World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Medicine recommendation system (Health Harbour)

Nagur Vali Shaik, Lokesh Karthik Varma Penmetsa *, Spandana Salveru, Praisey Bathula and Sahas Manikanta Madishetty

Department of CSE (Data Science), ACE Engineering college, Telangana, Hyderabad, India.

World Journal of Advanced Research and Reviews, 2025, 25(02), 195-203

Publication history: Received on 25 December 2024; revised on 31 January 2025; accepted on 02 February 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.25.2.0382

Abstract

Health Harbour is a machine-learning-based system designed to assist users in identifying possible health conditions and finding suitable medications. By analyzing symptoms entered by the user, it simplifies the process of symptom-based diagnosis and provides helpful health insights. Built using Python and Scikit-Learn, the model is trained on a dataset of 187 symptoms and achieves an impressive accuracy of 99.6%. Users can input four key symptoms, and the system will predict potential illnesses while suggesting appropriate medications. Additionally, it offers diet recommendations, necessary precautions, and workout plans to promote overall well-being. With an easy-to-use interface powered by Stream-Lit, Health Harbour ensures a smooth and interactive experience. By making healthcare guidance more accessible, this system helps users take proactive steps toward better health and informed decision-making.

Keywords: Medicine Recommendation System; Machine Learning; Symptom-based Diagnosis; Personalized Healthcare

1. Introduction

Staying healthy is a top priority, and getting the right treatment quickly is crucial. But, let's be honest, figuring out the right medicine when you're feeling under the weather can be a real challenge, even for doctors sometimes. That's where Health Harbour comes in. It's a new system that uses the power of machine learning to help you find the most suitable medications based on your specific symptoms. Think of it as a helpful guide to navigating your health.

Health Harbour is all about helping you find the best medicine when you're not feeling your best. You simply tell it what's bothering you – your symptoms – and it uses its advanced knowledge to figure out what might be going on and suggest the most appropriate treatments. But it goes beyond just medications. It also offers helpful advice on nutrition, preventative care, and even exercise, taking a well-rounded approach to your overall well-being.

You can select up to four symptoms from a comprehensive list of 187, which helps the system pinpoint potential health issues and recommend relevant medications. The system's intelligence comes from a machine learning model trained on a massive dataset of medical information, ensuring a high degree of accuracy in its recommendations. And to make it incredibly user-friendly, Health Harbour features a simple and intuitive interface built with Streamlit. This means that regardless of your tech skills, you can easily use it and understand the results.

Health Harbour aims to make healthcare more accessible to everyone, improve the accuracy of diagnoses, and reduce delays in getting the right treatment. It empowers individuals to make informed decisions about their health and promotes self-care, ultimately striving to make healthcare more efficient and available to a broader audience.

* Corresponding author: P. Lokesh Karthik Varma

2. Related Work

Healthcare is constantly evolving, and one of the most exciting developments is the use of machine learning. It's changing how we approach diagnosis and treatment in some really positive ways. Doctors and other healthcare professionals have always relied on their expertise, but let's face it, humans aren't perfect. We can make mistakes, and sometimes decisions take longer than they should. Machine learning offers a way to make things more accurate and efficient.

Think about Clinical Decision Support Systems, or CDSS. These are tools that help healthcare providers by analyzing all sorts of patient data – symptoms, medical history, test results, you name it – to come up with personalized treatment plans. IBM Watson Health is a good example. It uses AI to sift through massive amounts of medical information, helping doctors make better decisions, faster.

We've actually been trying to automate healthcare for a while now. Remember those early "expert systems" like MYCIN and DENDRAL? They were pretty cool for their time, but they had their limitations. They weren't great at handling huge amounts of data, and they struggled to keep up with new medical discoveries. But machine learning has changed all that. Now we have systems that are much more flexible, scalable, and data-driven. We're seeing things like MEDMNIST for analyzing medical images and symptom checkers like WebMD and Ada Health that help people get a better understanding of what might be going on with their health. The problem is, a lot of these tools require you to be pretty tech-savvy or have a deep understanding of medical terminology.

That's where Health Harbour comes in. We wanted to create something that's truly user-friendly and accessible to everyone. Instead of making you wade through complicated medical forms, Health Harbour lets you simply choose a few of your symptoms from a list. Our machine learning model, which has been trained on a ton of medical data, then gets to work, trying to figure out what might be the cause and suggesting appropriate medications. But we didn't want to stop there. We also include information on diet, preventative care, and exercise, because we believe in a holistic approach to health. And the best part? We've built Health Harbour with a super simple interface, so anyone can use it, regardless of their background. We're hoping Health Harbour can make a real difference by making healthcare more accessible, improving the speed and accuracy of diagnoses, and empowering people to take better care of themselves.

3. Existing System

Our current healthcare system has some real hurdles to overcome if we want it to be truly effective. One of the biggest problems is just how time-consuming everything is. Patients often find themselves bouncing from doctor to doctor, searching through countless websites and books, which can seriously delay getting the care they need. And let's be honest, healthcare professionals are sometimes working with outdated information, which only adds to the wait. Plus, many of the systems currently in use rely on what are called "decision tree models." These models make general recommendations, but they don't take into account a patient's unique medical history, genetics, or lifestyle. This can lead to diagnoses and treatment plans that just aren't right for the individual.

Another issue is that our current systems often treat health in a very fragmented way. They might recommend medication, but they don't necessarily consider the importance of diet, exercise, or preventative measures. A truly effective healthcare system needs to look at the whole picture. Those decision tree models I mentioned also struggle with complex medical conditions. They tend to oversimplify things, which can be a real problem when dealing with chronic illnesses or conditions that have multiple contributing factors. And they often fail to account for things like potential drug interactions or personalized dosages, which can be risky for patients.

On top of all that, most systems don't offer personalized advice on nutrition or lifestyle. This is a huge missed opportunity, because those things play a critical role in both recovery and prevention. What we really need are more advanced, integrated systems that can provide tailored, accurate recommendations while considering all aspects of a person's health.

4. Proposed Model

Health Harbour empowers you to take charge of your health by using machine learning to offer personalized health advice. By simply entering your symptoms, the system provides tailored recommendations for medications, diet, exercise, and preventative measures. At its core, Health Harbour uses the Random Forest algorithm to link your symptoms to potential health conditions, ensuring reliable and relevant suggestions.

Designed for ease of use, Health Harbour's interface allows anyone to quickly input up to four symptoms and receive personalized guidance. What sets it apart is its holistic approach—offering not just medical recommendations but also wellness and lifestyle tips for overall well-being.

While Health Harbour is a powerful tool, it is not a replacement for professional medical advice. Its accuracy depends on the details you provide, and it may not detect rare conditions due to dataset limitations. Think of it as a helpful starting point for understanding your health, with professional consultation always recommended for a diagnosis.

In short, Health Harbour demonstrates how AI can make healthcare more accessible, helping users make informed health decisions.

5. System Architecture

Our system is designed to work smoothly and efficiently, like a well-oiled machine. All the different parts – the user interface you interact with, the behind-the-scenes processing, the machine learning engine – are seamlessly integrated. This ensures that the system is not only user-friendly but also performs reliably and can handle a growing number of users. We've focused on creating a smooth and intuitive experience while maintaining robust performance under the hood.

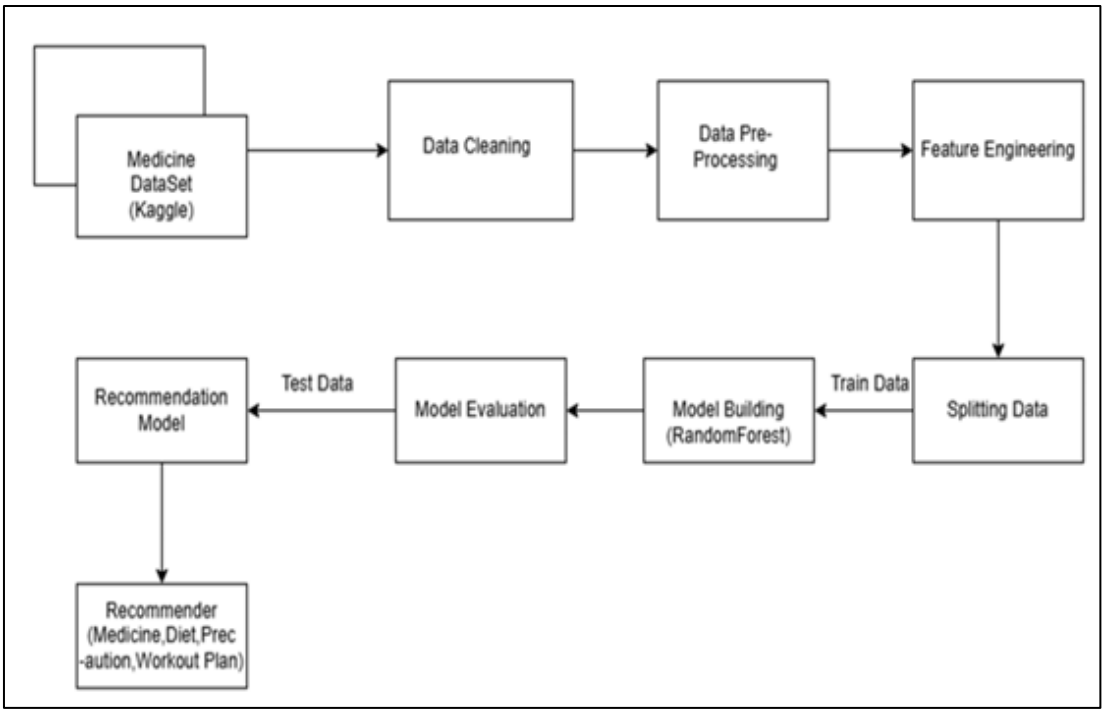

Figure 1 System Architecture

The following steps outline the development flow of the system:

- **Medicine Dataset (Kaggle):** The initial step involves gathering raw data from Kaggle, which includes details about various medicines, symptoms, and treatment patterns. This dataset serves as the foundation for building accurate recommendations.
- **Data Cleaning:** Next, the collected data is cleaned by removing any errors, inconsistencies, and irrelevant information. This step ensures that only accurate and reliable data is used for analysis.
- **Data Pre-Processing:** After cleaning, the data is pre-processed to make it ready for analysis. This includes tasks like normalizing data, handling missing values, and transforming data into a format that is compatible with machine learning models.
- **Feature Engineering:** In this step, additional features are created to enhance the performance of the machine learning model. These features are selected based on their potential to improve the model's ability to make accurate recommendations.

- **Splitting Data:** The dataset is split into two parts: one for training (70%) the model and another for testing (30%) it. This allows us to evaluate the model's performance on unseen data to ensure it generalizes well.
- **Model Building:** The training dataset is then used to build the recommendation model using the Random Forest algorithm. This machine learning model learns the patterns between symptoms and suitable treatments.
- **Model Evaluation:** After training the model, we assess its performance using various evaluation metrics, such as accuracy and precision. This ensures that the model makes reliable predictions based on the test data.
- **Recommendation Model:** Finally, the trained model generates personalized recommendations for the user, suggesting medicines, diet plans, necessary precautions, and suitable workout routines.

6. Methodology

The methodology for the Medicine Recommendation System is designed to leverage machine learning to predict medical conditions based on user-provided symptoms and generate personalized recommendations for medicine, diet, precautions, and workout plans.

Description of the Machine Learning Algorithms and Techniques Chosen: Libraries imported are:

LOADING NECESSARY LIBRARIES

```
!pip install scikit-learn
import numpy as np
import pandas as pd
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier
```

Figure 2 Importing Libraries

6.1. Data Collection and Pre-Processing

The first step in building a medicine recommendation system is collecting relevant data. We sourced our dataset from Kaggle, which includes information on symptoms, medicines, and treatment patterns. To ensure high-quality data, preprocessing is essential. This involves handling missing values, removing inconsistencies, and formatting the data to make it compatible with machine learning models. Normalization is also applied to bring all data into a standard format, improving model efficiency and accuracy.

6.2. Feature Selection

Feature selection plays a crucial role in optimizing model performance. It involves identifying the most relevant attributes from the dataset that contribute significantly to predicting the right medicine. By analyzing relationships between symptoms and treatments, irrelevant or redundant features are removed. This not only simplifies the model but also enhances its accuracy and efficiency by reducing computational complexity.

6.3. Model Selection

The core of the system relies on machine learning techniques to analyze patterns and provide recommendations. The model is built using algorithms like Random Forest, which is effective in handling complex relationships between symptoms and medicines. The training process allows the model to learn from past data and generalize well to new inputs. The choice of algorithm ensures reliable and personalized recommendations for users.

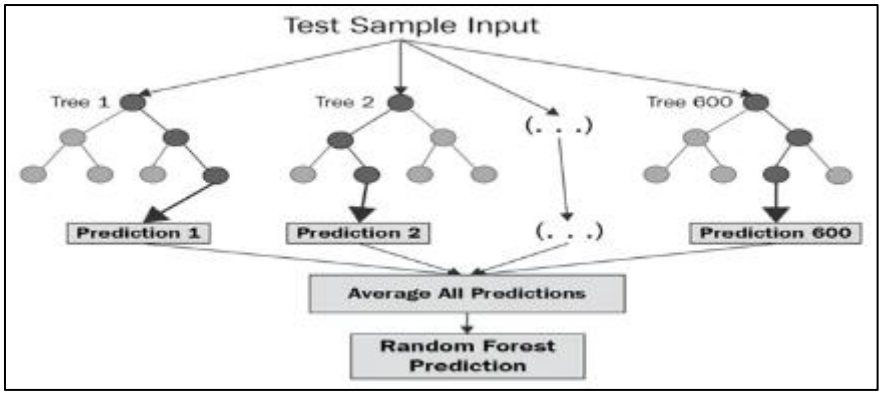

Figure 3 Random Forest

6.4. Model Training and Evaluation

Once the model is designed, it undergoes training using historical data. The dataset is divided into training, validation, and testing sets to ensure unbiased learning. The training phase helps the model recognize patterns, while validation data is used to fine-tune parameters. The model's performance is then evaluated using metrics such as accuracy, precision, recall, and F1-score. This ensures that the system is both reliable and applicable in real-world scenarios.

TRAIN-TEST SPLIT

```
# Splitting the data into training and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(encoded_df, disease, test_size=0.3, random_state=1)
```

```
# RandomForest Classifier model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predicting on test data
y_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Accuracy scores
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
train_accuracy = accuracy_score(y_train, y_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

# Display train and test accuracy
train_accuracy, test_accuracy
```

```
(0.9952486537852392, 0.9963045084996305)
```

Figure 4 Training the Model

6.5. Recommendation System

After the model is trained, it becomes capable of offering tailored recommendations. Users provide symptoms, and the trained system processes this input to recommend the most appropriate medications. Additionally, it offers personalized suggestions for diet, safety measures, and exercise routines, contributing to comprehensive health management. This stage is essential as it transforms user data into actionable and individualized health insights.

TESTING THE MODEL

```
# Function to get the key (disease name) for a given value (predicted value)
def get_disease_by_value(disease_map, value):
    for key, val in disease_map.items():
        if val == value:
            return key
    return "Disease not found"
diseasemap = {
    '(vertigo) Paroymsal  Positional Vertigo': 0, 'AIDS': 1, 'Acne': 2,
    'Alcoholic hepatitis': 3, 'Allergy': 4, 'Arthritis': 5, 'Bronchial Asthma': 6,
    'Cervical spondylosis': 7, 'Chicken pox': 8, 'Chronic cholestasis': 9,
    'Common Cold': 10, 'Dengue': 11, 'Diabetes': 12, 'Dimorphic hemmorhoids(piles)': 13,
    'Drug Reaction': 14, 'Fungal infection': 15, 'GERD': 16, 'Gastroenteritis': 17,
    'Heart attack': 18, 'Hepatitis B': 19, 'Hepatitis C': 20, 'Hepatitis D': 21,
    'Hepatitis E': 22, 'Hypertension': 23, 'Hyperthyroidism': 24, 'Hypoglycemia': 25,
    'Hypothyroidism': 26, 'Impetigo': 28, 'Jaundice': 29, 'Malaria': 30, 'Migraine': 31,
    'Osteoarthristis': 32, 'Paralysis (brain hemorrhage)': 33, 'Peptic ulcer disease': 34,
    'Pneumonia': 35, 'Psoriasis': 36, 'Tuberculosis': 37, 'Typhoid': 38,
    'Urinary tract infection': 39, 'Varicose veins': 40, 'hepatitis A': 41
}
# Assuming model.predict() returns an array of predicted values
predicted_values = loaded_model.predict([[0 ,0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,0, 0, 0, 0, 0,0,0,0,0,0,0]])

# Iterate through predicted values and get corresponding disease names
predicted_diseases = [get_disease_by_value(diseasemap, value) for value in predicted_values]
#print(predicted_diseases)
```

Figure 5 Testing the model

```
# Iterate through predicted values and get corresponding disease names
predicted_diseases = [get_disease_by_value(diseasemap, value) for value in predicted_values]
#print(predicted_diseases)

# Checking and printing corresponding medications from the 'medicine' DataFrame
for disease in predicted_diseases:
    if disease in medicine['Disease'].values:
        # Get the corresponding medication
        medication = medicine.loc[medicine['Disease'] == disease, 'Medication'].values[0]
        print(f"Disease: {disease}, Medication: {medication}")

    if disease in precaution['Disease'].values:
        precautions= precaution.loc[precaution['Disease']== disease, ['Precaution_1', 'Precaution_2', 'Precaution_3', 'Precautio
        print(f"Precaution:{precautions[0]},{precautions[1]},{precautions[2]},{precautions[3]}")

    if disease in diet['Disease'].values:
        diets=diet.loc[diet['Disease']==disease,'Diet'].values[0]
        print(f"Diet:{diets}")

    if disease in grouped_workout['disease'].values:
        wo = grouped_workout.loc[grouped_workout['disease'] == disease, 'workout'].values[0]
        print(f"workout:{wo}\n")
    else:
        print(f"Disease: {disease}, Medication not found")
```

```
Disease: Typhoid, Medication: ['Vaccination', 'Antiviral drugs', 'IV fluids', 'Blood transfusions', 'Liver transplant']
Precaution:eat high calorie vegitables,antiboitic therapy,consult doctor,medication
Diet:['Typhoid Diet', 'High-Calorie Diet', 'Soft and bland foods', 'Hydration', 'Protein-rich foods']
workout:['Consume easily digestible foods', 'Stay hydrated', 'Include nutrient-rich foods', 'Consume foods rich in vitamin C an
d antioxidants', 'Include protein-rich foods', 'Avoid fatty and greasy foods', 'Consult a healthcare professional', 'Gradually
resume normal diet', 'Follow medical recommendations', 'Avoid alcohol and caffeine']
```

Figure 6 Predicting the output

7. Results

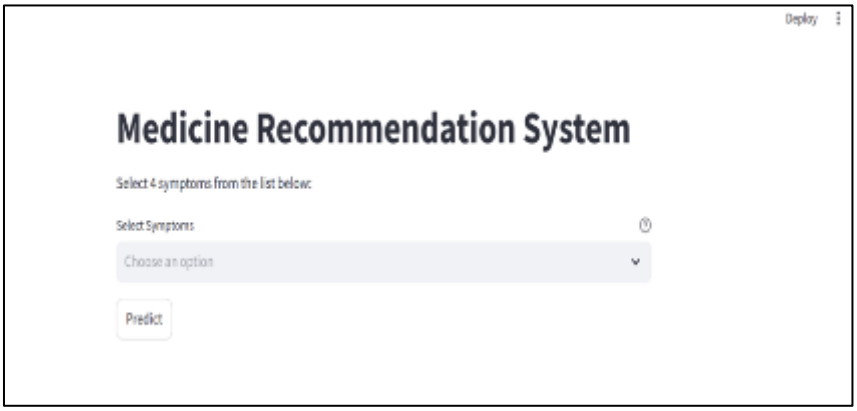

Figure 7 Interface

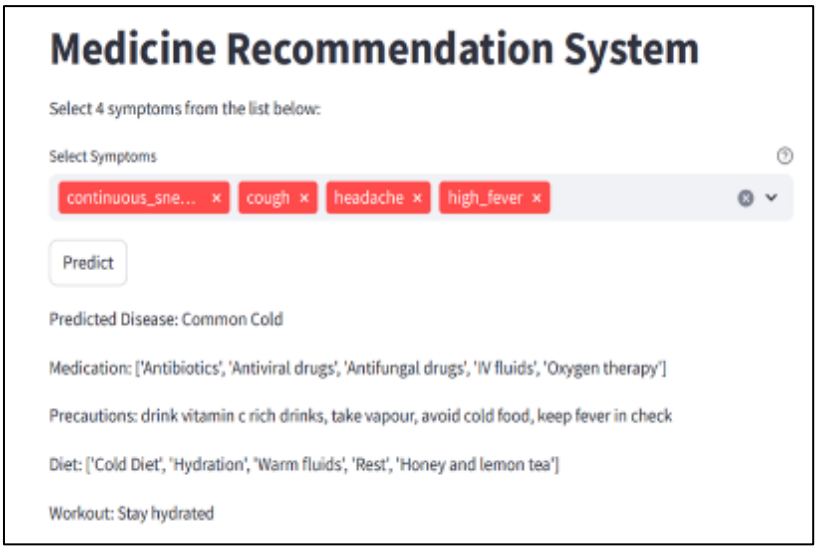

Figure 8 Sample Output-1

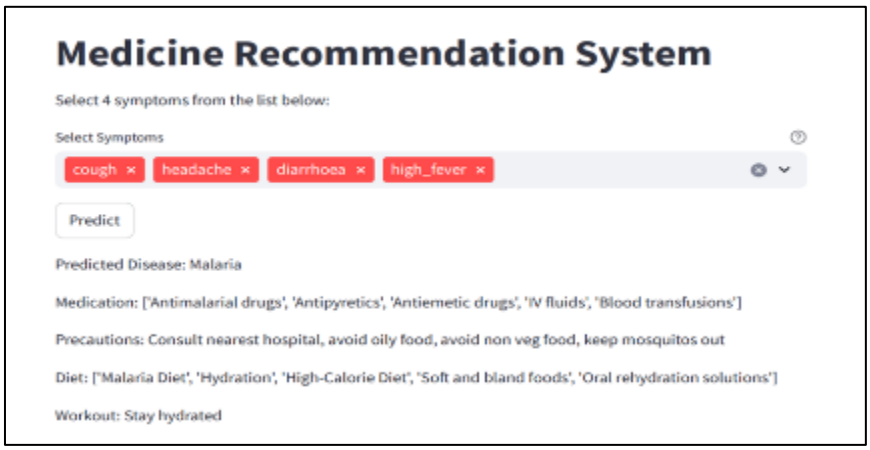

Figure 9 Sample Output-2

8. Discussion

The Medicine Recommendation System (Health Harbour) demonstrates how machine learning can revolutionize healthcare by utilizing the Random Forest algorithm for predicting diseases based on reported symptoms. This algorithm was chosen for its ability to manage large volumes of data, reduce overfitting, and deliver consistent, accurate predictions. Users can easily input up to four symptoms, which streamlines the diagnostic process and ensures an intuitive experience. In addition to recommending medications, the system also offers personalized advice on diet and exercise, encouraging a holistic approach to health improvement.

While effective, the system does have some limitations. Its performance relies heavily on the accuracy of the user input, and its training data is mainly based on common symptom-disease patterns, meaning it might struggle to identify rare conditions. While Random Forest is reliable in prediction, it doesn't always provide deep insights into more complex cases. Furthermore, symptom interactions in real life can be more complicated than those captured in the training data, and users may find it difficult to describe symptoms accurately. By enhancing the dataset and improving the system's interpretability, the tool could become even more effective in real-world healthcare applications.

9. Conclusion

The Medicine Recommendation System uses artificial intelligence to provide personalized health advice, including medications, diet plans, exercise routines, and prevention strategies based on the symptoms users report. By using the Random Forest algorithm, it links symptoms to possible health issues, ensuring that the recommendations are both accurate and data-driven. This AI-powered system simplifies access to crucial health information, making it more understandable and reliable for users.

Designed to be easy to use, the system allows users to input up to four symptoms. It then generates customized health suggestions, making it accessible to a broad range of users.

However, there are some challenges. The accuracy of the system depends on how well users describe their symptoms, and it might not detect less common diseases due to the data limitations. Therefore, while it provides helpful initial advice, it should not be considered a substitute for professional medical consultation.

Overall, the Medicine Recommendation System (Health Harbour) highlights how AI can improve healthcare by bridging the gap between users and medical insights, paving the way for more personalized and accessible healthcare solutions.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] "Medicine Recommendation Dataset,"Kaggle, 2020." [Online].

+Available: Medicine Recommendation

[2] A Literature Review on Medicine Recommender Systems – International Journal of Advanced Computer Science and Applications, Jan (2019)

[3] Smith et al., Symptom-based Disease Prediction Using Machine Learning Models", Journal of Healthcare Informatics Research, Springer, Berlin, 2021, PP 45-60.

[4] Anderson, A Survey on Disease Prediction Using Machine Learning Algorithms in Healthcare", https://arxiv.org/abs/2001.04587.

[5] Clinical-Decision-Support-Systems
https://library.oapen.org/bitstream/handle/20.500.12657/22918/1/1007243.pdf#page=153

[6] Webmb for medication and Symptom tracker Url: https://www.webmd.com

Author's short biography

Mr. Nagur Vali Shaik I'm Mr. Nagur Vali Shaik, an Assistant Professor in Computer Science and Engineering (Data Science) with 5.2 years of Industry Experience in domains like DevOps, AWS Cloud, and Data Science and 1.2 years Teaching Experience. Holding a B.Tech and M.Tech in Computer Science and Engineering. I inspire students and contribute to advancements in technology through my work.	
Lokesh Karthik Varma Penmetsa I am Lokesh Karthik Varma Penmetsa, a Final-Year B.Tech Student at ACE Engineering College, specializing in CSE (Data Science). I am passionate about coding and problem-solving in Java and Python. I strive to improve myself continuously and innovate new things in the tech world. My goal is to explore industry work cultures and contribute to impactful technological advancements.	
Spandana Salveru I am Spandana Salveru, A Final-Year BTech Student at ACE Engineering College, Specializing in CSE(Data Science). I am Passionate about coding and problem-solving in java and python, I Strive to improve myself and enjoy exploring new things. My goal is to explore various industry works and contribute to new technologies.	
Praisey Bathula I am Praisey Bathula a final-year student specialising in Data Science at ACE Engineering College. I have a strong interest in ALML and am passionate about Data Analytics. I aspire to bridge technology with real-world challenges, focusing on data-driven innovations to solve real world problems and make a meaningful impact.	
Sahas Manikanta Madishetty I am Sahas Manikanta Madishetty, A Final-Year B.Tech Student at ACE Engineering College, Specializing in CSE(Data Science). I have a strong interest in Python and enjoy working on UI/UX development. I love creating interesting websites to enhance my tech skills and explore new design possibilities