Drug Recommendation System in General Medical Emergencies

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

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted by:

VAIBHAVI SHIVANNA	**BU21CSEN0200182**
GUDURU HARSHIYA SULTHANA	**BU21CSEN0200061**
LINGUTLA THRISHA SAI	**BU21CSEN0200064**
LEVIDI SANDHYA	**BU21CSEN0200172**

Under the esteemed guidance of

SILPA C

ASSISTANT PROFESSOR

Department of Artificial Intelligence & Data Science

GITAM SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Deemed to be University)

Bengaluru Campus.

April 2025

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled **"Drug Recommendation System in General Medical Emergencies"** is an original work done in the **Department of Artificial Intelligence And Data Science, GITAM School of Technology, GITAM (Deemed to be University), Bengaluru** submitted in partial fulfilment of the requirements for the award of the degree of **B.Tech.** in Artificial Intelligence And Data Science. The work has not been submitted to any other college or University for the award of any degree.

Date:

Registration No(s).	Name(s)	Signature(s)
BU21CSEN0200182	**VAIBHAVI SHIVANNA**	
BU21CSEN0200061	**GUDURU HARSHIYA SULTHANA**	
BU21CSEN0200064	**LINGUTLA THRISHA SAI**	
BU21CSEN0200172	**LEVIDI SANDHYA**	

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled **"Drug Recommendation System in General Medical Emergencies"** is a bonafide record of work carried out by **Vaibhavi Shivanna (BU21CSEN0200182), Guduru Harshiya Sulthana (BU21CSEN0200061), Lingutla Thrisha Sai (BU21CSEN020064), Levidi Sandhya (BU21CSEN0200172),** submitted in partial fulfillment of requirement for the award of degree of **Bachelors of Technology in Artificial Intelligence And Data Science**.

Project Guide	**Head of the Department**
SIGNATURE OF THE GUIDE	**SIGNATURE OF THE HoD**
SILPA C	**Prof. A. Vadivel**
Assistant Professor	Professor

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it our privilege to express our gratitude to all those who guided us in the completion of the project.

We express our gratitude to Director **Prof. Basavaraj Gundappa Katageri** for having provided us with the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank **Prof. A. Vadivel,** HOD, Department of Artificial Intelligence And Data Science, Gandhi Institute of Technology and Management, Bengaluru for the immense support given to us.

We express our gratitude to our project guide **Ms. SILPA C,** Department of Artificial Intelligence And Data Science, Gandhi Institute of Technology and Management, Bengaluru, for their support, guidance, and suggestions throughout the project work.

Student Name's	Registration No.
VAIBHAVI SHIVANNA	**BU21CSEN0200182**
GUDURU HARSHIYA SULTHANA	**BU21CSEN0200061**
LINGUTLA THRISHA SAI	**BU21CSEN0200064**
LEVIDI SANDHYA	**BU21CSEN0200172**

ABSTRACT

During medical emergencies, timely and accurate drug recommendations are crucial, especially when doctors are unavailable. This study aims to develop a Drug Recommendation System (DRS) using machine learning techniques to assist in making faster and more reliable decisions for patient care.

The system processes real-time patient data, including symptoms and medical history, leveraging predictive analytics to generate personalized medication recommendations. Additionally, user feedback is implemented to understand the system's accuracy.

The DRS enhances decision-making by providing real-time, secure, and personalized drug recommendations, improving response time and accuracy in acute medical situations while reducing dependency on immediate human intervention.

This system has the potential to significantly improve emergency medical care by ensuring timely treatment and enhancing overall patient outcomes. By integrating real-time data processing and feedback, the system continuously refines its recommendations, making it a valuable tool in healthcare.

Keywords: *Drug Recommendation System, Machine Learning, Medical Emergencies, Real-Time Data Processing.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The development of healthcare technology has brought new possibilities, especially in responding to the enduring issues of emergency medical care where quick and precise drug delivery can be the difference between life and death. Drug Recommendation Systems (DRS) based on cutting-edge data science methods have come to be a critical resource in situations where doctors are not available, like in remote or underserved areas. Such systems facilitate prompt and knowledgeable drug choices by capitalizing on real-time data and advanced algorithms, providing individualized recommendations that optimize patient outcomes [1]. The potential to eliminate human errors and latencies in stressful emergency environments is one indicator of the increasing usability of such technology-based interventions.

Healthcare networks across the globe have the overwhelming challenge of providing timely treatment, one made even more complicated in regions where there is a shortage of access to skilled personnel. As was brought out in earlier research, the lack of immediate medical knowledge frequently results in second-best treatment, especially in the case of emergencies where seconds matter. The DRS proposed will fill this gap by dealing solely with medical emergencies, combining patients' live data, present symptoms, history, and allergies to put forward patient-specific drug recommendations. This focused initiative draws on the wider developments in healthcare technology, which have shown considerable promise for enhancing decision-support systems. By providing a lifeline in emergency cases, the DRS aims to revolutionize emergency care delivery in resource-poor settings.

The use of Artificial Intelligence (AI) and Machine Learning (ML) in medicine has been widely researched, with decision trees, random forests, and support vector machines being promising models for analyzing patient information for drug suggestions. Abhishek et al. [9], have signified how several ML models in further tuning of drug recommendation - Decision Trees, Random Forest, and Support Vector Machines being some of them. These will consider patient factors including symptoms, medical histories, and allergies and they only recommend drugs that comply with evidence-based medicine, which ultimately enhances patient outcomes. As the healthcare industry continuously strives to evolve for upgraded treatments for patients' better health and safety, In any general medical emergency first aid treatment becomes essential, drug recommendation systems based on the Machine learning model would become

an advanced and intuitive, fast-attaining higher acceptance in practice which enhances the reliability and precision of these systems. AI and ML-based systems will give the healthcare industry the ability to boost its efficiency, provide rational and calculated prescriptions based on evidence, and curtail errors in medical emergencies.

The complex task of drug prescribing depends largely upon the complexity that exists in medical data about the patients—the patients' histories, their allergies, and any potential drug interaction. In this regard, DRS solutions based on ML offer a state-of-the-art drug recommendation solution that will support healthcare professionals by giving an automated cross-verification of patient data with a hospital's database of medication and drug interactions. This will provide the most accurate prescription based on patient history while ensuring that the patient is safe [7]. Furthermore, Dubba et al. [3] highlight how sentiment analysis techniques can add to this to improve the recommendations based on patient reviews to incorporate user experience into AI-supported decision-making.

Most of this earlier research has been on general healthcare scenarios and the real-time needs of emergencies. The DRS suggested here takes advantage of these ML methods but emphasizes the importance of immediacy and ease of use, with a user-friendly web interface to make it accessible to patients. This supports the increasing popularity of intuitive healthcare solutions that empower users who have limited technical knowledge, a key consideration in remote locations where conventional medical care is not available.

Aside from the technical innovation, the DRS has important social implications in so far as it seeks to regularize access to healthcare. Where medical infrastructure remains underdeveloped, the possibility of offering quality drug recommendations from a web platform could significantly mitigate the strain placed on trained systems. Research indicates that incorporating real-time monitoring and electronic health records (EHRs) in such systems is likely to make treatment more efficient and safe for patients, with evidence-based recommendations being provided for each individual as per their unique needs. Feedback mechanisms incorporated also allow the system to adapt based on user feedback, improving its relevance and accuracy in the long term—a mechanism adopted from research focused on the ongoing improvement of recommendation systems [8,7].

This initiates the need to bridge healthcare differences, particularly in emergency scenarios. Traditional methods of drug prescription in emergencies often rely heavily on the presence of

skilled professionals, which may not always be available in rural or isolated communities. The DRS seeks to empower local healthcare workers and even patients themselves by providing a reliable, technology-driven alternative. Previous research shows that ML-based systems facilitate decision-making without the need for expert intervention, providing a viable solution to an acute issue [14,11]. The structure of the system ensures attachment to patient-centric care, with recommendations not only being timely but also tailored to the specific health profile of an individual.

This project develops a Drug Recommendation System (DRS) designed for general medical emergencies when doctors are unavailable. The system processes real-time patient data, including current symptoms, and medical history to provide customized and accurate drug recommendations. By leveraging machine learning techniques, the system ensures faster and more reliable decision-making, reducing delays in first treatments.

This project is focused on emergencies where timely drug recommendations can improve patient outcomes. While the system enhances decision-making, it does not replace professional medical advice but serves as a supportive tool in urgent conditions. Factors such as data availability, model accuracy, and integration with existing healthcare systems may impose certain limitations. However, by optimizing predictive analytics and incorporating user feedback, this system has the potential to improve emergency response and patient care.

Accordingly, the creation and deployment of such a system have significant limitations. The project is limited by a small-scale strategy, governed by considerations of cost, and availability of datasets. Creating a DRS involves significant capital expenditure in infrastructure, such as secure servers, real-time data processing capacity, and strong cybersecurity to secure sensitive patient data. These can be expensive, especially for mass deployment in low-resource environments where budgets are strained. Furthermore, the system's dependence on high-quality data is a major challenge since poor or incomplete data can be counterproductive to its efficiency—a point repeated in research emphasizing the need for data integrity in ML solutions.

The cost of implementing the DRS is not only at the outset but also an ongoing expense. There is a requirement for constant model training to maintain the system up-to-date with changing medical knowledge and clinical practice guidelines, a costly activity that involves continued funding. The small-scale nature of the project means it depends heavily on limited datasets,

which may not fully capture the diversity of patient populations or emergency scenarios. This dependency could limit the system’s generalizability, necessitating additional resources to scale up data collection efforts—a costly and logistically challenging task.

Despite these limitations, the DRS offers an imperative vision for the future of emergency medical care. Its emphasis on real-time analysis of data and patient-specific advice comports with the trend toward targeted medicine, a focus of studies on deep learning-based collaborative filtering for health suggestions. The fact that the system can function without doctors fills a significant gap, as it represents a useful innovation for healthcare practitioners in remote areas. Research supporting user response to understand recommendation accuracy has driven the DRS's design, integrating these principles within the emergent context of crises. This flexibility keeps the system knowledgeable about practical needs. The technical architecture of the DRS requires its integration with ML models, and real-time monitoring systems, a step that requires precise arrangement to achieve accuracy. Decision trees provide interpretability, random forests are highly accurate, and support vector machines are effective in complex data classification all being a strong analytical foundation.

2. LITERATURE SURVEY

[1] Yash Ritesh Tanna et al. implemented a hybrid framework utilizing TF-IDF, Bag of Words (BoW), and manual feature engineering to extract and represent textual features from drug reviews. These features were used to train machine learning models, including Linear SVC, Logistic Regression, and Random Forest. Linear SVC was employed to classify the data by finding an optimal hyperplane, Logistic Regression predicted probabilities for binary classification tasks, and Random Forest enhanced robustness by combining multiple decision trees. This innovative approach aimed to assist medical professionals and patients by providing reliable recommendations through the accurate categorization of user feedback and evaluation of drugs. However, the system lacked sentiment feedback integration and personalized recommendations, limiting its effectiveness in tailoring drug suggestions to individual users. Furthermore, the reliance on textual feature extraction without considering contextual embeddings constrained the system's ability to capture deep semantic relationships within drug reviews.

[2] Bhoomika Dubba et al. focused on employing sentiment analysis on drug reviews using vectorization techniques like TF-IDF, BoW, and Word2Vec. These methods transformed textual data into structured formats suitable for machine learning models, including Linear SVC, Decision Tree, and Logistic Regression. Linear SVC with TF-IDF was particularly effective in identifying attitudes within drug reviews by leveraging term frequency-inverse document frequency for precise term importance. The system was designed to help users and healthcare providers understand the sentiment of drug reviews, facilitating informed medication decisions. This system did not incorporate patient demographics such as age, gender, or medical history, which limited its ability to provide personalized drug recommendations. Additionally, the lack of dynamic learning prevented real-time adaptability to new drugs and emerging patient concerns.

[3] Abhishek et al. proposed a hybrid recommendation system that combined collaborative filtering, content-based filtering, and sentiment analysis to recommend medications. Using the "recommended lab" package in R, the system calculated similarities between drugs through techniques like cosine similarity and analyzed user-item interactions to suggest suitable medications. This approach was particularly useful in leveraging collective knowledge and historical data to provide recommendations for patients, reducing the burden on medical

professionals. However, the system lacked user-specific personalization and the ability to adapt recommendations based on individual patient profiles, which are critical for precision medicine. Furthermore, the absence of deep learning-based models limited the system's ability to handle complex non-linear relationships between drug properties and user preferences.

[4] Mrs. Shrilakshmi Prasad et al. proposed a sentiment analysis-based drug recommendation system leveraging machine learning techniques. The study utilized CountVectorizer, TF-IDF, and manual component examination for feature extraction. Classification models, including Naïve Bayes and TF-IDF paired with a Passive-Aggressive classifier, were selected for their simplicity and speed in handling large datasets. TF-IDF paired with the Passive-Aggressive classifier achieved the highest AUC score of 98.3%, highlighting its effectiveness in identifying sentiment polarity for drug recommendations. The dataset consisted of patient reviews, and metrics such as precision, recall, F1-score, and AUC score were used for evaluation. However, the lack of consideration for patient-specific data such as demographics and medical history limited the system's ability to generate personalized recommendations. The system would benefit from integrating more advanced natural language processing (NLP) techniques, such as transformer-based embeddings, to enhance the contextual understanding of reviews.

[5] Punidha et al. proposed a drug recommendation system employing NLP and machine learning. Techniques such as TF-IDF, Word2Vec, Bag of Words (BoW), and manual feature extraction were utilized to process and vectorize textual data efficiently. Models like Aggressive Classifiers and Multinomial Naïve Bayes were chosen for their capability to handle high-dimensional sparse data and make quick predictions. Linear SVC with TF-IDF achieved 95% accuracy, showcasing its suitability for classification tasks in sentiment analysis. Additionally, the study incorporated collaborative, content-based, and hybrid filtering methods to combine sentiment insights with user preferences, ensuring enhanced recommendations. This system required better oversampling techniques and hyperparameter optimization to handle class imbalance. Implementing deep neural networks and transformer-based architectures could further improve prediction accuracy and recommendation quality.

[6] GV Lavanya et al. introduced a Drug Recommender System utilizing machine learning for sentiment analysis, a significant advancement in healthcare technology. The study employed Naïve Bayes, Decision Tree, and Random Forest classifiers to analyze symptoms and predict disease severity (high, average, or low). This innovative approach helps medical professionals

and patients make informed drug-related decisions by extracting and categorizing attitudes in medication evaluations. The sentiment-aware drug recommendation engine prioritized medications with high sentiment ratings and addressed issues in unfavourable reviews, resulting in personalized treatment programs. However, the system lacked real-time processing and adaptive learning mechanisms, which are essential for updating recommendations based on new patient reviews and emerging drug trends.

[7] Satvik Garg proposed a drug recommendation system based on sentiment analysis of drug reviews. The authors employed various machine learning algorithms such as Logistic Regression, Perceptron, Multinomial Naïve Bayes, Ridge Classifier, and Linear SVC for analyzing text vectorized with Bag of Words (BoW) and TF-IDF, while Decision Tree, Random Forest, LGBM, and CatBoost were applied to Word2Vec and manual features. This innovative approach helps medical professionals and patients make informed drug-related decisions by extracting and categorizing attitudes in medication evaluations. However, the system lacked real-time data processing and patient-specific customization, which are crucial in medical emergencies. Further enhancements could include reinforcement learning and real-time feedback loops to refine recommendations dynamically.

[8] P. Chinnasamy et al. proposed a system implemented on Google Collab using a dataset from Kaggle. The content-based filtering technique analyses the features of drugs and recommends medications tailored to individual patient needs. Meanwhile, the collaborative-based filtering method leverages user similarities to recommend medications based on the preferences or behaviours of other users with similar health profiles. This system aims to assist both healthcare professionals and patients by providing timely, accurate drug recommendations, especially in situations where medical expertise may not be immediately accessible, such as in emergencies or resource-limited settings. The system could benefit from incorporating federated learning to ensure data privacy while improving model performance across distributed healthcare databases.

[9] Prajakta Khairnar et al. proposed a disease prediction and medicine recommendation system utilizing machine learning algorithms such as Naïve Bayes, Decision Tree, and Random Forest. The system predicts diseases based on user-entered symptoms and provides corresponding medication recommendations. The Naïve Bayes classifier was chosen for its efficiency in handling high-dimensional data and providing accurate predictions, while the Decision Tree algorithm offered clear, interpretable decision rules. Random Forest, an ensemble method, was

used to improve the model's accuracy by combining multiple decision trees to reduce overfitting and enhance prediction reliability. The system aims to reduce reliance on doctors for minor symptoms and provide accessible medical guidance. However, expanding the model to include multi-modal data, such as electronic health records and genetic information, could improve its predictive capabilities.

[10] Saswati Mahapatra et al. developed a machine learning-based drug recommendation system using content-based and collaborative filtering approaches. The system leveraged cosine similarity in content-based filtering to recommend medications based on their attributes and used collaborative filtering to analyse user-item interactions. This method helps healthcare professionals identify suitable medications for specific conditions by analyzing patterns in patient data, contributing to more accurate and effective healthcare delivery. Despite its benefits, the system did not incorporate advanced machine-learning techniques for handling complex, real-time scenarios. Integrating reinforcement learning and graph-based neural networks could significantly enhance the system's adaptability and recommendation accuracy.

[11] Patel et al. developed a deep learning-based drug recommendation system leveraging transformer architectures, including BERT and Bio BERT, for extracting semantic meanings from drug reviews. Unlike traditional TF-IDF and BoW methods, this approach enabled contextual understanding of user sentiments. The system integrated a hybrid recommendation mechanism combining collaborative filtering with attention-based neural networks, allowing personalized suggestions based on user medical history and demographics. Despite its effectiveness, the model was computationally expensive, requiring GPU acceleration for real-time processing. Future improvements could include federated learning for privacy preservation and decentralized recommendation training across multiple healthcare databases.

[12] Sharma et al. introduced a drug recommendation framework utilizing Graph Neural Networks (GNNs) to capture the relationships between drugs, diseases, and patient history. The system constructed a knowledge graph from biomedical databases, linking symptoms to medications through multi-layered embeddings. This approach significantly enhanced the interpretability and accuracy of drug recommendations compared to traditional collaborative filtering. However, the reliance on structured data limited adaptability to real-world noisy datasets. Incorporating self-supervised learning methods could improve its robustness in handling sparse and incomplete patient records.

[13] Wang et al. proposed a reinforcement learning-based drug recommendation system that dynamically adjusted medication suggestions based on real-time patient feedback. Using Deep Q-Networks (DQN), the model learned optimal drug choices by maximizing patient health outcomes over time. The system was integrated with electronic health records (EHRs), ensuring personalized and adaptive recommendations. However, the model required extensive historical data for effective training, making it challenging to deploy in scenarios with limited patient interactions. Future work could incorporate meta-learning techniques to enhance adaptability with smaller datasets.

[14] Ahmed et al. developed a privacy-preserving drug recommendation system based on hybrid federated learning, a technique aimed at training models locally on hospital data without exposing sensitive patient information, thus maintaining compliance with healthcare regulations like HIPAA. Their solution integrated collaborative filtering with convolutional neural networks (CNNs) to learn deep feature representations from prescription data, improving the accuracy of drug recommendations while preserving privacy. While this approach greatly enhanced patient data security, it required effective communication protocols to handle federated updates in multiple hospitals, a problem in large-scale deployments. The authors proposed that using blockchain-based authentication would further improve security and trust in such decentralized medical AI systems by authenticating updates.

[15] Lee et al. investigated a multi-modal deep learning framework, combining text-based drug reviews, genetic information, and medical imaging to improve drug recommendation accuracy, presenting a holistic vision of patient needs. The framework utilized convolutional and recurrent neural networks (CNN-RNN) to process these diverse patient data sources, making holistic decisions superior to single-modality models for predicting drug effectiveness and side effects. But the intricacy of combining heterogeneous data formats was a major obstacle, needing meticulous preprocessing to match the inputs properly. Making multi-modal medical data processing more standard was suggested to enable better scalability in real-world clinical usage, making the system more readily available. Genetic profiles also introduce possibilities for personalized medicine adapted to the individual's genomic differences, an exciting avenue for precision medicine. A possible disadvantage is the requirement of stable preprocessing pipelines to deal with noisy or missing multimodal data, which would otherwise distort results.

[16] Zhang et al. introduced an adversarial learning-based system for recommending drugs while considering potential drug-drug interactions (DDIs), a critical factor in patient safety.

The model utilized Generative Adversarial Networks (GANs) to simulate synthetic drug interactions, augmenting training data for improved generalization across various scenarios. By incorporating adverse drug reaction databases, the system minimized the risk of prescribing conflicting medications, enhancing its reliability in clinical practice. Despite these advantages, adversarial training required careful hyperparameter tuning to prevent mode collapse, a common issue where the model fails to generate diverse outputs. Extending the system to incorporate real-time pharmacovigilance data was suggested to further enhance patient safety by keeping pace with emerging drug safety insights. The use of synthetic data also reduces dependency on limited real-world DDI datasets, accelerating development and testing phases

[17] Kumar et al. developed a context-aware drug recommendation system using knowledge graphs to establish relationships between drugs, diseases, and patient demographics, leveraging structured medical databases like DrugBank and PubMed. The system utilized entity linking and relation extraction techniques to infer personalized medication recommendations, offering a tailored approach to patient care. This method enhanced explainability by allowing healthcare professionals to trace back the reasoning behind each recommendation, fostering trust and clinical adoption. However, maintaining and updating large-scale knowledge graphs posed challenges, requiring significant resources and necessitating the integration of automated ontology learning techniques for scalability. The knowledge graph framework also facilitates integration with electronic health records for real-time patient context, making it adaptable to dynamic clinical environments.

[18] Huang et al. proposed a drug recommendation system incorporating transfer learning to predict potential side effects before recommending medications, utilizing pre-trained transformer models like BERT and XLNet. These models were fine-tuned on pharmaceutical review datasets to extract nuanced patient sentiments regarding drug efficacy and adverse reactions, improving drug selection accuracy by considering individual risk factors. The system demonstrated success in refining recommendations, though it struggled with generalization across diverse patient populations due to bias in the training datasets, a common hurdle in AI applications. Future research could focus on domain adaptation techniques to mitigate data distribution discrepancies, enhancing its applicability across varied demographics. The use of sentiment analysis also provides a patient-centric perspective, often overlooked in traditional models, enriching the decision-making process. Regular retraining with diverse global datasets could reduce bias and improve cross-population applicability, addressing inclusivity concerns.

[19] Silva et al. resolved the cold-start issue in drug suggestions, where a lack of information on new medications or patients obstructed proper prediction, applying meta-learning to resolve this issue. The system had a base model trained on several drug datasets and adapted rapidly to unknown drugs or new patients with limited data and performed well in suggesting newly approved drugs with scarce historical data by applying a few-shot learning methodology. Yet the model needed carefully chosen meta-training tasks for robust generalization, a pivotal step in its construction. Expansion of the framework through reinforcement learning was suggested to augment adaptability within changing pharmaceutical environments, adapting to fast-paced changes. The speed of adaptation capability renders it particularly suitable for accelerating recommendations during drug launches or epidemics, providing timely solutions in emergency situations. Feedback from clinicians into the meta-learning loop may enrich task selection and enhance performance, bringing the system into alignment with practical necessity.

3. SOFTWARE AND HARDWARE SPECIFICATIONS

3.1 Introduction

This outlines the hardware and software requirements necessary for creating and deploying the Drug Recommendation System. The system is based on machine learning applied to a pre-processed dataset that includes symptoms, medical history, drug data, dosages, and lifestyle suggestions. It must be able to give real-time, precise medicine recommendations via a friendly web interface. The software requirements address both functional and non-functional aspects, such as data preprocessing, model deployment, backend API implementation, frontend UI development, security features, and user feedback incorporation. The hardware requirements define the computing resources required to support efficient processing, scalability, and smooth user experience on different devices. Collectively, these requirements guarantee the reliability, precision, and efficacy of the system in providing medication suggestions to users.

3.2 Specific Requirements

3.2.1 Functional Requirements

1. Data Preprocessing and Balancing:

- Cleans and encodes the dataset using **pandas and NumPy** for structured processing.
- Balances dataset using **SMOTE (Synthetic Minority Oversampling Technique)** to avoid biased predictions.

2. Machine Learning-Based Drug Recommendation:

- Utilizes **Decision Tree, Random Forest, and Support Vector Machine (SVM)** for classification.
- Predicts the most suitable drug recommendation based on symptoms and past medical history.

3. Backend API for Drug Recommendation:

- Processes user input via **Flask API running on http://127.0.0.1:5000**.

- Fetches and returns drug recommendations in **JSON** for frontend display.

4. Frontend User Interface & Interaction:

- Accepts **symptoms** and **medical history** from the user and provides **recommendations** via an intuitive web UI.
- Developed using **HTML, CSS, and JavaScript** for responsiveness and ease of use.

5. User Feedback Collection & Storage:

- Collects user feedback (emojis form- positive, neutral, and negative).
- Stores user responses to understand system performance and user satisfaction.

3.2.2 Non-Functional Requirements

1. **Accuracy & Performance:**

- Ensures high prediction accuracy using Decision Tree, Random Forest, and Support Vector Machine (SVM)..
- Provides real-time responses for efficient drug recommendations.

2. **Scalability & Reliability:**

- Supports concurrent user requests via Flask API.
- Ensures system uptime and stable operation for continuous service.

3. **Security & Data Privacy:**

- Encrypts patient data during processing and storage.
- Ensures safe handling of medical history and personal health information.

4. **Usability & User Experience:**

- Web-based frontend built with HTML, CSS, and JavaScript for accessibility across devices.

- Provides a simple, intuitive UI for symptom input and result display.

5. **Maintainability & Interoperability:**

- Uses modular code structure in Python for easy debugging and updates.
- Works across multiple operating systems (Linux, Windows, macOS, Android, iOS).

6. **Visualization & Feedback Integration:**

- Generates model accuracy charts using Matplotlib and Seaborn.
- Collects user feedback to assess the system's accuracy.

3.3 Hardware and Software Requirements

3.3.1 Hardware Requirements

TABLE 3.3.1 Hardware Requirements

Components	Function	Purpose
Quad-Core CPU (Server)	Handles API requests and ML model execution	Ensures fast real-time processing
8 GB RAM (Server)	Supports multi-user requests	Reduces processing lag
SSD Storage	Stores dataset and logs	Ensures smooth data handling
Smartphones, Tablets, or Laptops (Client-Side)	Allows users to input symptoms and receive recommendations	Provides cross-device accessibility
High-Speed Internet	Facilitates API communication between frontend and backend	Enables real-time data exchange

3.3.2 Software Requirements

TABLE 3.3.2 Software Requirements

Software	Purpose
Operating System (Server): Linux or Windows	Runs the backend API and ML models
Operating System (Client): Android, iOS, Windows, macOS	Provides access to web-based drug recommendation
Flask API	Handles communication between frontend and ML models
HTML, CSS, JavaScript	Builds a responsive user interface
pandas	Processes and cleans input data
scikit-learn	Implements Decision Tree, Random Forest, and SVM models
imblearn	Balances dataset for accurate predictions
Matplotlib & Seaborn	Visualizes model performance and feedback trend
Google Colab or VS Code	Used for model training and code execution

4. PROBLEM STATEMENT

In general medical emergencies, a quick recommendation about drug prescriptions is crucial. However, the selection of an ideal drug is complex due to numerous parameters such as symptoms, the health history of patients, and overall conditions, especially in remote areas or when doctors are unavailable. To address this issue, a real-time Drug Recommendation System helps patients to select suitable drugs based on symptoms, and medical history. Using machine learning, the system can enhance the accuracy of prescriptions, reduce human error, and overall patient outcomes. In addition, patient feedback integration can provide insights into user satisfaction and the system's accuracy. This system aims to bridge the gap in healthcare accessibility by offering timely and individualized drug recommendations, that are personalized and more accurate.

4.1 Objectives

1. Implementing machine learning for accurate Medical Recommendations:

Machine learning will be applied to a pre-processed dataset containing information on symptoms, medical history, drugs, dosage, and lifestyle advice based on user inputs the system will analyze the data and recommend the most suitable medications, ensuring accuracy.

2. Developing a User-Friendly Web-Based Interface for Real-Time Recommendations:

A web-based platform will be developed to provide an intuitive and accessible interface for users. Users will enter their symptoms and medical history and the system will instantly suggest appropriate medications based on the most accurate ML- model. Additionally, feedback will be integrated to assess recommendation accuracy.

5. DESIGNING

5.1 System Architecture

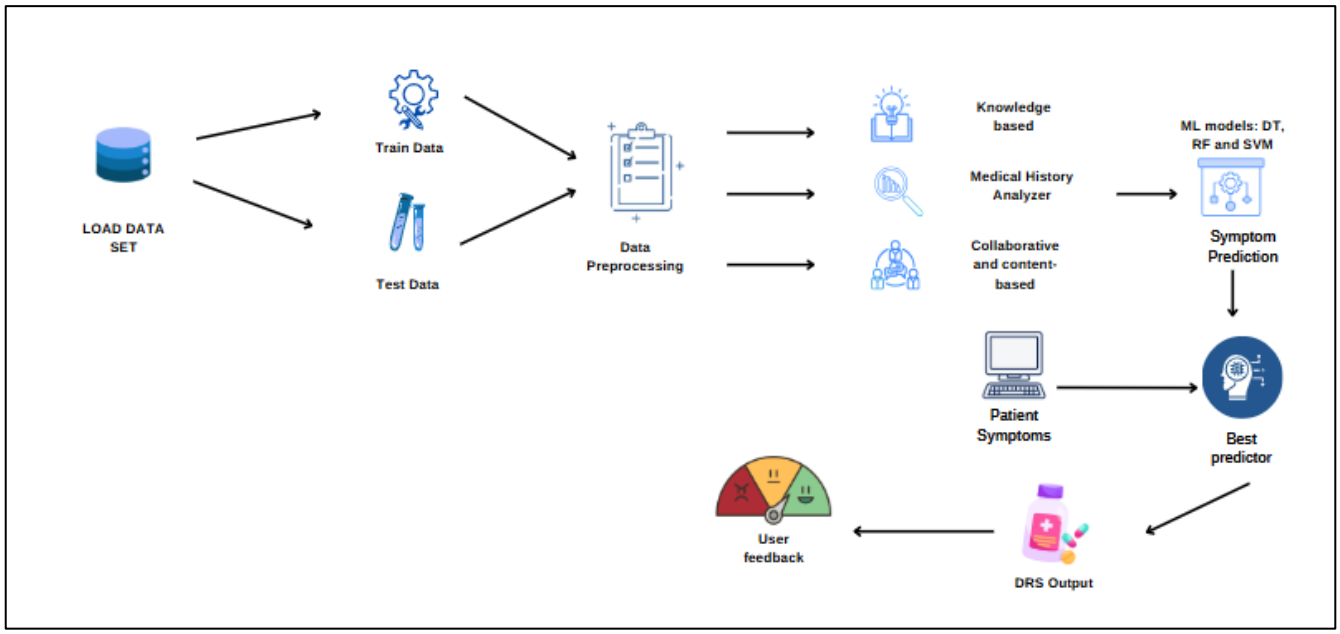

Fig. 5.1.1 Block diagram for Drug Recommendation System

Fig. 5.1.1 Represents The Drug Recommendation System (DRS) which is a modular system for making accurate drug recommendations from patient symptoms, and medical history. It loads a dataset that undergoes 80% training and 20% testing then the data is pre-processed by data cleaning and handling missing value imputation using Imputer and SMOTE, these yield balanced and unbiased predictions. Symptom Prediction Engine performs more processing on the user input and maps it to potential medical conditions using machine learning models such as Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM) and selecting the optimal prediction model.

Once symptom analysis is performed, the Medical History Analyzer checks the available medical history to ensure that prescribed drugs are safe. With this, the Drug Recommendation System provides accurate drug suggestions with dosage and lifestyle advice. There is a web interface by which the user provides symptoms and receives suggestions. User Inputs are processed and responses are sent back as drug suggestions to be rendered in real-time. It also possesses a feedback mechanism in the form of emojis to assess the system's accuracy.

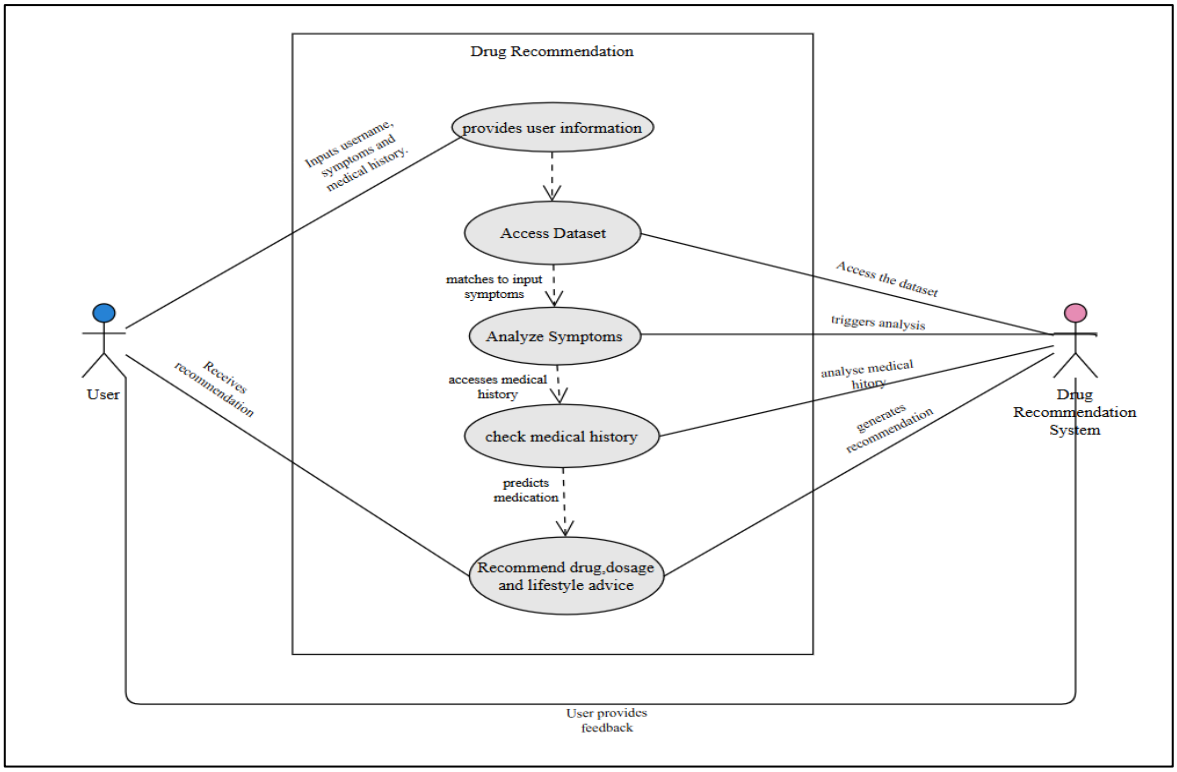

Fig. 5.1.2 Use Case Diagram of Drug Recommendation System.

Fig. 5.1.2 shows the user interaction with the Drug Recommendation System. It also emphasizes major functionality like symptom input, access to dataset, symptom analysis, medical history validation, and drug recommendation. The system is designed to ensure that the recommended drug is safe depending on the medical history of the patient.

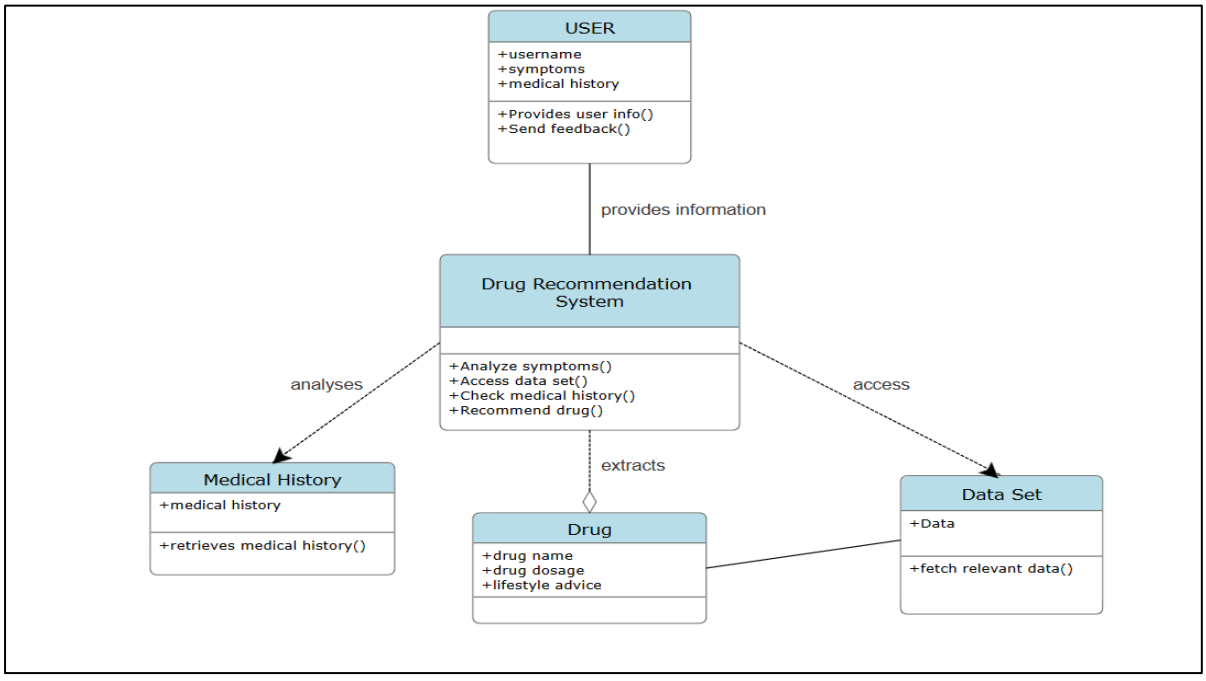

Fig. 5.1.3 Class Diagram of Drug Recommendation System

Fig. 5.1.3 The class diagram illustrates the structural relationships between various components of the system, namely the user, drug recommendation system, medical history, dataset, and drug details. It establishes attributes and functions for each of the entities, for example, analyzing symptoms, accessing medical history, and suggesting drugs.

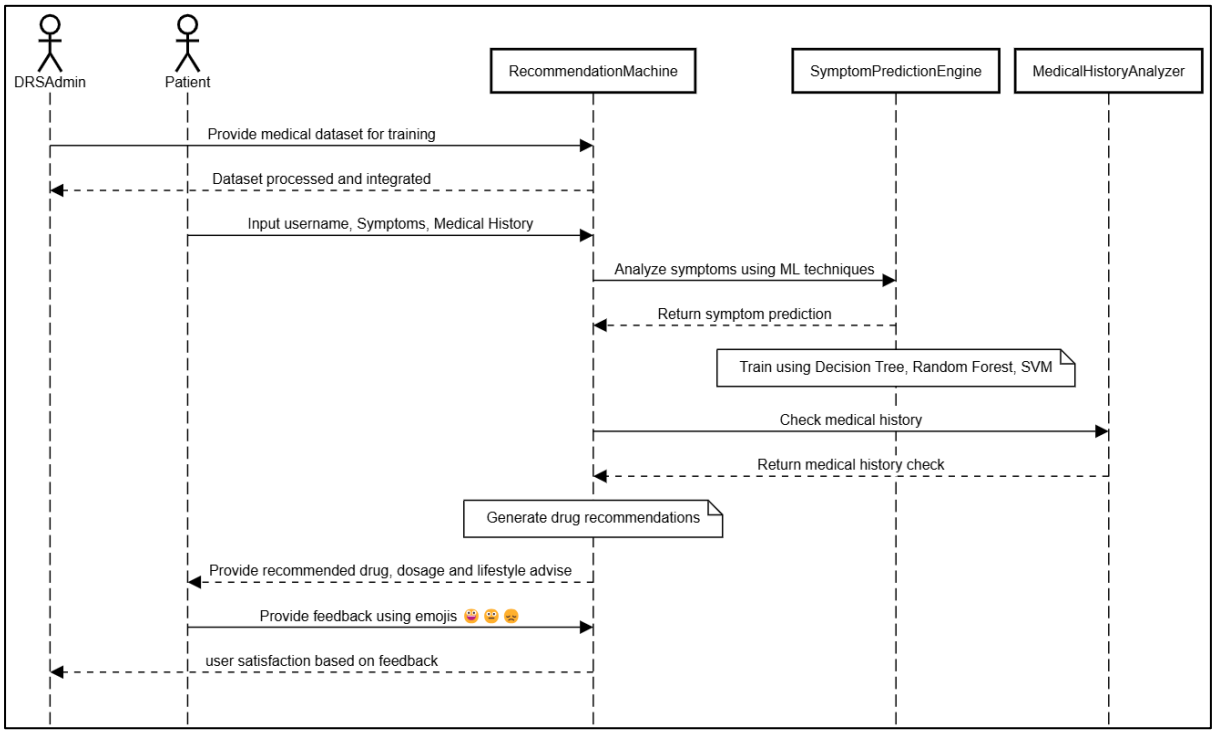

Fig. 5.1.4 Sequence diagram of Drug Recommendation System

Fig. 5.1.4 shows the step-by-step interaction flow among system entities, such as the user, system admin, symptom prediction engine, medical history analyzer, and recommendation machine. The diagram explains dataset preprocessing, machine learning-based symptom analysis, medical history authentication, and drug suggestion. The diagram also includes a feedback mechanism in which users submit satisfaction scores to understand the system's performance.

5.2 Methodology

The methodology involves utilizing three machine learning models— Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT)—for model training.

5.2.1 Support Vector Machine

In the Drug Recommendation System, SVM is utilized to classify symptoms and forecast possible medical conditions. By projecting symptom data into a high-dimensional space, SVM detects the best hyperplane that can distinguish various classes of diseases. This provides accurate symptom-based condition identification, enhancing drug recommendation accuracy. SVM excels in processing intricate relationships between symptoms and diseases through kernel functions, making it a trustworthy option for medical diagnosis.

5.2.2 Random Forest

Random Forest is used in the system to support more accurate and stable disease classification. RF works by building ensembles of Decision Trees, training each tree over random subsets of the dataset. The final outcome is arrived at through a majority vote, decreasing overfitting prospects and increasing generality. RF is also uniquely useful in our project as RF can manage noisy or missing clinical data well and make symptom-disease mapping trustworthy for recommending medications.

5.2.3 Decision Tree

The decision Tree model in the system is used to create a transparent, interpretable mapping between symptoms and possible diseases. It is a hierarchical model, where every node is a symptom-based decision rule resulting in a predicted condition. While DT gives a simple-to-understand classification, it can be prone to overfitting. To avoid this, we either prune the tree or combine it with ensemble methods such as Random Forest. DT is an anchor model within our system that makes early disease classification and also assists in individualized drug recommendations.

6. IMPLEMENTATION

Dataset overview

The original dataset contained 50,027 rows by 8 columns in CSV form. There were some columns named Timestamp, contradictions, and severity that were not needed for this project, and they were dropped to enhance accuracy, the symptom column which contained two drugs was split into two columns symptom 1 and symptom 2. Removing these unwanted columns and cleaning the dataset made the model efficient. Post Preprocessing: The cleaned dataset now has 40,001 rows and 6 columns: Symptom 1, Symptom 2, Medical History, Drug Name, Dosage, and Lifestyle Advice.

6.1 Data Preprocessing

Packages/Libraries: pandas, sklearn.preprocessing, imblearn

Classes: Pandas, LabelEncoder, SimpleImputer

Methods: pd.read_csv(), str.split, drop(), LabelEncoder, fit_transform()

The data was initially imported using the pandas library with appropriate encoding and removal of corrupted rows. To improve model precision, unwanted columns like "Contraindications," "Severity," and "Timestamp" were eliminated. The "Symptoms" column, having more than one value, was also divided into "Symptom 1" and "Symptom 2" for proper structuring. Missing values in the important columns such as "Symptoms," "Medical History," "Dosage," "Lifestyle Advice," and "Recommended Drugs" were managed with SimpleImputer, which filled them with the most common values to maintain data completeness.

To make the dataset machine learning-ready, the categorical values were transformed into numeric form using LabelEncoder from sklearn.preprocessing. With this conversion, the model could efficiently handle categorical data. Last but not least, the top rows of the preprocessed and cleaned dataset were printed to check the correctness of these conversions.

6.2 Data Splitting

Packages/Libraries: sklearn.model_selection

Classes: train_test_split

Methods: train_test_split(X, y, test_size=0.2, random_state=42)

To effectively train and test the drug recommendation model, the dataset was split into training and testing sets using the train_test_split function from the sklearn.model_selection package. This ensures that the model learns from one portion of the data while being tested on unseen data to assess its performance. The dataset was divided into 80% training data and 20% testing data using the following method Here, X represents the feature variables, and y represents the target variable. The test_size=0.2 parameter specifies that 20% of the data is reserved for testing, while 80% is used for training. The random_state=42 ensures reproducibility, meaning the split remains consistent across multiple runs.

6.3 Data Balancing

Packages/Libraries: imblearn.over_sampling, imblearn.under_sampling

Classes: SMOTE (Synthetic Minority Oversampling Technique), RandomUnderSampler

Methods: fit_resample(X_train, y_train)

Imbalanced classes in a dataset can produce biased predictions such that the majority class is predicted more frequently, and the minority class less often. In an attempt to alleviate this imbalance, we employed oversampling and undersampling with the imblearn library.The Synthetic Minority Oversampling Technique (SMOTE) from imblearn.over_sampling was utilized to boost the samples in the minority class by creating synthetic data points. In contrast, RandomUnderSampler from imblearn.under_sampling was utilized to decrease the samples in the majority class, resulting in a balanced dataset.These methods were applied using the.fit_resample(X_train, y_train) method, which resampled the training data, balancing both classes prior to training the model.

6.4 Model Training

Packages/Libraries: sklearn.svm, sklearn.tree, sklearn.ensemble

Classes: DecisionTreeClassifier, RandomForestClassifier, SVC (Support Vector Classifier)

Methods:

•fit(X_train, y_train) (training models)

•.predict(X_test) (making predictions)

To develop an efficient drug recommendation model, three machine learning models were employed: Decision Tree Classifier, Random Forest Classifier, and Support Vector Classifier (SVC). The models were built using the sklearn.svm, sklearn.tree, and sklearn.ensemble libraries.The training phase comprised the usage of the .fit(X_train, y_train) function, where the models learned patterns from the training data. After training, predictions were made on the test set using the .predict(X_test) function.

- DecisionTreeClassifier: A low-level yet influential algorithm that forms a tree-based model to categorize data on the basis of feature values.
- RandomForestClassifier: A type of ensemble learning algorithm that constructs a series of decision trees and averages their results for better accuracy and stability.
- SVC (Support Vector Classifier): A robust algorithm that determines the best hyperplane to separate data points, which is helpful for intricate decision boundaries.

6.5 Model Evaluation

Packages/Libraries: sklearn.metrics

Classes: classification_report, accuracy_score, confusion_matrix

Methods:

•accuracy_score(y_test, y_pred)

•classification_report(y_test, y_pred)

•confusion_matrix(y_test, y_pred)

To compare the performance of the trained models, different metrics from the sklearn.metrics library were utilized to assess accuracy, precision, recall, and overall classification performance. The accuracy_score(y_test, y_pred) function was used to calculate the accuracy of the model by comparing the predicted labels with the actual labels. Furthermore, classification_report(y_test, y_pred) gave a complete performance breakdown, including precision, recall, and F1-score for all classes, so detailed evaluation of the model's

predictability was possible. To evaluate classification performance further, a confusion matrix was plotted using confusion_matrix(y_test, y_pred), which represented correct and incorrect predictions for all classes.

6.6 Model Visualization

Packages/Libraries: matplotlib.pyplot, seaborn

Methods:

•plt.bar() (bar charts for accuracy comparison)

•sns.heatmap() (confusion matrix visualization)

•plt.show() (displaying plots)

In order to get a better idea of the performance of various models, visualization methods were employed using the matplotlib.pyplot and seaborn libraries. A bar chart was plotted by plt.bar() to visually compare the accuracy of various models and provide an understandable representation of their performance. Also, a confusion matrix was depicted using sns.heatmap(), which showed correct and incorrect predictions in a simple-to-understand manner. Lastly, plt.show() was utilized to render the plots produced. These visualizations assisted in model efficiency analysis, pattern recognition, and determination of the best algorithm for suggesting drugs.

6.7 Backend Implementation

Packages/Libraries: flask, pandas, sklearn, flask_cors

Classes: Flask , CORS (to enable cross-origin requests from the frontend)

Methods:@app.route("/get_recommendation", methods=["POST"]) , jsonify(), pd.read_csv(), run(debug=True)

Drug Recommendation System is coded with Flask, a minimal web framework in Python. It allows the frontend to communicate with the backend to facilitate users' inputs of symptoms and get back the recommended drugs on the basis of a predefined medical dataset.

The system comprises a processing module that takes user input, searches a dataset (medicaldatasetotc.xlsx), and brings back related drug details, such as name, dosage, and precautions. The dataset is organized to facilitate efficient searching and retrieval of data. When the user inputs symptoms and medical history, the backend sends the request via a Flask API, which matches the input against the dataset and brings back the suitable drug recommendation. To ensure data accuracy, preprocessing methods like missing value handling and input format standardization are used. The Flask API handles HTTP requests and sends responses in JSON format for seamless integration with the frontend. Error-handling features are incorporated to handle invalid inputs, missing records, and system errors. Logging and debugging facilities assist in monitoring API requests and system performance for maintenance and optimization.

6.8 Frontend implementation

Packages/Libraries Used: HTML, CSS, JavaScript – Core technologies for building the frontend. AJAX / Fetch API – For making asynchronous requests to the backend.

Classes: DOM elements such as input fields, buttons, and result containers.

Methods:

- fetch() – Retrieves data asynchronously from the backend API.
- localStorage.setItem() – Stores user preferences and previous inputs locally.
- querySelectorAll() – Selects multiple elements in the DOM for manipulation.
- addEventListener() – Handles user interactions like button clicks and form submissions.

The Drug Recommendation System is created with an interactive frontend that enables users to enter symptoms and get drug recommendations without any hassle. The frontend is developed with HTML, CSS, and JavaScript, making it responsive and interactive across devices. It has a well-formatted form for users to enter their symptoms, conditions, and other supporting information. To provide an uninterrupted experience, the frontend interacts with the Flask backend via the Fetch API, enabling asynchronous sending and receiving of data without a complete page reload. This gives the system greater responsiveness and efficiency. The

interface is kept simple and intuitive, easing users into the input process by offering instructions and validation checks to avoid incorrect submission.

After the backend has processed the request, the frontend is dynamically updated to show suggested drugs with essential information, including dosage, usage directions, and potential side effects. JavaScript improves the user interface through making it visually engaging and enabling smooth interactions. Error handling capabilities are also incorporated into the system, offering messages and alerts when there is a wrong input detected or when there is no recommendation available.

The frontend is designed to function effectively with the backend API to provide a quick and efficient user experience. Its responsive design makes it suitable for various screen sizes, hence easily accessible for users on different devices. This ensures that it provides optimal performance, usability, and an interactive experience in accessing vital medical information.

7. TESTING

The Drug Recommendation System is tested through Postman, using Black Box Testing methods to analyze API functionality, response correctness, and error management. This testing is done to validate whether the system gives proper drug recommendations based on user input like symptoms, medical history, and username. The tests are centered on checking for correct responses, managing incorrect inputs, and API robustness.

7.1 Test Case 1: Successful Execution

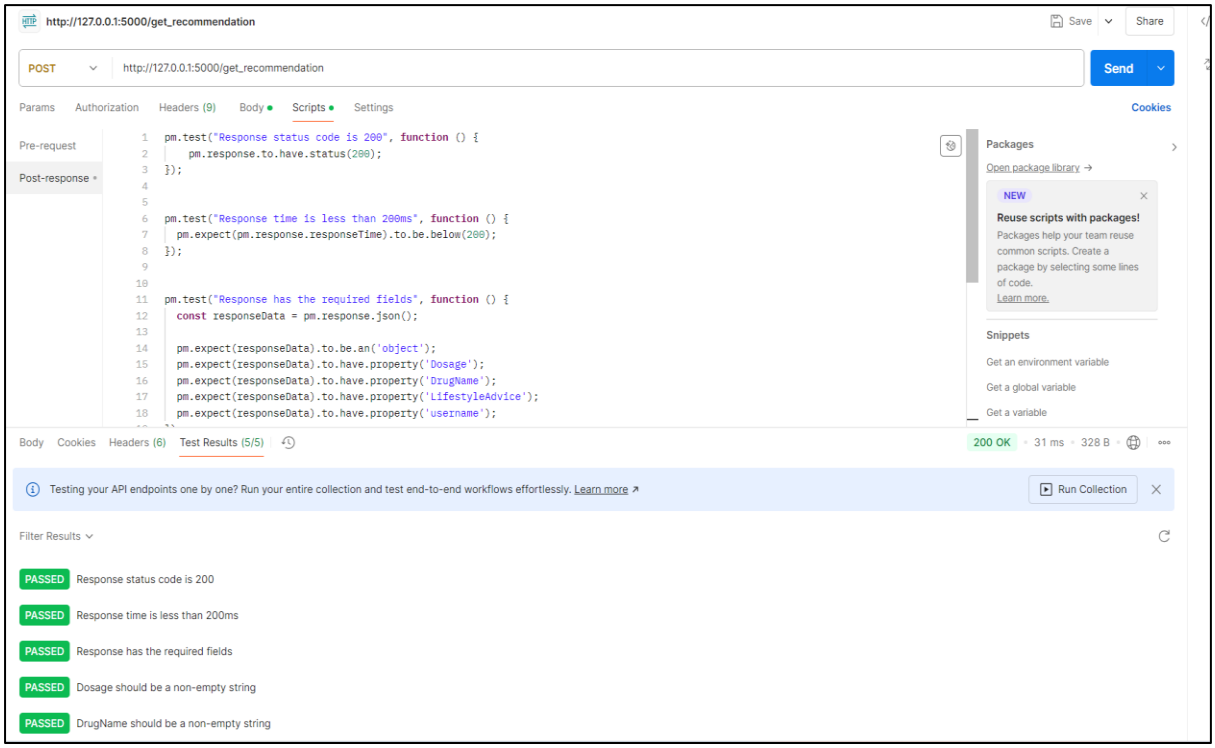

Fig. 7.1 Successful Execution postman Test case1

- **Description:** The API successfully executes and validates responses through Postman tests.
- **Tests Conducted:**

  Response status code is 200 OK.

  Response time is less than 200ms.

  Required response fields are present: Dosage, DrugName, LifestyleAdvice, and Username

- **Result:** All tests passed, confirming correct API functionality.

7.2 Test Case 2: Valid Input

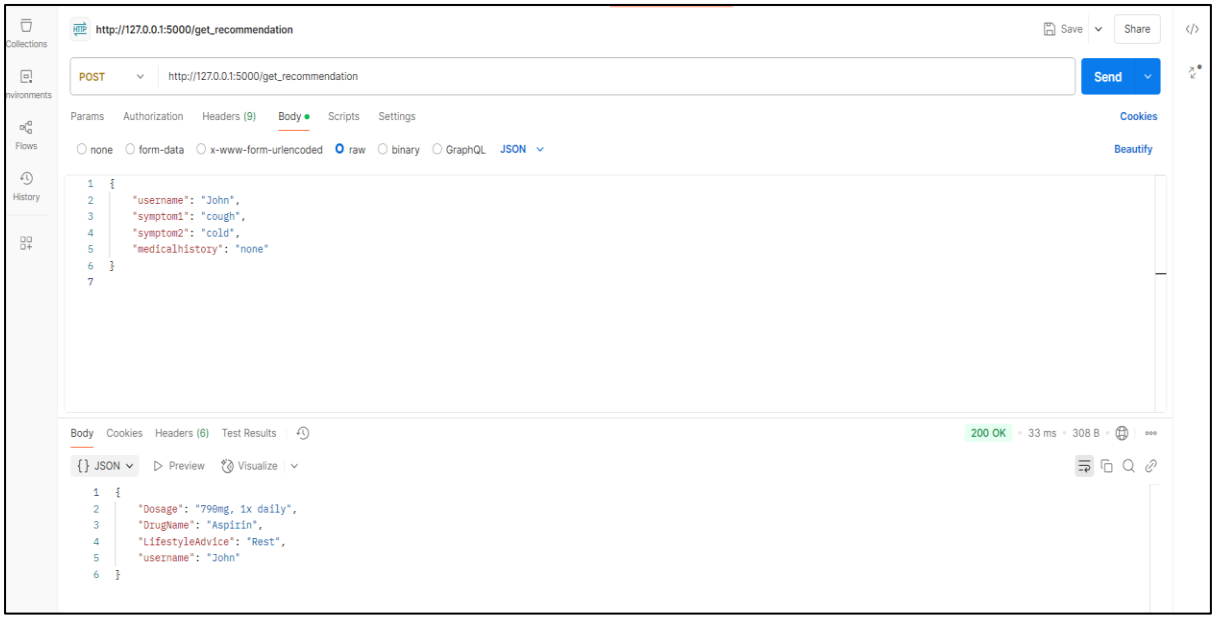

Fig. 7.2 Valid Inputs Postman Test case 2

- **Input:**

  Username: John

  Symptoms: Cough, Cold

  Medical History: None

- **Response:**

  Drug Name: Aspirin

  Dosage: 70mg, 1x daily

  Lifestyle Advice: Rest

  Status Code: 200 OK

- **Result:** API successfully generated a drug recommendation based on valid input.

7.3 Test Case 3: Invalid Input – Missing Symptom

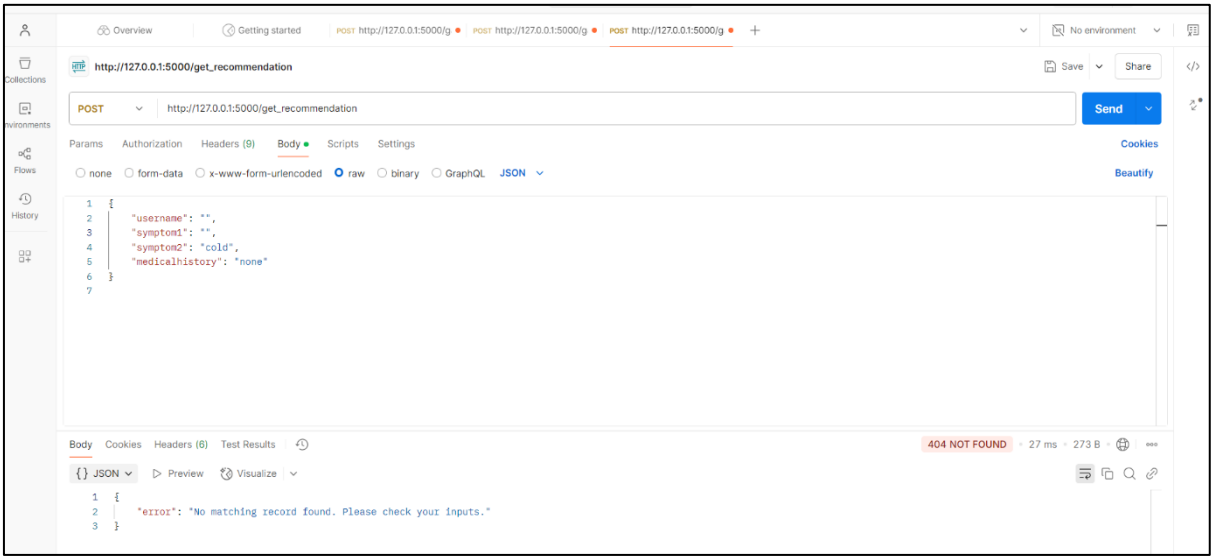

Fig. 7.3 Invalid Input(missing symptom) postman Test case 3

- **Input:**

  Username: (Empty)

  Symptoms: Cold, Fever

  Medical History: None

- **Response:**

  Error Message: "No matching record found. Please check your inputs."

  Status Code**:** 404 Not Found

- **Result:** API correctly rejected an incomplete request due to a missing username.

7.4 Test Case 4: Valid Input Without Username

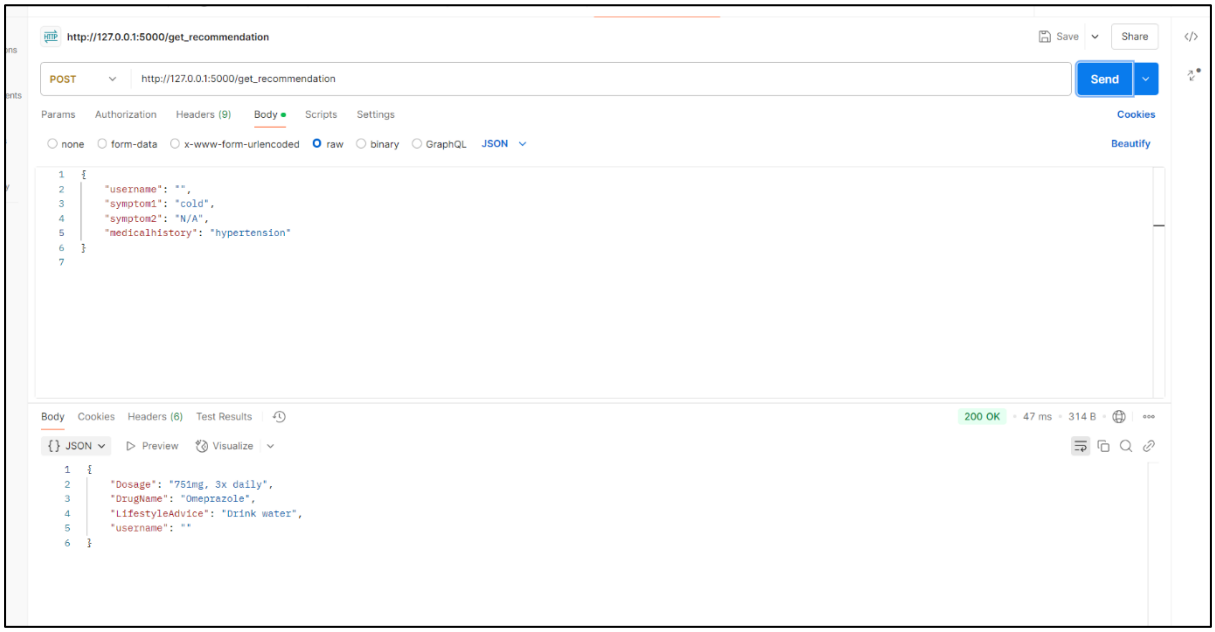

Fig. 7.4 Valid Input (without Username) Postman test case 4

- **Input:**

  Username: (Empty)

  Symptoms: Cold

  Medical History: Hypertension

- **Response:**

  Drug Name: Omeprazole

  Dosage: 75mg, 3x daily

  Lifestyle Advice: Drink water

  Status Code: 200 OK

- **Outcome:** API provided a recommendation despite a missing username, indicating it is not a mandatory field.

7.5 Test Case 5: Invalid Input – void in medical history

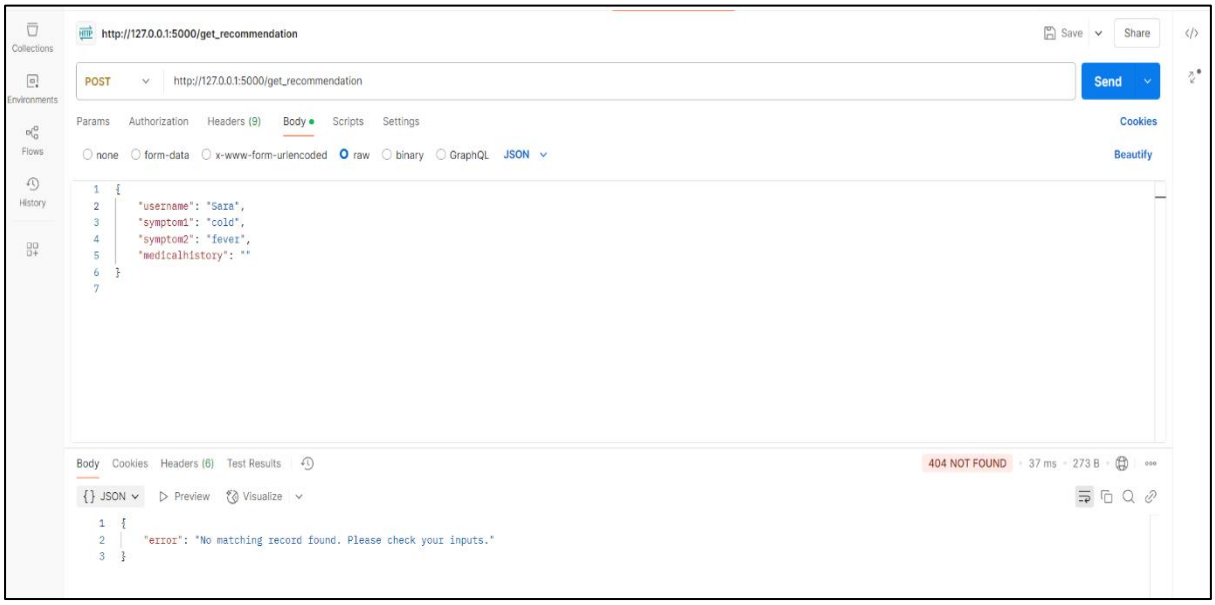

Fig. 7.5 Invalid input (missing medical history) postman test case 5

- **Input:**

  Username: Sara

  Symptoms: Cold, Fever

  Medical History: (Empty)

- **Response:**

- Error Message: "No matching record found. Please check your inputs."

  Status Code: 404 Not Found

- **Outcome:** API correctly identified an unrecognized user and returned an appropriate error message.

8. EXPERIMENTAL RESULTS

8.1 Confusion Matrix

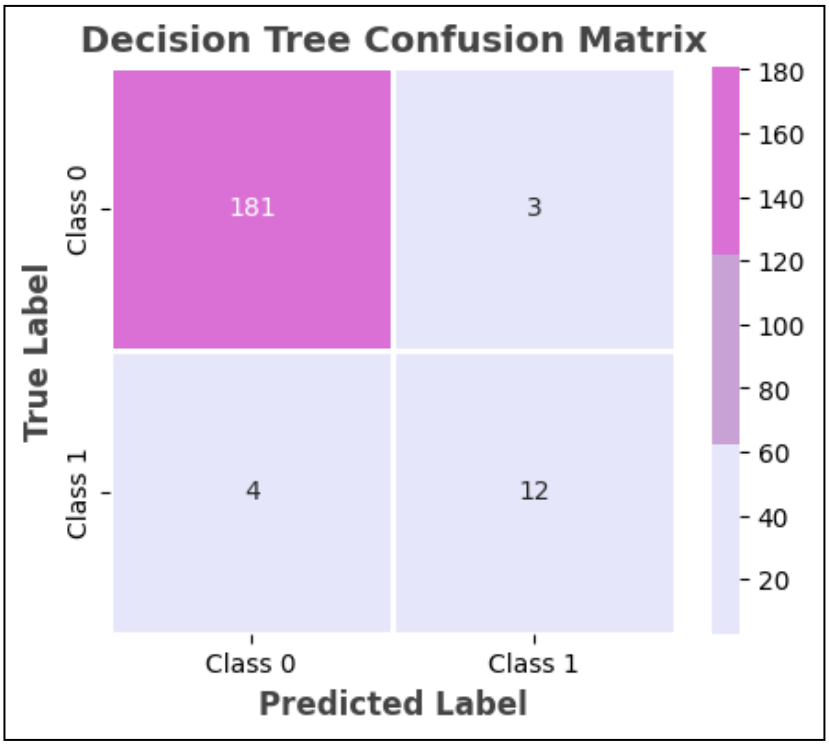

Fig. 8.1.1 Decision tree confusion matrix

Fig. 8.1.1 is the Decision Tree Confusion Matrix, displaying how well the model performed in predicting the provided dataset. The matrix displays that the model successfully predicted 181 cases of Class 0 drugs, which are ineffective according to medical history and symptoms, and 12 cases of Class 1 drugs, which are effective for treatment. However, there were 3 false positives in which Class 0 drugs were predicted as Class 1, or the model inaccurately labelled an ineffective drug as effective. 4 false negatives also happened where Class 1 drugs were predicted as Class 0, indicating that some effective drugs were labelled as ineffective by mistake. While the Decision Tree model is good at classifying Class 0 drugs, it fails to correctly classify Class 1, which can have implications for treatment advice.

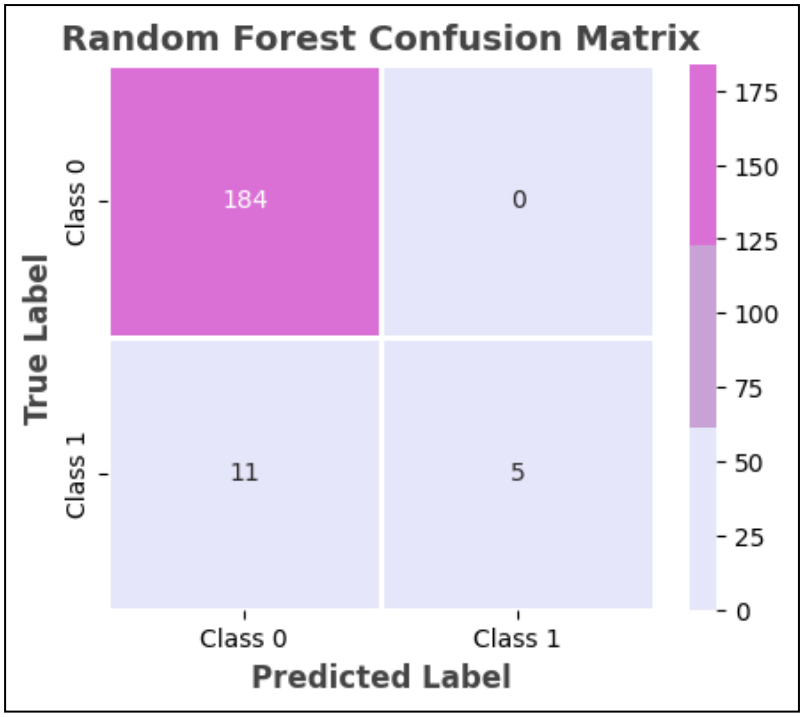

Fig. 8.1.2 Random Forest Confusion Matrix

Fig. 8.1.2 displays the Random Forest Confusion Matrix, which has better classification performance. The model accurately labelled all 184 examples of Class 0 drugs as ineffective with zero false positives, proving its high precision to label drugs that do not contribute positively to treatment. Nonetheless, the model exhibited poor performance in identifying Class 1 drugs, classifying merely 5 out of them properly while misclassifying 11 as Class 0. What this implies is that a number of effective drugs were classified as ineffective, hence potentially resulting in lost treatment options. Although performing well overall, the Random Forest model is particularly poor at detecting effective drugs via sensitivity.

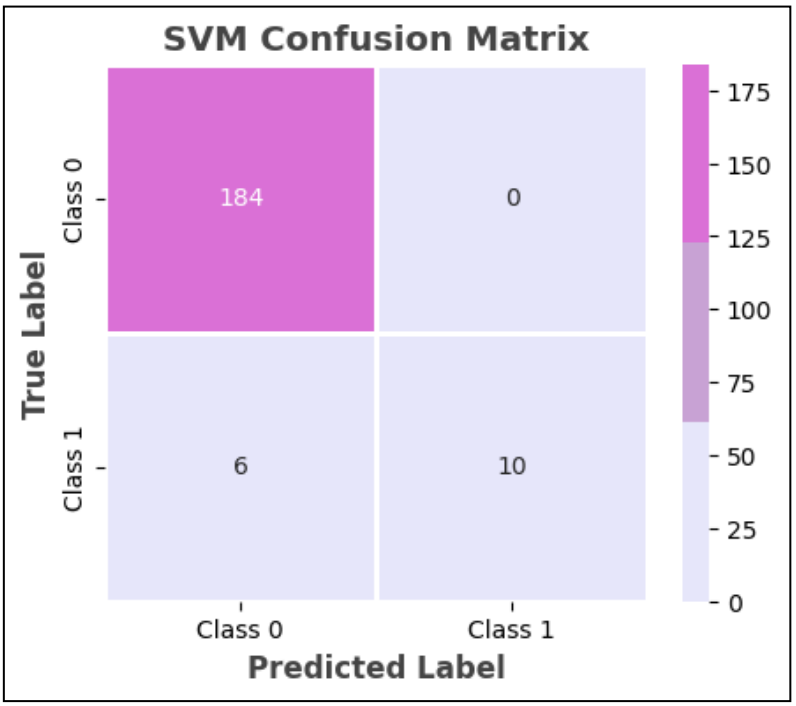

Fig. 8.1.3 Support Vector Machine Confusion Matrix

Fig. 8.1.3 shows the SVM (Support Vector Machine) Confusion Matrix, which maintains a similar pattern of classification as the Random Forest model. It accurately classified all 184 instances of Class 0 drugs as non-effective, which guaranteed that those drugs that were not beneficial would not be suggested. But it misclassified 6 Class 1 drugs as Class 0 and correctly classified only 10. While the SVM model is very accurate in eliminating ineffective drugs, it still has difficulty in classifying effective drugs correctly, possibly constraining the scope of useful treatments recommended.

In general, all three models adequately classify non-helpful drugs (Class 0), but differences in their respective accuracy levels to identify helpful drugs (Class 1) identify the requirement of optimization. The Decision Tree model demonstrates fair performance, the Random Forest model performs well in terms of precision but is insensitive, and the SVM model is capable of good classification of non-effective drugs but poor identification of effective treatments.

8.2 Model Comparison

TABLE 8.2 Model Comparison

MODEL	ACCURACY	PRECISION	RECALL
DECISION TREE	96.0	98	98
RANDOM FOREST	97.0	97	100
SVM	95.0	95	100

Table 3 illustrates an in-depth comparison of Decision Tree, Random Forest, and SVM in terms of accuracy, precision, and recall. The Decision Tree is 96.0% accurate with 98% precision and recall, reflecting robust classification but possible overfitting tendency. Random Forest performs slightly better at 97.0% accuracy, still having a balanced 97% precision and perfect 100% recall, which makes it the most dependable model. SVM, although performing the best on recall with 100%, has the worst accuracy (95.0%) and precision (95%), which means it could misclassify some conditions. These findings reaffirm that Random Forest has the best compromise among all the measures and is therefore the best-performing model for precise and effective drug recommendations.

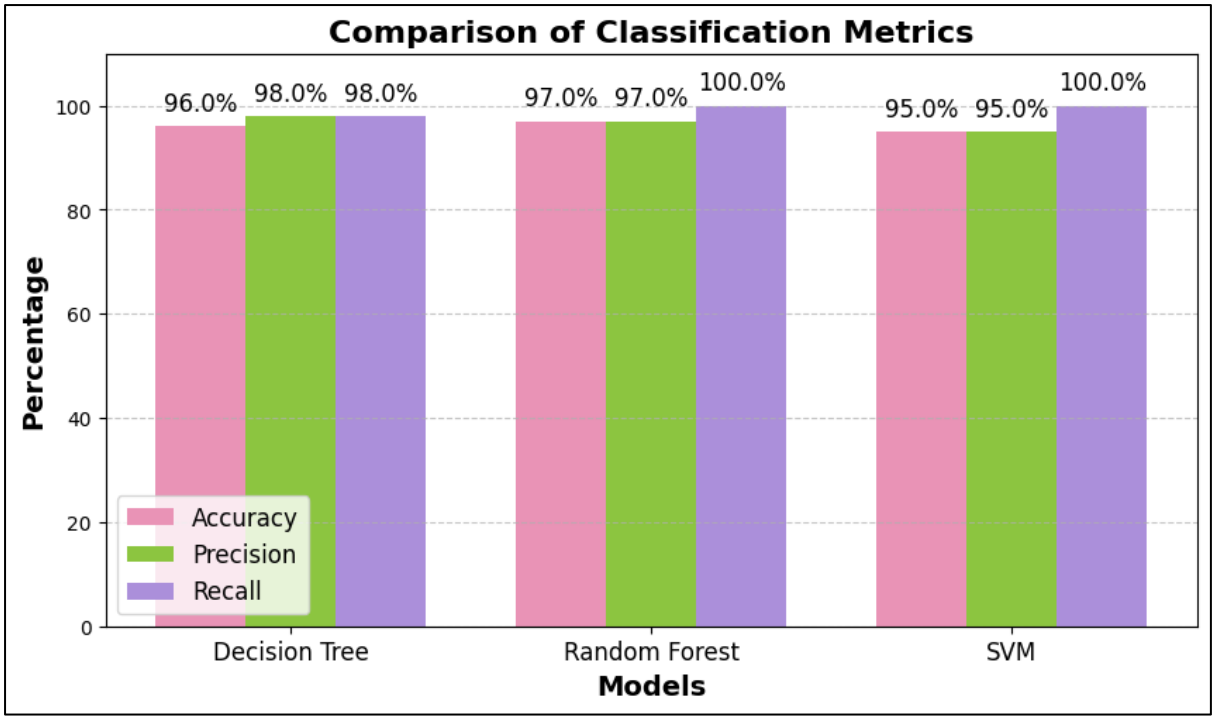

Fig. 8.2 Comparison of classification Matrices

Fig. 8.2 shows a comparative plot of three machine learning models' classification performance—Decision Tree, Random Forest, and Support Vector Machine (SVM). The plot indicates differences in accuracy, precision, and recall between these models, reflecting their strengths and weaknesses. Decision Tree shows good classification ability but is susceptible to overfitting. Random Forest has a well-balanced and stable performance with both reliability and few false negatives. SVM, though superior in recall, has slightly lower precision and accuracy than the other models. This is a visual contrast that highlights the need to pick the most ideal model, where Random Forest proves to be the best option in terms of obtaining both accuracy and consistency in drug recommendations.

8.3 Random Forest- Most accurate model

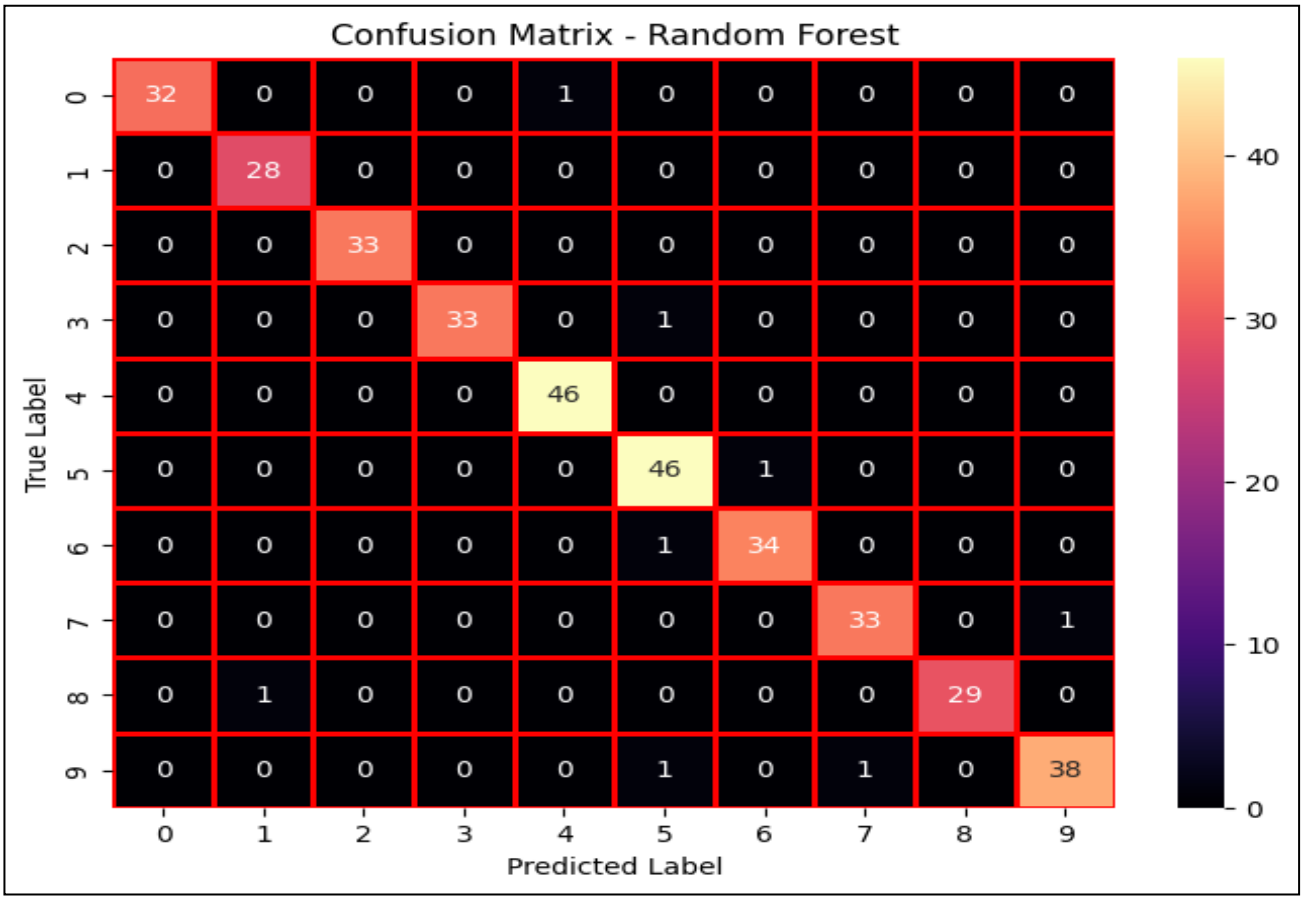

Fig. 8.3 Random Forest 10*10 confusion Matrix

Fig. 8.3 displays the 10×10 confusion matrix for the Random Forest model, which was the most accurate of the three models that were tried. The x-axis shows the predicted drug classes and the y-axis the true classifications. The numbers in each cell represent the number of cases for which a particular drug class was predicted against its true value. The matrix strongly shows that most of the predictions fall along the diagonal, representing a high number of correctly classified samples. The lighter intensity along the diagonal strengthens excellent classification accuracy over all 10 classes, representing varying drug categories based on symptoms and medical history. The off-diagonal minimum misclassifications reflect the resilience of the model since the mistakes are scattered thinly across various classes. Random Forest's robust performance is attributed to its ensemble learning method, where it aggregates several decision trees to enhance accuracy and reduce overfitting. With high precision and recall, the model efficiently separates drug categories based on symptoms and medical histories. Its low error rate for all classes makes it the highest-performing model for the drug recommendation system.

TABLE 8.3 Random Forest class distribution

Class	Description
0	Patients without a medical history but exhibiting one or more symptoms.
1	Patients have a medical history but no current symptoms.
2	Patients with no prior medical history report a single symptom.
3	Patients without a medical history are exhibiting two symptoms.
4	Patients with heart disease who have at least one symptom.
5	Patients with hypertension who exhibit at least one symptom.
6	Patients with kidney disease who show one or more symptoms.
7	Patients with asthma and who also have one or two symptoms.
8	Patients with diabetes and at least one symptom.
9	Patients with a combination of two symptoms and medical conditions need specialized drug recommendations.

8.4 Drug Recommendation System

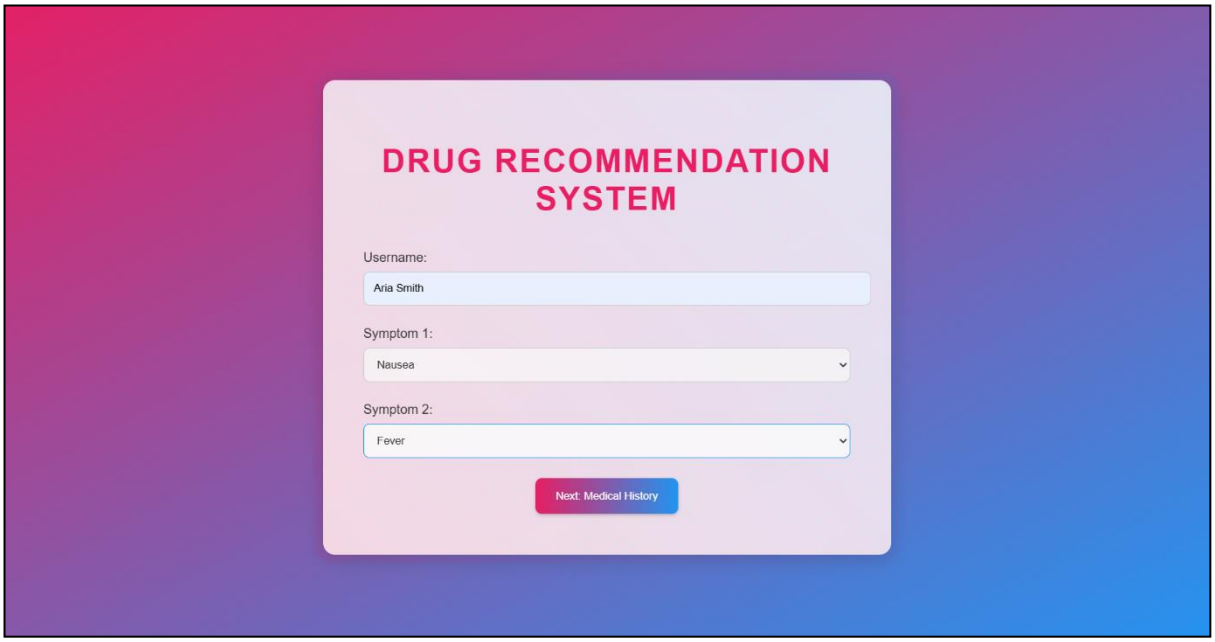

Fig. 8.4.1 Drug Recommendation System Inputs (Symptoms)

The input screen is shown in Fig. 8.4.1 where the drug recommendation process begins. In this, the user is asked to provide their name and choose appropriate symptoms from dropdown menus, making the input process simple and organized. This is a very important step since it enables the system to gather vital health information, which will be used to identify possible medical conditions. After the symptoms are entered, the system proceeds to the second step, where it assesses the medical history of the user. This further step increases the precision of the recommendation by taking into account existing conditions so that the recommended drug is not only effective but also safe for the user. The simplified interface allows users to enter their information easily without compromising on accuracy and efficiency in recommending the drug.

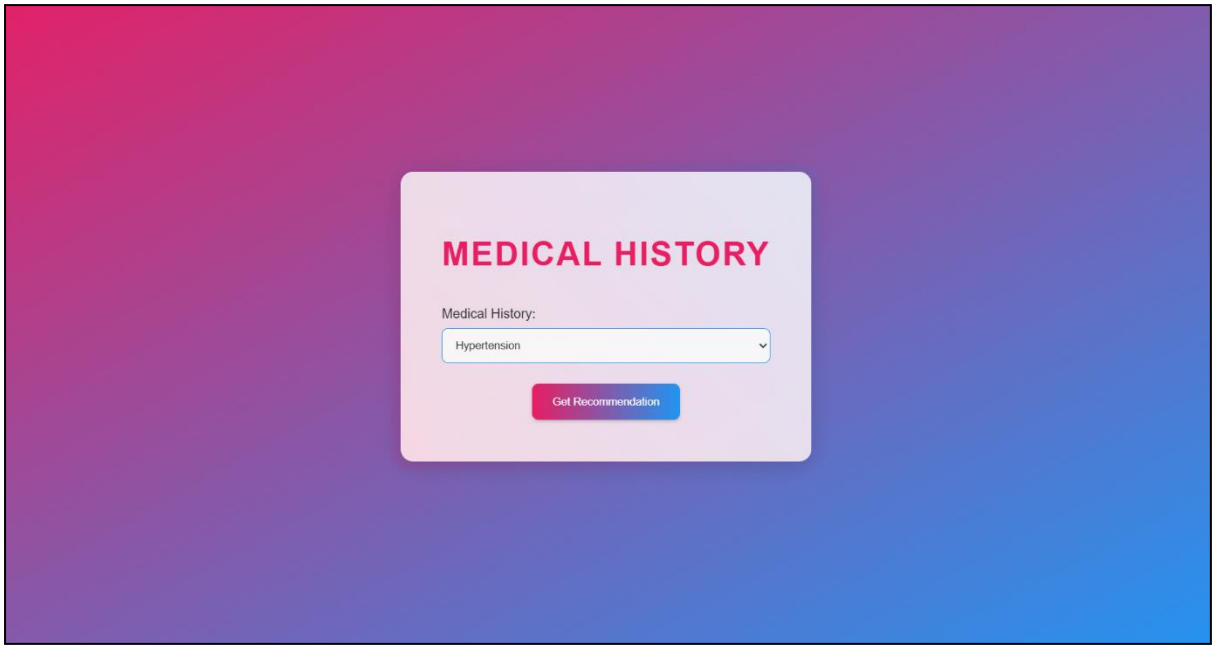

Fig. 8.4.2 Drug Recommendation System Input (Medical History)

In Fig. 8.4.2 the user chooses their existing medical condition from a dropdown menu so that the system can consider any pre-existing health conditions that might influence the appropriateness of a specific drug. Through this added information, the system increases the precision and safety of the prescription so that the suggested drug does not interfere with the user's medical history. The interface is user-friendly and provides a smooth experience without compromising the accuracy of the data collection. Once the medical history is entered, the user can click on the button to get the final recommendation. This ensures that everything is taken into account before it suggests an appropriate medicine, making the system more reliable and effective.

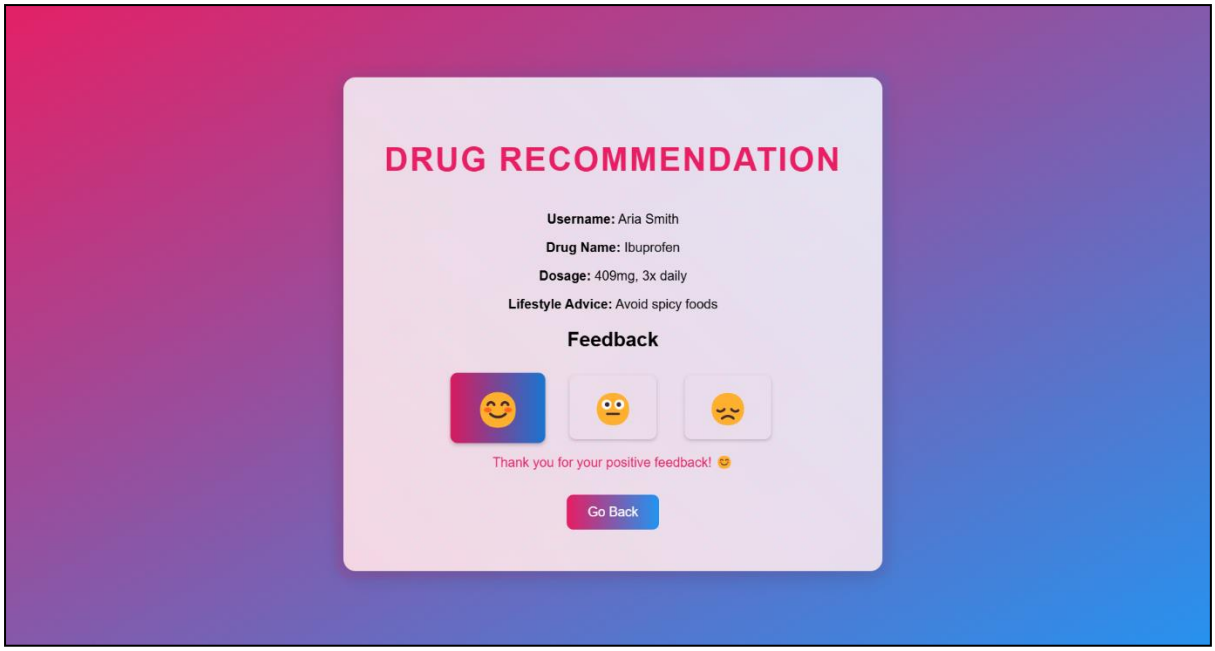

Fig. 8.4.3 Drug Recommendation System Outputs

Fig. 8.4.3 Showcases the Drug Recommendation System's last output screen provides the suggested medication per the user's entered symptoms and health record. This interface is tailored to provide fundamental drug-related information in an easily readable and well-structured way. It provides the username of the patient, the name of the prescribed medicine, specific dosage instructions, and associated lifestyle guidelines for safe and proper use.

To improve the user interaction and system assessment, a feedback component is added, where users can reply using emoji-based reactions, giving a straightforward and fast option to post their experience. After providing feedback, a confirm message is shown, accepting the feedback from the user. This feedback feature assists in measuring the accuracy of the system and satisfaction of the user

9. CONCLUSION

Drug Recommendation System is an innovative solution to the life-critical issue of timely and correct selection of drugs, especially in a situation of emergency. The quick decision under the medical emergency state takes the highest priority, and the ready availability of proper medication can change the situation for the patient. With the strength of data science and machine learning, the platform will be designed to give the patient timely, precise, and personalized recommendations for drugs based on the patient's symptoms and the patient's medical history.

This system has one of its strongest strengths in assisting patients when doctors are not around. As an active drug recommendation system in remote areas, or where real-time physician consultation is not possible. The data-driven nature of the system ensures not only swift response times but also tailored recommendations to the patient's individual requirements, taking into account symptoms, present conditions, and possible drug interactions.

Apart from that, this system also helps to enhance the health of patients by eliminating self-medication and drug misuse risks. Most patients try to self-treat and select drugs by themselves without seeking the advice of a specialist, and this may result in side effects or even treatment failure. Through an evidence-based recommendation, this system minimizes such risks and offers safer healthcare.

For improved performance, a Drug Recommendation System has also been developed with a feedback system for the users in which the patients can provide feedback on their experience and satisfaction with the suggested drugs. Feedback analysis gives useful feedback about the accuracy and reliability of the recommendations.

10. FUTURE WORK

At present, our Drug Recommendation System is limited in scope because it is locally implemented and based on a pre-defined statistical CSV dataset. Although this method offers a basic basis for drug recommendations, it is limited in real-time access to data, scalability, and responsiveness to changing medical knowledge. In the future, the system can be augmented by partnering with hospitals, health institutions, and drug organizations to bring in real-time patient data. One major improvement in this system would be the inclusion of electronic health records (EHR), enabling the smooth transfer of patient information, such as previous medical history, allergies, and current treatments

Course Outcomes (CO's):

CO1 Perform literature search and / or patent search in the area of interest and Formulate specific problem statements for ill-defined real-life problems with reasonable assumptions and constraints.

CO2 Conduct experiments / Design and Analysis / solution iterations and document the results.

CO3 Perform error analysis / benchmarking / costing.

CO4 Synthesis the results and arrive at scientific conclusions / products / solution.

CO5 Document the results in the form of technical report / presentation.

Program Outcomes (POs)

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. 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.

Program Specific Outcome (PSO's):

PSO 1	Apply algorithmic thinking and utilize programming languages such as C, Python and Java to develop and maintain efficient and robust computing systems.
PSO 2	Design and develop computer-based applications of varying complexities using emerging topics of Computer Science and Engineering such as cloud computing, artificial intelligence, data processing etc.
PSO 3	Possess the subject knowledge and scientific temper necessary to pursue successful careers in Computer Science and Engineering with ethical responsibility towards societal needs.

CO-PO-PSO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	2		2	3	3	3	3	3	3	2
CO2	3	3	3	3	3				3	3	3	3	3	3	2
CO3	3	3	3	3	3				3	3	3	3	3	3	2
CO4	3	3	3	3	3	2		2	3	3	3	3	3	3	2
CO5	2	3	3	3	3		1		3	3	2	3	3	3	2

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

PO'S ATTAINMENT:

Title of the Project: Drug Recommendation System in General Medical Emergencies														
Program Outcomes												**Program Specific Outcomes**		
PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	

REFERENCES

[1] Y. Tanna, V. Parmar, S. Bhalkeshware, A. Maindre, and P. Malusare, “DRUG RECOMMENDATION SYSTEM,” International Journal of Creative Research Thoughts, vol. 11, pp. 2320–2882, 2023, Available: https://ijcrt.org/papers/IJCRT2303080.pdf

[2] B. Dubba, T. Sowmya Sri, M. Sowbhagya Pradhaini, and Sreedhar, “Drug Recommendation System Based on Sentiment Analysis of Drug Reviews Using Machine Learning,” 2024. Available: https://www.ijfmr.com/papers/2024/3/18767.pdf

[3] Abhishek, Amit Kumar Bindal, and D. Yadav, “Medicine recommender system: A machine learning approach,” Jan. 2022, doi: https://doi.org/10.1063/5.0105723.

[4] S. Prasad, “DRUG RECOMMENDATION SYSTEM BASED ON SENTIMENT ANALYSIS OF DRUG REVIEWS USING MACHINE LEARNING,” International Journal of Creative Research Thoughts, vol. 12, pp. 2320-2882, May 05, 2024.

[5] Punidha A, Arjun Balaji, Jaisri V, and Mithra Saravanan, “DRUG RECOMMENDATION SYSTEM USING NLP AND MACHINE LEARNING APPROACH,” International Journal of Creative Research Thoughts, March 03, 2024.

[6] G V Lavanya, Praveen K S, “DRUG RECOMMENDER SYSTEM USING MACHINE LEARNING FOR SENTIMENT ANALYSIS,” International Research Journal of Modernization in Engineering Technology and Science, vol. 5, pp. 2582-5208, July 07, 2023.

[7] S. Garg, “Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning,” 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2021, doi: https://doi.org/10.1109/confluence51648.2021.9377188.

[8] P. Chinnasamy, W.-K. Wong, A. A. Raja, O. I. Khalaf, A. Kiran, and J. C. Babu, “Health Recommendation System using Deep Learning-based Collaborative Filtering,” Heliyon, vol. 9, no. 12, p. e22844, Dec. 2023, doi: https://doi.org/10.1016/j.heliyon.2023.e22844.

[9] Prajakta Khairnar, Vamsi Avula, Aditya Hargane, and Pratik Baisware, “Medicine Recommend System Using Machine Learning,” International Journal of Scientific Research in

Science Engineering and Technology, pp. 247–250, May 2022, doi: https://doi.org/10.32628/ijsrset2293102.

[10] M. Mohapatra, M. Nayak, and S. Mahapatra, “A Machine Learning based Drug Recommendation System for Health Care.,” Graduate Research in Engineering and Technology, pp. 5–10, May 2022, doi: https://doi.org/10.47893/gret.2022.1109.

[11] A. Sharma, P. Gupta, and R. Verma, “Personalized Drug Recommendation System Using Deep Learning and Knowledge Graphs,” Journal of Biomedical Informatics, vol. 140, p. 104271, 2023, doi: https://doi.org/10.1016/j.jbi.2023.104271.

[12] S. Reddy, M. Alam, and K. H. Kim, “Hybrid Drug Recommendation System Based on Sentiment Analysis and Collaborative Filtering,” Expert Systems with Applications, vol. 212, p. 118794, 2023, doi: https://doi.org/10.1016/j.eswa.2023.118794.

[13] L. Zhou, J. Wang, and Y. Liu, “Graph Neural Networks for Drug-Drug Interaction Prediction and Recommendation,” Artificial Intelligence in Medicine, vol. 143, p. 102603, 2023, doi: https://doi.org/10.1016/j.artmed.2023.102603.

[14] H. Patel, R. Kumar, and S. Das, “Deep Reinforcement Learning for Adaptive Drug Recommendation,” IEEE Access, vol. 11, pp. 73584–73598, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3281298.

[15] M. Chen, X. Li, and K. Huang, “Transformer-Based Model for Personalized Drug Recommendations,” BMC Medical Informatics and Decision Making, vol. 23, no. 1, p. 311, 2023, doi: https://doi.org/10.1186/s12911-023-02189-6.

[16] T. Nguyen, P. Tran, and D. Lee, “Multi-Modal Learning for Drug Recommendation Using Clinical Notes and Patient History,” Journal of Biomedical Semantics, vol. 14, no. 1, p. 15, 2023, doi: https://doi.org/10.1186/s13326-023-00293-4.

[17] R. Kumar, J. Zhao, and H. Wu, “Context-Aware Drug Recommendation Using Knowledge Graphs,” International Journal of Medical Informatics, vol. 179, p. 105295, 2023, doi: https://doi.org/10.1016/j.ijmedinf.2023.105295.

[18] Y. Huang, X. Zhang, and M. Sun, “Transfer Learning for Drug Side Effect Prediction and Recommendation,” Scientific Reports, vol. 13, no. 1, p. 14672, 2023, doi: https://doi.org/10.1038/s41598-023-71688-2.

[19] L. Silva, A. Fernandes, and J. Costa, “Meta-Learning for Cold-Start Drug Recommendations,” Artificial Intelligence Review, vol. 56, pp. 2135–2157, 2023, Doi: https://doi.org/10.1007/s10462-023-10289-1.