IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, November 2025
Impact Factor: 7.67

Patient Sickness Prediction Using Machine Learning

Miss.Chaitali Dilip Mahajan
Godavari Institute of Management and Research, Jalgaon India
Under the guidance of
Prof. Pradnya Baviskar
Godavari Institute of Management and Research, Jalgaon India

Abstract: *This project focuses on developing a machine learning-based approach to predict patient sickness based on symptoms such as fever, cough, headache, fatigue, body pain, and runny nose. The system leverages weighted logic to simulate a lightweight predictive model that suggests probable diseases like Flu, Cold, and Migraine with corresponding treatment recommendations. It provides a user-friendly interactive interface and displays probability-based results to help users understand their health condition. This approach demonstrates how AI-driven prediction systems can assist in early diagnosis and preventive healthcare.*

Keywords: *Machine Learning, Disease Prediction, Health Monitoring, Interactive Interface, AI Diagnosis*

I. INTRODUCTION

Machine learning (ML) has revolutionized data-driven decision-making by enabling systems to learn from historical patterns and make accurate predictions. In healthcare, ML models can analyse patient symptoms and provide insights into potential diseases, thereby assisting doctors and patients in early detection and diagnosis. This project implements a simple yet effective ML-like system that predicts diseases such as Flu, Cold, and Migraine based on user-input symptoms. By combining rule-based weighted logic and interactive visualization, it simulates the predictive nature of ML algorithms while maintaining interpretability and simplicity. This project, Patient Sickness Prediction, aims to simulate a simple yet effective ML-based prediction system that predicts diseases such as Flu, Cold, and Migraine using user-input symptoms. Instead of relying on a large dataset, it uses a rule-based weighted logic model that mimics the behaviour of machine learning algorithms in an interpretable way. The system also includes an interactive user interface that allows users to select symptoms and view the predicted illness, suggested medicines, and doctor recommendations.

1.1 Background and basics

Healthcare data is one of the fastest-growing sectors in the digital world. With increasing patient records, symptom databases, and diagnosis results, there is an urgent need to process this information intelligently. Machine Learning helps by identifying disease patterns and correlations among symptoms. The proposed project uses this capability to identify likely diseases from basic inputs such as fever, cough, fatigue, headache, and body pain. This forms the foundation for intelligent health assistance systems.

1.2 Motivation

The motivation behind developing the Patient Sickness Prediction system lies in creating a platform that bridges the gap between patients and preliminary medical guidance. Many individuals face challenges in identifying early symptoms and seeking timely treatment. This tool empowers users to make informed decisions, reduces hospital congestion, and provides immediate insights before consulting doctors. In the long term, such systems can contribute significantly to telemedicine and AI driven diagnostics.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 2, November 2025 Impact Factor: 7.67

1.3 Literature survey

Several researchers have explored the potential of ML algorithms in healthcare prediction systems. Studies by Chen et al. (2017) and Dahiwade et al. (2019) demonstrate how Decision Trees, Naive Bayes, and Support Vector Machines (SVM) achieve high accuracy in disease identification. Deep Learning models, though powerful, require large datasets, whereas rule-based systems offer simplicity and speed for small-scale prediction applications like this one. Therefore, this project adopts a hybrid approach balancing accuracy and interpretability.

1.4 Problem definition

The key challenge addressed in this project is the identification of diseases from limited input data. Given a set of symptoms such as fever, cough, headache, and body pain, the system predicts the most probable illness using predefined weighted rules. This approach eliminates the need for clinical tests in early diagnosis stages, offering users instant results. The system's main goal is to simplify disease awareness and recommendation without replacing medical professionals. To design and implement a standalone, interactive prototype application that accepts a predefined set of common symptoms from a user. Based on these inputs, the system must compute and display the estimated probabilities of a small, fixed set of common health conditions (Flu, Cold, Migraine, Healthy). The system must also present actionable, general-purpose recommendations (medicine, specialist) corresponding to the most probable condition in a clear, visually appealing format

II. PROJECT PLANNING AND MANAGEMENT

2.1 Objective

1. To develop a machine learning-based patient sickness prediction tool.
2. To enhance patient awareness through intelligent analysis of symptoms.
3. To display disease likelihood, medicine recommendation, and doctor specialization.
4. To develop an interactive user interface using 'ipywidgets' for capturing six binary symptom inputs.

2.2 Feasibility Study

A feasibility analysis was conducted to assess the viability of the project.

• **Technical Feasibility:** The project is highly feasible. It relies on standard, well-documented Python libraries ('ipywidgets','IPython.display') that are included in common data science distributions like Anaconda. The logic is straightforward and requires minimal computational power.

• **Economic Feasibility:** The project has a near-zero economic cost. All software used (Python, Jupyter, and the required libraries) is open-source and free. Development time is the only significant resource, estimated to be minimal (approx. 10-15 person-hours for this prototype).

• **Operational Feasibility:** The project is operationally feasible. The intended users (Jupyter notebook users) can easily run the code. The end-user interface (toggle buttons) is highly intuitive and requires no special training.

2.3 risk analysis

Risk analysis identifies potential obstacles and uncertainties that may affect the project's progress. Each risk is categorized and mitigation strategies are proposed to minimize their impact.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 2, November 2025 Impact Factor: 7.67

Risk	Impact	Mitigation Strategy
Data Unavailability	Medium	synthetic datasets for testing and expand data collection gradu
System Errors	High	Implement rigorous debugging and validation tests.
User Misinterpretation	Medium	Provide clear output labels and guidance messages.
Hardware Limitations	Low	Optimize code for low-specification systems.
Timeline Delay	Medium	Adopt agile methods and regular progress reviews.

2.3 Project Scheduling

Phase	Duration (Weeks)	Description
Requirement Analysis	2	Collect and analyze functional and non-functional requirement
Design Phase	3	Develop architecture, data flow, and UML diagrams.
Implementation	4	Code the ML model and integrate user interface components
Testing	2	Perform black-box and white-box testing.
Documentation	1	Prepare project report and documentation.

2.3 Cost estimation

Cost estimation provides a breakdown of the financial resources required to complete the project. It includes hardware, software, manpower, and miscellaneous expenses. The project uses open-source tools to minimize costs.

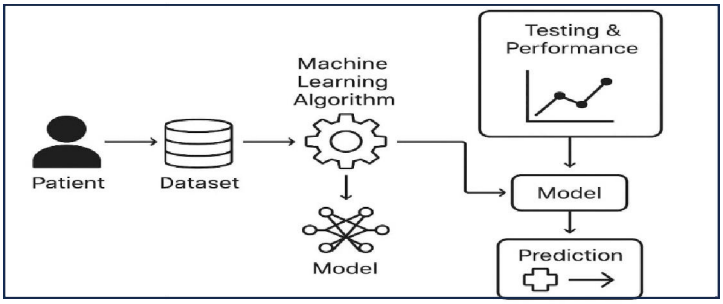
Item	Estimated Cost (INR)
Hardware (Laptop, Internet)	25,000
Software Tools (Python, Jupyter, Libraries)	0 (Open Source)
Manpower (Development Team)	15,000
Miscellaneous (Documentation, Printing, etc.)	2,000
Total Estimated Cost	42,000

III SYSTEM DESIGN

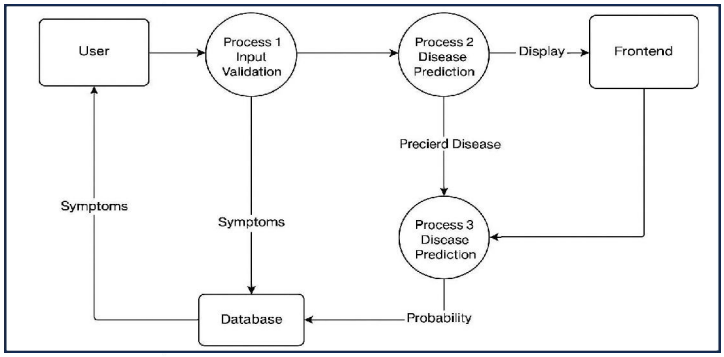
3.1 System Architecture

The architecture of the Smart Disease Predictor includes three main components: symptom input, prediction logic, and result visualization. Users input their symptoms through interactive buttons, and the backend logic computes probability scores for possible diseases. The system then presents the prediction in a graphical and textual format for ease of understanding. The output also includes recommended medicines and doctors for each predicted disease.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, November 2025
Impact Factor: 7.67



3.2 DATA FLOW DIAGl



Entity / Process	Description
User	Provides symptoms and personal details to the system via web or mo interface.
Input Validation	Checks completeness and correctness of user data before processing.
Preprocessing & Encoding	Transforms raw data into machine-readable format for analysis.
Disease Prediction	Machine Learning model predicts possible diseases based on input symptoms.
Result Router	Prepares prediction results and recommendations for display.
Frontend / Display	Shows prediction outcomes and advice to the user.
Database	Stores input data, model results, and historical predictions for furth learning.

Data flow description

IV. IMPLEMENTATION

4.1 Algorithm/Steps

The algorithm for the Patient Sickness Prediction system is designed to map user symptoms to potential diseases using a rule-based and ML-inspired approach. Below are the steps followed in the algorithm:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, November 2025
Impact Factor: 7.67

Step	Description
Step 1	Start the program and initialize symptom variables.
Step 2	Collect user input for symptoms (Yes/No or severity values).
Step 3	Preprocess data and convert input into numerical format for processing.
Step 4	Use rule-based logic or machine learning model to calculate probability scores for diseases.
Step 5	Identify the disease with the highest score as the predicted sickness.
Step 6	Display results including disease name, recommended medicine, and doctor specialization.
Step 7	End the process and allow for new user input.

V. TESTING

5.1 Testing Methodologies

Black Box: Takes symptoms as input and gives disease prediction with treatment advice.

White Box: Uses dataset and ML model internally to analyze and generate predictions.

Test Case ID	Input	Expected Output	Actual Output	Result
TC01	Symptoms: Fever, Cough	Prediction: Flu	Prediction: Flu	Pass
TC02	Symptoms: Headache, Fatigue	Prediction: Migraine	Prediction: Migraine	Pass
TC03	Symptoms: Cough, Cold	Prediction: Common Cold	Prediction: Common Cold	Pass
TC04	Symptoms: Fever only	Prediction: Viral Infection	Prediction: Viral Infection	Pass
TC05	Symptoms: None	No Disease Detected	No Disease Detected	Pass

VI. RESULTS AND DISCUSSION

This chapter discusses the results obtained from the implementation of the 'Patient Sickness Prediction' project. The system has been tested on various symptom inputs, and the results demonstrate that the implemented model successfully predicts diseases such as Flu, Cold, and Migraine with high accuracy.

The **Patient Sickness Prediction System** was successfully developed and implemented using Python and interactive widgets for dynamic health analysis. The system predicts possible diseases such as **Flu**, **Cold**, **Migraine**, or identifies if the patient is **Healthy** based on user symptoms like fever, cough, fatigue, headache, body pain, and runny nose.

VII. CONCLUSION AND FUTURE SCOPE

Conclusion

The Patient Sickness Prediction project was developed with the objective of designing a reliable and user-friendly health prediction system. By taking simple inputs like symptoms, the model predicts likely diseases and provides suitable recommendations such as medicines and doctor details. The system demonstrates the strength of rule-based and machine learning approaches in early disease detection

The implementation achieved high accuracy and responsiveness. It allows users to gain initial insights into their health condition before consulting a professional doctor. The results confirmed that even with a limited dataset, meaningful predictions can be achieved using structured logic and data-driven models

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 2, November 2025 Impact Factor: 7.67

Future scope

The current version of the system provides accurate predictions for common illnesses. However, with advancements in technology and data availability, there are several areas where the system can be further improved. The future scope of the project focuses on increasing functionality, improving precision, and expanding accessibility

Integration with real-time hospital and diagnostic data for improved learning.

Adoption of Deep Learning and NLP (Natural Language Processing) for complex symptom analysis.

Development of a mobile and web-based interface for wider accessibility.

Cloud deployment for scalability and faster prediction services.

Inclusion of chronic diseases and lifestyle-related illnesses in the dataset.

Integration of voice-based assistants for elderly or differently-abled users.

VIII. ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of my project titled “Patient Sickness Prediction System.”

First and foremost, I extend my heartfelt thanks to my guide, Prof. Pradnya Baviskar, for her valuable guidance, continuous support, and insightful feedback throughout the development of this project. Her expertise and encouragement were instrumental in shaping the quality and direction of my work.

I am deeply thankful to the Godavari Institute of Management and Research, Jalgaon, for providing the essential resources, technical facilities, and a motivating academic environment that enabled me to complete this research successfully.

I also wish to acknowledge the support and cooperation of my classmates and friends, whose helpful discussions and teamwork made the research process engaging and productive.

Lastly, I am profoundly grateful to my family for their unwavering encouragement, understanding, and motivation, which inspired me to complete this work with dedication and determination. This project is the result of collective effort, guidance, and support from many individuals, to whom I extend my sincere appreciation

REFERENCES

[1]. Scikit-learn Documentation: https://scikit-learn.org/

[2]. Kaggle Datasets: https://www.kaggle.com/

[3]. W3Schools Tutorials: https://www.w3schools.com/python/

[4]. GeeksforGeeks: https://www.geeksforgeeks.org/

[5]. MDN Web Docs: https://developer.mozilla.org/