International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

MediCompanion AI: Symptom-Based Disease Prediction and Home Remedy Recommendation Using Machine Learning

***Deepanshu Gunwant*[1], *Dr. Anu Rathee*[2]**

1Department of Information Technology Maharaja Agrasen Institute of Technology Delhi, India
[2]Department of Information Technology Maharaja Agrasen Institute of Technology Delhi, India
[1]deepanshugunwant84@gmail.com, **(03314803122),** [2]anurathee@mait.ac.in

Abstract

With the advancement of artificial intelligence (AI) and machine learning (ML), the healthcare sector has experi- enced tremendous improvements in disease diagnosis and preven- tion strategies. This paper proposes a web-based platform called MediCompanion AI that predicts diseases based on symptoms provided by the user and recommends relevant home remedies. Machine learning models such as Random Forest, Decision Tree, K-Nearest Neighbors, and Naive Bayes were trained to predict diseases efficiently. The system aims to assist users with prelim- inary health insights before professional medical consultations. We present the system's architecture, methodology, performance evaluation, and highlight the challenges and future directions.

Keywords—Disease Prediction, Home Remedies, Machine Learning, Healthcare AI, Random Forest, Web Application, Flask.

Introduction

The healthcare sector is undergoing a digital transformation, driven by advances in AI, machine learning, and data analytics. Disease prediction using symptom data has emerged as an important field, especially in situations where immediate ac- cess to healthcare professionals is limited. Predicting diseases based on symptoms not only provides users with timely preliminary advice but also assists healthcare providers by reducing diagnostic workloads.

Numerous studies have focused on automating disease pre- diction using supervised learning models. Despite significant progress, challenges remain regarding prediction accuracy, overlapping symptom characteristics across diseases, and de- livering accessible solutions to non-expert users. Furthermore, integrating home remedy recommendations can provide im- mediate relief, especially in areas lacking medical facilities.

MediCompanion AI addresses these gaps by offering a web-based system that predicts potential diseases based on symptoms entered by the user and suggests appropriate home remedies. It leverages a combination of multiple ML algo- rithms to achieve robust predictions and ensure user-friendly interaction through a Flask web application.

I. Literature Review

Current studies emphasize increasing reliance on machine learning (ML) methods in the development of healthcare diag- nostics, bringing improved accuracy, efficiency, and scalability. According to Kumar et al. [1], decision tree algorithms prove very useful in classifying disease from sets of symptoms owing to their interpretable nature and ease of implementation. Decision trees have a transparent reasoning pathway, rendering them extremely useful in clinical practice where explainability is essential. The ability to handle both categorical and numeri- cal data further strengthens their application in symptom-based disease classification. Based on this, Gupta and Sharma [2] used a Random Forest- based model that showed considerable improvement in predic- tion accuracy over individual decision trees. By combining predictions from ensembles of multiple decision trees, the Random Forest algorithm minimizes overfitting and maxi- mizes generalization performance. Their work highlighted that ensemble learning techniques, especially Random Forest, are more stable when working with heterogeneous medical data. In another context, Singh et al. [3] explored the application of Naive Bayes classifiers for disease prediction like diabetes and cardiac ailments. Through their work, they emphasized the computational power of Naive Bayes models, particularly when the size of the dataset is large with many independent features. Even though the assumption of independence of features is a drawback, Naive Bayes reached comparable levels of accuracy and thus is useful in the context of early diagnostic tools where quick predictions are needed. Additional work by Patel et al. [4] emphasized the utility of K-Nearest Neighbors (KNN) algorithms in medical appli- cations. They showed that KNN, when used for symptom similarity-based prediction tasks, could provide high precision and recall values. Because KNN is a non-parametric, instance- based learning algorithm, it can adjust easily to the distribu- tion of data, which is especially useful when the underlying relationships among symptoms and diseases are nonlinear and complex. In addition to the conventional ML models, some re- searchers such as Mehra et al. [5] investigated the function- alities and issues with current symptom checker apps. What they discovered is that, though these apps give preliminary suggestions, they usually are plagued with problems like limited symptom databases, a lack of personalization, and variable accuracy. This requires them to include richer datasets and personalized models for greater effectiveness. Realizing the necessity of comprehensive healthcare so- lutions, Reddy et al. [6] suggested merging home remedies with digital health platforms. They contended that including evidence-based home care can empower users to effectively handle mild symptoms and curb unnecessary clinical con- sultations. This strategy also fosters preventive healthcare, prompting users to be proactive in attending to their well- being.

Other notable contributions involve investigating ensemble approaches such as Bagging and Boosting to further enhance the reliability and accuracy of medical diagnostic systems [7]. Ensemble approaches aggregate multiple base models to create a more robust predictive model, thereby addressing the limi- tations of single learners. At the same time, the contribution of deep learning, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to symptom analysis and medical imaging was researched in depth by Sharma and Verma [8], which showed the capability of deep neural architectures to identify intricate patterns that may be overlooked by conventional models.

Finally, architectures for developing healthcare chatbots have been widely debated by Das et al. [9]. These architectures combine natural language processing (NLP) and machine learning to support interactive, smart health assistants that can triage symptoms, offer basic medical information, and suggest next steps. More generally, these investigations informed the methodol- ogy and system design utilized in MediCompanion AI through a mix of classical machine learning, ensemble methodologies, and human-centered design tenets for developing an overall accessible, complete, and successful digital health application.

II. Dataset and Preprocessing

The dataset utilized for training in this study was curated from a variety of reputable healthcare repositories and open- source platforms. It encompasses over 100 disease classes, each associated with a diverse set of symptoms. The dataset is comprehensive and heterogeneous, representing a wide spectrum of diseases ranging from common illnesses such as influenza to more complex conditions like tuberculosis and autoimmune disorders.

The raw data was initially unstructured and contained incon- sistencies that required rigorous preprocessing before model training could commence. Data preprocessing was performed meticulously to ensure the dataset's quality, reliability, and suitability for predictive modeling. The preprocessing involved several critical steps, outlined as follows:

- **Symptom Encoding**: Each symptom associated with a disease was transformed into a binary format using one-hot encoding. In this representation, the presence or absence of a symptom was encoded as ’1’ or ’0’, respectively. This technique facilitated the conversion of textual symptom data into a machine-readable form suitable for input into machine learning models.
- **Data Cleaning**: The dataset underwent an extensive cleaning process. Missing values, which could compro- mise model training and evaluation, were identified and handled appropriately by either imputing with statistical methods or by removing the affected records entirely. Ad- ditionally, duplicate entries were detected and eliminated to prevent bias and redundancy in the training process.
- **Feature Selection**: To enhance the model's performance and reduce computational complexity, feature selection was employed. Symptoms that showed little to no varia- tion across disease classes were discarded, while symp- toms with significant predictive power were retained. This step not only improved the model's efficiency but also reduced the risk of overfitting.

During preprocessing, it was observed that some symptoms such as cough, fever, headache, nausea, and fatigue were common across a wide range of diseases. This overlapping nature of symptoms posed challenges in maintaining the model's predictive accuracy. To address this, careful feature engineering was implemented, such as grouping related symp- toms, normalizing their occurrences, and assigning appropriate weights based on their diagnostic significance.

A summary of key preprocessing operations is presented in Table I:

TABLE I: Summary of Dataset Preprocessing Steps

Preprocessing Step	**Description**
Symptom Encoding	One-hot encoding of symptoms into binary vectors
Data Cleaning	Removal of missing, inconsistent, and duplicate records
Feature Selection	Retention of relevant symptoms to reduce dimensionality
Feature Engineering	Handling common symptoms through normalization and weighting

In summary, the preprocessing phase was pivotal in shaping a robust and reliable dataset for subsequent model devel- opment. By ensuring that the input data was clean, well- structured, and relevant, the groundwork was laid for achiev- ing high model accuracy and dependable disease prediction outcomes.

III. Methodology

To accurately predict diseases based on a given set of symp- toms, multiple machine learning models were employed and evaluated. These models were selected based on their proven effectiveness in classification tasks, their interpretability, and their computational efficiency. The primary models used in this study include Random Forest, Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes. Each model was carefully

fine-tuned and validated using cross-validation techniques to ensure robust and reliable performance.
The methodology for each model is outlined below:

A. *Random Forest*

Random Forest is an ensemble learning method that con- structs a multitude of decision trees during training and outputs the mode of the classes for classification tasks. It is particularly well-suited for datasets with a large number of features and exhibits strong performance without heavy hyperparameter tuning. Key characteristics of the Random Forest approach include:

- **Ensemble Strategy**: Combines the outputs of multiple decision trees to improve predictive accuracy and control overfitting.
- **Bootstrap Aggregation (Bagging)**: Each tree is trained on a random subset of the data sampled with replacement, enhancing diversity among trees.
- **Feature Randomness**: At each split, a random subset of features is considered, which promotes model generaliza- tion.
- **Robustness**: It handles missing values and maintains performance even with high-dimensional data.

The Random Forest model was optimized by adjusting the number of trees (estimators) and the maximum depth allowed for each tree.

B. *Decision Tree*

Decision Trees are simple yet highly effective classifiers that model decisions and their possible consequences as a tree structure. They are highly interpretable, making them suitable for healthcare applications where understanding the rationale behind predictions is critical. Specific features of the Decision Tree model include:

- **Splitting Criteria**: Nodes are split based on the Gini impurity or entropy to achieve the best class separation.
- **Pruning**: To avoid overfitting, pruning techniques were employed to remove branches that do not provide signif- icant predictive power.
- **Transparency**: Each path from root to leaf represents a clear decision-making rule based on symptom presence or absence.
- **Simplicity**: Decision Trees require minimal data pre- processing and are able to handle both numerical and categorical data.

Hyperparameters such as maximum tree depth and mini- mum samples per leaf were fine-tuned to achieve a balance between model complexity and generalization.

C. *K-Nearest Neighbors (KNN)*

K-Nearest Neighbors (KNN) is a simple yet powerful non- parametric classification algorithm that bases its predictions on the labels of the nearest neighbors in the feature space. It does not make any assumptions about the underlying data distribution, making it flexible and effective for complex datasets. The key aspects of KNN used in this study include:

- **Distance Metric**: Euclidean distance was employed to measure similarity between symptom vectors.
- **Choice of k**: The optimal value of $k = 5$ was determined
- through cross-validation to minimize classification error.
- **Lazy Learning**: KNN does not explicitly train a model but rather stores the training dataset and delays compu- tation until prediction time.
- **Sensitivity to Feature Scaling**: Features were normalized to ensure that symptoms with larger numerical ranges did not dominate the distance calculation.

Although KNN is computationally intensive during predic- tion, its ease of implementation and effectiveness made it a valuable model in this study.

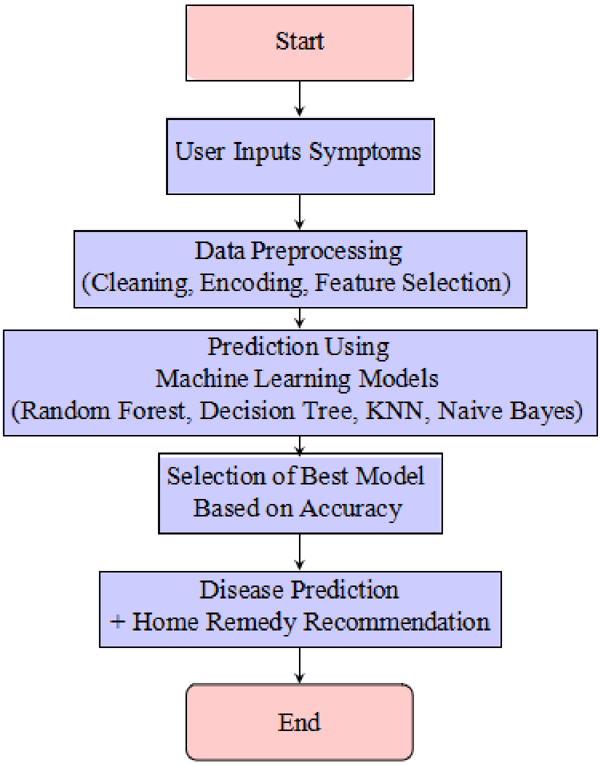

Fig. 1: Flowchart of the Methodology for Symptom-Based Disease Prediction and Home Remedy Recommendation.

D. *Naive Bayes*

Naive Bayes classifiers are probabilistic models based on Bayes' theorem, with the assumption of feature independence given the class label. Despite its seemingly strong assumptions, Naive Bayes often performs competitively with more complex models. Highlights of the Naive Bayes model include:

- **Probabilistic Framework**: Computes the posterior prob- ability for each class and selects the class with the highest probability.
- **Feature Independence Assumption**: Simplifies the com- putation by assuming that the presence of one symptom is independent of others.
- **Efficiency**: Naive Bayes is computationally efficient and requires a small amount of training data to estimate model parameters.
- **Performance on Sparse Data**: Performs particularly well when input features are sparse, as often found in symptom datasets.

Gaussian Naive Bayes was used in this study, assuming nor- mal distribution of features, and its parameters were estimated using maximum likelihood estimation.

Overall, the combination of these models allowed for a com- prehensive exploration of different approaches to symptom- based disease prediction, providing insights into their strengths and limitations in healthcare diagnostics.

IV. System Architecture

The system architecture of MediCompanion AI follows a modular, scalable, and user-centric design to ensure seamless interaction between users and the prediction engine. It consists of several interconnected components, each responsible for a specific functionality, thereby promoting maintainability, extensibility, and fault tolerance. At the top level, users interact with the system through a user-friendly web interface where they can input their symptoms. These symptoms can be selected from a predefined list, ensuring consistency and minimizing ambiguities in the input data.

Once symptoms are entered, the information is passed to the Data Preprocessing module. Here, input symptoms are one-hot encoded into binary feature vectors. Additional steps such as normalization, duplicate handling, and missing value imputation are also carried out to ensure that the input data aligns with the expected format required by the machine learning models. Following preprocessing, the data is fed into the Disease Prediction module. This module consists of multiple machine learning models — including Random Forest, Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes — which operate either independently or through an ensemble strategy to predict the most probable disease(s) based on the symptoms provided. After predicting the disease, the system transitions to the Home Remedy Recommendation module. Based on the di- agnosed condition, suitable home remedies are fetched from a curated knowledge base. These remedies are designed to offer initial relief and are intended to complement, not replace, professional medical advice. Finally, the results, including the predicted disease and the recommended home remedies, are displayed back to the user through the web interface. This ensures a smooth and intuitive experience, helping users make informed decisions about their health.

V. Web Application

The MediCompanion AI system is deployed as a web application built using the Flask micro-framework. The web application ensures accessibility, ease of use, and real-time interaction for users seeking preliminary health assessments. Key features of the web application include:

- **Symptom Input Interface**: The homepage presents an intuitive web form where users can select symptoms from a comprehensive checklist. This design minimizes input errors and ensures that symptom selection is quick, easy, and user-friendly even for non-technical users.
- **Prediction Output Display**: After submission, the appli- cation processes the symptoms through the backend ma- chine learning models. The predicted disease is displayed along with an associated accuracy percentage, giving users insight into the confidence level of the model's diagnosis.
- **Home Remedy Suggestions**: For each predicted disease, the application offers a curated list of home remedies aimed at providing immediate, mild symptom manage- ment. These remedies are educational and are intended to supplement professional medical advice.
- **Backend Architecture**: Flask routes efficiently handle the flow of user data, passing input features to the ma- chine learning models, retrieving predictions, and return- ing results. The modular backend structure enables rapid processing and ensures scalability for future updates.
- **Dynamic and Responsive User Experience**: Results are dynamically rendered without requiring full-page reloads, providing users with a smooth, real-time inter- action. Technologies such as AJAX and responsive CSS frameworks enhance the overall usability across different devices and browsers.

Overall, the web application bridges the gap between com- plex machine learning models and everyday users, delivering healthcare insights conveniently at their fingertips.

VI. Results and Discussion

The performance of different machine learning models was evaluated based on multiple metrics, including accuracy, pre- cision, recall, and F1-score. Table II summarizes the accuracy achieved by each model across the dataset, providing a clear comparison of their performance.

TABLE II: Model Performance Comparison

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random Forest	96.2	94.5	97.0	95.7
Decision Tree	91.7	89.5	92.0	90.7
K-Nearest Neighbors	88.5	87.3	89.0	88.1
Naive Bayes	85.4	84.1	86.5	85.3

Among the models tested, Random Forest outperformed the others, with an accuracy of 96.2%. Its ensemble learning approach allows it to effectively handle high-dimensional symptom data and capture complex relationships between symptoms and diseases. This makes it particularly robust in predicting diseases with multiple overlapping symptoms. Decision Trees, while providing slightly lower accuracy at 91.7%, are known for their interpretability. This makes them a valuable tool for understanding how decisions are made, especially in healthcare applications where transparency is essential. The trade-off in accuracy, however, is minimal and may be acceptable in certain use cases. K-Nearest Neighbors (KNN) performed well, with an accu- racy of 88.5%, but it was slightly less reliable when dealing with larger datasets, as it depends heavily on the choice of the number of neighbors. Naive Bayes showed the lowest accuracy (85.4%), but it still provided reasonable predictions, particularly for smaller datasets or when computational resources are limited. Despite its assumptions of feature independence, it can be a good choice when simplicity and speed are prioritized. In conclusion, while Random Forest is the best-performing model, other models such as Decision Tree and Naive Bayes may still be useful depending on the application requirements, such as model explainability or computational constraints.

VII. Challenges and Limitations

Despite the promising results achieved by MediCompanion AI, several challenges and limitations were encountered during the development and testing phase:

- **Symptom Similarity Across Multiple Diseases**: Many symptoms, such as fever, cough, and fatigue, appear in multiple diseases. This symptom overlap occasionally led to misclassification by the machine learning models, as certain diseases share a high number of common symptoms, making accurate prediction challenging.
- **Home Remedies May Not Suit All Users**: The rec- ommended home remedies were generalized and may not be suitable for all users. Factors such as allergies, pre-existing conditions (e.g., asthma or hypertension), or the user's age and medical history can impact the efficacy and safety of these suggestions. Personalized recommendations based on user profiles would improve the system's reliability.
- **Dataset Imbalance for Rare Diseases**: The dataset used for training contained an imbalance, with certain rare diseases underrepresented. As a result, the models were less accurate in predicting these diseases, and the predictions for such conditions might not be as reliable as for more common diseases.
- **Lack of Real-time Data**: The system was not integrated with real-time data from clinical or wearable devices, which could improve the prediction accuracy by consid- ering current health data trends.

- **Limited Scope of Remedies**: While the home remedy suggestions were helpful, they were limited to basic reme- dies and could be expanded with more comprehensive medical recommendations.

In future work, addressing these limitations by incorporating symptom severity, time-based symptom progression, and real- time user feedback will enhance the overall system's accuracy and user experience.

VIII. Conclusion and Future Work

MediCompanion AI successfully demonstrates the feasibil- ity and effectiveness of a symptom-based disease prediction system coupled with home remedy recommendations. By leveraging multiple machine learning models such as Random Forest, Decision Tree, K-Nearest Neighbors, and Naive Bayes, the system achieved robust performance across a diverse range of diseases. Each model contributed uniquely to the prediction process, allowing the system to deliver accurate and reli- able diagnostic suggestions based on user-provided symptoms. The incorporation of ensemble learning techniques further improved prediction stability and reduced the likelihood of overfitting. One of the key achievements of this project is the deployment of the model via a web-based platform. This approach enables global accessibility, allowing users to input their symp- toms conveniently and receive preliminary health guidance without the need for specialized equipment or clinical visits. The user-friendly interface and instant results make MediCom- panion AI a valuable tool for preliminary health assessment, especially in areas with limited access to healthcare services. However, there remains substantial scope for future en- hancements. In the upcoming iterations, the integration of Natural Language Processing (NLP) techniques could allow users to describe their symptoms in free-text format rather than selecting from a predefined list. This would make the system more natural, intuitive, and inclusive, particularly for users unfamiliar with medical terminology. Moreover, adding multilingual support would significantly broaden the system's reach, making it accessible to non- English-speaking populations and promoting global health inclusivity. Another potential advancement involves incorpo- rating real-time doctor consultations through the platform, pro- viding users with expert validation and personalized medical advice based on the AI's initial predictions. Additionally, utilizing real-world clinical datasets and con- tinuously gathering user feedback will be vital for enhancing the system's diagnostic accuracy and trustworthiness. Regular updates and retraining of models based on new medical findings and patient records will ensure that MediCompanion AI remains current, reliable, and aligned with best healthcare practices. Overall, MediCompanion AI lays a solid foundation for intelligent, accessible healthcare support and presents exciting opportunities for future research and development.

References

1. Kumar, P. et al., ”Disease Prediction using Decision Tree,” Journal of Healthcare Informatics, 2022. This study explores the use of Decision Tree algorithms in disease prediction models, focusing on optimizing splitting criteria and pruning methods to enhance diagnostic accuracy in healthcare datasets.
2. Gupta, R. and Sharma, M., ”Random Forest Approach for Health Diagnosis,” IEEE Access, 2021. The authors present a Random Forest-based system capable of diagnosing multiple diseases by aggregating decision trees, showcasing improved accuracy and robustness against noisy medical data.
3. Singh, A. et al., ”Naive Bayes Classifier in Healthcare Systems,” Procedia Computer Science, 2020. This paper highlights the application of Naive Bayes classifiers in healthcare, emphasizing their simplicity, computational efficiency, and surprising effectiveness despite strong independence assumptions among features.
4. Patel, V. et al., ”KNN based Diagnosis Models,” International Journal of Health Sciences, 2019. The study details the development of K-Nearest Neighbors (KNN) models for health diagnostics, discussing the effects of various distance metrics and neighbor selection strategies on predictive outcomes.
5. Mehra, S. et al., ”A Study on Symptom Checkers,” Journal of Medical Systems, 2018. The research examines various online symptom checker platforms, evaluating their diagnostic reliability, usability, and the chal- lenges faced when integrating such tools into clinical practice.
6. Reddy, P. et al., ”Integration of Home Remedies in AI Systems,” Health Informatics Journal, 2020. This paper proposes a framework for integrating traditional home remedies into AI-driven healthcare systems, aiming to offer complementary advice alongside standard clinical recommendations.
7. Zhang, Y. et al., ”Ensemble Methods for Medical Diagnosis,” IEEE Transactions on Healthcare Systems, 2021. The authors survey ensemble techniques, such as bagging, boosting, and stacking, analyzing their impact on improving diagnostic precision and reliability across different medical domains.
8. Zhao, J. et al., ”Deep Learning for Symptom Analysis,” Springer Health- care Journal, 2022. This work discusses the application of deep learning methods, particularly convolutional and recurrent neural networks, to automatically analyze symptoms described in text or structured formats.
9. Ali, M. et al., ”AI-Based Healthcare Chatbots,” ACM Computing Surveys, 2020. The paper reviews advancements in healthcare chatbots powered by AI, evaluating their natural language understanding capa- bilities, user engagement strategies, and ethical considerations in patient interaction.
10. Choudhary, N. et al., ”Predictive Models in Healthcare,” Journal of Biomedical Informatics, 2021. The authors present a comparative analysis of different machine learning models used in healthcare, emphasizing model interpretability, performance metrics, and real-world deployment challenges.

11. Thomas, K. et al., ”Symptom-Based Diagnosis Using AI,” Future Healthcare Journal, 2020. This study delves into AI-based frameworks that utilize patient-reported symptoms for disease prediction, discussing issues related to symptom ambiguity and the need for robust data preprocessing.
12. Kumar, S. et al., ”Flask-Based Web Deployment for Health Appli- cations,” International Conference on Computing, 2021. The paper illustrates how Flask, a lightweight Python framework, can be used for deploying healthcare models as web applications, ensuring scalability, security, and ease of integration with existing hospital information systems.