationsModalLabel">Medications</h5>
                    <button type="button" class="m_btn-close" aria-label="Close"></button>
                </div>
                <div class="m_modal-body">
                    <ul>
                        {% for i in dis_med %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>
         <div class="m_modal m_fade" id="m_workoutsModal" aria-labelledby="workoutsModalLabel" aria-hidden="true">
        <div class="m_modal-dialog">
            <div class="m_modal-content">
```

```
                <div class="m_modal-header" style="background-color:#020606;color:white">
                    <h5 class="m_modal-title" id="m_workoutsModalLabel">Workouts</h5>
                    <button type="button" class="m_btn-close" aria-label="Close"></button>
                </div>
                <div class="m_modal-body">
                    <ul>
                        {% for i in dis_wrkout %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>


    <div class="m_modal m_fade" id="m_dietsModal" aria-labelledby="dietsModalLabel" aria-hidden="true">
        <div class="m_modal-dialog">
            <div class="m_modal-content">
                <div class="m_modal-header" style="background-color:#020606;color:white">
                    <h5 class="m_modal-title" id="m_dietsModalLabel">Diets</h5>
                    <button type="button" class="m_btn-close" aria-label="Close"></button>
                </div>
                <div class="m_modal-body">
                    <ul>
                        {% for i in dis_die %}
                            <li>{{ i }}</li>
                    {% endfor %}
                  </ul>
                </div>
              </div>
            </div>
          </div>
        </div>
        </body></html>
```

(iii) scraping.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"><html><head><META http-equiv="Content-Type"
content="text/html; charset=utf-8"><style></style></head><body><u></u>

<div>

  <div class="m_container m_my-4 m_mt-4">

    <h2 class="m_text-center"> Building a Symptom-Based Disease Diagnosis
Web App with Flask and Machine Learning</h2>

    <p>In the age of technology and information, access to accurate and timely
healthcare is more critical than ever. With the increasing importance of remote
healthcare solutions, we embarked on a journey to develop a symptom-based
disease diagnosis web application. Leveraging Flask for the backend and a
Decision Tree Classifier model, we created a user-friendly platform that can help
users identify potential illnesses based on their reported symptoms.</p>

    <h3>The Problem</h3>

    <p>The project began with recognizing a common issue: people often
experience symptoms and want quick answers about their health concerns. It can
be challenging to differentiate between various diseases, especially when
symptoms overlap. Our goal was to provide a convenient solution for users to
input their symptoms and receive potential diagnoses.</p>

    <h3>The Solution</h3>

    <p>We developed a web app that allows users to enter a list of symptoms
```

```
they are experiencing. The app then uses a pre-trained Decision Tree Classifier model to predict the most likely disease based on the provided symptoms. Here&#39;s how it works:</p>

    <ol>

        <li><b><strong>Symptom Input:</strong></b> Users enter their symptoms through a user-friendly interface. The web app supports a wide range of symptoms, making it versatile for different scenarios.</li>

        <li><b><strong>Machine Learning Model:</strong></b> We trained a Decision Tree Classifier using a dataset containing symptoms and corresponding diseases. The model is capable of predicting diseases based on input symptoms.</li>

        <li><b><strong>Prediction:</strong></b> The app uses the model to predict the most likely disease, providing users with instant information about potential health concerns.</li>

        <li><b><strong>Additional Information: </strong></b> To enhance user experience, the app also provides additional information about the predicted disease. This includes a description of the disease, recommended precautions, medications, dietary advice, and workout tips.</li>

    </ol>

    <h3>Key Features</h3>

    <ul type="circle">

        <li><b><strong>User-Friendly Interface:</strong></b> The web app boasts an intuitive and easy-to-navigate interface, ensuring that users can input their symptoms without any hassle.</li>

        <li><b><strong>Accurate Predictions:</strong></b> Leveraging machine learning, our model provides accurate disease predictions based on the symptoms provided by the user.</li>

        <li><b><strong>Comprehensive Information: </strong></b>In addition to predictions, the app offers comprehensive information about the predicted disease, including descriptions, precautions, medications, diet recommendations, and workout tips.</li>

        <li><b><strong>ducational Content:</strong></b> To empower users with knowledge, the app includes informative blog posts about various health topics.</li>
        </ul>

        <br>

        <h3>Conclusion</h3>

        <p>Our Symptom-Based Disease Diagnosis Web App brings the power of
machine learning and healthcare information to the fingertips of users. It serves as
a valuable resource for individuals looking to gain insights into their health
conditions quickly and conveniently. By combining technology and healthcare,
we aim to make healthcare more accessible and user-centric.

In an era where remote healthcare solutions are gaining prominence, this project
represents a significant step forward in providing accessible and reliable
healthcare information to the masses. We hope that this project will contribute to
better health awareness and ultimately improve the well-being of individuals
worldwide.</p>

    </div>

  </div>

</body></html>
```

II. BACKEND CODE

```
from flask import Flask,request,render_template
import numpy as np
import pandas as pd
import pickle


#load database
sym_des = pd.read_csv("dataset/symtoms_df.csv")
precautions = pd.read_csv("dataset/precautions_df.csv")
workout = pd.read_csv("dataset/workout_df.csv")
description = pd.read_csv("dataset/description.csv")
medications = pd.read_csv('dataset/medications.csv')
diets = pd.read_csv("dataset/diets.csv")

#load Model
svc = pickle.load(open('models/svc.pkl','rb'))

app = Flask(__name__)

#============================================================
# custome and helping functions
#==========================helper funtions================
def helper(dis):
    desc = description[description['Disease'] == dis]['Description']
    desc = " ".join([w for w in desc])
    pre = precautions[precautions['Disease'] == dis][['Precaution_1',
'Precaution_2', 'Precaution_3', 'Precaution_4']]
    pre = [col for col in pre.values]

    med = medications[medications['Disease'] == dis]['Medication']
    med = [med for med in med.values]

    die = diets[diets['Disease'] == dis]['Diet']
    die = [die for die in die.values]

    wrkout = workout[workout['disease'] == dis] ['workout']
    return desc,pre,med,die,wrkout


symptoms_dict = {'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2,
'continuous_sneezing': 3, 'shivering': 4, 'chills': 5, 'joint_pain': 6,
'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue': 9, 'muscle_wasting':
10, 'vomiting': 11, 'burning_micturition': 12, 'spotting_ urination': 13,
'fatigue': 14, 'weight_gain': 15, 'anxiety': 16, 'cold_hands_and_feets':
17, 'mood_swings': 18, 'weight_loss': 19, 'restlessness': 20, 'lethargy':
21, 'patches_in_throat': 22, 'irregular_sugar_level': 23, 'cough': 24,
'high_fever': 25, 'sunken_eyes': 26, 'breathlessness': 27, 'sweating':
28, 'dehydration': 29, 'indigestion': 30, 'headache': 31,
'yellowish_skin': 32, 'dark_urine': 33, 'nausea': 34, 'loss_of_appetite':
35, 'pain_behind_the_eyes': 36, 'back_pain': 37, 'constipation': 38,
'abdominal_pain': 39, 'diarrhoea': 40, 'mild_fever': 41, 'yellow_urine':
42, 'yellowing_of_eyes': 43, 'acute_liver_failure': 44, 'fluid_overload':
45, 'swelling_of_stomach': 46, 'swelled_lymph_nodes': 47, 'malaise': 48,
'blurred_and_distorted_vision': 49, 'phlegm': 50, 'throat_irritation':
51, 'redness_of_eyes': 52, 'sinus_pressure': 53, 'runny_nose': 54,
'congestion': 55, 'chest_pain': 56, 'weakness_in_limbs': 57,
'fast_heart_rate': 58, 'pain_during_bowel_movements': 59,
'pain_in_anal_region': 60, 'bloody_stool': 61, 'irritation_in_anus': 62,
'neck_pain': 63, 'dizziness': 64, 'cramps': 65, 'bruising': 66,
'obesity': 67, 'swollen_legs': 68, 'swollen_blood_vessels': 69,
'puffy_face_and_eyes': 70, 'enlarged_thyroid': 71, 'brittle_nails': 72,
'swollen_extremeties': 73, 'excessive_hunger': 74,
'extra_marital_contacts': 75, 'drying_and_tingling_lips': 76,
'slurred_speech': 77, 'knee_pain': 78, 'hip_joint_pain': 79,
'muscle_weakness': 80, 'stiff_neck': 81, 'swelling_joints': 82,
'movement_stiffness': 83, 'spinning_movements': 84, 'loss_of_balance':
85, 'unsteadiness': 86, 'weakness_of_one_body_side': 87, 'loss_of_smell':
88, 'bladder_discomfort': 89, 'foul_smell_of urine': 90,
'continuous_feel_of_urine': 91, 'passage_of_gases': 92,
'internal_itching': 93, 'toxic_look_(typhos)': 94, 'depression': 95,
'irritability': 96, 'muscle_pain': 97, 'altered_sensorium': 98,
'red_spots_over_body': 99, 'belly_pain': 100, 'abnormal_menstruation':
101, 'dischromic _patches': 102, 'watering_from_eyes': 103,
'increased_appetite': 104, 'polyuria': 105, 'family_history': 106,
'mucoid_sputum': 107, 'rusty_sputum': 108, 'lack_of_concentration': 109,
'visual_disturbances': 110, 'receiving_blood_transfusion': 111,
'receiving_unsterile_injections': 112, 'coma': 113, 'stomach_bleeding':
114, 'distention_of_abdomen': 115, 'history_of_alcohol_consumption': 116,
'fluid_overload.1': 117, 'blood_in_sputum': 118,
'prominent_veins_on_calf': 119, 'palpitations': 120, 'painful_walking':
121, 'pus_filled_pimples': 122, 'blackheads': 123, 'scurring': 124,
'skin_peeling': 125, 'silver_like_dusting': 126, 'small_dents_in_nails':
```

```
127, 'inflammatory_nails': 128, 'blister': 129, 'red_sore_around_nose':
130, 'yellow_crust_ooze': 131}
diseases_list = {15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9:
'Chronic cholestasis', 14: 'Drug Reaction', 33: 'Peptic ulcer diseae', 1:
'AIDS', 12: 'Diabetes ', 17: 'Gastroenteritis', 6: 'Bronchial Asthma',
23: 'Hypertension ', 30: 'Migraine', 7: 'Cervical spondylosis', 32:
'Paralysis (brain hemorrhage)', 28: 'Jaundice', 29: 'Malaria', 8:
'Chicken pox', 11: 'Dengue', 37: 'Typhoid', 40: 'hepatitis A', 19:
'Hepatitis B', 20: 'Hepatitis C', 21: 'Hepatitis D', 22: 'Hepatitis E',
3: 'Alcoholic hepatitis', 36: 'Tuberculosis', 10: 'Common Cold', 34:
'Pneumonia', 13: 'Dimorphic hemmorhoids(piles)', 18: 'Heart attack', 39:
'Varicose veins', 26: 'Hypothyroidism', 24: 'Hyperthyroidism', 25:
'Hypoglycemia', 31: 'Osteoarthristis', 5: 'Arthritis', 0: '(vertigo)
Paroymsal  Positional Vertigo', 2: 'Acne', 38: 'Urinary tract infection',
35: 'Psoriasis', 27: 'Impetigo'}

# Model Prediction function
def get_predicted_value(patient_symptoms):
    input_vector = np.zeros(len(symptoms_dict))
    for item in patient_symptoms:
        input_vector[symptoms_dict[item]] = 1
    return diseases_list[svc.predict([input_vector])[0]]


#creating routes
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST','GET'])
def predict():
    if request.method == 'POST':
        symptoms = request.form.get('symptoms')
        user_symptoms = [s.strip() for s in symptoms.split(',')]
        # Remove any extra characters, if any
        user_symptoms = [symptom.strip("[]' ") for symptom in
user_symptoms]
        predicted_disease = get_predicted_value(user_symptoms)

        desc, pre, med, die, wrkout = helper(predicted_disease)

        return
render_template('index.html',predicted_disease=predicted_disease,dis_des=
desc,dis_pre=pre,dis_med=med, dis_die=die,dis_wrkout=wrkout)
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/blog')
def blog():
    return render_template('blog.html')

@app.route('/contact')
def contact():
    return render_template('contact.html')

@app.route('/developer')
```

```
def developer():
    return render_template('developer.html')

#python main
if __name__=="__main__":
    app.run(debug=True)
```

B. SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

• Operating System	:	Window 10 or above, Linux
• Technologies	:	Python, HTML, Bootstrap JS
• IDE	:	PyCharm
• Python Version	:	Python 3.9 or above

HARDWARE REQUIREMENTS

• CPU	:	Intel i7 or Ryzen 5 or higher
• GPU	:	GTX 1650 or higher
• Speed	:	2.6 GHz
• RAM	:	8 GB

C.TECHNOLOGIES

PYTHON

Python is a high-level, interpreted programming language known for its simplicity and readability. Created by Guido van Rossum in the late 1980s, Python has gained immense popularity and has become one of the most widely used programming languages in the world. It has a vast ecosystem of libraries and frameworks that make it suitable for various applications, ranging from web development to data analysis and artificial intelligence.

Python's versatility and cross-platform compatibility are other notable characteristics. It supports multiple operating systems, including Windows, macOS, and Linux, making it accessible to a wide range of users. Furthermore, Python integrates seamlessly with other programming languages, enabling developers to combine Python code with modules written in C, C++, or Java, among others.

The Python Standard Library provides a rich set of modules and functions that cover a wide range of tasks, from file manipulation to network programming and web development. This comprehensive library, combined with the extensive third-party packages available through the Python Package Index (PyPI), makes Python a powerful language for rapid application development.

Python's popularity in the field of data science and machine learning is particularly noteworthy. Libraries such as NumPy, Pandas, and Matplotlib provide efficient and convenient tools for data manipulation, analysis, and visualization. Additionally, frameworks like TensorFlow and PyTorch offer powerful capabilities for building and training machine learning models.

Web development is another domain where Python shines. Frameworks like Django and Flask provide developers with the necessary tools to build robust and scalable web applications. These frameworks offer features such as URL routing, template rendering, database integration, and user authentication, simplifying the development process and boosting productivity.

Python's community and ecosystem play a crucial role in its success. The Python community is known for its inclusivity and helpfulness, providing support through online forums, mailing lists, and conferences. The open-source nature of Python encourages collaboration and the sharing of code, resulting in a vast collection of libraries and resources that benefit developers worldwide.

Python's popularity and demand in the job market have been steadily increasing over the years. Many companies and organizations, including Google, Facebook, and NASA, rely on Python for their critical projects. Its ease of use, combined with its ability to handle complex tasks, makes it an excellent choice for both beginners and experienced programmers.

PANDAS

Pandas is a powerful open-source data manipulation and analysis library for Python. It provides easy-to-use data structures and data analysis tools that make working with structured data, such as tabular or time series data, efficient and convenient. Developed by Wes McKinney in 2008, Pandas has become a go-to library for data scientists and analysts due to its versatility and extensive functionality.

At the heart of Pandas are two primary data structures: Series and DataFrame. A Series is a one-dimensional labeled array that can hold any data type. It is similar to a column in a spreadsheet or a single column of a SQL table. A DataFrame, on the other hand, is a two-dimensional labeled data structure, resembling a table or a spreadsheet, where each column can contain different data types. DataFrames are especially useful for handling structured data and performing complex data manipulations.

Pandas provides a wide range of functionalities to manipulate and transform data. It allows users to handle missing values, filter data based on conditions, sort and group data, merge and join datasets, and perform arithmetic and statistical operations. These operations can be performed with concise and expressive syntax, making data manipulation tasks more efficient.

One of the key strengths of Pandas is its ability to handle time series data effectively. It offers built-in functionality for time-based indexing, resampling, and time zone handling. This makes it a valuable tool for analyzing and processing data with temporal components, such as stock prices, weather data, or sensor readings.

Pandas integrates seamlessly with other popular Python libraries, such as NumPy, Matplotlib, and SciPy, enabling a comprehensive data analysis workflow. NumPy provides the underlying array processing capabilities, Matplotlib allows for data visualization, and SciPy extends the functionality with advanced statistical operations. Together with Pandas, these libraries form a powerful ecosystem for data analysis and scientific computing.

Pandas also supports reading and writing data from various file formats, including CSV, Excel, SQL databases, and more. This makes it easy to import data into Pandas from different sources, perform analysis, and export the results in a desired format. Additionally, Pandas supports interoperability with other data analysis tools and libraries, allowing for seamless integration into existing workflows.

The Pandas community is vibrant and actively contributes to the library's development and improvement. The community provides extensive documentation, tutorials, and resources, making it easier for users to learn and utilize Pandas effectively. This collaborative environment ensures that the library remains up-to-date, reliable, and continually evolves to meet the needs of its users.

Pandas' versatility and usability have made it a popular choice not only in data science and analytics but also in a wide range of industries. It is widely used in finance, marketing, social sciences, and many other domains that require data manipulation and analysis. Its ability to handle large datasets efficiently has also made it a preferred tool for big data processing.

NUMPY

NumPy (Numerical Python) is a powerful open-source library for numerical computing in Python. It provides high-performance multidimensional array objects, along with a wide range of mathematical functions and tools for efficient array manipulation. NumPy serves as a fundamental building block for scientific computing and data analysis in Python.

At the core of NumPy is the ndarray (n-dimensional array) object, which is a homogeneous container for storing and manipulating large datasets in a memory-efficient manner. The ndarray enables efficient storage and vectorized operations on arrays, allowing for fast numerical computations. Arrays can have any number of dimensions, from one-dimensional vectors to multi-dimensional matrices.

NumPy provides a comprehensive set of functions and methods for creating, manipulating, and reshaping arrays. It supports a variety of data types, including integers, floats, and complex numbers. NumPy arrays can be easily reshaped, sliced, and indexed, allowing for efficient data extraction and manipulation.

One of the key advantages of NumPy is its ability to perform vectorized operations on arrays. Vectorization eliminates the need for explicit loops, resulting in faster and more concise code. NumPy provides a wide range of mathematical functions that can be applied element- wise to arrays, such as arithmetic operations, trigonometric functions, exponential and logarithmic functions, and more. These vectorized operations significantly speed up numerical computations.

NumPy also offers powerful tools for array broadcasting, which allows for efficient operations on arrays with different shapes and dimensions. Broadcasting automatically aligns the dimensions of arrays, eliminating the need for explicit array expansion or replication. This feature simplifies complex array operations and makes code more readable and efficient.

In addition to array manipulation, NumPy provides linear algebra routines, Fourier transform capabilities, random number generation, and tools for working with polynomials and integration. These functionalities make NumPy a versatile library for various scientific computing tasks, including signal processing, image analysis, optimization, and simulation.

NumPy seamlessly integrates with other scientific computing libraries in Python, such as SciPy, pandas, and matplotlib. SciPy builds on top of NumPy and provides additional functionality for scientific computing, including numerical integration, optimization, interpolation, and signal processing. pandas, a popular data manipulation library, relies on NumPy arrays as the underlying data structure for efficient data processing. matplotlib, a powerful visualization library, accepts NumPy arrays for generating high-quality plots and charts.

The NumPy community is highly active and constantly contributes to the development and improvement of the library. The community provides extensive documentation, tutorials, and examples, making it easy for users to learn and utilize NumPy effectively. NumPy also benefits from the wide adoption in the scientific community, ensuring ongoing support and updates.

Jupyter

Jupyter, an umbrella term, encompasses a suite of open-source software tools designed to facilitate interactive computing across diverse programming languages. It empowers users to seamlessly blend code execution, rich text formatting, visualizations, and multimedia elements within a single document format known as a Jupyter Notebook.

Key Components of the Jupyter Ecosystem:

- **Jupyter Notebook:** The cornerstone of the Jupyter experience, Jupyter Notebook is a web-based application that functions as a computational notebook. It presents a structured interface where users can interweave code cells with textual explanations, equations, and graphics, all within the same document. This interactive environment fosters iterative exploration, experimentation, and effective communication of data science and scientific computing workflows.
- **Jupyter Kernels:** The backbone of Jupyter's language-agnostic execution capabilities, kernels act as bridges between the Jupyter Notebook and various programming languages. When a code cell is executed, the corresponding kernel interprets and runs the code, returning the output back to the notebook. Jupyter supports a vast array of kernels, encompassing popular languages like Python (the most commonly used), R, Julia, Scala, and many more.
- **Project Jupyter:** The driving force behind the development and evolution of Jupyter tools, Project Jupyter is a non-profit organization that fosters a thriving community of developers, educators, and data scientists. Through open collaboration, Project Jupyter ensures the continuous advancement of Jupyter technologies and their widespread adoption.

Benefits of Utilizing Jupyter:

- **Interactive and Exploratory Environment:** Jupyter excels at enabling rapid prototyping, testing ideas, and iteratively refining code. The interactive nature allows for immediate feedback on code execution, streamlining the development process.
- **Language Agnosticism:** Project Jupyter's commitment to open standards and cross-language support empowers users to leverage their preferred programming languages within Jupyter Notebooks. This flexibility caters to diverse skillsets and project requirements.
- **Reproducibility and Collaboration:** The self-contained nature of Jupyter Notebooks, with code, explanations, and results housed within the same document, fosters reproducibility, making it easier to share and replicate analyses. Moreover, Jupyter Notebooks can be effectively utilized for collaborative work.
- **Rich Media Integration:** Jupyter Notebooks go beyond code execution. They permit the incorporation of text formatting, equations, images, charts, and other multimedia content, making notebooks highly informative and visually appealing. This enriches data storytelling and knowledge sharing.

PyCharm

PyCharm is an integrated development environment (IDE) specifically designed for Python programming. It's developed by JetBrains, a well-regarded company known for creating programmer-friendly tools.

Key Features of PyCharm:

- **Code Analysis:** PyCharm offers real-time code analysis, highlighting syntax errors, potential issues, and suggesting improvements as you type.
- **Graphical Debugger:** It includes a graphical debugger to step through your code line by line, inspect variables, and identify bugs effectively.
- **Integrated Unit Tester:** PyCharm allows you to write and run unit tests directly within the IDE, ensuring the functionality of your code.
- **Version Control Integration:** It integrates seamlessly with popular version control systems like Git, Subversion, and Mercurial, enabling version tracking and collaboration.
- **Web Development Support:** PyCharm provides built-in support for popular web frameworks like Django and Flask, streamlining web development workflows.
- **Scientific Tools:** The Professional edition offers features geared towards scientific computing, including integration with Jupyter notebooks and scientific libraries like Pandas and NumPy.

Benefits of Using PyCharm:

- **Increased Productivity:** PyCharm's features like code completion, refactoring, and navigation tools significantly speed up development.
- **Improved Code Quality:** The code analysis and debugging capabilities help you write cleaner, more maintainable code.
- **Streamlined Web Development:** Built-in web development features make creating and managing web applications in Python more efficient.
- **Enhanced Scientific Programming Experience:** (Professional Edition) Scientific tools streamline data science workflows and analysis.

Licensing:

- **PyCharm Community Edition:** Free and open-source, ideal for basic Python development and learning.
- **PyCharm Professional Edition:** Paid license with extended features for web, scientific, and enterprise development. However, there are free options available for open-source projects and educational use.

Getting Started with PyCharm:

1. Download and install PyCharm from the JetBrains website https://www.jetbrains.com/pycharm/download/.
2. Create a new project and configure your Python interpreter.
3. Start writing Python code and leverage PyCharm's features to enhance your development experience.

Scikit

scikit-learn (often abbreviated as sklearn) is a free and open-source software library that simplifies machine learning tasks in Python. It offers an extensive collection of efficient algorithms for various machine learning problems, including:

- **Classification:** Classifying data points into predefined categories. Examples include spam detection, image recognition, customer segmentation, and credit risk assessment.
- **Regression:** Predicting continuous-valued attributes based on existing data. Applications encompass stock price forecasting, weather prediction, real estate valuation, and sales forecasting.
- **Clustering:** Grouping data points into sets (clusters) based on their similarities. This is useful for exploratory data analysis, market segmentation, and anomaly detection.
- **Dimensionality Reduction:** Transforming data into a lower-dimensional space while preserving essential information. It can improve model performance, reduce computational complexity, and visualize high-dimensional data.
- **Model Selection and Evaluation:** Choosing the best model for a specific task and evaluating its performance through metrics like accuracy, precision, recall, and F1-score.

Key Features of scikit-learn:

- **Ease of Use:** scikit-learn is renowned for its user-friendly API, making it accessible to beginners and experienced developers alike.
- **High-Quality Implementations:** The library provides well-tested and optimized algorithms, ensuring reliable results.
- **Scalability:** scikit-learn can handle datasets of varying sizes, efficiently supporting smaller-scale projects and larger-scale data analysis.
- **Interoperability:** It integrates seamlessly with other popular scientific Python libraries like NumPy, SciPy, and Matplotlib, facilitating a cohesive data science workflow.
- **Extensibility:** scikit-learn allows for customization and advanced use cases by providing mechanisms to create custom estimators, transformers, and pipelines.

Benefits of Using scikit-learn:

- **Rapid Prototyping:** The streamlined API enables quick machine learning model development and experimentation.
- **Efficient Model Building:** The library offers a wide range of algorithms, reducing the need for extensive implementation from scratch.
- **Robust Performance Evaluation:** Built-in metrics and functionalities assist in rigorously assessing model efficacy.
- **Active Community:** scikit-learn boasts a large and active community that provides extensive documentation, tutorials, and support.

HTML

HTML, or Hyper Text Markup Language, is the standard markup language used to create web pages. It's a combination of Hypertext, which defines the link between web pages, and Markup language, which is used to define the text document within tags to structure web pages. This language is used to annotate text so that machines can understand and manipulate it accordingly. HTML is human-readable and uses tags to define what manipulation has to be done on the text.

HTML Elements and Tags

HTML uses predefined tags and elements that instruct the browser on how to display the content. HTML elements include an opening tag, some content, and a closing tag. It's important to remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML Page Structure

The basic structure of an HTML page is shown below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

<!DOCTYPE html> – This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

<html> – This is called the HTML root element. All other elements are contained within it.

<head> – The head tag contains the "behind the scenes" elements for a webpage. Elements within the head aren't visible on the front end of a webpage. HTML elements used inside the <head> element include:

<style> – This HTML tag allows us to insert styling into our web pages and make them appealing to look at with the help of CSS.

<title> – The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.

<base> – It specifies the base URL for all relative URL's in a document.

<noscript> – Defines a section of HTML that is inserted when the scripting has been turned off in the user's browser.

<script> – This tag is used to add functionality to the website with the help of JavaScript.

<link> – The 'link' tag is used to tie together HTML, CSS, and JavaScript. It is self-closing.

<body> – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.

CSS

CSS (Cascading Style Sheets) is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to HTML documents. It describes how a webpage should look. It prescribes colors, fonts, spacing, etc. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

HTML uses tags and CSS uses rulesets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

CSS modules

As there are so many things that you could style using CSS, the language is broken down into modules. You'll see reference to these modules as you explore MDN. Many of the documentation pages are organized around a particular module. For example, you could take a look at the MDN reference to the Backgrounds and Borders module to find out what its purpose is and the properties and features it contains. In that module, you will also find a link to Specifications that defines the technology (also see the section below).

At this stage, you don't need to worry too much about how CSS is structured; however, it can make it easier to find information if, for example, you are aware that a certain property is likely to be found among other similar things, and is therefore, probably in the same specification.

CSS specifications

All web standards technologies (HTML, CSS, JavaScript, etc.) are defined in giant documents called specifications (or "specs"), which are published by standards organizations (such as the W3C, WHATWG, ECMA, or Khronos) and define precisely how those technologies are supposed to behave.

CSS is no different — it is developed by a group within the W3C called the CSS Working Group. This group is made of representatives of browser vendors and other companies who have an interest in CSS. There are also other people, known as invited experts, who act as independent voices; they are not linked to a member organization.

New CSS features are developed or specified by the CSS Working Group — sometimes because a particular browser is interested in having some capability, other times because web designers and developers are asking for a feature, and sometimes because the Working Group itself has identified a requirement. CSS is constantly developing, with new features becoming available. However, a key thing about CSS is that everyone works very hard to never change things in a way that would break old websites. A website built in 2000, using the limited CSS available then, should still be usable in a browser today!

As a newcomer to CSS, it is likely that you will find the CSS specs overwhelming — they are intended for engineers to use to implement support for the features in user agents, not for web developers to read to understand CSS. Many experienced developers would much rather refer to MDN documentation or other tutorials. Nevertheless, it is worth knowing that these specs exist and understanding the relationship between the CSS you are using, the browser support (see below), and the specs.

BOOTSTRAP

Bootstrap is a widely-used open-source front-end framework for web development, providing a collection of HTML, CSS, and JavaScript components and tools that enable developers to build responsive, mobile-first websites with ease.

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers – Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

By using this framework, we can easily manipulate the styling of any web page, like font style, text color, background color, flex, grid system, etc. Bootstrap Vesrion 4 & Vesrion 5 are the most popular versions. There are lots of other CSS frameworks like Tailwind CSS, Bulma, and Foundation but among them, this framework is the most popular because of below mentioned features:

1. It is Faster and Easier way for Web-Development.
2. It creates Platform-independent web-pages.
3. It creates Responsive Web-pages.
4. It designs responsive web pages for mobile devices too.
5. It is a free and open-source framework available on www.getbootstrap.com

Applications of Bootstrap

- Responsive Web Design: Bootstrap empowers developers to create websites that seamlessly adapt to different screen sizes and devices, providing a consistent and optimal user experience.

- Mobile-First Development: Bootstrap’s mobile-first approach ensures that websites are designed and optimized for mobile devices, catering to the increasing usage of smartphones and tablets.

- Efficient Prototyping: With its extensive collection of pre-designed components and templates, Bootstrap facilitates rapid prototyping, enabling developers to quickly create functional website layouts and UIs.

- Consistent Cross-Browser Compatibility: Bootstrap’s standardized CSS and JavaScript codebase ensures consistent rendering and functionality across various web browsers, saving developers time and effort in browser-specific troubleshooting.

- Customizable Themes and Styling: Bootstrap offers a wide range of customizable themes and styles, allowing developers to create visually appealing and unique designs that align with their brand or project requirements.

- Time and Cost Efficiency: By leveraging the power of Bootstrap, developers can save significant time and effort in front-end development, resulting in faster project delivery and cost savings.

RESOURCES