Health Care Prediction System

BACHELOR OF COMPUTER APPLICATION (BCA)

SUBMITTED BY: ASHISH KOTHARI, ANUJ NEGI, ADITYA SINGH RAWAT

ENROLMENT NO: PV-22210283, PV-22041712, PV-22210087

Roll No: 17, 10, 4

Course: BCA

Section: D1

Batch-2022-2025

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

Synopsis

1. Introduction

The **Health Care Prediction System** is a web-based application designed to predict possible diseases based on user symptoms using machine learning models. The system allows users to input symptoms manually or through speech recognition, and then leverages trained AI models to provide health insights including the predicted disease, its description, precautions, recommended medications, workouts, and diets. This platform serves as an initial diagnostic tool and health information system, helping users take preliminary actions before consulting a medical professional.

---

2. Objective

- To build an intelligent web application that predicts diseases based on user-reported symptoms.
- To offer a seamless user experience through a clean interface and speech recognition integration.
- To provide additional health recommendations such as precautions, medications, workouts, and diets.
- To use **Flask** as the backend framework and integrate a trained **machine learning model** for prediction.

---

3. Technologies Used

Frontend:

- HTML5, CSS3
- JavaScript
- Bootstrap

Backend:

- Python
- Flask Web Framework

Machine Learning:

- Pickle

- Pandas, NumPy for data handling.
- Trained classification model: Support Vector Classifier, Random Forest Classifier, Gradient Boosting.

Other Features:

- Speech Recognition using the speech recognition Python library.
- CSV-based symptom-disease dataset.

---

4. System Architecture

1. **User Interface**: Accepts symptoms input through text or speech.
2. **Flask Backend**: Receives the input and processes it.
3. **ML Model**: Predicts the disease based on symptoms.
4. **Response Generation**: Fetches related information (description, medications, precautions, etc.) from pre-structured datasets.
5. **Results Display**: Outputs the prediction and advice on the frontend in user-friendly format.

---

5. Workflow

1. User navigates to the homepage and enters symptoms.
2. Optionally, the user can use speech input to describe symptoms.
3. Upon submission, the system sends the symptoms to the Flask server.
4. The server passes the symptoms through the ML model.
5. The model predicts the most probable disease.
6. The system retrieves and displays:
    - Disease Name
    - Description
    - Precautions
    - Medications
    - Workouts
    - Diet suggestions

---

6. Dataset

- A dataset containing diseases and their associated symptoms.
- Each entry includes:
  - Symptoms
  - Disease name
  - Precautions
  - Description
  - Medications
  - Workouts and diet

---

7. Modules

- **Symptom Input Module**
- **Speech Recognition Module**
- **Disease Prediction Engine**
- **Health Advice Generator**
- **Results Display Interface**
- **Admin/Developer Info and Contact Pages**

---

8. Features

- Predict diseases based on multiple symptoms.
- Voice recognition input.
- UI for non-technical users.
- AI-driven backend processing.
- Personalized health recommendations.

---

9. Future Enhancements

- Connect with doctors for online consultation.

- Real-time chatbot for health queries.

---

10. Conclusion

This project provides an innovative approach to preliminary healthcare diagnostics using AI and machine learning. It empowers users with quick insights into their health status and promotes health awareness. The project is scalable and can evolve into a complete digital healthcare assistant.

Bibliography / References

1. **Scikit-learn Developers.**
   Scikit-learn: Machine Learning in Python.
   URL: https://scikit-learn.org/

2. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011).**
   Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

3. **Breiman, L. (2001).**
   Random Forests. Machine Learning, 45(1), 5–32.
   DOI: https://doi.org/10.1023/A:1010933404324

4. **Friedman, J. H. (2001).**
   Greedy Function Approximation: A Gradient Boosting Machine.
   Annals of Statistics, 29(5), 1189–1232.

5. **Cortes, C., & Vapnik, V. (1995).**
   Support-vector networks. Machine Learning, 20(3), 273–297.
   DOI: https://doi.org/10.1007/BF00994018

6. **Python Software Foundation.**
   Python Language Reference, version 3.x.
   URL: https://www.python.org/

7. **Kaggle Datasets.**
   Disease Symptom Prediction Dataset.
   URL: https://www.kaggle.com/

8. **World Health Organization (WHO).**
   URL: https://www.who.int/health-topics
   (Used for disease descriptions or precautionary info)

9. **Python Speech Recognition Library Documentation**
   URL: https://pypi.org/project/SpeechRecognition/

10. **Flask Documentation.**
    Flask Web Development Framework.
    URL: https://flask.palletsprojects.com/