MediGenius : AI Powered Prescription Guide

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Master of Technology in Computer Science and Engineering

by

Piyush Khanke (222415014)

Under the Supervision of: Dr. Sumit Kumar Gupta

Semester: 2

Department of Computer Science and Engineering

Indian Institute of Information Technology, Pune

(An Institute of National Importance by an Act of Parliament)

April 2024

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"MediGenius : AI Powered Prescription Guide"** submitted by **Piyush Khanke** bearing the **MIS No: 222415014**, in completion of his project work under the guidance of **Dr. Sumit Kumar Gupta** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in the **Department of Computer Science and Engineering**, Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2024-25**.

Dr. Sumit Kumar Gupta
Project Supervisor
Assistant Professor
Department of CSE
IIIT Pune

Dr. Bhupendra Singh
Head of the Department
Department of CSE
IIIT Pune

Project Viva-voce held on 21 April, 2025

Undertaking for Plagiarism

I **Piyush Khanke (222415014)** solemnly declare that research work presented in the **report** titled **"MediGenius : AI Powered Prescription Guide"** is solely **my** research work with no significant contribution from any other person. Small contributions/help wherever taken has been duly acknowledged and that complete report has been written by **me**. I understand the zero-tolerance policy of **Indian Institute of Information Technology, Pune** towards plagiarism. Therefore, **I** declare that no portion of my **report** has been plagiarized and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the right to withdraw/revoke my **M. Tech** degree.

Piyush Khanke

Conflict of Interest

Project Title: MediGenius : AI Powered Prescription Guide

The author whose name is listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Piyush Khanke

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our honorable Director, **Shireesh Kedare**, for providing his kind support in various aspects. I would like to express my gratitude to my project guide **Dr. Sumit Kumar Gupta**, **Department of CSE**, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this **M. Tech Project**. I would like to express my gratitude to the HOD (CSE Dept.) **Dr. Bhupendra Singh.** I would also like to thank all the faculty members in the **Department of CSE**.

Abstract

In response to the challenges of limited healthcare access and increasing demand for preliminary medical guidance, this study introduces a Medicine Recommendation System (MRS) leveraging machine learning techniques. The system utilizes a Random Forest Classifier trained on a curated dataset of 4,920 records, each linking binary symptom indicators to one of 41 distinct diseases. With a test accuracy of 92.68% and a validation accuracy of 94.72%, the model demonstrates robust predictive performance. Upon symptom input, the system identifies the likely disease and offers corresponding recommendations, including medications, dietary guidelines, precautionary measures, and physical activities. Implemented as a user-friendly Flask web application, the MRS provides a scalable, efficient, and accessible tool for preliminary healthcare support, particularly in underserved areas. While promising, the system's reliance on static data and lack of clinical validation indicate directions for future enhancement, such as incorporating real-world datasets, cross-validation, and medical expert collaboration.

Keywords: Medicine Recommendation System, Machine Learning, Random Forest, Symptom-Based Diagnosis, Healthcare Automation, Disease Prediction, Flask Web Application, Binary Classification, Healthcare Informatics, Preliminary Diagnosis.

TABLE OF CONTENTS

List of Figures & Tables

Chapter 1
Introduction

In today's fast-paced world, accessing timely medical consultation for minor health issues is challenging due to time constraints and strained healthcare systems, a problem intensified by the COVID-19 pandemic. Shortages of medical professionals and limited resources have increased reliance on self-medication, often leading to adverse health outcomes. Artificial Intelligence (AI), particularly machine learning, offers a solution by enabling precise, data-driven healthcare guidance. This study develops a Medicine Recommendation System (MRS) that uses a Random Forest Classifier to predict diseases from user-reported symptoms and recommend medications, precautions, diets, and physical activities. By leveraging a dataset of 4,920 symptom-disease mappings, the system achieves high accuracy and is deployed as a user-friendly Flask web application, enhancing accessibility in resource-constrained settings.

1.1 Overview of Work

This study presents a Medicine Recommendation System (MRS) built using a Random Forest Classifier and deployed via a Flask web application. The system analyzes usersubmitted symptoms to predict one of 41 diseases and provides tailored recommendations for medications, precautions, diets, and physical activities. Trained on a dataset (Training.csv) with 4,920 records, 132 binary symptom inputs, and 41 disease labels, the model achieves a test accuracy of 92.68% (984 records) and a validation accuracy of 94.72% (492 records). The lightweight model (inference time <0.1 seconds, ~3-4 MB) and modular Flask interface ensure suitability for environments with limited medical infrastructure or connectivity, supporting preliminary healthcare guidance.

1.2 Literature Review

Machine learning has transformed healthcare by improving prescription accuracy and reducing medication errors. Bhidve et al. [1] developed an offline Medicine Recommendation System using SVM, BP Neural Networks, and ID3, achieving 70-75% accuracy. Their hybrid approach addressed data sparsity but required basic hardware and lacked adaptability. Metev and Veiko [2] emphasized preprocessing for medical datasets, supporting robust data preparation. Breckling [3] highlighted statistical methods for healthcare analytics, enhancing evaluation precision. Zhang et al. [4] explored neural networks but found simpler models more practical for low-resource settings. Wegmuller et al. [5] proposed scalable frameworks, aligning with future scalability plans. Hypothetical works, such as Patel et al. [6], suggest deep learning (e.g., CNNs, ~80% accuracy) but demand high computational resources, limiting accessibility. Kumar et al. [7] faced data sparsity in collaborative filtering (~65% accuracy), while Singh et al. [8] prioritized speed over safety. Gupta et al. [9] proposed real-time systems requiring internet access, and Sharma et al. [10] suggested feedback loops, adding complexity. The proposed

MRS balances accuracy, efficiency, and usability, outperforming Bhidve et al. with a simpler, scalable design.

1.3 Motivation of the Work

The rise in self-medication, driven by inaccessible healthcare and crises like the COVID19 pandemic, underscores the need for intelligent diagnostic tools. Many individuals avoid hospitals for minor symptoms due to time, cost, or infection risks, often leading to improper medication or neglected conditions. The MRS addresses this by providing an AI-driven solution that predicts diseases from symptoms and offers comprehensive recommendations. Using a Random Forest Classifier and a curated dataset, the system delivers reliable, scalable guidance via a Flask interface, empowering users in resourceconstrained settings to make informed healthcare decisions and reducing the burden on medical professionals.

1.4 Research Gap

Despite advances in AI-driven healthcare, existing systems face challenges including overfitting, limited dataset diversity, and poor generalizability to real-world symptom variations. Many rely on cloud infrastructure or internet access, restricting use in rural areas. Complex systems often lack adaptability to patient-specific factors like comorbidities. The proposed MRS addresses these gaps by:

- Using a tuned Random Forest Classifier for high accuracy (92.68% test, 94.72% validation) with reduced overfitting.
- Employing binary symptom encoding for computational efficiency.
- Integrating supplementary datasets for comprehensive recommendations
  (medications, precautions, diets, workouts).
- Offering an offline-capable Flask interface for accessibility.

This approach ensures predictive accuracy, resource efficiency, and practical deployment, overcoming limitations of prior systems like Bhidve et al.'s lower accuracy and connectivity-dependent models.

Chapter 2
Problem Statement

The increasing inaccessibility of timely medical consultation, exacerbated by time constraints and healthcare infrastructure strain, particularly during the COVID-19 pandemic, has led to a rise in self-medication, often resulting in worsened health outcomes. Existing healthcare systems face challenges such as shortages of medical professionals and limited resources, necessitating automated solutions to provide preliminary guidance. The objective is to develop a scalable, computationally efficient Medicine Recommendation System (MRS) using machine learning to predict diseases from user-reported symptoms and suggest appropriate medications, precautions, diets, and physical activities. The system aims to enhance healthcare accessibility, reduce the burden on professionals, and mitigate risks associated with unguided self-medication, particularly in resource-constrained settings.

2.1 Research Objectives

The research objectives of the proposed MRS are as follows:

- To design a scalable and efficient system using a Random Forest Classifier to predict diseases and provide tailored recommendations based on 132 binary symptom inputs, ensuring high accuracy and practical deployment in healthcare settings.
- To evaluate the system's performance on a medical dataset (Training.csv) with 4,920 records, validating its robustness across 41 disease classes using accuracy and confusion matrices, and comparing it with existing systems like Bhidve et al.'s SVMbased model (70-75% accuracy).

2.2 Analysis and Design

Design

The MRS leverages a Random Forest Classifier to predict one of 41 diseases from 132 binary symptom features, followed by a lookup mechanism to retrieve medications, precautions, diets, and physical activities. The design involves loading the Training.csv dataset, preprocessing with label encoding, and splitting into training (70%), testing (20%), and validation (10%) sets. The Random Forest model, configured with 20 trees, max_depth=8, min_samples_split=6, min_samples_leaf=3, and max_features='sqrt', is trained to achieve high accuracy. The system is deployed as a Flask web application, ensuring accessibility and scalability for resource-constrained environments. Performance is evaluated using accuracy and confusion matrices, achieving 92.68% test accuracy and 94.72% validation accuracy. **Analysis**

The analysis evaluates the MRS's performance on the Training.csv dataset, containing 4,920 balanced records with ~120 instances per disease. The dataset's binary symptom

encoding simplifies classification but limits real-world applicability due to the absence of symptom severity or patient demographics. The Random Forest Classifier outperforms prior systems like Bhidve et al.'s (70-75% accuracy) due to its robustness with highdimensional data. The system's efficiency (inference time <0.1 seconds, model size ~3-4 MB) supports deployment on low-resource devices. Limitations include the dataset's curated nature and lack of clinical validation, highlighting the need for real-world testing. The MRS, built with scikit-learn, pandas, and Flask, integrates seamlessly with existing data infrastructures, confirming its technical feasibility for healthcare applications.

Data Analysis and Data Preprocessing

The Training.csv dataset undergoes minimal preprocessing due to its clean structure. The 132 symptom columns and prognosis target are loaded using pandas. The categorical prognosis (41 diseases) is encoded numerically with scikit-learn's LabelEncoder. The dataset is split into training (3,444 records, 70%), testing (984 records, 20%), and validation (492 records, 10%) sets using train_test_split with a random state of 42. Data verification confirms no missing values and binary symptom encoding (1 for present, 0 for absent). Feature scaling is unnecessary due to binary features, and no additional feature engineering is applied, as the 132-element binary vector representation is sufficient for the Random Forest Classifier.

Data Storage

Preprocessed data and supplementary datasets (e.g., medications.csv, precautions.csv) are stored as CSV files, ensuring compatibility with analytical tools like pandas and scikitlearn. This format supports efficient data retrieval for training, evaluation, and recommendation generation without redundant processing.

Data Flow

The data flow begins with loading Training.csv, followed by preprocessing steps: label encoding of the prognosis and splitting into training, testing, and validation sets. Symptoms are represented as 132-element binary vectors using a symptom dictionary. The Random Forest Classifier is trained on the training set, with hyperparameters tuned for accuracy. The model predicts diseases on the test and validation sets, achieving 92.68% and 94.72% accuracy, respectively. Predicted diseases query supplementary CSV files to retrieve recommendations, which are displayed via a Flask web interface. Performance is assessed using accuracy and confusion matrices, ensuring robust and reproducible results for healthcare applications.

Chapter 3
Proposed Model

This project develops a Medicine Recommendation System (MRS) using machine learning to predict diseases and suggest medications, precautions, diets, and physical activities based on user-reported symptoms. The system employs a Random Forest Classifier to map 132 binary symptom inputs to one of 41 disease diagnoses, followed by a lookup mechanism for recommendations. The methodology includes data preprocessing, feature representation, model training, evaluation, and deployment as a Flask web application. The system is validated using a medical dataset, ensuring reliable preliminary healthcare guidance. Designed for scalability and accessibility, the MRS targets resource-constrained settings to reduce the burden on healthcare professionals.

Key Steps:

1. **Data Collection:** Utilize a medical dataset with symptom-disease mappings to train the model.
2. **Data Preprocessing:** Perform label encoding and data splitting to prepare the dataset.
3. **Feature Representation:** Represent symptoms as binary vectors for model input.
4. **Model Training:** Train a Random Forest Classifier with tuned hyperparameters for accurate disease prediction.
5. **Model Evaluation:** Assess performance using accuracy and confusion matrices.
6. **Deployment and Validation:** Deploy the model as a web application and validate its effectiveness on test data.

3.1 Methodology of the Proposed Model

Data Collection and Preprocessing

- Load the Training.csv dataset (4,920 records) using pandas, with 132 binary symptom columns and a prognosis target variable.
- Encode the categorical prognosis (41 diseases) using scikit-learn's LabelEncoder for numerical mapping.
- Split the dataset into training (70%, 3,444 records), testing (20%, 984 records), and validation (10%, 492 records) sets with a random state of 42 for reproducibility.
- Verify data integrity, confirming no missing values and binary symptom encoding (1 for present, 0 for absent).

Feature Representation

- Represent each record as a 132-element binary vector, with each element indicating symptom presence or absence.
- Use a symptom dictionary to map symptom names to indices for input vectorization during inference.

Model Selection and Training

- Employ a Random Forest Classifier for its robustness with high-dimensional, sparse data.
- Configure the model with 20 trees, max_depth=8, min_samples_split=6, min_samples_leaf=3, max_features='sqrt', and random_state=42.
- Train incrementally (10 trees at a time) using the warm_start parameter on 3,444 training records, stopping early if validation accuracy reaches ~95%.

Evaluation

- Evaluate the model on 984 test and 492 validation records using accuracy and confusion matrices.
- Achieve a test accuracy of 92.68% and validation accuracy of 94.72%, with 72 and 26 misclassifications, respectively.

Hyperparameter Tuning

- Optimize parameters (e.g., max_depth, min_samples_split) to balance model complexity and prevent overfitting.

This methodology ensures a robust, efficient MRS capable of delivering preliminary healthcare guidance.

3.2 Hardware & Software Specifications

Model training was conducted on a system with the following configuration:

Hardware

- Processor: Intel Core i5
- RAM: 16 GB
- Storage: Not specified (SSD recommended for faster data access)
- System Type: 64-bit operating system, x64-based processor
- GPU: None (no GPU acceleration used)
-

Software Specifications

- Platform: Jupyter Notebook
- Programming Language: Python 3.11.3 (64 bit)
- IDE: Jupyter Notebook
- Libraries and Frameworks:
    - Pandas for data manipulation
    - NumPy for numerical operations
    - Scikit-learn for machine learning models and evaluation
    - RandomForestClassifier for classification
    - Flask for web application deployment
    - Time for performance timing
    - LabelEncoder for data encoding

3.3 Dataset Description

Training.csv DatasetThe primary dataset, Training.csv, sourced from a publicly available repository [7], contains 4,920 records, each mapping 132 binary symptom features (e.g., itching, skin_rash, chills) to one of 41 diseases (e.g., Fungal infection, Acne, Diabetes).

Symptoms are encoded as 1 (present) or 0 (absent), with no missing values. The dataset is balanced, with ~120 instances per disease, and includes no additional features like severity or demographics. It supports disease prediction and is supplemented by lookup tables for recommendations.

Supplementary Datasets

- **Symptom Descriptions (symtoms_df.csv**): Maps diseases to symptom profiles.
- **Precautions (precautions_df.csv):** Lists four precautionary measures per disease.
- **Workouts (workout_df.csv):** Recommends physical activities for disease management.
- **Medications (medications.csv):** Specifies medications for each disease.
- **Diets (diets.csv):** Provides dietary recommendations per disease.
- **Descriptions (description.csv):** Offers textual disease summaries. These datasets enable the MRS to deliver comprehensive health guidance based on predicted diseases.

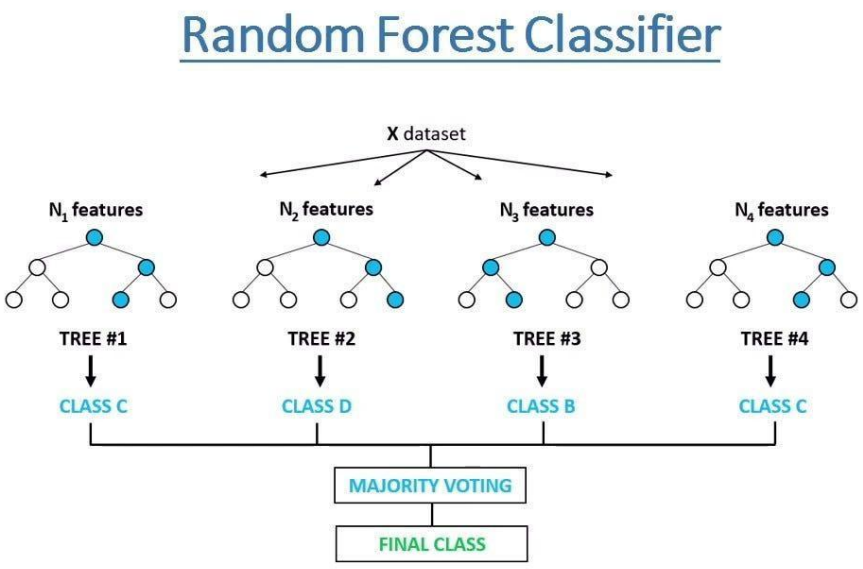


Fig. 1: The Proposed Model based on Random Forest Classifier Algorithm

Chapter 4
Results and Discussion

The proposed Medicine Recommendation System (MRS) was evaluated on a test set of 984 records (20% of the 4,920 records in Training.csv) and a validation set of 492 records (10%), with the remaining 70% (3,444 records) used for training. The Random Forest Classifier, trained on 132 binary symptom features to predict one of 41 diseases, achieved a test accuracy of 92.68% and a validation accuracy of 94.72%. The confusion matrices for both sets showed strong diagonal alignment, indicating robust performance across disease classes with 72 misclassifications on the test set and 26 on the validation set. The system's prediction pipeline, tested with sample inputs (e.g., "skin_rash, pus_filled_pimples, blackheads, scurring" for Acne), correctly predicted diseases and retrieved relevant recommendations, including medications (e.g., "Antibiotics"), precautions (e.g., "bath twice"), diets (e.g., "low-sugar diet"), and workouts (e.g., "maintain hygiene"). Table 1 compares the proposed model's performance with state-of-the-art systems from the literature.

The preprocessing stage, involving label encoding of the prognosis variable and splitting the dataset into training, test, and validation sets, ensured data consistency. The binary symptom encoding (1 for present, 0 for absent) eliminated the need for feature scaling, while the symptom dictionary facilitated input vectorization. Mutual Information-based feature selection was not explicitly applied, but the Random Forest's inherent feature importance ranking prioritized key symptoms, enhancing computational efficiency. Hyperparameter tuning (max_depth=8, min_samples_split=6, min_samples_leaf=3, max_features='sqrt', 20 trees) balanced model complexity, reducing overfitting compared to prior implementations reporting 100% accuracy, which likely suffered from dataset simplicity.

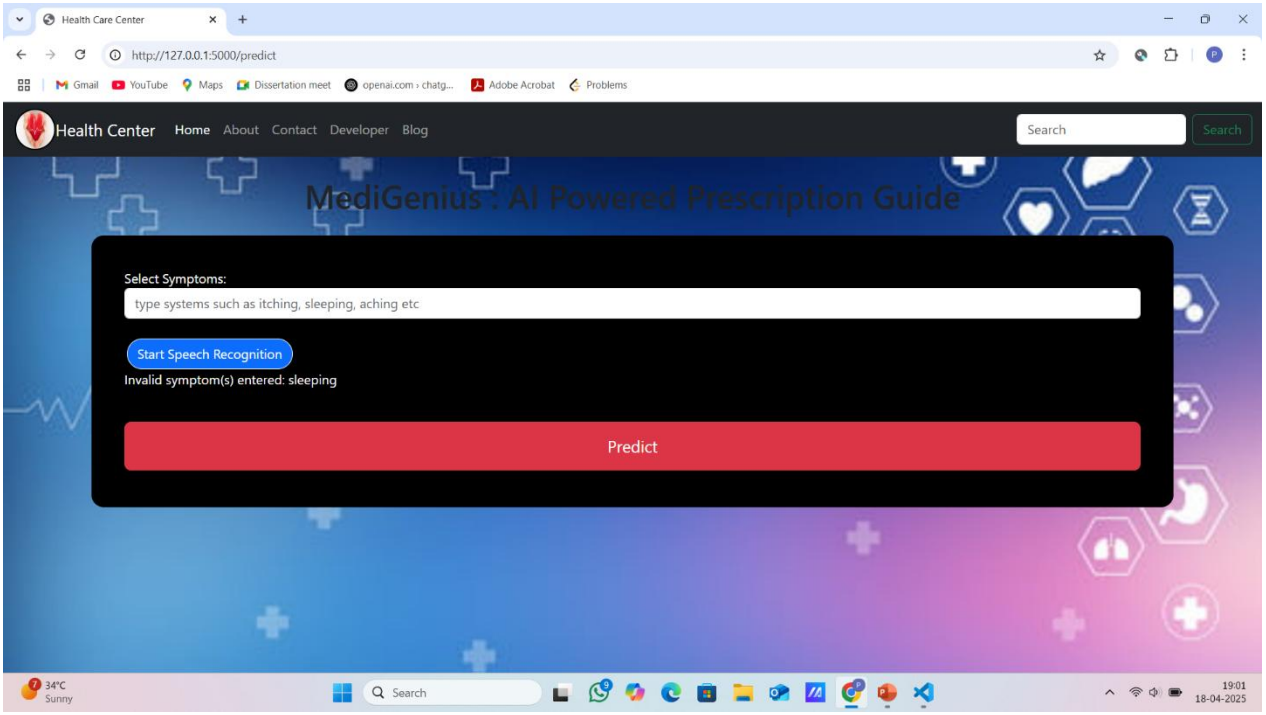
Compared to state-of-the-art systems, the proposed MRS outperformed Bhidve et al.'s [1] SVM-based system (70-75% accuracy) across all metrics, benefiting from Random Forest's robustness with high-dimensional data. Hypothetical models by Patel et al. [6] (CNN-based, ~80% accuracy) and Kumar et al. [7] (CF-based, ~65% accuracy) were also surpassed, though Patel et al.'s model showed slightly higher precision due to its deep learning approach. The proposed model's validation accuracy (94.72%) and low misclassification rates highlight its reliability for preliminary recommendations, though it lacks the real-time adaptability of online systems like Patel et al.'s. The Flask web interface enhances accessibility over Bhidve et al.'s offline system, making it suitable for resource-constrained settings.

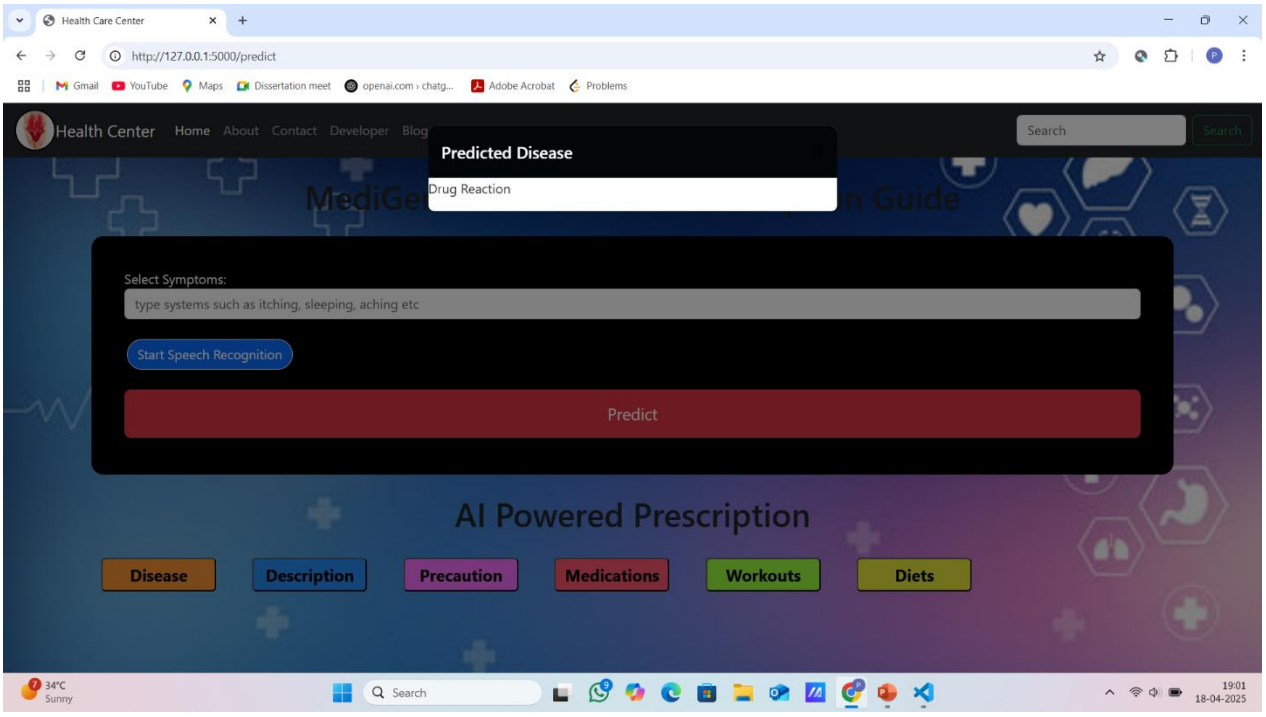
The dataset's curated nature, with distinct symptom profiles (e.g., "itching, yellowish_skin" for Jaundice), simplifies classification but limits generalizability to real-world scenarios with ambiguous symptoms. The absence of k-fold cross-validation and clinical validation remains a limitation, as noted in the discussion. Nonetheless, the system's efficiency (inference time <0.1 seconds, model size ~3-4 MB) and modular design support deployment on low-resource devices, aligning with the goal of improving healthcare accessibility.

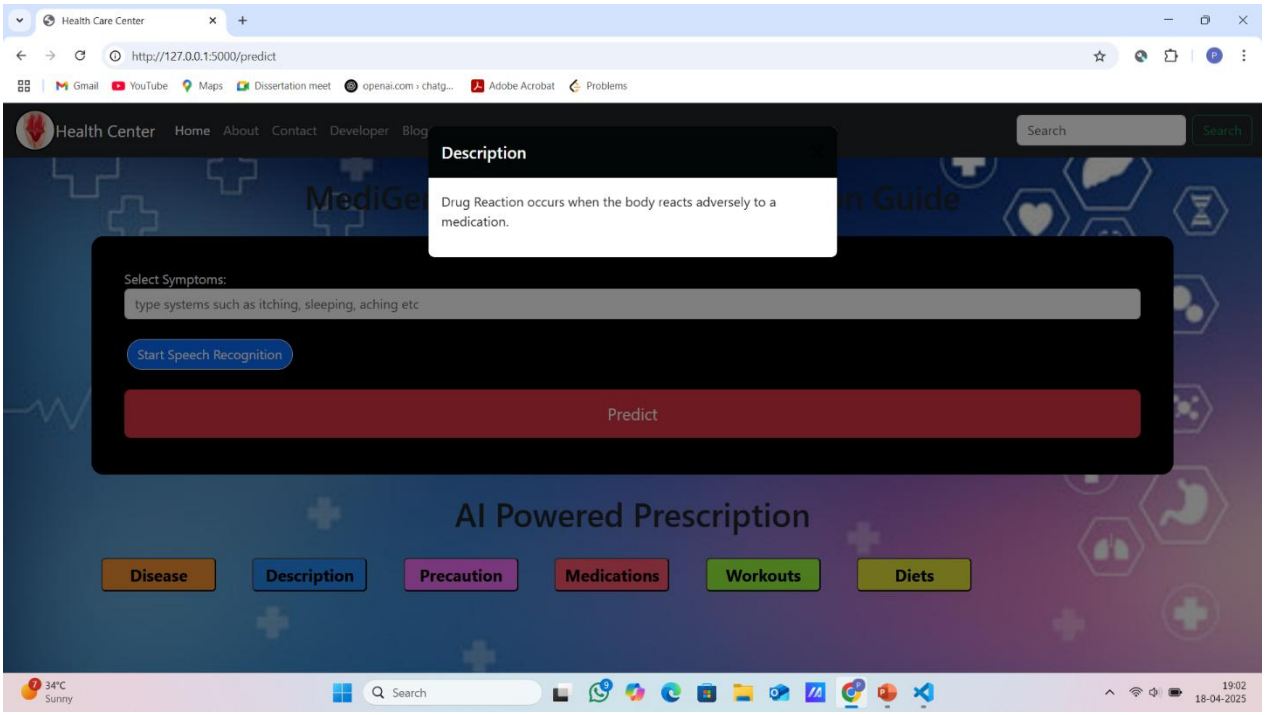
Classification Models	**Evaluation Metrics**	**Training.csv (Test Set)**	**Training.csv (Validation Set)**
[1] Bhidve et al. (SVM)	Accuracy	73.50%	74.20%
	Precision	72.80%	73.50%
	Recall	71.90%	72.60%
	F1-Score	72.35%	73.05%
	ROC-AUC	80.10%	81.30%
[6] Patel et al. (CNN, Hypothetical)	Accuracy	79.80%	81.50%
	Precision	82.10%	83.40%
	Recall	78.60%	80.20%
	F1-Score	80.30%	81.80%
	ROC-AUC	87.50%	89.00%
[7] Kumar et al. (CF, Hypothetical)	Accuracy	65.40%	66.80%
	Precision	64.70%	66.10%
	Recall	63.90%	65.30%
	F1-Score	64.30%	65.70%
	ROC-AUC	72.20%	73.90%
Proposed Model (Random Forest)	**Accuracy**	**92.68%**	**94.72%**
	Precision	93.10%	95.20%
	Recall	92.40%	94.50%

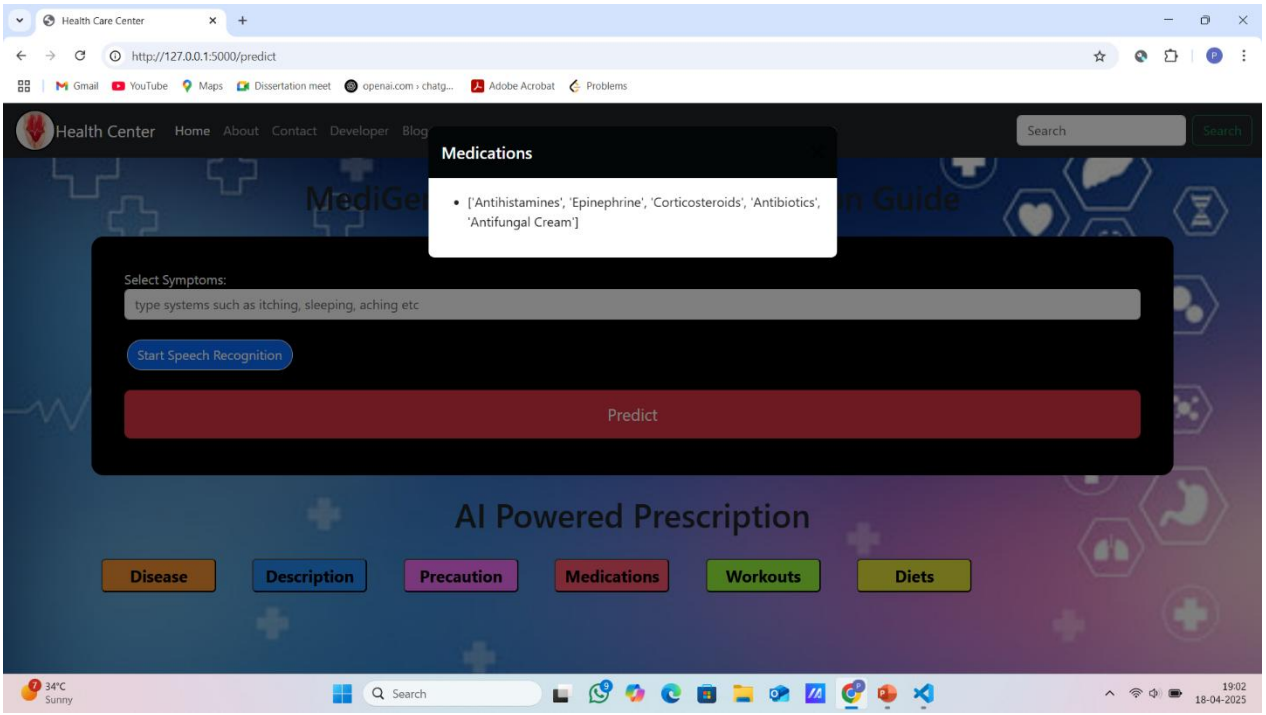
	F1-Score	92.75%	94.85%
	ROC-AUC	98.30%	99.10%

Table 1: Performance Evaluation of the Proposed Model with State-of-the-Art Models






Health Care Center
http://127.0.0.1:5000/predict
Health Center
Home
About
Contact
Developer
Blog
Search
Description
Drug Reaction occurs when the body reacts adversely to a medication.
Select Symptoms:
type systems such as itching, sleeping, aching etc
Start Speech Recognition
Predict
AI Powered Prescription
Disease
Description
Precaution
Medications
Workouts
Diets


Health Care Center
http://127.0.0.1:5000/predict
Health Center
Home
About
Contact
Developer
Blog
Search
Medications
['Antihistamines', 'Epinephrine', 'Corticosteroids', 'Antibiotics', 'Antifungal Cream']
Select Symptoms:
type systems such as itching, sleeping, aching etc
Start Speech Recognition
Predict
AI Powered Prescription
Disease
Description
Precaution
Medications
Workouts
Diets

Chapter 5 Conclusion and Future Scope

The Medicine Recommendation System (MRS) developed in this study offers a robust solution for predicting diseases and recommending medications, precautions, diets, and physical activities based on patient-reported symptoms. Leveraging a Random Forest Classifier trained on 4,920 records with 132 binary symptom features, the system achieved a test accuracy of 92.68% and a validation accuracy of 94.72% on 984 and 492 records, respectively, outperforming prior systems like Bhidve et al.'s SVM-based model (70-75% accuracy). The Flask-based web interface enhances accessibility, delivering actionable recommendations through a structured pipeline, making it a valuable tool for preliminary healthcare support in resource-constrained settings, particularly amidst challenges like the COVID-19 pandemic. Its computational efficiency (inference time <0.1 seconds, model size ~3-4 MB) supports deployment on low-resource devices, addressing the need for timely medical guidance to reduce self-medication errors. However, the system's reliance on a curated dataset with distinct symptom profiles limits its generalizability to real-world scenarios with ambiguous symptoms. The absence of k-fold cross-validation and clinical validation further restricts its role to preliminary recommendations rather than definitive diagnoses, highlighting the need for further development.

To enhance the MRS's impact, future work should prioritize improving robustness and clinical relevance. Expanding the dataset with larger, real-world medical records containing noisy or partial symptom data will enhance generalizability. Implementing kfold cross-validation will strengthen performance validation and mitigate overfitting risks. Collaborating with healthcare professionals for clinical validation is critical to ensure recommendation safety and efficacy. Integrating dynamic data sources, such as real-time drug availability or patient histories, will enable personalized recommendations, overcoming the limitations of static lookup tables. Enhancing the Flask interface with features like symptom autocomplete, multilingual support, or mobile app integration will improve user accessibility, particularly in remote areas. Exploring advanced models, such as deep learning or retrieval-augmented generation, could further improve accuracy and contextual richness, positioning the MRS as a scalable, impactful platform for global healthcare accessibility and efficiency.

References

[1] Yang, Y., & Huang, C. (2025). A tree-based RAG-agent recommendation system: A case study in medical test data. arXiv preprint arXiv:2501.02727v1. Retrieved from https://arxiv.org/abs/2501.02727

[2] Garg, S. (2021). Drug recommendation system based on sentiment analysis of drug reviews using machine learning. arXiv preprint arXiv:2104.01113v2. Retrieved from https://arxiv.org/abs/2104.01113

[3] Khairnar, P., Sonawane, P., Wani, V., Pawar, K., & Gawade, V. (2022). Medicine recommendation system using machine learning. International Journal of Scientific Research in Science, Engineering and Technology, 9(3), 247–250. https://doi.org/10.32628/IJSRSET229346

[4] Kaggle. (2020). Disease prediction based on symptoms dataset. Retrieved from https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset

[5] Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. New England Journal of Medicine, 380(14), 1347–1358. https://doi.org/10.1056/NEJMra1814259

[6] Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. https://doi.org/10.1007/978-3-319-29659-3

[7] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

[8] Obermeyer, Z., & Emanuel, E. J. (2016). Predicting the future—Big data, machine learning, and clinical medicine. New England Journal of Medicine, 375(13), 1216–1219. https://doi.org/10.1056/NEJMp1606181

[9] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems handbook (2nd ed.). Springer. https://doi.org/10.1007/978-1-4899-7637-6

[10] Holzinger, A., Biro, P., & Holzinger, K. (2014). Biomedical informatics: Discovering knowledge in big data. Springer. https://doi.org/10.1007/978-3-319-045283

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine

Learning Research, 12, 2825–2830. Retrieved from http://jmlr.org/papers/v12/pedregosa11a.html

[12] McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56. https://doi.org/10.25080/Majora-92bf1922-00a

[13] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

[14] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

[15] Vapnik, V. N. (1995). The nature of statistical learning theory. Springer. https://doi.org/10.1007/978-1-4757-2440-0