BATCH NO:MI2380

DRUG ADVISORY SYSTEM USING ML

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

C.HARSHITH CHOWDARY	(22UEIN0004)	**(VTU23230)**
S.SIVA MALLIKA	(22UEIN0023)	**(VTU22673)**
T.SAI LAKSHMI LALASA	(22UEIN0024)	**(VTU21867)**

Under the guidance of
Dr.T.KAMALESHWAR,Ph.D
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

BATCH NO:MI2380

DRUG ADVISORY SYSTEM USING ML

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

C.HARSHITH CHOWDARY	(22UEIN0004)	**(VTU23230)**
S.SIVA MALLIKA	(22UEIN0023)	**(VTU22673)**
T.SAI LAKSHMI LALASA	(22UEIN0024)	**(VTU21867)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

CERTIFICATE

It is certified that the work contained in the project report titled ”DRUG ADVISORY SYSTEM USING ML” by ”C.HARSHITH CHOWDARY (22UEIN0004), S.SIVA MALLIKA (22UEIN0023), T.SAI LAKSHMI LALASA (22UEIN0024)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

of Supervisor
Dr.T.KAMALESHWAR
Associate Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of Head/Assistant Head of the Department
Dr. N. Vijayaraj/Dr. M. S. Murali dhar
Professor & Head/ Professor &Assistant Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of the Dean
Dr. S P. Chokkalingam
Professor & Dean
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(C.HARSHITH CHOWDARY)

Date: / /

(Signature)

(S.SIVA MALLIKA)

Date: / /

(Signature)

(T.SAI LAKSHMI LALASA)

Date: / /

APPROVAL SHEET

This project report entitled HEALTH ADVISORY USING ML by C.HARSHITH CHOWDARY (22UEIN0004), S.SIVA MALLIKA (22UEIN0023), T.SAI LAKSHMI LALASA (22UEIN0024) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.T.KAMALESHWAR,Ph.D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for their blessings.

We express our sincere thanks to our respected Chairperson and Managing Trustee **Mrs. RANGARAJAN MAHALAKSHMI KISHORE,B.E., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Associate Dean,School of Computing, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Computer Science & Engineering, Dr. N. VIJAYARAJ, M.E., Ph.D.,** and **Professor & Assistant Head, Department of Computer Science & Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,**for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Supervisor name,degree.,(in capital letters)** for his/her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Dr. SADISH SENDIL MURUGARAJ,Professor, Dr.C.SELVARATHI,M.E,Ph.D., Mr. V. ASHOK KUMAR, B.E,M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

C.HARSHITH CHOWDARY (22UEIN0004)
S.SIVA MALLIKA (22UEIN0023)
T.SAI LAKSHMI LALASA (22UEIN0024)

ABSTRACT

Advisor systems (AS) suggest the right item to the right user. It predicts the user's output to an item based on their input. AS provides the suggestion to users. The system is designed to assist healthcare providers by suggesting potential medica tions based on patient-specific data according to their current symptoms. In today's world many online applications are already using the Advisor system that provides a recommendation for a particular item like books, movies, music etc. The project employs a content-based filtering approach, where the system matches patient pro f iles with drug characteristics to recommend the most appropriate medications. This aims to propose a system that helps to find the best suitable Medicine advise accord ing to diseases by using library "data sets". This study develops a system that gives the best medicine available according to the user rating available in database. User gets their medication and workouts. By their provided input diseases. Due to un availability, individuals started taking medication independently without appropriate consultation, making the health condition worse than usual. To evaluate the performance of the proposed system, extensive experiments are con ducted using real-world clinical datasets. The results demonstrate the efficiancy of the SVM-based approach in accurately predicting patient responses to various treatment regimens, thereby facilitating personalized medicine recommendations. Moreover, comparative analyses with alternative machine learning methods high light the su periority of SVM in terms of predictive accuracy and generalization ability. this project offers advantages in terms of cost-effectiveness compared to the existing sys tem. This could involve reducing unnecessary medical tests, optimizing treatment plans, and improving overall healthcare resource utilization.

Keywords:

Advisor Systems, Machine Learning Models, Random Forest Clas sifier, User Interface

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Networks
FS	Feature Selection
HTML	Hypertext Markup Language
ML	Machine Learning
MT	Model Training
RF	Random Forest
SVC	Support vector classifier
SVM	Support Vector Machine
TTS	Treatment Tailoring Strategies
UI	User Interface

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Health Advisor system involves tailoring medical treatments and interventions to individual patients based on their unique characteristics, such as ge netics, lifestyle factors, environmental influences, and personal preferences. This approach recog nizes that each patient responds differently responds to treatments and that ”one size-fits-all” approaches may not be the most effective or efficient.The benefits of advisor systems are significant. It can lead to improved treat ment outcomes by ensuring that patients receive therapies that are most likely to be effective for them, minimizing the risk of adverse reactions or ineffective treatments. Additionally, personalized medicine can enhance patient satisfaction and engage ment by involving them in the decision-making pro cess and tailoring treatments to align with their preferences and values. Overall, personal health advising system represents a paradigm shift in healthcare, moving away from a one-size-fits-all approach towards more precise, targeted, and effective interventions that maximize patient outcomes and quality of life.

1.2 Aim of the project

Health Advisor System using machine learning is to harness the power of com putational algorithms to tailor medical treatments and interventions to individual pa tients based on their unique characteristics. By analyzing vast amounts of patient specific data, machine learning algorithms can identify patterns and correlations that may not be apparent to human practitioners. This enables the development of highly personalized treatment plans that are more likely to be effective for individ ual patients, improving overall treatment outcomes. By avoiding trial-and-error ap proaches and minimizing unnecessary treatments, machine learning can help reduce healthcare costs and improve the overall efficiency of the healthcare system.

1.3 Scope of the Project

The scope of this project focused on Health Advisor Using Machine Learning is vast and continually expanding as technology advances and our understanding of medical science deepens.Machine learning algorithms can analyze genomic data to identify genetic variations associated with disease susceptibility, drug response, and treatment outcomes. This information can be used to tailor treatments to individual patients based on their genetic profile. The Machine learning model can predict pa tient outcomes, such as disease progression, treatment response, and risk of adverse events, by analyzing various patient data types, including clinical, genomic, imaging, and lifestyle data. This also includes addressing ethical and regulatory challenges related to data privacy, informed consent, algorithm transparency, and bias mitiga tion to ensure the responsible and equitable implementation of these technologies in healthcare. This project can analyze data from wearable devices and remote monitoring sys tems to track patient health metrics in real-time, identify early signs of disease pro gression or complications, and provide personalized interventions for lifestyle mod ifications and also analyzes population-level health data to identify patterns, and risk factors associated with disease incidence, prevalence, and outcomes. This informa tion can be used to develop targeted interventions and public health strategies.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

The increasing prevalence of health advisor systems has underscored the critical need for effective strategies. Machine Learning techniques have emerged as a primary techniques for the enhanced advisory results. [1] S. K. Nayak et.al (2023) Proposes a medicine advice machine that makes use of information mining and machine mastering strategies to research scientific databases and offer sufferers with personalized remedy pointers primarily based on their signs and symptoms. The machine makes use of herbal language processing, sensitivity evaluation, and probabilistic methods to make drug recommendations considering feasible facet effects and device learning algorithms, the gadget objec tives to provide drug guidelines determine is correct and dependable, and could in the end boom patient effects The treasured of the choice gives the technique.

[2]V. K. Dailya et.al (2021) Explains the optimization of multivariable linear regression for the prediction of the development of diabetic illness in 442 patients. Models efficiency of the optimized systems are compared with a host of regression techniques proves much better, thus showing its ability to offer patients. The results suggest that this strategy could be a useful tool for health professionals to select out high-risk patients and develop tailored interventions to slow down the progression of disease.

[3]Tae-Ho Hwang et.al (2020) Proposes an Interactive Healthcare System using artificial intelligence, and chatbots in providing users personalized health advice. By measuring and analyzing the biological information exhibited by users including, but not limited to, body temperature, oxygen saturation, and electrocar diogram, the chatbot provides health advice.

2.2 Gap Identification

[1] Hakilo Sabit (2020) It explained a drug recommender system is proposed in which sentiment analysis will be used to analyze user-generated reviews of drugs and provide personalized recommendations to patients. In the applied system, machine learning algorithms like Decision Tree were used for generating ratings on drugs using user reviews, patient conditions, and sentiment polarity. The model was chosen for rating generation because it has an accuracy and efficiency. It then uses a hybrid recommendation model to list the suitable medication for patients according to their symptoms.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system of drug Advisory system Using Machine Learning helped in advisory mechanism of medicines to be used, but it was not very accurate. It used the Decision Tree, which gave prediction accuracy score of 96 percentage, which may lead to few wrong results. The existing system used a very little amount of data to classify the data i.e, they chose minimum information to build the model, which led to a low scalability. The existing system lacked the user-friendly user interfaces. Before the advent of machine learning-based system, healthcare relied on various traditional methods, including established clinical guidelines, manual decision-making by physicians, and pharmaceutical databases. Physicians made recommendations based on their training and patient history, while drug interaction checkers provided limited interaction alerts. Complex cases often required specialist consultations, and pharmacists offered medication advice based on their knowledge. Patient education materials provided general information but lacked personalization. These existing systems faced challenges such as variability in treatment approaches, inefficiencies, potential for human error, and limited personalization, ultimately leading to delays and inappropriate medication choices. The introduction of machine learning aimed to enhance accuracy, efficiency, and personalization in medication advising by leveraging large datasets and advanced algorithms.

3.2 Problem statement

Health is essential and one of the prime focuses in humans, it occupies an essential role in the quality of every individual's lifeand cannot be neglected. An balanced health could be more complex due to diagnosis and management of different medical conditions. Technology has improved now, and high-speedinternet is available all over the world, so more number of people using ansmart phone only to use social me-

dia sites or do shopping'sand time passing.Networking gives a lot of things available and we did not understand that which product or services is good for us. No Medicine Prescription available to cure our illness. Taking Own medications are often now-a-days because absence of a trusted health advisor,and using over usage of drugs or online social media advice without proper consultation of professional guidance, which leads to receiving the incorrect 4 dosing and adverse reactions.The traditional approaches for getting the treatment have a several drawbacks. Moreover, a doctor may or may not have the ability to analyse different forms of data.The diagnosis may be impacted by the poor quality, and frequent differences from other machines. A varied datasets with relevant features gathered, the data must be preprocessed, machine learning models must be assessed. The system involves the development of a user-friendly web page that combines the trained models for advising medication, workouts, description of the illness and the diets. The web page will allow users to input their data and receives error-free results. The project scope is to focus on providing an efficient tool that is easy and accessible using machine learning techniques to provide good health for everyone

3.3 System Specification

3.3.1 Hardware Specification

• RAMCapacity : 8GB or higher
• SSD: 256GB or higher
• GPU:NVIDIAGPUs
• Processor : Intel Core i5 or i7, or AMD Ryzen 5 or 7
• Hard Disk : 512 GB SSD Google Cloud Storage(Optional)

3.3.2 Software Specification

• Data Visualization : Matplotlib or Seaborn
• Key Libraries : Pandas , NumPy
• Algorithms : RF and LR
• Model Training : Google Colab, VS code
• RAM:8GBandhigher
• Language : Python

3.3.3 Standards and Policies

When using Google Colab to implement Predictive Modeling there are sev eral important policies and considerations. These cover data privacy, ethical AI practices, compliance with healthcare regulations, and best practices for model development.Although advanced models may offer high predictive power, simpler models (e.g., decision trees) can be more interpretable. Depending on stakeholder requirements, consider balancing accuracy with interpretability.

Standard Used: MHD 1165

Jupyter

When developing predictive models for Health IAdvising System using ML techniques in a Jupyter environment, it's essential to maintaining data security, ensuring model accuracy and transparency, and promoting ethical considerations. Random Forest models are generally interpretable; use this to explain the impact of features on predicted health medicine. This helps in understanding which factors most significantly influence advising.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Proposed System

The proposed health advisor system employs sophisticated algorithms and ma chine learning to handle extensive patient data, genomic information and medical literature, thereby improving health care efficiency. It allows for specific and instant advice which helps in interventions that take place on time. This enables the system to predict treatment outcomes more precisely, choose the best course of action, elim inate guesswork, avoid futile treatments and minimize drug side-effects. As a result, patients conditions become better with increased fulfilment that helps the human health to balance their daily routine. Data Collection and Preparation includes arranging the datasets with the rel evant features to process in disease prediction. Collecting the data on the different disease types, symptoms, health conditions. Cleaning and preprocessing of data. Limiting access to the sensitive data of patients allowing admin login with Two Step authentication. Enforcing strong restrictions on user access enables protection against breach of sensitive data of the patients. Performing vulnerability and security assessments regularly can aid in exposing the weaknesses and flaws in the system. Developing interfaces for seamless integration with existing hospital databases. Using proper communication with the healthcare administrators. Enable real time data retrieval from hospital databases to ensure access to up-to-date medical his tories. Enhancing compatibility with current hospital databases increases the ability of healthcare workers to make informed treatment choices, streamlines operations in the clinic, as well as improves the quality of healthcare provided to the patients thanks to individualized treatment and better results. Feature Selection describes the controls those features that are most useful in the prediction of the diseases at hand and equips them into the feature selection algorithm. Depending on the type of machine learning algorithm used, feature importance ranking, recursive feature removal, or correlation analysis may be employed.

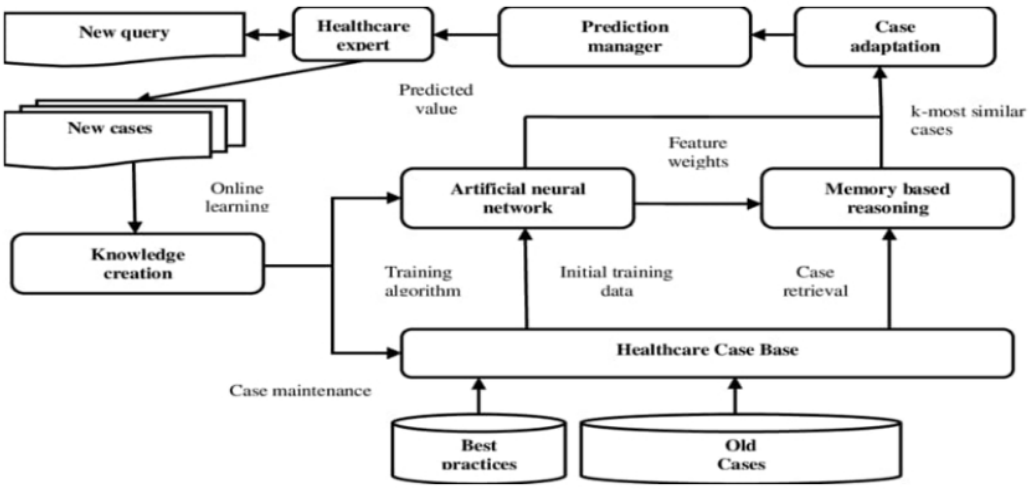


Figure 4.1: **Architecture diagram**

4.2 General Architecture

The Figure 4.1 describes the Health Advisor system Using Machine Learning Architecture diagram. This mainly consist of four sections: Data Collection, Data Processing, Output Prediction and Data Presentation. Initially the data is collected from the user and then the processing takes place in which the data gets scaled and encoded for better learning of the model. Later the model i.e, the Support Vector Ma chine Classifier predicts the best output of the given data and sends the information to the webpage. This architecture aims to create a robust advising system and proac tive monitoring. Implementing these components and processes can significantly enhance an organization's security posture. Using proper communication with the healthcare administrators. Enable real time data retrieval from hospital databases to ensure access to up-to-date medical his tories. Enhancing compatibility with current hospital databases increases the ability of healthcare workers to make informed treatment choices, streamlines operations in the clinic, as well as improves the quality of healthcare provided to the patients thanks to individualized treatment and better results. Feature Selection describes the controls those features that are most useful in the prediction of the diseases at hand and equips them into the feature selection algorithm. Depending on the type of machine learning algorithm used, feature importance ranking, recursive feature removal, or correlation analysis may be employed.

4.3 Design Phase

4.3.1 Data Flow Diagram

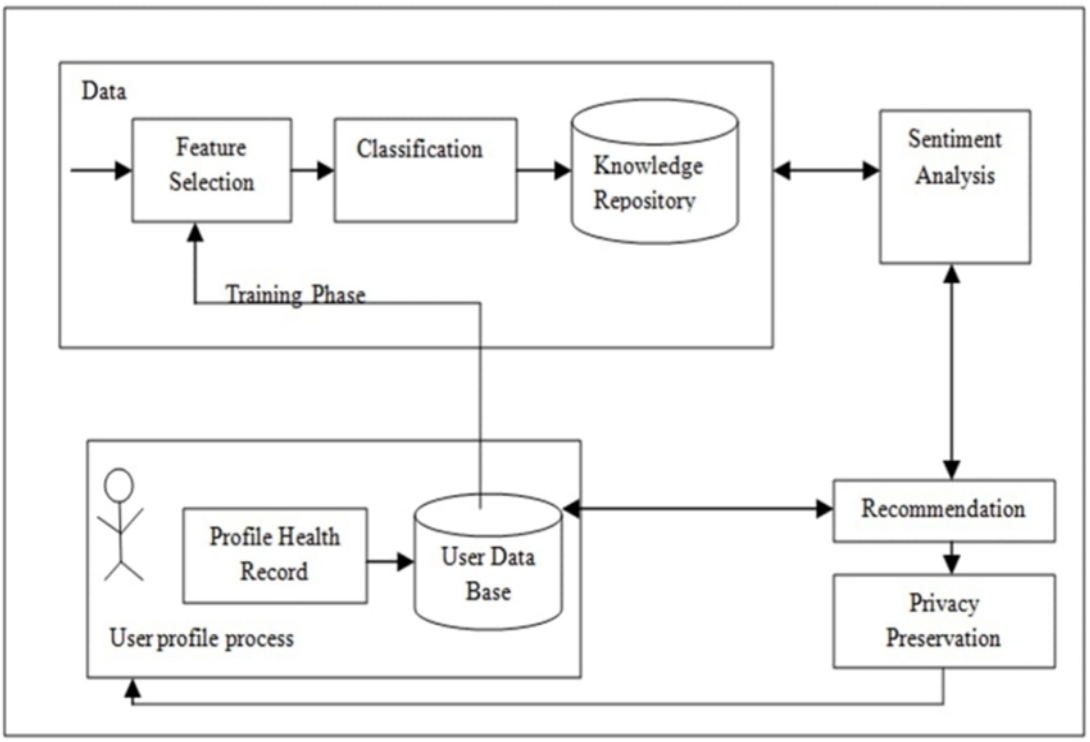


Figure 4.2: **Data flow daiagram**

The Figure 4.2 shows the dataflow diagram of the Health Advisor system operates through a series of stages, beginning with the input of various parameters.Once the input data is collected, it undergoes a systematic process within the system.The data is then analyzed using support vector machine classifier algorithm. The output from this is decrypted by the flask framework where a user-defined function generates the output. This output is printed on the user interface using routing by flask. This split allows the model to be trained on one portion of the data and tested on another to evaluate its accuracy

4.3.2 Use Case Diagram

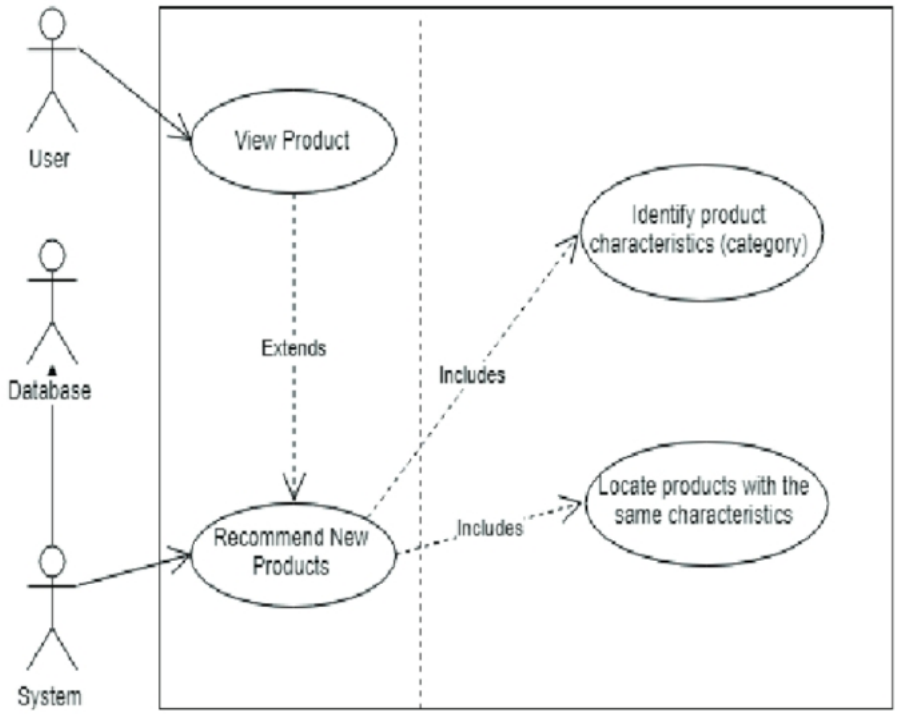


Figure 4.3: **Use case diagram**

The above Figure 4.3 represents the use case diagram for Health Advisor system Using Machine Learning in which the user i.e, the user accesses the website and enters the symptoms which gets processed by the machine learning model and the model returns the predicted output to the website and the website displays the result. Once the data is input or updated, the system processes it, preparing it for further analysis or prediction. The user can then view the results of these predictions, aiding them in decision making applications.

4.3.3 Class Diagram

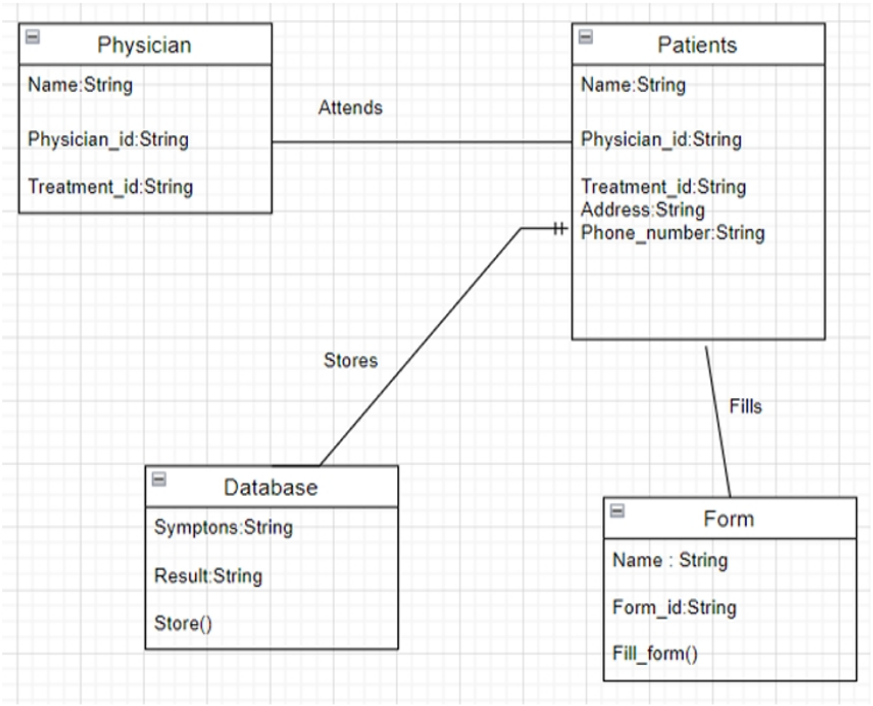


Figure 4.4: **Class diagram**

The above Figure 4.4 represents the class diagram of the Health Advisor system Using Machine Learning. It consists of the patient and the medicine classes. The Health Advising system recommends classes consists of patient, medicine, disease, symptoms and related related attributes along with the function prediction which implies to the predict class. The predict class contains the result function which dis plays the result. This entity stores the results of the prediction. It contains a Result, a reference to the patient, the prediction of the medicine (the predicted medicine) and the prediction model used.

4.3.4 Sequence Diagram

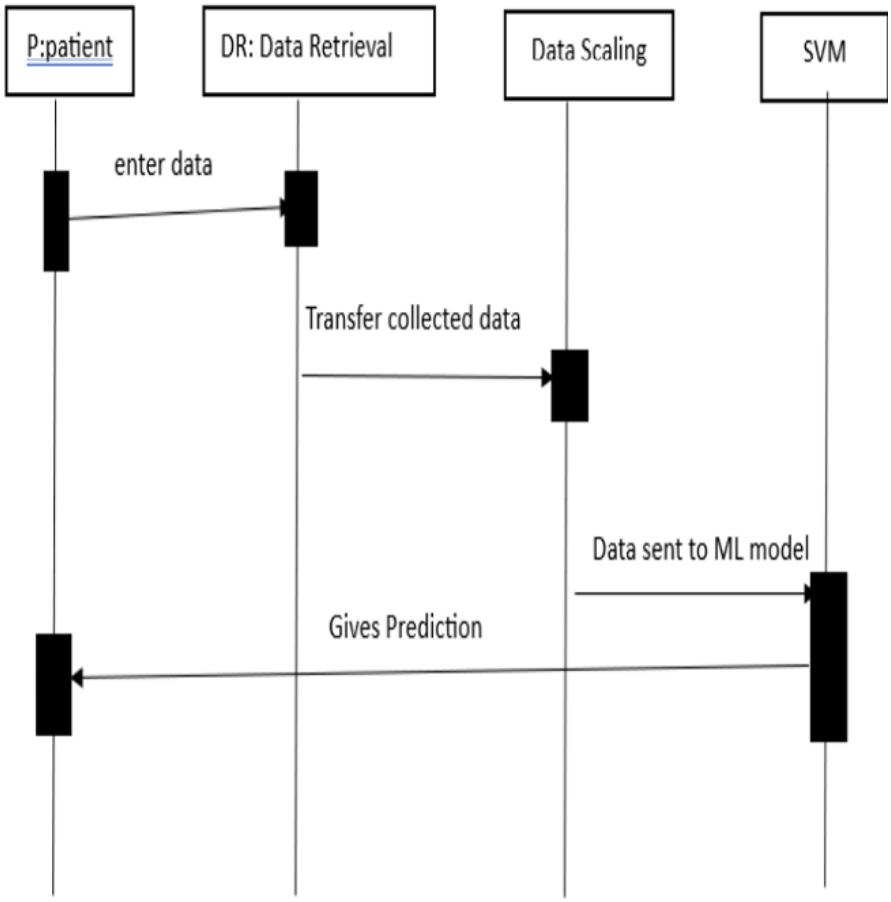


Figure 4.5: **Sequence diagram**

The figure 4.5 tells about a sequence diagram of the Health Advisor System Us ing Machine Learning that illustrates the chronological flow of interactions and mes sages exchanged between various components or modules within a system. In this, initially the patient gives the data to the website which gets interpreted by the back end program that is integrated with the help of flask framework. The data is scaled using standard and min-max scalar methods. After that, the support vector classifier processes this scaled data as input and predicts an appropriate medicine

4.3.5 Collaboration diagram

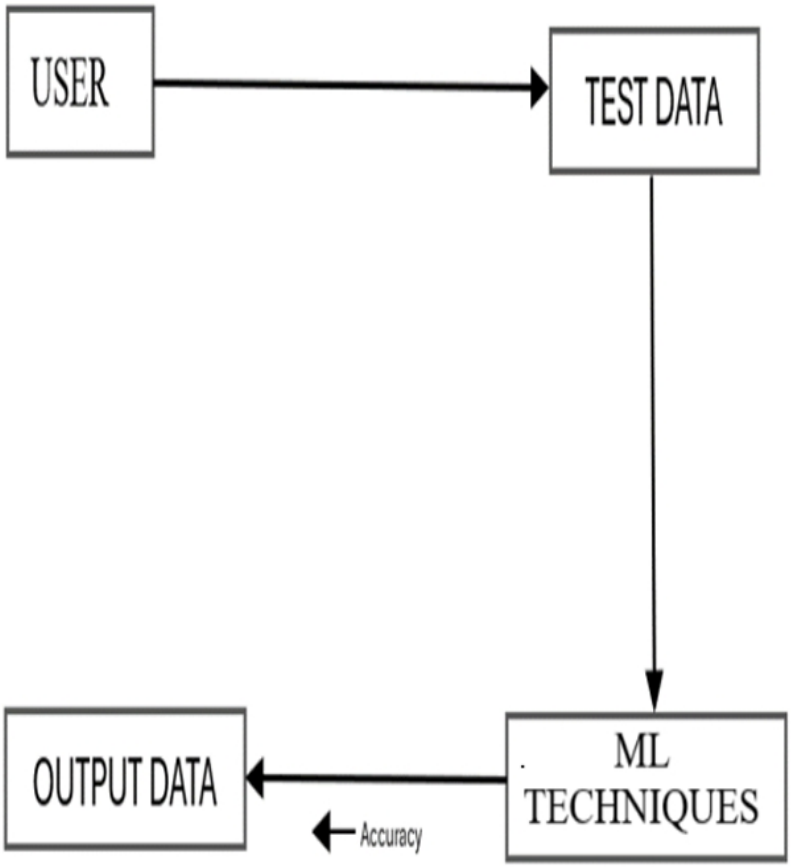


Figure 4.6: **Fig. Name**

The Figure 4.6 illustrating the process of using ML Techniques for data analysis. It begins with the "USER" who provides "TEST DATA." This test data is then fed into a "ML Approaches" model. The output from the ML model is labeled as "OUT PUTDATA". Additionally, there is a feedback loop where the accuracy of the output data is evaluated and sent back to the ML model to improve its performance. This diagram is relevant as it visually represents the workflow of a ML process, specif ically how user-provided data is processed through a ML model to produce output data and how accuracy feedback is used to refine the model.

4.3.6 Activity Diagram

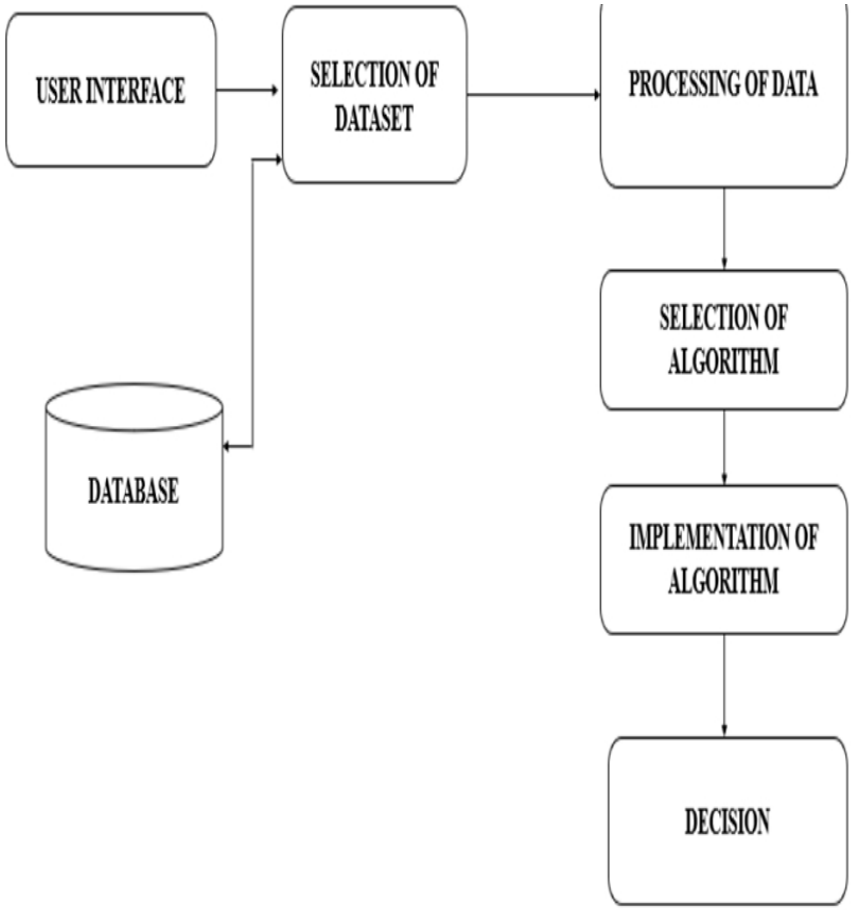


Figure 4.7: **Fig. Name**

4.4 Algorithm & Pseudo Code

4.4.1 Algorithm

STEP1: Explore the distribution of each feature through visualizations.
STEP2: Separate the independent variables from the dependent variable.
STEP3: Use the ML models to train and fit the model to the training data.
STEP4: Use the trained model to predict values on the training data.
STEP5: Predict values on the test data and calculate the test set.
STEP6: Make Predictions:Take a new input (e.g., a symptoms status).
STEP7: Reshape the input data to match the format of the model.
STEP8: Use the trained model to predict the disease for this input.

4.4.2 Pseudo Code

define route ”/about” AS about:
return template(”about.html”)
16 define route ”/contact” AS contact:
return template(”contact.html”)
define route ”/developer” AS developer:
return template(”developer.html”)
define route ”/blog” AS blog:
return template(”blog.html”)

4.4.3 Data Set / Generation of Data (Description only)

For the “Health Advisor SystemUsing Machine Learning Techniques” the dataset will consist of a diverse combination of demographic, health and socio economic information. The data will include both numerical and categorical features to enable the ML models to make accurate and personalized predictions. Rare dis eases can be advised through the datasets which are having the recent information provided with the data sets. Combining the predictions of multiple models trained on different datasets this can help improve the overall performance and robustness of the system. Obtaining datasets that specifically focus on rare diseases or symptoms. This can help the model learn from a more diverse range of cases and improve its performance on rare diseases or symptoms. Utilize external knowledge bases (e.g., Clinical guidelines) to augment training data. The descriptions and the medications were described on the basis of the given disease. The detailed description of the dataset is given above. The dataset consists of the information of the data about different diseases, workouts and the required information. The sets are taken from Kaggle website which is available on internet. It contains rows and columns of the symptoms. As told a dataset is the collection of data which is used for analysis and modelling and typically organized in a structured format. Together, this rich and varied dataset will enable the ML model to generate precise and comprehensive health predictions.

4.5 Module Description

4.5.1 Module1

The Data Collection Module is responsible for gathering, managing and preparing the necessary datasets required for building and training ML models aimed at pre dicting health insurance claims. This module plays a critical role in ensuring that the data used for the prediction process is relevant, accurate and comprehensive. The data can come from multiple sources, including patient records, disease databases, healthcare providers, or external datasets

4.5.2 Module2

This module include data preprocessing pipelines for cleansing and harmoniz ing heterogeneous data sources, model deployment frameworks for hosting machine learning models in scalable and secure environments, and decision support interfaces for presenting healthcare providers in an intuitive and interpretable manner. It en sures the seamless integration of structured and unstructured data while maintaining data privacy and integrity. This module presents the design and implementation of an integration module for health advisor system using machine learning. The mod ule serves as a bridge between disparate data sources, machine learning models, and clinical decision-making processes, facilitating the translation of predictive insights into actionable recommendations.

4.5.3 Module3

The RFalgorithm is an effective ML model for predicting patient health insurance status. It works by creating an ensemble of decision trees, each trained on a different subset of the dataset. This technique, known as bagging, reduces overfitting and improves the model's generalization to new data. In the context of predicting health insurance, RF takes into account various features, such as demographic factors (age, gender, education), socioeconomic status (income, employment) and health-related data (chronic conditions, number of doctor visits).

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

Sections to input details about the patient's medical history, including past diag noses, existing medical conditions, and any previous treatments or surgeries. Areas for describing the patient's current symptoms, complaints, or reasons for seeking medical advice. Areas for the patient to express their treatment preferences, goals, and concerns, ensuring a patient-centered approach to care. This GUI is designed using the package called Flask in Python. Few HTML pages were designed, describ ing about, blog, contacts and the other for output. The GUI contain the inputs for all the attributes in dataset. This system will detects patient' disease medication and the workouts, diets. This interface will useful tool for medical staff,normal people in the pandemic times and the uneven medical knowledge one's and the validation of the disease in patients. A picture of the developed GUI

5.1.2 Output Design

The output of the input given by the user which is describing about the disease description and the particular workouts, diets and the medication of the given disease and it is used to operate in the cloud based system. It will be available through a user based input system to work on easy with the help of the user input. It specifies the result of the challenges that addressed to be developed in a way to monitor the patient health and for the better health care to predict. It can be decorated through an best result with more accuracy and the precision percentage which having the best performance.

5.2 Testing

Testing involves the entire web page for the correctness of all operations in order to predict accurate results.Deploy the web page in a server or in the cloud so that it is available for access by users. Monitor the working of the page and troubleshoot issues and errors as they happen.Continue to update and improve the model and web page based on new information or methods. First, the dataset used for training the model must be verified for correctness, consistency and quality. Techniques such as cross-validation and train-test splitting are applied to divide the data into training and testing sets to prevent over fitting and to evaluate how well the model generalizes to unseen data.

5.3 Types of Testing

5.3.1 Unit testing

Input

```
import unittest
2 import pandas as pd
3 dataset = pd. read csv (    Training    . csv    )
4 dataset
5 # vals = dataset . values . flatten ()
6 dataset . shape
7 from sklearn . model selection import train
t e s t split
8 from sklearn . preprocessing import LabelEncoder
9 X = dataset .drop(    prognosis        , axis=1)
10 y = dataset [    prognosis       ]
11 # ecoding prognonsis
12 le = LabelEncoder ()
13 le . fit (y)
14 Y = le . transform (y)
15 X
t rain , X test , y train , y
t e st = train
t e s t split (X, Y, test size =0.3, random state=20)
16 from sklearn . datasets import make classification
17 from sklearn . model selection import train
18 from sklearn .svm import SVC
t e s t split
19 from sklearn . ensemble import RandomForestClassifier , GradientBoostingClassifier
20 from sklearn . neighbors import KNeighborsClassifier
21 from sklearn . naive bayes import MultinomialNB
```

```
from sklearn . metrics import accuracy score , confusion matrix
import numpy as np
# Create a dictionary to store models
models = {

   SVC    : SVC( kernel=    linear    ) ,
 20
      RandomForest    : RandomForestClassifier(n estimators=100, random state=42) ,
        GradientBoosting      : GradientBoostingClassifier(n estimators=100, random state=42) ,
       KNeighbors    : KNeighborsClassifier(n neighbors=5) ,
       MultinomialNB     : MultinomialNB()
}
# Loop through the models , train , test , and print results
for model name, model in models. items() :
# Train the model
model. fit (X train , y train)
# Test the model
predictions = model.predict (X test)
# Calculate accuracy
accuracy = accuracy score(y test , predictions)
print ( f  {model name} Accuracy: {accuracy}   )
# Calculate confusion matrix
cm= confusion matrix(y test , predictions)
print ( f  {model name} Confusion Matrix:   )
print (np. array2string(cm, separator=    ,    ))
print (   \ n   +    =   *40 +    \ n   )
# selecting svc
svc = SVC(kernel=  linear       )
svc. fit (X train ,y train)
ypred = svc.predict (X test)
accuracy score(y test ,ypred)
# save svc
import pickle
pickle.dump(svc ,open(  svc  .pkl     ,    wb   ))
# load model
svc = pickle. load(open(  svc  .pkl     ,    rb   ))
# test 1:
print (   predicted    disease :   ,svc.predict (X test . iloc[0].values . reshape(1, 1 )))
print (   Actual    Disease :   , y test [0])
# test 2:
print (   predicted    disease :   ,svc.predict (X test . iloc[100].values . reshape(1, 1 )))
print (   Actual    Disease :   , y test [100])
symdes = pd. read csv(  symtoms   df. c s v )
precautions = pd. read csv(   precautions    df . c s v )
workout = pd. read csv(  workout   df . c s v )
description = pd. read csv(    description    . c s v )
medications = pd. read csv(    medications    . c s v )
diets = pd. read csv(   diets    . c s v )
#============================================================
# custome and helping functions
```

```
70 #=========================helper funtions================
71 def helper(dis):
72 desc = description[description[ Disease ] == predicted disease][ Description ]
73 desc = . join([w for w in desc])
74 pre = precautions[precautions[ Disease ] == dis][[ Precaution 1 , Precaution 2 , Precaution 3
, Precaution 4 ]]
75 pre = [col for col in pre.values]
21
76 med = medications[medications[ Disease ] == dis][ Medication ]
77 med = [med for med in med.values]
78 die = diets[diets[ Disease ] == dis][ Diet ]
79 die = [die for die in die.values]
80 wrkout = workout[workout[ disease ] == dis] [ workout ]
81 return desc ,pre ,med,die ,wrkout
82 symptoms dict = { itching : 0, skin rash : 1, nodal skin eruptions : 2, neck pain : 63,
dizziness : 64, cramps : 65, bruising : 66, obesity : 67, swollen legs : 68,
swollen blood vessels : 69, puffy face and eyes : 70, enlarged thyroid : 71, brittle nails :
72, swollen extremeties : 73, excessive hunger : 74, extra marital contacts : 75,
drying and tingling lips : 76, slurred speech : 77, knee pain : 78, hip joint pain : 79,
muscle weakness : 80, stiff neck : 81, swelling joints : 82, movement stiffness : 83,
spinningmovements : 84, loss of balance : 85, unsteadiness : 86, weakness of one body side :
87, loss of smell : 88, bladder discomfort : 89, palpitations : 120, painful walking : 121,
pus filled pimples : 122, blackheads : 123, scurring : 124, skin peeling : 125,
silver like dusting : 126, small dents in nails : 127, inflammatory nails : 128, blister :
129, red sore around nose : 130, yellow crust ooze : 131}
83 diseases list = {15: Fungal infection , 4: Allergy , 16: GERD , 9: Chronic cholestasis , 14:
Drug Reaction , 33: Peptic ulcer diseae , 1: AIDS , 12: Diabetes , 17: Gastroenteritis , 6:
Bronchial Asthma , 23: Hypertension , 30: Migraine , 7: Cervical spondylosis , 32:
Paralysis (brain hemorrhage) , 28: Jaundice , 29: Malaria , 8: Chicken pox , 25:
Hypoglycemia , 31: Osteoarthristis , 5: Arthritis , 0: (vertigo) Paroymsal Positional
Vertigo , 2: Acne , 38: Urinary tract infection , 35: Psoriasis , 27: Impetigo }
84 # Model Prediction function
85 def get predicted value(patient symptoms):
```

```
86 input vector = np.zeros(len(symptoms dict))
87 for item in patient symptoms:
88 input vector[symptoms dict[item]] = 1
89 return diseases list [svc.predict ([ input vector ]) [0]]
90 symptoms = input( Enter your symptoms . . . . . )
91 user symptoms = [s. strip() for s in symptoms. split ( , )]
92 # Remove any extra characters , if any
93 user symptoms = [symptom. strip( [] ) for symptom in user symptoms]
94 predicted disease = get predicted value(user symptoms)
95 desc , pre , med, die , wrkout = helper(predicted disease)
96 print ( ==========predicted disease========== )
97 print (predicted disease)
98 print ( ==========description========== )
99 print (desc)
100 print ( ==========precautions========== )
101 i = 1
102 for p i in pre[0]:
103 print (i , : , p i)
104 i += 1
105 print ( ==========medications========== )
106 for m i in med:
107 print (i , : , m i)
108 i += 1
109 print ( ==========workout========== )
110 for w i in wrkout:
22
111
print ( i , : , w i)
112
i += 1
113 print ( ==========diets========== )
114 for d
115
i in die :
print ( i , : , d i )
116
117
i += 1
# let s use pycharm flask app
118 # but install this version
119 import sklearn
120 print ( sklearn .
version
)
```

Test result

5.3.2 Integration testing

Input

```
from flask import Flask , request , render template , jsonify # Import jsonify
 2 import numpy as np
 3 import pandas as pd
 4 import pickle
 5 # flask app
 6 app = Flask(
 name
 )
 7 # load databasedataset==================================
 8 sym des = pd. read csv( datasets /symtoms df . c s v )
 23
9 precautions = pd. read csv( datasets /precautions df . c s v )
 10 workout = pd. read csv( datasets /workout df . c s v )
 11 description = pd. read csv( datasets /description. c s v )
 12 medications = pd. read csv( datasets /medications . c s v )
 13 diets = pd. read csv( datasets /diets . c s v )
 14 # load model=========================================
 15 svc = pickle. load(open( models /svc.pkl , rb ))
 16 #============================================================
 17 # custome and helping functions
 18 #==========================helper funtions================
 19 def helper(dis):
 20 desc = description[description[ Disease ] == dis][ Description ]
 21 desc = . join([w for w in desc])
 22 pre = precautions[precautions[ Disease ] == dis][[ Precaution 1 , Precaution 2 ,
        Precaution 3
 , Precaution 4 ]]
 23 pre = [col for col in pre.values]
 24 med = medications[medications[ Disease ] == dis][ Medication ]
 25 med = [med for med in med.values]
 26 die = diets[diets[ Disease ] == dis][ Diet ]
 27 die = [die for die in die.values]
 28 wrkout = workout[workout[ disease ] == dis] [ workout ]
 29 return desc ,pre ,med,die ,wrkout
 30 symptoms dict = { itching : 0, skin r a s h : 1, nodal skin e r u p t i o n s : 2,
        continuous s n e e z i n g : 3,
    shivering : 4, chills : 5, joint p a i n : 6, stomach p a i n : 7, acidity : 8,
        ulcers on tongue
    : 9, muscle w a s t i n g : 10, vomiting : 11, burning m i c t u r i t i o n : 12, spotting
     u r i n a t i o n :
 13, fatigue : 14, weight g a i n : 15, anxiety : 16, cold hands and f e e t s : 17,
     mood s w i n g s :
 18, weight l o s s : 19, restlessness : 20, lethargy : 21, patches in throat :
    22,
```

```
irregular sugar level : 23, cough : 24, high fever : 25, sunken eyes : 26, breathlessness :
27, sweating : 28, dehydration : 29, indigestion : 30, headache : 31, yellowish skin : 32,
dark urine : 33, nausea : 34, loss of appetite : 35, pain behind the eyes : 36, back pain :
37, constipation : 38, abdominal pain : 39, diarrhoea : 40, mild fever : 41, yellow urine
: 42, yellowing of eyes : 43, acute liver failure : 44, fluid overload : 45,
swelling of stomach : 46, swelled lymph nodes : 47, malaise : 48,
blurred and distorted vision : 49, phlegm : 50, throat irritation : 51, redness of eyes : 52,
sinus pressure : 53, runny nose : 54, congestion : 55, chest pain : 56, weakness in limbs
: 57, fast heart rate : 58, pain during bowel movements : 59, pain in anal region : 60,
bloody stool : 61, irritation in anus : 62, neck pain : 63, dizziness : 64, cramps : 65,
bruising : 66, obesity : 67, swollen legs : 68, swollen blood vessels : 69,
puffy face and eyes : 70, enlarged thyroid : 71, brittle nails : 72, swollen extremeties :
73, excessive hunger : 74, extra marital contacts : 75, drying and tingling lips : 76,
slurred speech : 77, knee pain : 78, hip joint pain : 79, muscle weakness : 80, stiff neck :
81, swelling joints : 82, movement stiffness : 83, spinningmovements : 84, loss of balance
: 85, unsteadiness : 86, weakness of one body side : 87, loss of smell : 88,
bladder discomfort : 89, foul smell of urine : 90, continuous feel of urine : 91,
passage of gases : 92, internal itching : 93, toxic look (typhos) : 94, depression : 95,
irritability : 96, muscle pain : 97, altered sensorium : 98, red spots over body : 99,
belly pain : 100, abnormal menstruation : 101, dischromic patches : 102, watering from eyes
: 103, increased appetite : 104, polyuria : 105, family history : 106, mucoid sputum : 107,
rusty sputum : 108, lack of concentration : 109, visual disturbances : 110}
31 # Model Prediction function
24
32 def get predicted value(patient symptoms):
33 input vector = np.zeros(len(symptoms dict))
34 for item in patient symptoms:
35 input vector[symptoms dict[item]] = 1
36 return diseases list [svc.predict ([ input vector ]) [0]]
37 # creating routes========================================
38 @app. route( / )
39 def index() :
40 return render template( index .html )
41 # Define a route for the home page
```

```
@app. route(    /predict       , methods=[    GET      ,     POST     ])
def home() :
if request .method ==     POST     :
symptoms = request .form.get(     symptoms      )
# mysysms = request .form.get(     mysysms      )
# print (mysysms)
print (symptoms)
if symptoms ==     Symptoms     :
message =     Please     either write symptoms or you have written misspelled symptoms
return render template(   index   .html    , message=message)
else:
# Split the users input into a list of symptoms (assuming they are comma separated )
user symptoms = [s. strip() for s in symptoms. split (     ,     )]
# Remove any extra characters , if any
user symptoms = [symptom. strip(    []          ) for symptom in user symptoms]
predicted disease = get predicted value(user symptoms)
dis des , precautions , medications , rec diet , workout = helper(predicted disease)
my precautions = []
for i in precautions[0]:
my precautions.append(i)
return render template(   index   .html      , predicted disease=predicted disease , dis des=
dis des , my precautions=my precautions ,medications=medications ,
my diet=rec diet ,workout=workout)
return render template(   index   .html      )
# about view funtion and path
@app. route(      /about      )
def about () :
return render template(   about     .html )
# contact view funtion and path
@app. route(      /contact      )
def contact () :
return render template(    contact      .html )
# developer view funtion and path
@app. route(      /developer      )
def developer() :
return render template(    developer      .html )
# about view funtion and path
@app. route(      /blog )
def blog() :
return render template(   blog    .html )
25
if name ==       main      :
app. run(debug=True)
```

Test result

5.3.3 System testing

Input

```
<!doctype html>
2<html lang= en >
3 <head>
4 <meta charset= utf8 >
5 <meta name= viewport content= width = d e v i c e width , i n i t i a l scale =1 >
6 <title>Health Care Center</title>
7 <link href= https ://cdn. jsdelivr .net /npm/bootstrap@5.3.1/ dist /css/bootstrap.min. c s s
rel=
s t y l e s h e e t i n t e g r i t y= sha3844bw +/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
PtmoHDEXuppvnDJzQIu9 c r o s s o r i g i n= anonymous >
8 </head>
9 <style>
10 . logo {
11 width: 50px;
12 height : 50px;
13 color : black;
14 m a r g i n top : 0;
15 m a r g i n left : 2px;
16 }
17 .myimg {
18 width: 50px;
19 height : 50px;
20 border: 2px solid black;
21 b o r d e r radius : 25px;
22 }
26
23 </style>
24 </head>
25<body>
26 <! Navbar >
27 <nav class= navbar n a v b a r expandlg n a v b a r dark b g dark >
28 <div class= containerfluid >
29 <! Logo at the t o p left corner >
30 <div class= logo >
31 <img class= myimg src= {{ url for( static , filename= img . p n g ) }} alt=
>
32 </div>
33 <a class= navbarbrand href= # >Health Center</a>
34 <button class= navbartoggler type= button d a t a bstoggle = collapse
d a t a bstarget = #
navbarSupportedContent a r i a controls = navbarSupportedContent a r i a expanded = false
a r i a label = Toggle n a v i g a t i o n >
35 <span class= navbartogglericon ></span>
36 </button>
```

```
37 <div class= collapse navbar collapse id= navbarSupportedContent >
38 <ul class= navbarnav me auto mb 2mblg0 >
39 <li class= navitem >
40 <a class= navlink active aria current = page href= # >Home</a>
41 </li>
42 <li class= navitem >
43 <a class= navlink href= /about >About</a>
44 </li>
45 <li class= navitem >
46 <a class= navlink href= /contact >Contact</a>
47 </li>
48 <li class= navitem >
49 <a class= navlink href= /developer >Developer</a>
50 </li>
51 <li class= navitem >
52 <a class= navlink href= /blog >Blog</a>
53 </li>
54 </ul>
55 <form class= dflex role= search >
56 <input class= formcontrol me 2 type= search placeholder= Search aria label
    =
Search >
57 <button class= btn btn outlinesuccess type= submit >Search</button>
58 </form>
59 </div>
60 </div>
61 </nav>
62 <! main form of page >
63<h1 class= mt4my4 text center text green >Health Care Center</h1>
64<div class= container my 4 mt 4 style= background : black; color : white;
    border radius : 15px;
padding: 40px; >
65 <form action= /predict method= post >
66 <div class= formgroup >
67 <label for= symptoms >Select Symptoms:</label>
27
68
<input type= text class= formcontrol , id= symptoms name= symptoms
    placeholder= type
systems such as itching , sleeping , aching etc >
69
70
71
72
73
74
75
76
77
78
```

```
79
80
81
82
83 </div>
</div>
<br>
<button type= button id= startSpeechRecognition class= btn b t n primary style=
    marginleft
: 3 px ; border :1px solid white ; border radius :20px; >
Start Speech Recognition
</button>
<br>
<! Display the transcribed text here >
<div name= mysysms id= transcription ></div>
{% if message %}
<p>{{ message }}</p>
{% endif %}
<br>
<button type= submit class= btn b t n danger b t n lg style= width : 100%; padding : 14
    px;
m a r g i n bottom : 5px; >Predict </button>
</form>
84 {% if predicted
85 <! Results >
disease %}
86 <h1 class= text center m y 4 m t 4 >Our AI System Results </h1>
87 <div class= container >
88
89
90
91
92
93
94
95
96
<div class= result container >
<! Buttons to toggle display >
<button class= toggle button data bstoggle = modal data bstarget = #
    diseaseModal style=
padding :4px; margin : 5px 40px 5px 0; f o n t size :20px; font weight : bold ; width:140px;
border radius :5px; background :#F39334; color : black ; >Disease </button>
<button class= toggle button data bstoggle = modal data bstarget = #
    descriptionModal
s t yle= padding :4px; margin : 5px 40px 5px 0; font size :20px; font weight : bold ; width
    :140px
;
border radius :5px; background:#268AF3 ; color : black ; >Description </button>
```

```
<button class= toggle button data bstoggle = modal data bstarget = #
    precautionModal style
= padding :4px; margin : 5px 40px 5px 0; font size :20px; font weight : bold ; width:140px;
border radius :5px; background :#F371F9 ; color : black ; >Precaution </button>
<button class= toggle button data bstoggle = modal data bstarget = #
    medicationsModal
s t yle= padding :4px; margin : 5px 40px 5px 0; font size :20px; font weight : bold ; width
    :140px
; border radius :5px; background :#F8576F ; color : black ; >Medications </button>
<button class= toggle button data bstoggle = modal data bstarget = #
    workoutsModal style=
padding :4px; margin : 5px 40px 5px 0; font size :20px; font weight : bold ; width:140px;
border radius :5px; background:#99F741 ; color : black ; >Workouts</button>
<button class= toggle button data bstoggle = modal data bstarget = #
    dietsModal style=
padding :4px; margin : 5px 40px 5px 0; font size :20px; font weight : bold ; width:140px;
border radius :5px; background :#E5E23D; color : black ; >Diets </button>
</div>
97 </div>
98 {% endif %}
99 <! Disease Modal >
100
101
<div class= modal f a d e id= diseaseModal tabindex= 1 aria labelledby =
        diseaseModalLabel aria
  hidden = true >
<div class= modaldialog >
28
102 <div class= modalcontent >
103 <div class= modalheader style= backgroundcolor : #020606; color :white; ><!
        Set
header background color inline >
104 <h5 class= modaltitle id= diseaseModalLabel >Predicted Disease</h5>
105 <button type= button class= btnclose d a t a bsdismiss = modal a r i a label =

Close ></button>
106 </div>
107 <div class= modalbody style= backgroundcolor : # m o d a l bodycolor ; ><!
    Set modal
body background color inline >
108 <p>{{ predicted disease }}</p>
109 </div>
110 </div>
111 </div>
112 </div>
113 <! Description Modal >
114 <div class= modal f a d e id= descriptionModal tabindex= 1 a r i a labelledby =
descriptionModalLabel a r i a hidden = true >
115 <div class= modaldialog >
116 <div class= modalcontent >
```

```
117 <div class= modalheader style= backgroundcolor : #020606; color :white; >
118 <h5 class= modaltitle id= descriptionModalLabel >Description </h5>
119 <button type= button class= btnclose d a t a bsdismiss = modal a r i a label =

Close ></button>
120 </div>
121 <div class= modalbody >
122 <p>{{ dis des }}</p>
123 </div>
124 </div>
125 </div>
126 </div>
127 <! Precaution Modal >
128 <div class= modal f a d e id= precautionModal tabindex= 1 a r i a labelledby =
    precautionModalLabel
    a r i a hidden = true >
129 <div class= modaldialog >
130 <div class= modalcontent >
131 <div class= modalheader style= backgroundcolor : #020606; color :white; >
132 <h5 class= modaltitle id= precautionModalLabel >Precaution </h5>
133 <button type= button class= btnclose d a t a bsdismiss = modal a r i a label =

Close ></button>
134 </div>
135 <div class= modalbody >
136 <ul>
137 {% for i in my precautions %}
138 <li>{{ i }}</li>
139 {% endfor %}
140 </ul>
141 </div>
142 </div>
143 </div>
144 </div>
29
145 <! Medications Modal >
146 <div class= modal f a d e id= medicationsModal tabindex= 1 a r i a labelledby =
medicationsModalLabel a r i a hidden = true >
147 <div class= modaldialog >
148 <div class= modalcontent >
149 <div class= modalheader style= backgroundcolor : #020606; color :white; >
150 <h5 class= modaltitle id= medicationsModalLabel >Medications </h5>
151 <button type= button class= btnclose d a t a bsdismiss = modal a r i a label =

Close ></button>
152 </div>
153 <div class= modalbody >
154 <ul>
155 {% for i in medications %}
156 <li>{{ i }}</li>
```

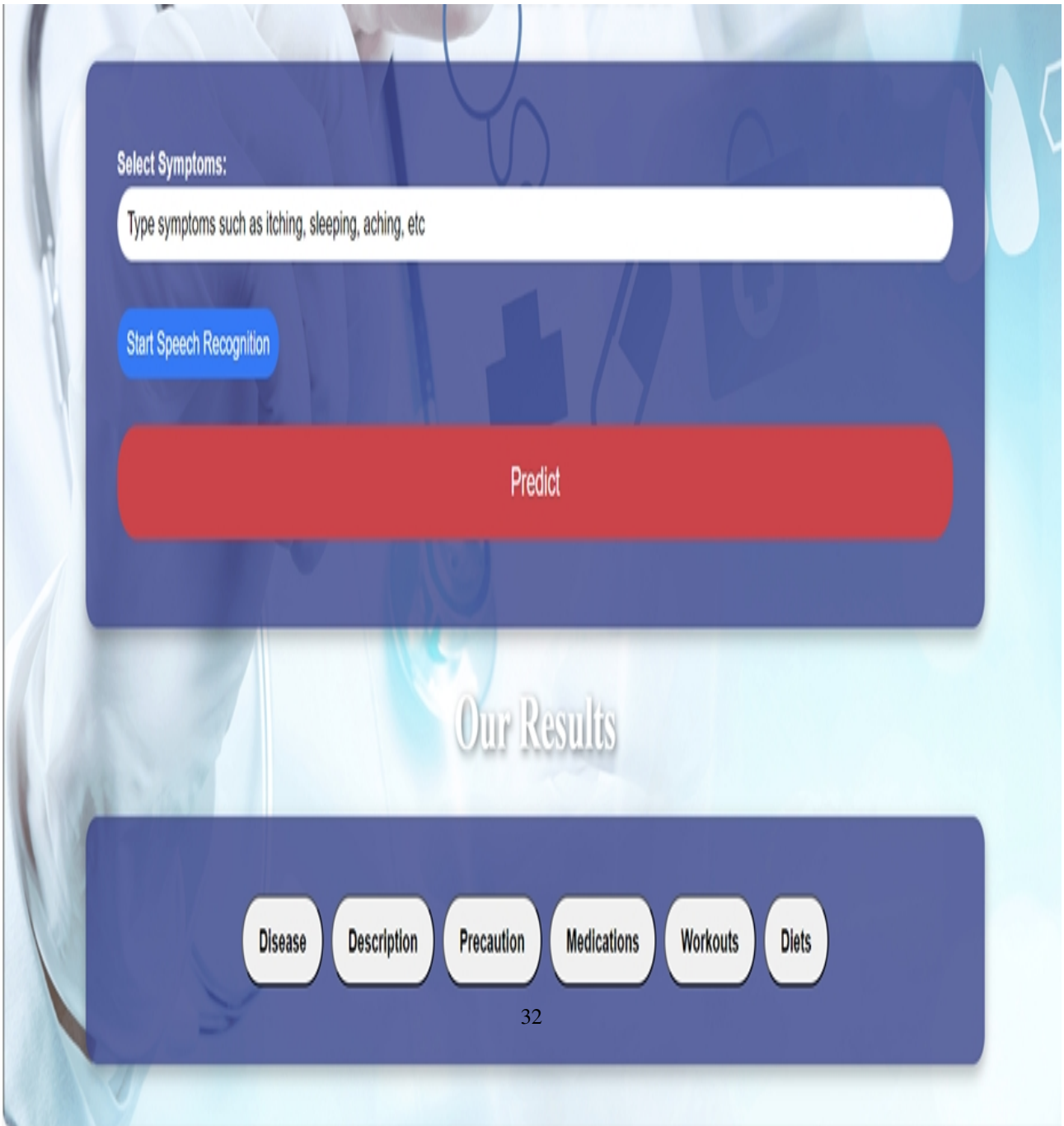
```
157 {% endfor %}
158 </ul>
159 </div>
160 </div>
161 </div>
162 </div>
163 <! Workouts Modal >
164 <div class= modal fade id= workoutsModal tabindex= 1 aria labelledby =
        workoutsModalLabel
 aria hidden = true >
165 <div class= modaldialog >
166 <div class= modalcontent >
167 <div class= modalheader style= backgroundcolor : #020606; color :white; >
const recognition = new webkitSpeechRecognition () ; // Use webkitSpeechRecognition for
compatibility
recognition . lang = enUS ; // Set the language for recognition
recognition . onresult = function ( event ) {
const result = event . results [0][0]. transcript ;
transcriptionDiv . textContent = result ;
};
recognition . onend = function () {
console . log ( Speech recognition ended . ) ;
};
recognition . start () ;
}
201 </script >
202
<script src= https ://cdn . jsdelivr . net /npm/ bootstrap@5 .3.1/ dist / js / bootstrap .
    bundle .min. js
integrity= sha384HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm
crossorigin= anonymous ></script>
203 </body>
204 </html>
```

Test Result

```
Enter your symptoms.......yellow_crust_ooze,red_sore_around_nose,small_dents_in_nails,inflammatory_nails,blist
================predicted disease============
Impetigo
================description==================
Impetigo is a highly contagious skin infection causing red sores that can break open.
================precautions==================
1 :  soak affected area in warm water
2 :  use antibiotics
3 :  remove scabs with wet compressed cloth
4 :  consult doctor
================medications==================
5 :  ['Topical antibiotics', 'Oral antibiotics', 'Antiseptics', 'Ointments', 'Warm compresses']
================workout==================
6 :  Maintain good hygiene
7 :  Stay hydrated
8 :  Consume nutrient-rich foods
9 :  Limit sugary foods and beverages
10 :  Include foods rich in vitamin C
11 :  Consult a healthcare professional
12 :  Follow medical recommendations
13 :  Avoid scratching
14 :  Take prescribed antibiotics
15 :  Practice wound care
================diets==================
16 :  ['Impetigo Diet', 'Antibiotic treatment', 'Fruits and vegetables', 'Hydration', 'Protein-rich foods']
```

Figure 5.1: **Architecture**

5.3.4 Test Result



Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed health advisor system employs sophisticated algorithms and machine learning to handle extensive patient data, genomic information and medical literature, thereby improving health care efficiency. It allows for specific and instant advice which helps in interventions that take place on time which is having the efficiancy. The prediction accuracy of each model for Drug Medication With an accuracy of over 90 percent, the Collaborative Filtering model is by far the best performing model,Gradient Boosting comes in second at about 65 percent. With an accuracy of just over 60 percent , the KNN model likewise performs admirably. SVC and Random Forest, on the other hand, have comparatively lower accuracy rates, at roughly 57 and 45 percent, respectively. Based on their accuracy results, this comparison aids in determining which model is best suited for precise Drugadvising. This enables the system to predict treatment outcomes more precisely, choose the best course of action, eliminate guesswork, avoid futile treatments and minimize drug side-effects. As a result, patients conditions become better with increased fullfilment that helps the human health to balance their daily routine. Enhancing compatibility with current hospital databases increases the ability of healthcare workers to make informed treatment choices, streamlines operations in the clinic, as well as improves the quality of healthcare provided to the patients thanks to individualized treatment and better results. Depending on the type of machine learning algorithm used, feature ranking, recursive feature removal, or correlation analysis may be employed. Training the data set was separated into two different sets namely train test set and test set. That would have the largest proportion, which is about 60–70 percentage, while the remainder, 30–40 percentage, was used for testing.Choosing an appropriate learning algorithm such as random forest algorithm, collaborative filtering or SVM that could generate health advice using multiple data techniques.Training with a selected model using the trained set of data.

6.2 Comparison of Existing and Proposed System

Sample attached

Existing system:(Decision tree)

The existing system of advisor system helped in recommending the medicines to be used, but it was not very accurate. It used the Decision Tree, which gave pre diction accuracy score of 96 percentage, which may lead to few wrong recommen dations. The existing system used a very little amount of data to classify the data i.e, they chose minimum information to build the model, which led to a low scalability. The existing system lacked the user-friendly user interfaces. The advantages of the decision tree are model is very easy to interpret we can know that the variables and the value of the variable is used to split the data. But the accuracy of decision tree in existing system gives less accurate output that is less when compared to proposed system.

Proposed system:(Random forest algorithm)

The proposed personalized medicine recommendation system integrates advanced technologies and data analytics to enhance patient interaction and com munication while being cost-effective by minimizing unnecessary medical tests and optimizing treatment plans. It employs machine learning, specifically a Support Vector Classifier (SVC) that achieves around 99 percent accuracy, alongside a Random Forest algorithm that also demonstrates high predictive performance. Both models analyze a comprehensive dataset of patient profiles, genetic markers, clinical history, and lifestyle factors to tailor treatment recommendations. Utilizing the Flask framework, the system features a user-friendly interface that presents clear, actionable recommendations along with confidence scores and detailed explanations, ensuring accessibility for users with varying levels of technological expertise.

```
import numpy as np
2 import pandas as pd
3 import pickle
4 # flask app
5 app = Flask(
name
)
6 # load databasedataset==================================
7 sym des = pd. read csv( datasets /symtoms df . c s v )
8 precautions = pd. read csv ( datasets / precautions df . c s v )
9 workout = pd. read csv ( datasets / workout df . c s v )
10 description = pd. read csv ( datasets / description . c s v )
```

```
11 medications = pd. read csv (      datasets / medications . csv     )
12 diets = pd. read csv (    datasets     / diets . c s v )
13 # load model==========================================
34
14 svc = pickle. load(open(   models   /svc.pkl      ,      rb     ))
15 #============================================================
16 # custome and helping functions
17 #=========================helper funtions================
18 def helper(dis):
19 desc = description[description[     Disease     ] == dis][       Description       ]
20 desc =          . join([w for w in desc])
21 pre = precautions[precautions[     Disease     ] == dis][[     Precaution     1     ,     Precaution     2     ,
          Precaution     3
,     Precaution     4     ]]
22 pre = [col for col in pre.values]
23 med = medications[medications[     Disease     ] == dis][       Medication       ]
24 med = [med for med in med.values]
25 die = diets[diets[     Disease     ] == dis][     Diet        ]
26 die = [die for die in die.values]
27 wrkout = workout[workout[     disease     ] == dis] [     workout     ]
28 return desc ,pre ,med,die ,wrkout
29 symptoms dict = {     itching     : 0,     skin     r a s h : 1,     nodal     s k i n     e r u p t i o n s : 2,
          continuous     s n e e z i n g : 3,
      shivering     : 4,     chills     : 5,     joint     p a i n : 6,     stomach     p a i n : 7,     acidity     : 8,
          ulcers     on tongue
    : 9,     muscle     w a s t i n g : 10,     vomiting     : 11,     burning     m i c t u r i t i o n : 12,     spotting
      u r i n a t i o n :
13,     fatigue     : 14,     weight     g a i n : 15,     anxiety     : 16,     cold     hands and     f e e t s : 17,
       mood     s w i n g s :
18,     weight     l o s s : 19,     restlessness     : 20,     lethargy     : 21,     patches     in throat     :
     22,
irregular sugar level     : 23,     cough     : 24,     high     f e v e r     : 25,     sunken     e y e s : 26,
          breathlessness     :
27,     sweating     : 28,     dehydration     : 29,     indigestion     : 30,     headache     : 31,
        yellowish     s k i n : 32,
   dark     u r i n e : 33,     nausea     : 34,     loss     of     a p p e t i t e : 35,     pain     behind the     e y e s : 36,
        back     p a i n :
37,     constipation     : 38,     abdominal     p a i n : 39,     diarrhoea     : 40,     mild     f e v e r     : 41,
       yellow     u r i n e
: 42,     yellowing     of     e y e s : 43,     acute     l i v e r     f a i l u r e : 44,     fluid     o v e r l o a d : 45,
swelling of s t o m a c h : 46,     swelled     lymph     n o d e s : 47,     malaise     : 48,
blurred and d i s t o r t e d     v i s i o n : 49,     phlegm     : 50}
30
31 # Model Prediction function
32 def get predicted value(patient symptoms):
33 input vector = np.zeros(len(symptoms dict))
34 for item in patient symptoms:
35 input vector[symptoms dict[item]] = 1
36 return diseases list [svc.predict ([ input vector ]) [0]]
37 # creating routes========================================
```

```
38 @app. route(    /    )
39 def index() :
40 return render template(  index   . h t m l )
41 # Define a route for the home page
42 @app. route(    /predict       , methods=[    GET      ,      POST     ])
43 def home() :
44 if request .method ==     POST     :
45 symptoms = request .form.get(     symptoms      )
46 # mysysms = request .form.get(     mysysms      )
47 # print (mysysms)
48 print (symptoms)
49 if symptoms ==     Symptoms     :
50 message =    Please     either write symptoms or you have written misspelled s y m p t o m s
51 return render template(  index   . h t m l   , message=message)
35
52
else :

# Split the user   s   input into a list of symptoms (assuming they are   c o m m a separated   )
user symptoms = [s. strip () for s in symptoms. split (       ,      )]
# Remove any extra characters , if any
user symptoms = [symptom. strip (    []          ) for symptom in user symptoms]
predicted
disease = get predicted
value ( user symptoms)
dis des , precautions , medications , rec diet , workout = helper ( predicted
my precautions = []
f or i in precautions [0]:
my precautions . append( i )
r eturn render template (      index . html      , predicted
dis des ,
disease=predicted
disease )
disease , dis des=
my precautions=my precautions , medications=
medications , my diet=rec diet , workout=workout)
r eturn render template (      index . html      )
64 # about view funtion and path
65 @app. route (      / about      )
66 def about () :
67
r eturn render template (  about     . h t m l )
68 # contact view funtion and path
69 @app. route (      / contact      )
70 def contact () :
71
r eturn render template (  contact     . h t m l )
72 # developer view funtion and path
73 @app. route (      / developer      )
74 def developer () :
```

```
103 75
104 r eturn render template ( developer . h t m l )
105 76 # about view funtion and path
106 77 @app. route ( / blog )
107 78 def blog () :
108 79
109 80 if
110 81
111 r eturn render template ( blog . h t m l )
112 name
113 ==
114 main
115 app . run (debug=True )
```

Output

```
Enter your symptoms.......itching,skin_rash,nodal_skin_eruptions
================predicted disease============
Fungal infection
================description=================
Fungal infection is a common skin condition caused by fungi.
================precautions=================
1 :  bath twice
2 :  use detol or neem in bathing water
3 :  keep infected area dry
4 :  use clean cloths
================medications=================
5 :  ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']
================workout=================
6 :  Avoid sugary foods
7 :  Consume probiotics
8 :  Increase intake of garlic
9 :  Include yogurt in diet
10 :  Limit processed foods
11 :  Stay hydrated
12 :  Consume green tea
13 :  Eat foods rich in zinc
14 :  Include turmeric in diet
15 :  Eat fruits and vegetables
================diets=================
16 :  ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']
```

Figure 6.1: **Output 1**

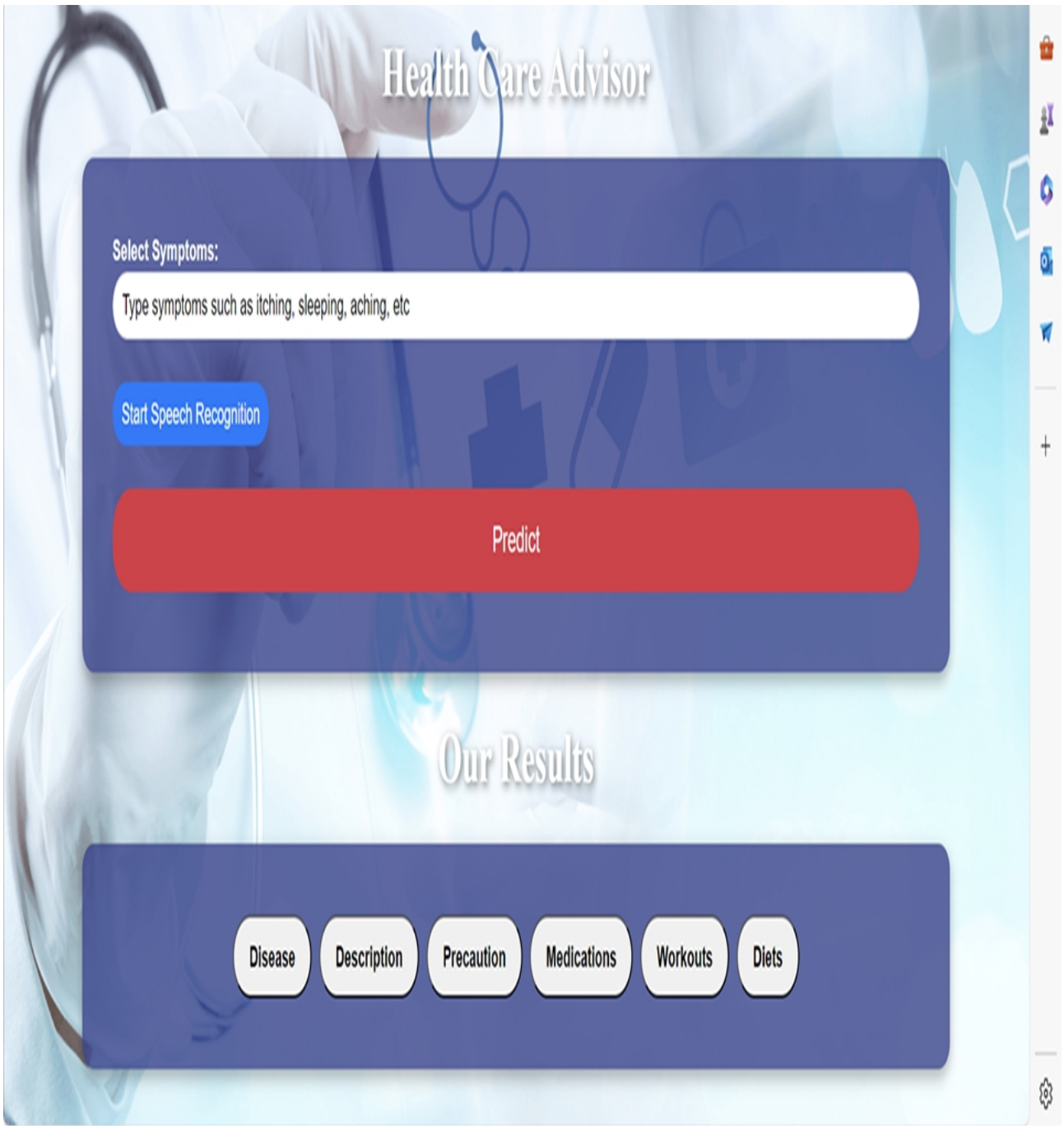


Figure 6.2: **Output 2**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Development and deployment of Health Advisor powered by a machine learning model can help revolutionize health-care delivery based on tailored treatment rec ommendations in view of patient features and medical needs, driving anoptimized decision-making, improved patient outcomes, and the efficiency of healthcare. The web page will thus act more like an interactive platform where the relevant infor mation, such as symptoms, could be fed by the user. It's then processed by the ML model on the page, which generates predictions from learned patterns in a labelled dataset. The five learning algorithms are used in work including Collaborative Fil tering Algorithm, Random Forest Algorithm, KNN, SVM, and Gradient Boosting Classifier. This model's accuracy scoring indicates how good model in use in pro ducingsystem predictions. Constructing a feedback loop meant to integrate practical outcomes and current actions made in this area. Working hand-in-hand with med-ical personnel to evaluate and improve updates of models. Taking into account the ultimate goal of this work, which is to modify or evolve a model's updates using ex-plainable AI. Emphasizing the growing negative repercussions of failing to respond to the evolving patterns of the patients' outcomes and progress in medicine. This results turned out best with an accuracy of 86 percentage for the Collaborative Fil-tering Algorithm. Compared to earlier studies, this experiment proved that the CFA is extremely effective. This would imply that the CFA is used to develop a GUI that all hospitals and medical staff can use as a kind of medical instrument. The web page and the ML model shall be updated and improved frequently in the future to improve the accuracy of Drug advise. Challenges to be addressed include data privacy concerns, interoperability issues, and equability in access of the data.

7.2 Future Enhancements

Future Enhancements Results the high-end systems that are to be developed for the advanced health advisors can be aimed at several areas to further strengthen the capabilities of the system and the efficiency of delivering healthcare. For example, it starts with the use of more advanced artificial intelligence techniques and method ologies, such as deep learning and reinforcement learning, in order to optimize the accuracy and strength of key predictions through these models. Indeed, deep learn ing and reinforcement learning proved to be promising approaches in dealing with complex, high-dimensional data providing good patterns and relationships within health care datasets. Further, data from real-time monitoring by wearable devices and IoT sensors representing the patients would be incorporated, which might introduce the possi bility of dynamic and adaptive recommendations when changes are observed in the patients' health status or their responses to the treatments. This would, in this regard, enable recommendation systems to provide proactive interventions and responses as treatments change over time to meet the dynamic needs of a patient through contin uous capture and analysis of patient-generated health data. The block chain tech nology can make health care data safer, more transparent, and traceable and resolve privacy issues and enable secure sharing and interoperability across different health systems. Blockchain-based solutions will allow patients to retain health records and have control of them while, at the same time, allowing permissible parties to access and add to care.

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagrism report**

Appendices

Appendix A

Complete Data / Sample Data / Sample Source Code / etc

```
<!doctype html>
2<html lang= en >
3 <head>
4 <meta charset= utf8 >
5 <meta name= viewport content= width = device width , initial scale =1 >
6 <title>Health Care Center</title>
7 <link href= https ://cdn. jsdelivr .net /npm/bootstrap@5.3.1/ dist /css/bootstrap.min. css
    rel=
stylesheet integrity= sha3844bw +/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
PtmoHDEXuppvnDJzQIu9 crossorigin= anonymous >
8 </head>
9 <style>
10 . logo {
11 width: 50px;
12 height : 50px;
13 color : black;
14 margin top : 0;
15 margin left : 2px;
16 }
17 .myimg {
18 width: 50px;
19 height : 50px;
20 border: 2px solid black;
21 border radius : 25px;
22 }
23 </style>
24 </head>
25<body>
26 <! Navbar >
27 <nav class= navbar navbar dark bg dark >
28 <div class= containerfluid >
29 <! Logo at the top left corner >
30 <div class= logo >
31 <img class= myimg src= {{ url for( static , filename= img .png ) }} alt=
    >
32 </div>
33 <a class= navbarbrand href= # >Health Center</a>
```

```
34 <button class= navbartoggler type= button d a t a bstoggle = collapse
    d a t a bstarget = #
navbarSupportedContent a r i a controls = navbarSupportedContent a r i a expanded = false
 a r i a label = Toggle n a v i g a t i o n >
35 <span class= navbartogglericon ></span>
36 </button>
37 <div class= collapse n a v b a r collapse id= navbarSupportedContent >
43
38 <ul class= navbarnav m e auto m b 2mblg0 >
39 <li class= navitem >
40 <a class= navlink a c t i v e a r i a current = page href= # >Home</a>
41 </li>
42 <li class= navitem >
43 <a class= navlink href= / a b o u t >About</a>
44 </li>
45 <li class= navitem >
46 <a class= navlink href= / c o n t a c t >Contact</a>
47 </li>
48 <li class= navitem >
49 <a class= navlink href= / d e v e l o p e r >Developer</a>
50 </li>
51 <li class= navitem >
52 <a class= navlink href= / b l o g >Blog</a>
53 </li>
54 </ul>
55 <form class= dflex role= search >
56 <input class= formcontrol m e 2 type= search placeholder= Search a r i a label
    =
 S e a r c h >
57 <button class= btn b t n outlinesuccess type= submit >Search</button>
58 </form>
59 </div>
60 </div>
61 </nav>
62 <! main form of page >
63<h1 class= mt4my4 t e x t center t e x t green >Health Care Center</h1>
64<div class= container m y 4 m t 4 style= background : black; color : white;
    b o r d e r radius : 15px;
padding: 40px; >
65 <form action= / p r e d i c t method= post >
66 <div class= formgroup >
67 <label for= symptoms >Select Symptoms:</label>
68 <input type= text class= formcontrol , id= symptoms name= symptoms
    placeholder= type
systems such as itching , sleeping , aching e t c >
69 </div>
70 <br>
71 <button type= button id= startSpeechRecognition class= btn b t n primary style=
        marginleft
:3px;border:1px solid white; b o r d e r radius :20px; >
```

```
81  72 Start Speech Recognition
82  73 </button>
83  74 <br>
84  75 <!        Display the transcribed text here        >
85  76 <div name=    mysysms     id=      transcription      ></div>
86  77 {% if message %}
87  78 <p>{{ message }}</p>
88  79 {% endif %}
89  80 <br>
90  81 <button type=    submit     class=  btn    b t n danger    b t n lg   style=  width   : 100%; padding:
        14px;
91   m a r g i n bottom : 5px;    >Predict </button>
92  82 </form>
93  44
94 83 </div>
95   disease %}
96  84 {% if predicted
97  85 <!        Results        >
98  86 <h1 class=  text      center     m y 4    m t 4 >Our AI System Results </h1>
99  87 <div class=     container     >
100
101 <div class=  result          container     >
102 <!        Buttons to toggle display        >
103  {% endif %}
104 99 <!        Disease Modal        >
105 100
106 101
107 102
108 103
109 104
110 105
111 106
112 107
113 108
114 109
115 110
116 111
117 112
118 113
119 114
120 <div class=  modal    f a d e   id=     diseaseModal       tabindex=     1       aria     labelledby    =
            diseaseModalLabel        aria
121    hidden   =     true     >
122 <div class=       modaldialog       >
123 <div class=       modalcontent       >
124 <div class=       modalheader        style=     backgroundcolor       : #020606; color : white ;    > <!
        Set
125  header background color inline        >
126 <h5 class=    modal     title     id=      diseaseModalLabel     >Predicted Disease </h5>
```

```
<button type= button class= btnclose data bsdismiss = modal aria label =

Close ></button>
</div>
<div class= modalbody style= backgroundcolor : # modal bodycolor ; > <!
    Set modal
body background color inline >
<p>{{ predicted
</div>
</div>
</div>
</div>
<! Description Modal >
disease }}</p>
<div class= modal fade id= descriptionModal tabindex= 1 aria labelledby =
descriptionModalLabel aria hidden = true >
115
<div class= modaldialog >
45
116 <div class= modalcontent >
117 <div class= modalheader style= backgroundcolor : #020606; color :white; >
118 <h5 class= modaltitle id= descriptionModalLabel >Description </h5>
119 <button type= button class= btnclose data bsdismiss = modal aria label =

Close ></button>
120 </div>
121 <div class= modalbody >
122 <p>{{ dis des }}</p>
123 </div>
124 </div>
125 </div>
126 </div>
127 <! Precaution Modal >
128 <div class= modal fade id= precautionModal tabindex= 1 aria labelledby =
        precautionModalLabel
    aria hidden = true >
129 <div class= modaldialog >
130 <div class= modalcontent >
131 <div class= modalheader style= backgroundcolor : #020606; color :white; >
132 <h5 class= modaltitle id= precautionModalLabel >Precaution </h5>
133 <button type= button class= btnclose data bsdismiss = modal aria label =

Close ></button>
134 </div>
135 <div class= modalbody >
136 <ul>
137 {% for i in my precautions %}
138 <li>{{ i }}</li>
139 {% endfor %}
140 </ul>
```

```
141 </div>
142 </div>
143 </div>
144 </div>
145 <!        Medications Modal        >
146 <div class= modal fade id= medicationsModal tabindex= 1 aria labelledby =
medicationsModalLabel aria hidden = true >
147 <div class= modaldialog >
148 <div class= modalcontent >
149 <div class= modalheader style= backgroundcolor : #020606; color :white; >
150 <h5 class= modaltitle id= medicationsModalLabel >Medications </h5>
151 <button type= button class= btnclose data bsdismiss = modal aria label =

Close ></button>
152 </div>
153 <div class= modalbody >
154 <ul>
155 {% for i in medications %}
156 <li>{{ i }}</li>
157 {% endfor %}
158 </ul>
159 </div>
160 </div>
46
161 </div>
162 </div>
163 <!  Workouts  Modal  >
164 <div class= modal fade id= workoutsModal tabindex= 1 aria labelledby =
workoutsModalLabel
aria hidden = true >
165 <div class= modaldialog >
166 <div class= modalcontent >
167 <div class= modalheader style= backgroundcolor : #020606; color :white; >
168 <h5 class= modaltitle id= workoutsModalLabel >Workouts </h5>
169 <button type= button class= btnclose data bsdismiss = modal aria label =

Close ></button>
170 </div>
171 <div class= modalbody >
172 <ul>
173 {% for i in workout %}
174 <li>{{ i }}</li>
175 {% endfor %}
176 </ul>
177 </div>
178 </div>
179 </div>
180 </div>
181 <!        Diets Modal        >
```

```
182 <div class= modal fade id= dietsModal tabindex= 1 aria labelledby =
        dietsModalLabel aria
183 hidden= true >
184 <div class= modaldialog >
185 <div class= modalcontent >
186 <div class= modalheader style= backgroundcolor : #020606; color :white; >
187 <h5 class= modaltitle id= dietsModalLabel >Diets </h5>
188 <button type= button class= btnclose data bsdismiss = modal aria label =
    Close ></button>
189 </div>
190 <div class= modalbody >
191 <ul>
192 {% for i in my diet %}
193 <li>{{ i }}</li>
194 {% endfor %}
195 </ul>
196 </div>
197 </div>
198 </div>
199 </div>
200 <script>
201 const startSpeechRecognitionButton = document .getElementById( startSpeechRecognition );
202 const transcriptionDiv = document .getElementById( transcription );
203 startSpeechRecognitionButton.addEventListener( click , startSpeechRecognition);
204 function startSpeechRecognition() {
205 const recognition = new webkitSpeechRecognition() ; // Use webkitSpeechRecognition for
    compatibility
47 recognition. lang = enUS ; // Set the language for recognition
206
r ecognition . onresult = function ( event ) {
215 </script >
216
217 </body>
218 </html>
```

References

[1] Ayushi Sharma and Rakesh Kumar (2023)Artificial Intelligence in Health Care Sector and Future Scope2023 International Conference on 2023 Innovative Data Communication Technologies and Application (ICIDCA). IEEE (2023) pp. 1-5.

[2] Castiglione F, Nardini C, Onofri E, Pedicini M, Tieri P (2023). Explainable drug repurposing approach from biased random walks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 20(2), 341-353.

[3] Daliya V K, Ramesh T K, Seok-BumKo (2021). An Optimised Multivariable Regression Model for Predictive Analysis of Diabetic Disease Progression. IEEE Access, 9,Pp : 1-13.

[4] Herrero P, Pesl P, Reddy M, Oliver N, Georgiou Toumazou C (2020). Advanced Insulin Bolus Advisor Based on Run-To-Run Control and Case-Based Reason ing. IEEE Journal of Biomedical and Health Informatics, 19(3), 651-659.

[5] Hajjej,F.Javeed, M. Ksibi, A. Alarfaj, M. Alnowaiser, K.Jalal, A. Alsufyani, N. Shorfuzzaman, Park, J. (2022). Deep human motion detection and multi-features analysis for smart healthcare learning tools. IEEE Access, 10, 1-13.

[6] Hakilo Sabit, Md. Shafiul Azam, Md Jahan Ali, Md. Deloar Hos sain(2020).Drugs Rating Generation and Recommendation from Sentiment Analysis of Drug Reviews using Machine Learning. 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)(IEEE) pp. 1782-1820.

[7] H. Wu, T. Wiesner-Hanks, E. L. Stewart, C. DeChant, N. Kaczmar, M. A. Gore, R. J. Nelson, and H. Lipson, “Autonomous detection of plant disease symptoms directly from aerial imagery,” Plant Phenome J., vol. 2,no. 1, pp. 1–9, Jan. 2022.

[8] J. Sun, Y. Yang, X. He, and X. Wu, "Northern maize leaf blight detection under complex field environment based on deep learning," IEEE Access, vol. 8, pp. 33679–33688, 2024.

[9] J. P. Gupta, A. Singh and Ravi Kanth Kumar (2021).A Computer-Based Dis ease Prediction And Medicine Recommendation System Using Machine Learn ing Approach. International Journal of Advanced Research in Engineering and Technology (IJARET) Vol.12, No. 03 (2021)Pp : 673-683.

[10] Kondaveeti H K, Dalai A K, Selavaraj P Reddy G V (2021). Habit tracker and advisor using mobile application for health monitoring. In 2021 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5).

[11] . Zhang, F. Wang, X. Zhao, and Z. Liu, "A hybrid deep learning framework for real-time detection of crop diseases using UAV imagery," Comput. Electron. Agric., vol. 210, p. 107874, Apr. 2022.

[12] Murumkar A D, Singh A, Chachar B R, Bagade, P D, Zaware G (2023). Artifi cial intelligence (AI) based nutrition advisor using an app. In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS) (pp. 1-5).

[13] M. T. Ahmed, H. Li, and J. Song, "Application of deep reinforcement learning for precision agriculture: soil moisture prediction and irrigation control," Expert Syst. Appl., vol. 212, p. 118734, Jul.

[14] Yoon J, Davtyan C, van der Schaar M (2020). Discovery and Clinical Decision Support for Personalized Healthcare. IEEE Journal of Biomedical and Health Informatics, 21(4), 831-843.

[15] Y. Xie, et al (2021) A deep learning model integrating mutation and transcription features for cancer survival prediction IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 9, pp. 2535-2545, 2021