Voice Assistant for Disease Diagnosis Using Machine Learning and Natural Language Processing

A PROJECT REPORT

Submitted by,

SMITHA REDDY S - 20191CCE0061

Under the guidance of,

Dr. SWATI SHARMA
Associate Professor

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

JUNE 2023

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project report **"Voice Assistant for Disease Diagnosis Using Machine Learning and Natural Language Processing"** being submitted by *"***Smitha Reddy S***"* bearing roll number *"***20191CCE0061***"* in partial fulfilment of requirement for the award of degree of Bachelor of Technology in **Computer Science and Engineering** is a bonafide work carried out under my supervision.

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. MD. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

Dr. SWATI SHARMA
Associate Professor
School of CSE&IS
Presidency University

DR. ZAFAR ALI KHAN N
HOD
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Voice Assistant for Disease Diagnosis Using Machine Learning and Natural Language Processing** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr Swati Sharma, Associate Professor, School of Computer Science & Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

	Smitha Reddy S 20191CCE0061

ABSTRACT

The use of voice assistants in healthcare has become increasingly popular due to their ability to provide remote and personalized care. The proposed idea is to develop a voice assistant model to predict acute diseases using the Random Forest algorithm. The system utilizes the Natural Language Processing (NLP) techniques to record and predict symptoms reported by patients. The Random Forest algorithm is used for feature selection and classification, and its performance is compared with other machine learning algorithms such as Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. The experimental results show that the Random Forest algorithm outperformed the other algorithms, achieving higher accuracy. The proposed model can be applied in the development of future healthcare systems that leverage voice assistant technology for improved disease detection and diagnosis. The study highlights the potential of voice assistants in remote areas, making healthcare more accessible and efficient for patients.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science & Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Dean **Dr. C. Kalaiarasan,** Professor **Dr. T K Thivakaran,** University Project-II In-charge, School of Computer Science & Engineering, Presidency University for rendering timely help for the successful completion of this project.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Co-Ordinators **Mr. Mrutyunjaya MS, Mr. Sanjeev P Kaulgud, Mr. Rama Krishna K and Dr. Madhusudhan MV**.

We are greatly indebted to our guide **Dr Swati Sharma, Associate Professor**, School of Computer Science & Engineering, Presidency University for her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Smitha Reddy S (20191CCE0061)

List of Figures

Table of Contents

Chapter 1

INTRODUCTION

1.1 Introduction

Artificial Intelligence (AI) is making significant changes in healthcare by offering novel ways to gather and analyze patient information, improve medical decision-making, and personalize medical treatments. One of the most promising AI applications in healthcare is the use of voice assistants, which are powered by machine learning algorithms and natural language processing (NLP), to improve healthcare delivery and patient outcomes. These voice assistants allow patients to communicate their symptoms, obtain medical information, and receive personalized medical advice through voice-based communication.

The use of voice assistants with machine learning algorithms in healthcare can enhance the accessibility, convenience, and efficiency of medical care. These can be particularly beneficial for individuals living in remote areas, where access to medical care may be limited due to geographical barriers, lack of healthcare facilities, and shortage of medical professionals. Moreover, the technology can alleviate the workload of healthcare professionals, allowing them to focus on critical cases and improving overall healthcare delivery.

Supervised Machine Learning (ML) models are increasingly being used in disease diagnosis to improve the accuracy of predictions and enhance patient outcomes. The models are trained on labeled datasets and make predictions based on input data. By analyzing large amounts of patient data, these models can identify patterns and risk factors that may not be immediately apparent to human clinicians. Additionally, they can continuously learn and adapt to new data, making them effective tools for disease diagnosis and prediction.

This paper explores the potential of voice assistant systems utilizing ML models and NLP techniques in disease prediction for remote areas healthcare development. It discusses the effectiveness of supervised machine learning models in accurately predicting diseases, leading to more effective interventions and treatments. Furthermore, it highlights the potential of OpenAI in unlocking the full potential of voice assistants in the medical field. Lastly, the paper offers recommendations for future research and development in supervised machine learning models for disease prediction.

1.1 Motivation

This project presents the development of a voice assistant to predict acute diseases, based on Artificial Intelligence and Natural Language Processing techniques. The motivation is driven by the goal of improving healthcare accessibility and empowering individuals to take control of their health. Access to quality healthcare is not always equitable, with many individuals facing barriers such as distance, cost, or lack of medical expertise. This project aims to address these challenges by creating a voice assistant that can accurately interpret spoken symptoms and provide reliable diagnostic suggestions, treatment, and diet recommendations. By leveraging advanced technologies like ML and NLP, the system aims to provide preliminary medical guidance, helping individuals make informed decisions about their health and seek appropriate medical care. The ultimate motivation is to improve healthcare outcomes in remote areas, enable early detection of diseases, and contribute to better health management for individuals, regardless of their location or medical knowledge.

1.2 Problem Statement

Diagnosing diseases poses challenges, particularly for those without medical expertise and to those who are in remote areas. Common approaches involve searching for symptoms online or seeking professional medical advice. However, these methods may not always yield accurate or timely results. To tackle this issue, this project focuses on developing an intelligent voice assistant that can accurately analyze the symptoms spoken by the user and provide diagnosis with reliable treatment options and diet recommendations. By using advanced technologies such as Machine Learning and Natural Language Processing, the voice assistant aims to improve the accessibility of disease diagnosis, empowering individuals to make informed healthcare decisions.

Chapter 2

LITERATURE SURVEY

The literature review holds immense importance for any project as it plays a pivotal role in shaping the project's direction and ensuring its success. It serves as a foundation of knowledge, providing insights into existing research, theories, and best practices in the field. It enables project teams to identify gaps in current knowledge, formulate research questions, and define objectives. By synthesizing and analyzing relevant literature, the review helps in developing a robust theoretical framework, informing methodology choices. The literature review ensures that the project builds upon existing knowledge, avoids redundancy, and leverages successful approaches, ultimately leading to a more effective and impactful project. It also provides a context for interpreting project findings and contributes to the overall credibility and validity of the project outcomes.

2.1 Literature Review

SI No	Papers	Published Year	Review
1.	Speech to text and text to speech recognition systems-Areview	2018	*Various text-to-speech (TTS) and speech-to-text (STT) conversion techniques are reviewed and it is concluded that Format Synthesis and Hybrid Machine Translation (HMT) works best respectively.*
2.	A Self-Diagnosis Medical Chatbot Using Artificial Intelligence	2018	*A medical chatbot using Artificial Intelligence is proposed to diagnose and provide basic details about diseases before consulting a doctor, in order to reduce healthcare costs and improve accessibility to medical knowledge.*

3.	Development of machine learning model for diagnostic disease prediction based on laboratory tests	2021	*A disease diagnostic system using light gradient boosting machine (LightGBM) and extreme gradient boosting (XGBoost) ML models and a Deep Neural Network (DNN) model with a F1-score of 81% and prediction accuracy of 92% for the five most common diseases.*
4.	IntelliDoctor – AI based Medical Assistant	2019	*An AI-based personal medical assistant known as IntelliDoctor is designed to analyze symptoms, predict medical conditions, generates treatments, tracks user's health activities, and displays periodic health reports.*
5.	Text Messaging Based Medical Diagnosis Using Natural Processing and Fuzzy Logic	2020	*A chatbot service is developed using fuzzy logic rules, Telegram bot Application Programming Interface (API), knowledge base, NLP and short message service (SMS).*
6.	Identification and Prediction of Chronic Diseases Using Machine Learning Approach	2022	*The suggested system delivers disease prognosis based on patient's symptoms by employing machine learning (ML) algorithms such as convolutional neural network (CNN) for automatic feature extraction and disease prediction and K-nearest neighbor (KNN).*

7.	Comparing different supervised machine learning algorithms for disease prediction	2019	*This paper offers a thorough analysis of the relative efficacy of various supervised machine learning algorithm models for disease prediction.*
8.	NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality	2018	*In this paper a human-level quality TTS model called NaturalSpeech is developed. It is demonstrated that the model achieves human-level quality with Comparative Mean Opinion Score (CMOS) evaluation.*

2.2 Methodologies

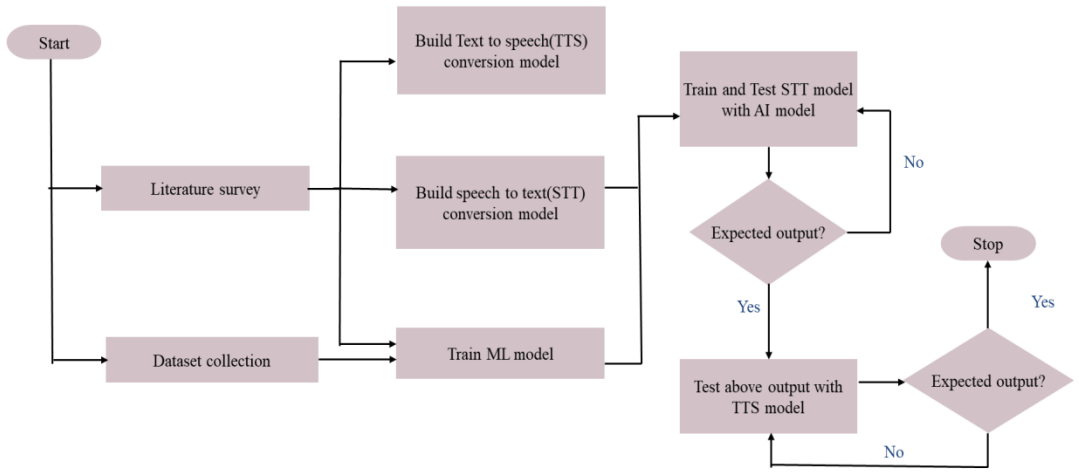

Fig 2.1: Methodology Chart of the Project

Phase 1 – Research

- Literature Survey / Research
    - Conducting a thorough review/research on material, articles, papers related to Machine Learning and Natural Language Processing.
- Dataset collection
    - Collecting relevant datasets to train and test the ML model.

Phase 2 – Development

In this phase the implementation of all the necessary steps for the project takes place. Such as,

- Data Preposing.
- Building Text to Speech (TTS) model.
- Building Speech to Text (STT) model.
- Building Machine Learning model.
- Developing functions to access the OpenAI models.

Phase 3 – Testing

- Training and testing the ML, TTS, STT, OpenAI models to check for accuracy and errors.
- Debugging and rectifying the errors to deliver a reliable system.

Chapter 3

PROPOSED SYSTEM

3.1 Introduction

In recent years, advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized various industries, including healthcare. The project outlines the intricacies of the system design, encompassing three key components: a voice assistant, a symptom checker, and a disease diagnosis module. These components work in tandem to provide users with an interactive and personalized diagnostic experience. Leveraging the capabilities of NLP, the voice assistant allows users to articulate their symptoms in a conversational manner, facilitating a more natural and intuitive interaction. The symptom checker, powered by ML algorithms, analyses the reported symptoms, and generates a list of potential diseases that match those symptoms. Subsequently, the disease diagnosis module utilizes additional data, such as medical history and test results, to refine the list and offer an accurate diagnosis.

3.2 Existing System Drawbacks

1. **Lack of Medical Expertise:** They may provide general information on health topics, but they do not have the knowledge or training to accurately diagnose a specific medical condition.

2. **Limited Questioning Capabilities:** General-purpose chatbots may not be able to ask the right questions to identify the underlying cause of a medical issue**.**

3. **Risk of Misdiagnosis:** Without the proper training and expertise, general-purpose chatbots may provide incorrect or misleading information. Misdiagnosis can lead to serious health consequences, and the liability for such errors would ultimately fall on the user or developer of the chatbot.

4. **Privacy Concerns:** Self-diagnosis of diseases involves sensitive and personal information, and there are concerns about how this data would be collected, stored, and used by general-purpose chatbots.

5. **Lack of Personalization:** General-purpose chatbots may not be able to provide personalized advice based on an individual's medical history, lifestyle, or other factors that could impact their health. This could lead to ineffective or even harmful advice.

3.3 Proposed System

In the proposed system (Fig 4.1), a disease prediction model is built using a Machine Learning algorithm (Random Forest), NLP techniques such as Text-to-Speech, Speech-to-Text and GPT platform like OpenAI. Based on the symptoms that are input by the user, the disease is predicted and the drug that is most prescribed by the doctor is suggested.

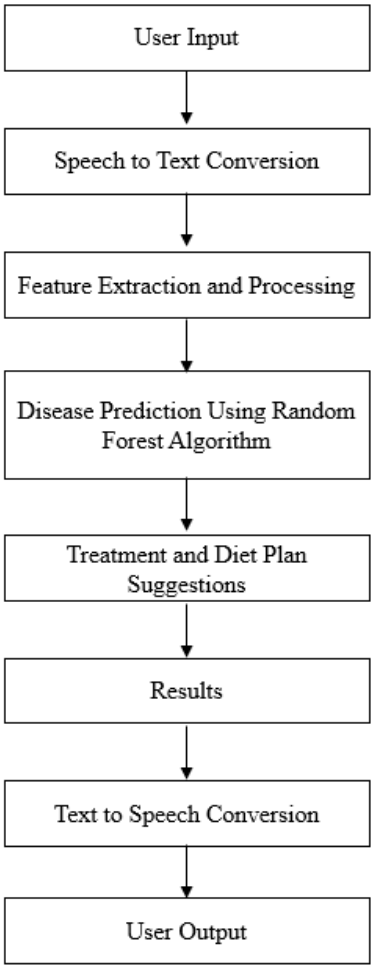

Fig 3.1: Architecture of proposed Voice Assistant for disease diagnosis system.

Speech-to-Text Model (STT):

This model converts spoken words into text, which allows the voice assistant to understand what the user is saying. It is used to translate the patients' symptoms into a digital format that can be processed and analyzed.

Text-to-Speech Model (TTS):

This model converts written text into spoken words. This allows the voice assistant to respond to the user in a natural and human-like way. The model communicates the predicted information or advice to patients in a way that is easy to understand and accessible.

Machine Learning Model:

It is used to analyze patients' data and identify patterns that may indicate the presence of disease. For example, the model is trained using a dataset that includes patient data. It can then use this data to predict the likelihood of a patient developing the disease based on their individual characteristics.

Generative Pre-Trained Transformer Functions:

These functions are used to get treatment options and diet recommendations responses from the OpenAI platforms GPT Models. The responses are according to the predicted disease.

3.4 Technology Details

The "Voice Assistant for Disease Diagnosis using ML and NLP" project leverages several cutting-edge technologies to achieve its objectives. These technologies include:

1. **Machine Learning (ML):** Machine learning algorithms are utilized to train the disease prediction model. ML algorithms are trained on comprehensive datasets containing symptom information and corresponding disease labels. This enables the voice assistant to accurately predict diseases based on the gathered symptoms.
2. **Natural Language Processing (NLP):** NLP techniques are employed to analyze and process spoken symptoms. These techniques enable the voice assistant to extract relevant information from the spoken input, classify symptoms based on severity and duration, and understand user queries and commands.
3. **Voice Recognition:** The project integrates voice recognition technology to convert spoken symptoms into text format. Voice recognition algorithms are employed to accurately transcribe spoken input, overcoming challenges related to diverse speech patterns, accents, and languages.
4. **OpenAI or Similar Platforms:** The voice assistant makes use of OpenAI or similar platforms to enhance its capabilities. These platforms provide access to pre-trained

models and natural language generation capabilities. By integrating OpenAI, the voice assistant can generate accurate treatment options and personalized diet recommendations based on the diagnosed disease.

5. **Software Development:** The project involves software development using programming languages such as Python, which is widely used in ML and NLP applications. Various libraries and frameworks, such as TensorFlow or PyTorch for ML, and NLTK or spaCy for NLP, may be utilized to streamline the development process.

4.5 Proposed System Advantages

1. Unlike Siri, Alexa, Google this provides accurate diagnosis based on the patient's symptoms as it is trained.

2. Can help healthcare professionals save time by analyzing the patient's data quickly and efficiently.

3. Provides necessary information on treatments and diet plans in accordance to predicted disease.

4. Can be used in remote areas where healthcare facilities are scarce, providing access to medical diagnosis and treatment information.

Chapter 4

OBJECTIVES

4.1 Project Objectives

1. Develop a voice recognition module: Build a voice recognition system that accurately converts spoken symptoms provided by the user into text format. This module will enable seamless interaction with the voice assistant.
2. Utilize Machine Learning (ML) algorithms for disease prediction: Train ML models using a comprehensive dataset of symptoms and diseases to predict the most likely disease based on the gathered symptoms. Explore and optimize ML algorithms such as decision trees, random forests, or deep learning neural networks to achieve accurate disease predictions.
3. Implement NLP for communication: Utilize NLP techniques to collect symptoms and communicate the predicted disease and relevant information to the user in a clear and understandable manner. This will involve generating human-readable diagnosis reports and explanations.
4. Integrate OpenAI for treatment options and diet recommendations: Utilize OpenAI or similar platforms to suggest treatment options based on the predicted disease. Additionally, leverage the power of OpenAI to generate personalized diet recommendations tailored to the user's specific health condition.
5. Ensure accuracy and reliability: Conduct thorough testing and validation to ensure the accuracy and reliability of the voice assistant system. Fine-tune and optimize the ML and NLP models to enhance the system's performance in disease prediction, diagnosis communication, treatment suggestion, and diet recommendation.
6. Give reliable medical facility in remote areas. Provide immediate and proper medication so that anyone with an Internet facility can access healthcare regardless of distance, place, education, etc.

By accomplishing these objectives, the project aims to build a comprehensive voice assistant system that accurately gathers symptoms through NLP, predicts diseases using ML algorithms, communicates diagnoses via NLP, suggests treatment options using OpenAI or similar platforms, and generates personalized diet recommendations. The system intends to provide reliable and accessible healthcare support, to those in need.

Chapter 5

REQUIREMENT ANALYSIS

5.1 Introduction

System requirements encompass the minimum and/or maximum hardware and software specifications necessary for a system or application to function effectively. To determine the system requirements for a specific system or application, it is crucial to carefully analyse its functional and non-functional requirements, considering any constraints or factors that may impact its performance. Requirement analysis primarily involves software requirements, hardware requirements, and functional requirements.

The key purposes of using system requirements are as follows:

1. Clearly and accurately defining the requirements for a system or application.
2. Serving as a reference point throughout the development process.
3. Facilitating communication and collaboration among team members.
4. Ensuring cost-effective development of the system or application.

Software requirements involve the process of collecting, understanding, evaluating, and documenting the requirements provided by the client. It is an iterative process consisting of several steps, including:

1. Requirements Elicitation: Gathering information about the needs and expectations of stakeholders for the software system.
2. Requirements Analysis: Analysing the gathered information to identify the high-level goals and objectives of the software system.
3. Requirements Specification: Documenting the identified requirements in a clear, consistent, and unambiguous manner.
4. Requirements Verification and Validation: Checking the completeness, consistency, and accuracy of the requirements.
    - Verification: Ensuring the software correctly implements specific functions.
    - Validation: Ensuring the built software aligns with customer requirements.

5. Requirements Management: Analysing, documenting, tracking, prioritizing, and agreeing on requirements, while controlling communication with relevant stakeholders.

Hardware requirements refer to the specifications of a hardware device. Integrating the software with hardware is essential for completing the development process. These requirements include factors such as minimum processor speed, memory, and disk space, as well as the need for a microphone, speaker, processor (i3 or later), RAM (4 GB or more), and network connectivity.

Functional requirements are related to the functional aspects of the software. They specify the functionality that the system must provide, such as input validation, data storage, and user interface. Functional requirements may involve calculations, data manipulation, technical details, and data processing.

Non-functional requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of. They specify the quality attributes of the system, such as performance, reliability, usability, and security.

5.2 Software and Hardware Requirements

Software Requirements: Listed below are the software requirements for performing time series analysis on the fraud data:

1. **Operating System:** Windows 8 and above (64 bit) operating system is required.
2. **Anaconda:** Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.
3. **Jupyter Notebook:** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text.
4. **Data Set:** Two datasets are used in this project, one for training and another for testing the model. Each dataset contains 132 common symptoms matched to 41 unique diseases.
5. **Voice Recognition Software:** A reliable and accurate voice recognition software or

library, such as Google Cloud Speech-to-Text, SpeechRecognition or Pyttsx3, to convert spoken symptoms into text and resultants into speech.

6. **Machine Learning (ML) Framework:** Employ an ML framework like scikit-Learn, TensorFlow or PyTorch to develop and train ML models for disease prediction based on symptoms.
7. **OpenAI Integration:** Integrate OpenAI or similar platforms to access their APIs and leverage their capabilities for treatment options and diet recommendation generation.

Hardware Requirements

1. Processor: Intel i3 and later
2. GPU (preferred): dedicated GPU from NVIDIA or AMD with 4GB VRAM
3. Network Connectivity: bandwidth ~ 10 Mbps 3 75 Mbps
4. Microphone
5. Speaker
6. RAM: 4G GB and above

5.3 Functional and Non-Functional Requirements

Functional Requirements

1. **Voice Recognition:** The system should accurately recognize and convert spoken symptoms provided by the user into text format. It should handle variations in speech patterns, accents, and languages to ensure reliable symptom gathering.

2. **Disease Prediction:** The system should utilize ML algorithms trained on a comprehensive dataset of symptoms and diseases to predict the most likely disease based on the gathered symptoms.

3. **Diagnosis Communication:** The system should utilize NLP techniques to communicate the predicted disease and relevant information to the user in a clear and understandable manner. It should generate human-readable diagnosis reports and explanations, providing users with a comprehensive understanding of their health condition.

4. **Treatment Option Suggestion:** The system should integrate OpenAI or similar platforms to suggest appropriate treatment options based on the predicted disease. It

should provide relevant and reliable treatment recommendations, considering factors such as medical guidelines, drug interactions, and individual preferences.

5. **Diet Recommendation:** The system should leverage OpenAI or similar platforms to generate personalized diet recommendations based on the user's specific health condition and the predicted disease. It should provide dietary guidance and suggestions tailored to the user's needs, including considerations for allergies, intolerances, and nutritional requirements.

Non-functional Requirements

1. **Performance:** The system should respond quickly and provide real-time feedback to ensure a seamless user experience.

2. **Accuracy:** The system's voice recognition module, NLP analysis, disease prediction, and diagnosis communication should achieve a high level of accuracy. The ML models used for disease prediction should be trained on reliable and up-to-date datasets to ensure accurate predictions.

3. **Security and Privacy:** The system should prioritize the security and privacy of user data, adhering to applicable data protection regulations. It should implement robust security measures, such as data encryption and access control, to safeguard sensitive user information.

4. **Compatibility:** The system should be compatible with commonly used platforms, operating systems, and devices to ensure broad accessibility.

Chapter 6

METHODOLOGY

6.1 Introduction

The methodology phase of the "Voice Assistant for Disease Diagnosis using ML and NLP" project marks a significant step towards bringing the envisioned solution to life. This phase involves translating the design and requirements into a functional software system that effectively analyzes spoken symptoms, predicts diseases, and provides accurate diagnosis reports, treatment options, and personalized diet recommendations. During the implementation phase, various technical components and processes will be involved. These include developing and integrating the voice recognition module to accurately convert spoken symptoms into text format. Machine learning algorithms will be utilized to train the disease prediction model on comprehensive datasets, enabling accurate disease identification based on the gathered symptoms.

The integration of OpenAI or similar platforms will enhance the system's capabilities, allowing it to provide reliable treatment options and generate personalized diet recommendations based on the diagnosed disease. This integration will leverage the power of artificial intelligence and natural language generation to deliver relevant and meaningful information to the users.

6.2 Design Procedure

The complete workflow is depicted in Figure 6.1, where the voice assistant captures the user's input query. The input text undergoes Natural Language Processing for text processing. After the processing, the chatbot identifies and classifies the symptoms specified by the user, leading to the prediction of the respective disease. Additionally, the chatbot can provide suggestions regarding food and physical activity to the user after predicting the disease.

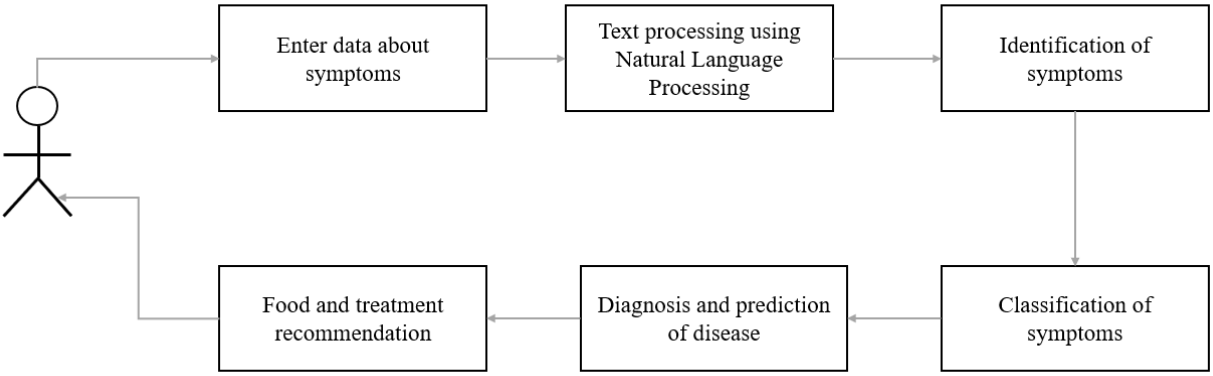

Fig 6.1: Architecture of the Proposed System

1. **Analyzing the problem statement & requirements:** Analyze the problem in terms of what we want to predict and what kind of observation data we must make those predictions. Predictions are generally a label or a target answer; it may be a yes/no label (binary classification) or a category (multiclass classification) or a real number (regression).

2. **Collect and clean the data:** Identify what kind of historical data we have for prediction modelling; the next step is to collect the data from datasets or from any other data sources.

3. **Prepare data for ML application:** Transform the data in the form that the Machine Learning system can understand.

4. **Train the model:** Before training the model, it is essential to split the data into training and evaluation sets, as we need to monitor how well a model generalizes to unseen data. Now, the algorithm will learn the pattern and mapping between the feature and the label.

5. **Evaluate and improve model accuracy:** Accuracy is a measure to know how well or bad a model is doing on an unseen validation set. Based on the current learning, evaluate the model on validation sets.

6. **Test the model:** Test the model on unknown data. After the system starts working properly, the model is complete.

6.3 Algorithm

Algorithms gives us the step-by-step procedure to build any project. It is used to set requirements before the actual implementation of code. The algorithm followed for the said project is given below:

Step 1: Import libraries and modules:

Initially, import the necessary libraries and modules, including the datasets module from sklearn, to make use of the built-in data.

Step 2: Load and pre-process the dataset:

Step 3: Evaluate machine learning models through train-test split:

We will utilize the train-test split procedure to assess machine learning models on standard classification and regression predictive datasets. Finally, we will apply the train-test split procedure to assess the performance of various machine learning models on standard classification and regression predictive modelling datasets.

Step 4: Compare models and select the final one based on accuracy.

Step 5: Set API key to access OpenAI.

Step 6: Set the engine for text-to-speech conversion.

Step 7: Define functions:

1. Create functions for speech-to-text conversion, text-to-speech conversion, generating a diet chart, and generating treatments.
2. Define a function for disease prediction.

Step 8: Define the main function:

1. Create a while loop to listen for the phrase "Hello doctor."
2. If the phrase is heard, prompt the user to enter their name.
3. Greet the user by name and ask, "How can I assist you?"
4. If the user requests symptom prediction, call the Predict_Symptoms() function.

5. Call the generate_treatment() function.
6. If the user requests a diet chart, call the Diet_chart() function.
7. If the user says "Stop" or "Bye," exit the loop.

Step 9: Call the main() function.

Step 10: End the program.

Random Forest Algorithm:

The Random Forest algorithm, belonging to the supervised learning technique, is a popular machine learning algorithm. It can be utilized for both Classification and Regression problems. It operates based on the concept of ensemble learning, which involves combining multiple classifiers to address complex problems and enhance model performance.

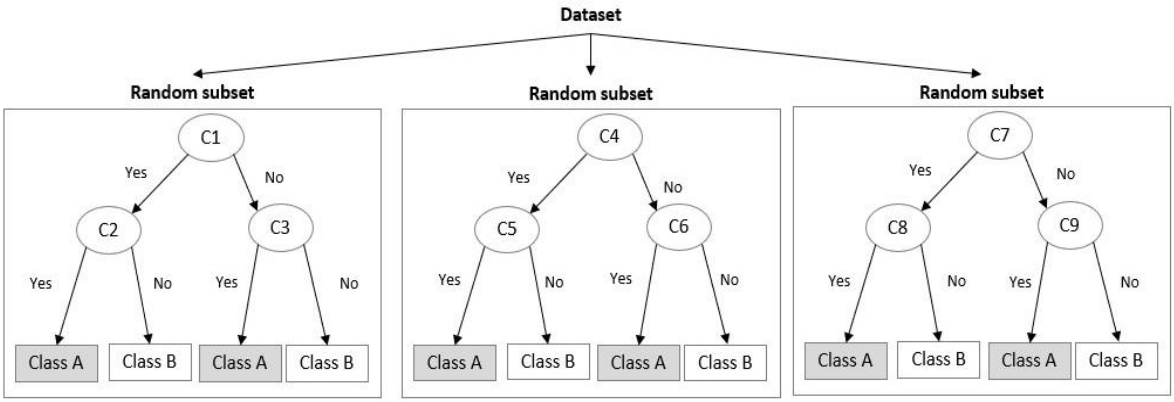

Fig 6.2: An illustration of a Random Forest which consists of three different trees

The working process can be explained through the following steps and fig 6.2:

Step 1: Randomly select K data points from the training set.

Step 2: Build decision trees associated with the selected data points (subsets).

Step 3: Determine the desired number of decision trees, N, to be constructed.

Step 4: Repeat Steps 1 and 2.

Step 5: For new data points, find the predictions of each decision tree and assign the new data points to the category that receives the majority votes.

6.4 Packages/Libraries Used

1. **Speech Recognition:**

   Speech Recognition is a Python library that enables the conversion of spoken words into text through the process of speech recognition. It offers a user-friendly interface for working with different speech recognition engines, including Google Speech Recognition, CMU Sphinx, and Wit.ai.

2. **Scikit-learn:**

   Scikit-learn, also known as sklearn, is a widely used open-source machine learning library for Python. It can be utilized for various natural language processing (NLP) tasks, such as text classification, which plays a significant role in supervised machine learning. The library provides a wide range of tools for data preprocessing, feature selection, model selection, evaluation, as well as supervised and unsupervised machine learning algorithms.

3. **Pandas:**

   Pandas is an open-source library designed for efficient and intuitive manipulation of relational or labeled data. It offers diverse data structures and operations for handling numerical data and time series. Built on top of the NumPy library, Pandas is known for its speed, high performance, and productivity. It provides powerful tools for data cleaning, reshaping, and analysis, including built-in methods for grouping, combining, and filtering data.

4. **NumPy:**

   NumPy, short for "numeric python," is a Python package focused on computational and processing tasks involving multidimensional and single-dimensional arrays. It offers various functions capable of performing numeric computations at a high speed. NumPy implements powerful data structures such as multi-dimensional arrays and matrices, which are optimized for efficient array and matrix operations.

5. **Open AI:**

   The OpenAI Python library serves as an interface for accessing a wide range of OpenAI models and tools. It finds extensive application in fields like natural language processing, chatbots, and AI-powered customer service. Its popularity stems from its user-friendly interface, state-of-the-art models, and support for customization and integration with other Python libraries.

6. **Pyttsx3 (Text to Speech):**

   Pyttsx3 is a Python library utilized for converting text into synthesized speech. It provides a straightforward interface for generating speech from text strings. Pyttsx3 is commonly employed in text-to-speech applications, including voice assistants, audiobooks, and accessibility tools. Its popularity is attributed to its flexibility, ease of use, and support for multiple speech engines and customization options.

Chapter 7

TIMELINE FOR EXECUTION OF PROJECT

7.1 Gantt Chart

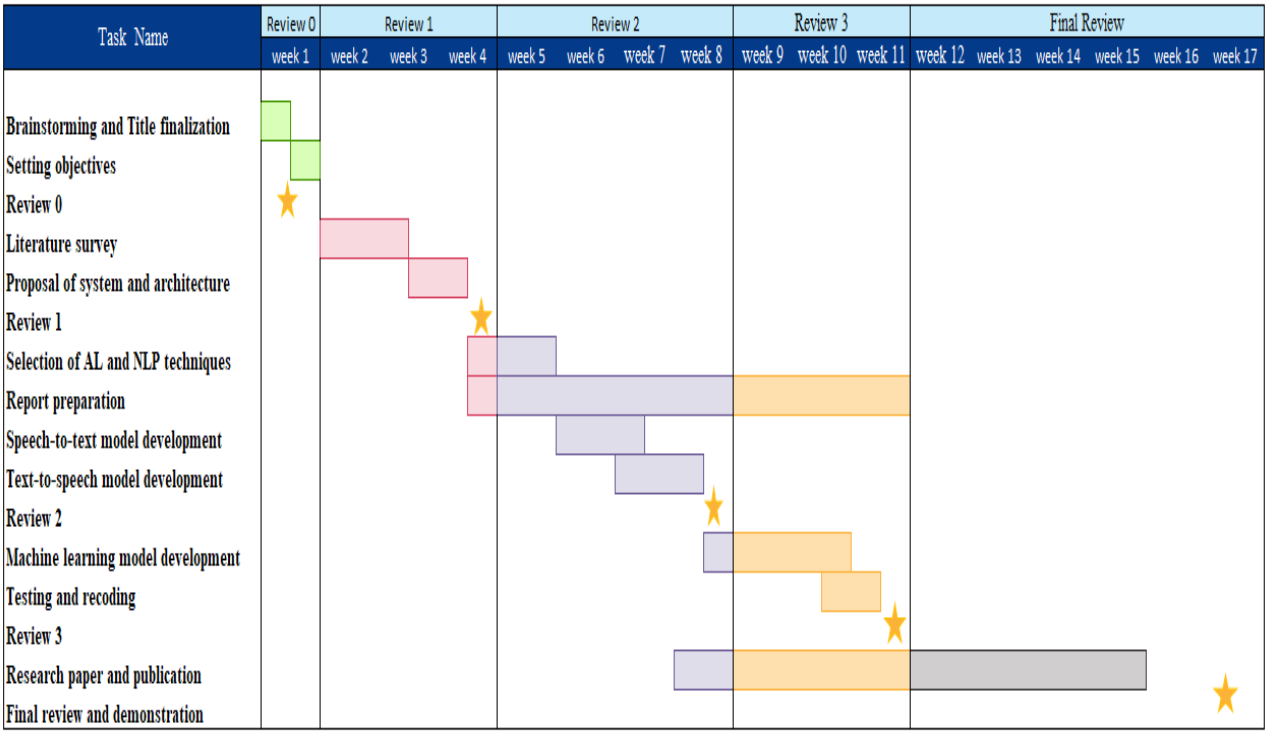

Fig 7.1: Gantt Chart for the Execution of the project

Chapter 8

OUTCOMES

8.1 Project Outcomes

The "Voice Assistant for Disease Diagnosis using ML and NLP" project has successfully achieved several key outcomes that have the potential to revolutionize disease diagnosis and healthcare accessibility. These outcomes include:

1. **Accurate Disease Diagnosis:** Through rigorous testing on a dedicated test dataset, the voice assistant demonstrated an impressive accuracy of 92.68% in predicting acute diseases. This high level of accuracy provides confidence in the system's ability to accurately diagnose diseases based on the spoken symptoms provided by users, ensuring reliable and trustworthy results.

2. **Enhanced Communication:** The successful implementation of the speech-to-text and text-to-speech conversion modules has significantly improved communication between users and the voice assistant. This functionality enables users to effortlessly communicate their symptoms through speech, while the voice assistant effectively translates the spoken input into text format. Similarly, the voice assistant can provide diagnosis reports, treatment options, and diet recommendations in an easily understandable voice format, facilitating seamless and user-friendly interactions.

3. **Reliable Treatment and Diet Recommendations:** The integration of the OpenAI library has been a resounding success, allowing the voice assistant to retrieve pertinent information and provide accurate treatment options and personalized diet plans. Leveraging the power of artificial intelligence and natural language generation, the voice assistant delivers reliable and contextually relevant recommendations, ensuring users receive tailored guidance for their specific health conditions.

4. **Improved Healthcare Accessibility:** By empowering individuals without medical expertise, the voice assistant has significantly improved healthcare accessibility. Users can easily gather symptom information, receive accurate diagnostic suggestions, and access personalized treatment and diet recommendations, regardless of their geographical location or access to medical professionals. This project's outcomes bridge the gap between individuals and healthcare services, offering valuable support and information to a wider audience.

5. **Efficiency and Knowledge Enrichment:** The voice assistant's accurate disease prediction, efficient symptom analysis, and personalized recommendations have enhanced the efficiency of the diagnosis process. By automating these tasks, healthcare professionals can allocate their time and resources more effectively, focusing on critical cases and complex medical conditions. Furthermore, the project's extensive literature review and research contribute to the knowledge enrichment of disease diagnosis methodologies, voice recognition, natural language processing, and treatment recommendation systems, fostering advancements and potential research opportunities in the field.

In conclusion, the "Voice Assistant for Disease Diagnosis using ML and NLP" project has achieved remarkable outcomes. With a high accuracy rate in disease prediction, successful speech-to-text and text-to-speech conversion, reliable treatment and diet recommendations, improved healthcare accessibility, and increased efficiency in diagnosis, the voice assistant has the potential to significantly improve disease management and patient outcomes. By harnessing the power of advanced technologies and personalized approaches, this project marks a crucial step towards a more accurate, accessible, and efficient healthcare ecosystem.

Chapter 9

RESULTS AND DISCUSSION

9.1 Outputs

Prompt: "Hello doctor"

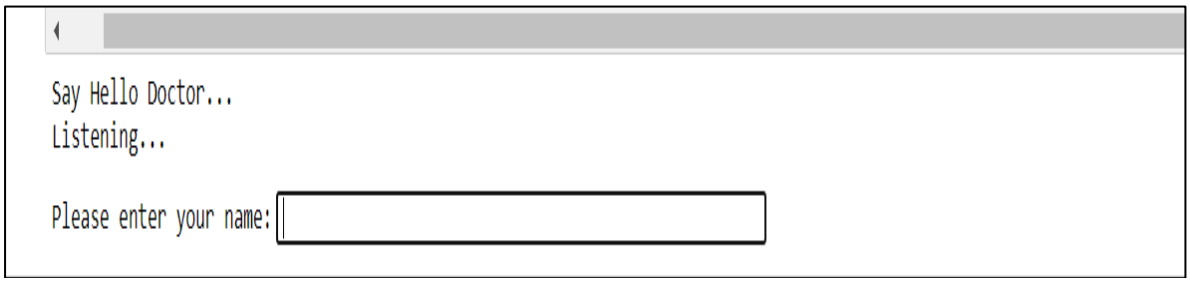

Fig 9.1: Output 1- Requesting the user to enter their name.

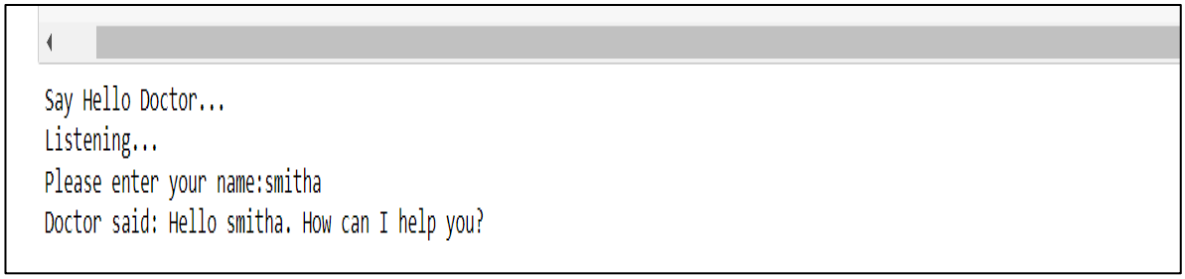

Fig 9.2: Output 2- Greeting the user and asking user's concern.

Prompt: "Predict my symptoms/ predict disease/ symptoms/ predict symptoms"

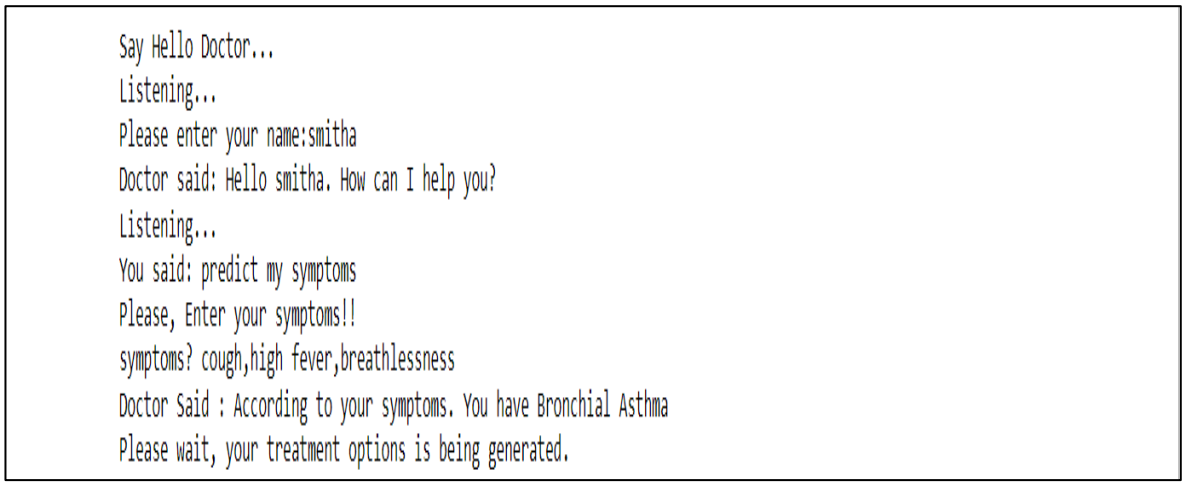

Fig 9.3: Output 3a- Disease Prediction based on the symptoms entered by the user.

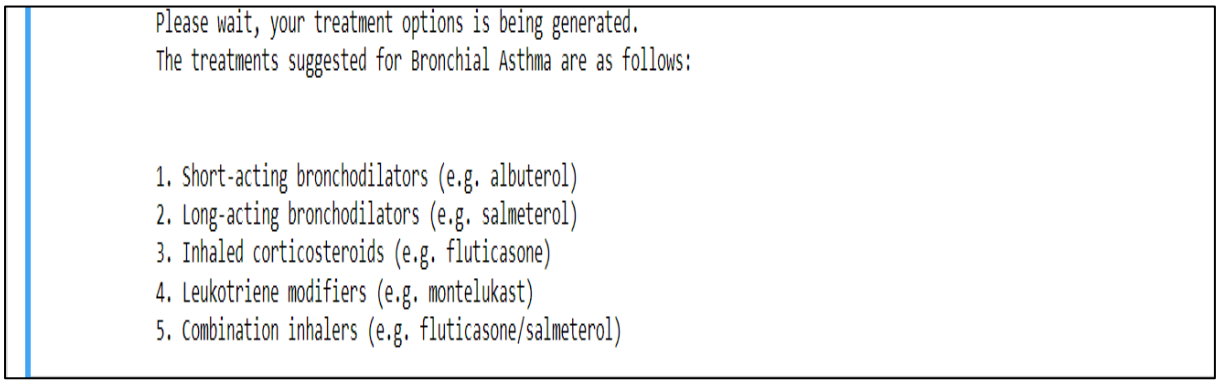

Fig 9.4: Output 3b – Top 5 treatment recommendations for the predicted disease.

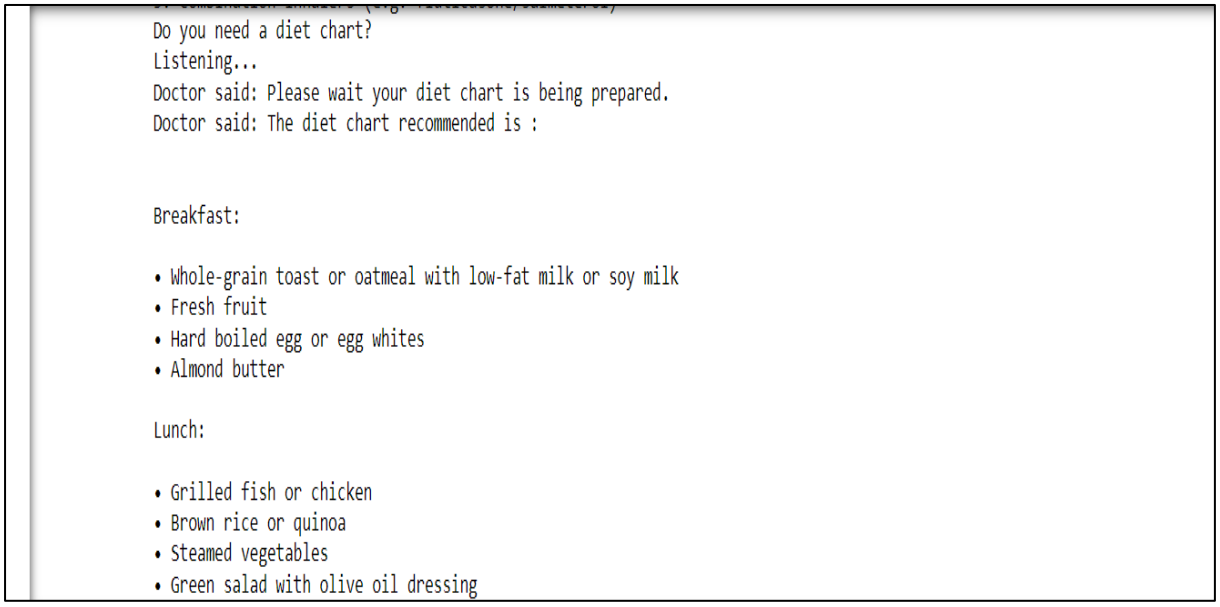

Fig 9.5: Output 3c- Diet chart recommended for the predicted disease.

Prompt: "Prepare a diet chart/ diet chart/ create a diet chart"

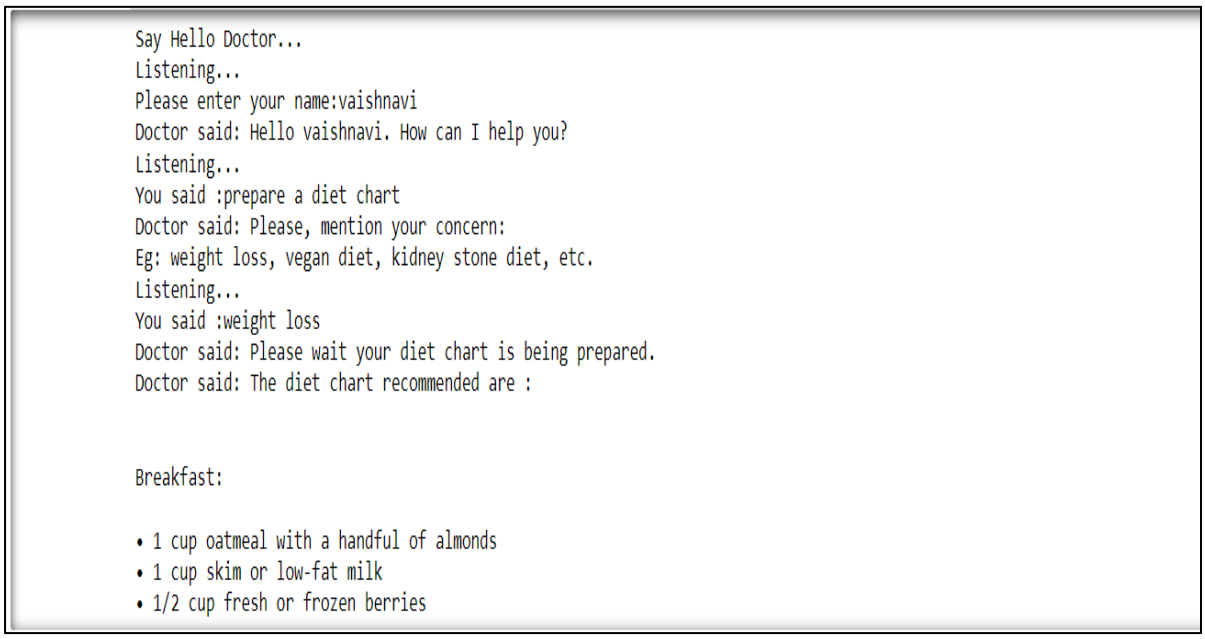

Fig 9.6: Output 4a- Diet recommendations based on individual's concern (e.g. weight loss).

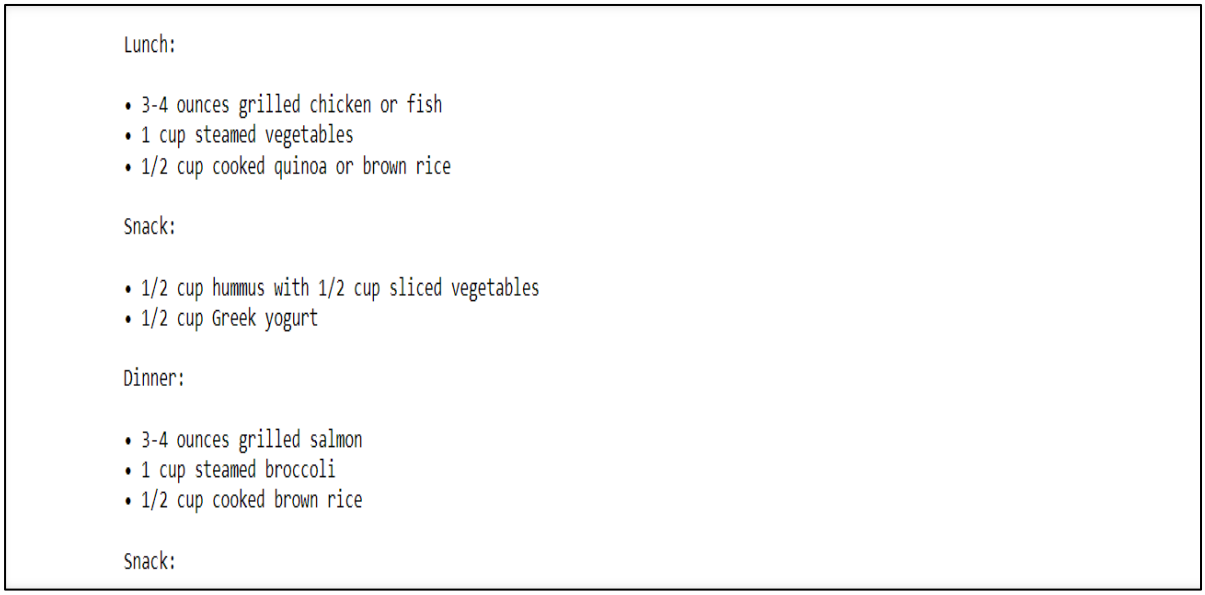

Fig 9.7: Output 4b- Diet recommendations based on individual's concern (e.g. weight loss).

Prompt: "what are the treatments/ how can I cure this/ treatments"

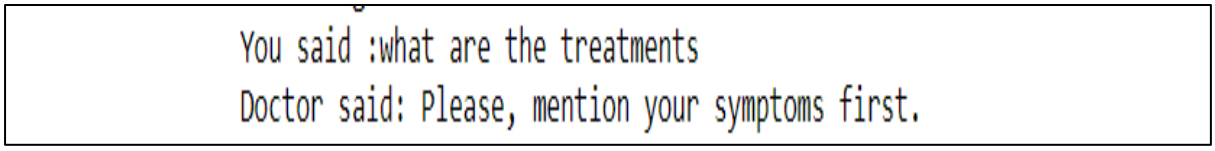

Fig 9.8: Output when user requests for treatment options without mentioning the symptoms.

Prompt: " "

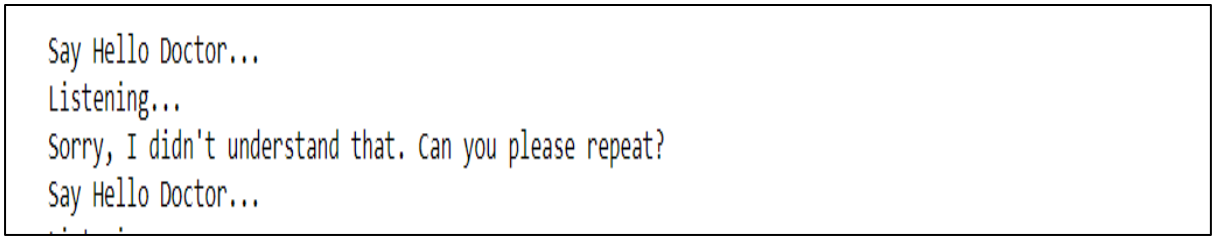

Fig 9.9: Output when there is no response from the user.

Prompt: “stop/ exit/ bye”

```
Say Hello Doctor...
Listening...
You said :bye
Doctor Said: Goodbye! Have a good day
```

Fig 9.10: Output when the user wants to exit.

9.2 Further Discussions

The obtained results from the "Voice Assistant for Disease Diagnosis using ML and NLP" project demonstrate the system's effectiveness and potential in disease diagnosis. With a disease prediction accuracy of 92.68% as shown in the Fig 9.11, the voice assistant proves its proficiency in analysing spoken symptoms and accurately predicting acute diseases. The speech-to-text and text-to-speech conversion modules exhibit robust functionality, ensuring seamless communication between users and the system. The successful integration of the OpenAI library enables the retrieval of relevant treatment and diet information, providing personalized recommendations for users.

```
In [19]: y_pred = clf_rf.predict(x_test)
         f1 = f1_score(y_test, y_pred, average='weighted')
         recall = recall_score(y_test, y_pred, average='weighted')
         print("Random Forest:")
         print("Accuracy",clf_rf.score(m,n))

         Random Forest:
         Accuracy 0.926829268292683
```

Fig 9.11: Accuracy achieved with the Random Forest Algorithm

The implications of the results are significant for healthcare accessibility and disease management. The voice assistant empowers individuals without medical expertise, improving their access to accurate diagnostic suggestions and personalized recommendations. It has the potential to bridge the gap between individuals and healthcare services, especially in regions with limited access to medical professionals. However, limitations and future improvements were identified, such as expanding the training dataset, enhancing voice recognition algorithms, and exploring advanced natural language processing techniques.

Overall, the results and discussions highlight the system's performance, potential impact, and areas for further enhancement, contributing to advancements in voice-assisted disease diagnosis and healthcare accessibility.

Chapter 10

CONCLUSION

10.1 Conclusion

In this project, we proposed a voice assistant system to predict diseases and make recommendations on treatments and diets based on the user's symptoms. It seeks to convert the input audio to machine understandable form using NLP techniques, predict symptoms using the Random Forest algorithm, and make recommendation using the OpenAI. Through significant milestones achieved, including successful voice recognition and accurate symptom analysis, the project demonstrates the potential to empower individuals without medical expertise. The integration of machine learning algorithms and OpenAI enhances the system's capabilities, enabling reliable treatment suggestions and personalized diet recommendations. System requirements analysis has ensured the alignment of software and hardware components, facilitating scalability, performance optimization, and user-friendly interface design. The successful implementation of this project has far-reaching implications for healthcare accessibility and disease diagnosis. It has the potential to alleviate the burden on healthcare professionals, improve diagnosis efficiency, and enhance patient outcomes. While challenges such as voice recognition accuracy and availability of comprehensive datasets exist, the project's achievements and potential impact underscore the importance of embracing innovative solutions like this voice assistant to create a more accurate, efficient, and inclusive healthcare ecosystem.

In conclusion, this is an innovative approach that has the potential to revolutionize the healthcare industry. By providing accurate diagnoses based on patient symptoms and medical history, voice assistants can help healthcare professionals save time, reduce costs, and provide accessible healthcare to patients. With further development and integration into healthcare systems, voice assistants have the potential to improve patient outcomes and the overall healthcare experience in urban and rural areas.

10.2 Future Scope

Future work can involve,

1. Expanding of dataset to include a wider range of diseases and symptoms, and incorporating other technologies such as image recognition.
2. Incorporation with neural networks to achieve better understanding of user's input and generate more accurate disease diagnosis.
3. To include other regional and international languages.
4. Future systems could explore the potential use of the system in clinical settings, where it could aid healthcare professionals in making more informed diagnosis and treatment decisions.

APPENDIX

Appendix 1: Code

#importing libraries

```
import  numpy as np
import pandas as pd

from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import  train_test_split, cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
```

#Reading and pre-processing datasets

```
train=pd.read_csv('Training.csv')
test=pd.read_csv('Testing.csv')

test=test.drop(['Unnamed: 133'],axis=1)#droping last column
test=pd.DataFrame(test)
train=pd.DataFrame(train)

test=test.rename(columns=lambda x: x.replace('_',' '))
train=train.rename(columns=lambda x: x.replace('_',' '))
```

#splitting

```
y=train['prognosis'] #target
x=train.drop(['prognosis'],axis=1) #symptoms
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.33,random_state=42)
```

#Machine Learning Models

```
print ("Naive Bayes")
nb=MultinomialNB()
clf_nb=nb.fit(x_train,y_train)
print ("Acurracy: ", clf_nb.score(x_test,y_test))

print ("DecisionTree")
dt = DecisionTreeClassifier(min_samples_split=20)
clf_dt=dt.fit(x_train,y_train)
```

```
print ("Acurracy: ", clf_dt.score(x_test,y_test))


print ("Random Forest")
rf=RandomForestClassifier(n_estimators=150,n_jobs=25,random_state=33,criterion="entrop
y")
clf_rf=rf.fit(x_train,y_train)
print ("Acurracy: ", clf_rf.score(x_test,y_test))

#splitting
n=test['prognosis'] #target
m=test.drop(['prognosis'],axis=1) #symptoms
x_train,x_test,y_train,y_test=train_test_split(m,n,test_size=.33,random_state=42)

y_pred = clf_rf.predict(x_test)
f1 = f1_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
print("Random Forest",clf_rf.score(m,n))
print("F1 score:", f1)
print("Recall:", recall)
print(classification_report(y_test, y_pred))
```

#Building the Voice Assistant

```
import openai
import pyttsx3
import speech_recognition as sr
import os

openai.api_key="API-Key "
```

#Text-to-Speech engine

```
engine= pyttsx3.init()
```

#Defining necessary functions

```
def stt(filename):
    recognizer= sr.Recognizer()
    with sr.AudioFile(filename) as source:
        audio= recognizer.record(source)
    try:
        return recognizer.recognize_google(audio)
    except:
        print("Unknown error")
```

```
def generate_response(prompt):
    response = openai.Completion.create(

  engine="text-davinci-003",
        prompt="what are the treatment options and precautions for "+prompt+". write in points",
        max_tokens=500,
        best_of=5,
        n=1,
        stop=None,
        temperature=0.5,
    )
    return response.choices[0].text

def diet_chart(concern):
    response2 = openai.Completion.create(
        engine="text-davinci-003",
        prompt="what are the ingredients i must eat and avoid for "+concern+". write 4 points each",
        max_tokens=1000,
        best_of=5,
        n=1,
        stop=None,
        temperature=0.8,
    )
    return response2.choices[0].text


def speak_text(text):
    engine.say(text)
    engine.runAndWait()

def Predict_disease():
    symptoms=input("symptoms? ").split(', ')
    if len(symptoms)==1:
        return symptoms
    else:
        # Create empty dataframe to hold symptoms
        symptoms_df = pd.DataFrame(columns=x_train.columns)
        # Add input symptoms to dataframe
        symptoms_df.loc[0] = 0
        for symptom in symptoms:
            symptoms_df[symptom]
```

```
    # Make prediction using trained model
        pred = rf.predict(symptoms_df)

  return pred[0]
```

#Main Function

```
import warnings

warnings.filterwarnings("ignore")

def main():
    while True:
        print("Say Hello Doctor...")
        with sr.Microphone() as source:
            r=sr.Recognizer()
            audio = r.listen(source)
        print("Listening...")
        filename = "input.wav"
        with open(filename, "wb") as f:
            f.write(audio.get_wav_data())
        text = stt(filename)
        if text and text.lower() == "hello doctor":
            speak_text("Please enter your name?")
            name=input("Please enter your name:")
            print(f"Doctor said: Hello "+name+". How can I help you? ")
            speak_text("Hello "+name+" How can I help you?")
            with sr.Microphone() as source:
                audio = r.listen(source)
            print("Listening...")
            filename = "input.wav"
            with open(filename, "wb") as f:
                f.write(audio.get_wav_data())
            text = stt(filename)
            if text and ("Predict my symptoms" in text or "predict disease" in text or "symptoms"
in text or "predict symptoms" in text):
                print("You said: "+text)
                print("Please, Enter your symptoms!!")
                speak_text("Please, Enter your symptoms")
                predict = Predict_disease()
                string="".join(predict)
                print(f"Doctor Said : According to your symptoms. You have "+string)
                speak_text("According to your symptoms. You have "+string)
```

```
            print("Please wait, your treatment options is being generated.")
            speak_text("Please wait, your treatment options is being generated.")
            treatment  = generate_response(string)
            speak_text("The Treatments suggested for "+string+" are as follows")

    print(f"The treatments suggested for "+string+" are as follows:\n"+treatment)
            speak_text(treatment)
            speak_text("Do you need a diet chart?")
            print("Do you need a diet chart?")
            with sr.Microphone() as source:
                audio = r.listen(source)
                print("Listening...")
            filename = "input.wav"
            with open(filename, "wb") as f:
                f.write(audio.get_wav_data())
            text = stt(filename)
            print("You said:"+text)
            if text and ("Yes" in text or "yes" in text):
                print("Doctor said: Please wait your diet chart is being prepared.")
                speak_text("Please wait your diet chart is being prepared.")
                dietchart=diet_chart(string)
                print(f"Doctor said: The diet chart recommended:\n {dietchart}")
                speak_text("This is the diet chart recommended for "+string)
                speak_text("Please note that this is just an example and it's important to consult with a healthcare professional or registered dietitian to create a personalized plan that meets your needs and goals.")
            elif text == "no":
                speak_text("Goodbye, have a good day.")
                print("Doctor said: Goodbye!")
                break
          elif text and ("Prepare a diet chart" in text or "Create a diet chart" in text or "diet chart" in text):
            print("You said :"+text)
            speak_text("What is your concern?")
            print("Doctor said: What is your concern?")
            print("Eg: weight loss, vegan diet, kidney stone diet, etc.")
            with sr.Microphone() as source:
                audio = r.listen(source)
                print("Listening...")
            filename = "input.wav"
            with open(filename, "wb") as f:
                f.write(audio.get_wav_data())
```

```
 text = stt(filename)
            print("You said :"+text)
            print("Doctor said: Please wait your diet chart is being prepared.")
            speak_text("Please wait, your diet chart it being created!!")
            dietchart=diet_chart(text)
            print(f"Doctor said: The diet chart recommended is:\n {dietchart}")

speak_text(dietchart)
            speak_text("Please note that this is just an example and it's important to consult with a healthcare professional or registered dietitian to create a personalized plan that meets your needs and goals.")
        elif text and ("what are the treatments?" in text or "treatments" in text or "How to cure this?" in text):
            print("You said :"+text)
            speak_text("Doctor said: Please, mention your symptoms first.")
            print("Doctor said: Please, mention your symptoms first.")

        elif text==" ":
            speak_text("Doctor said: Sorry, I didn't get you. Mention your symptoms again.")
      elif text and ("stop" in text or "exit" in text or "bye" in text):
        print("You said :"+text)
        speak_text("Goodbye, have a good day.")
        print("Doctor Said: Goodbye! Have a good day")
        break
      else:
        speak_text("Sorry, I didn't understand that. Can you please repeat?")
        print("Sorry, I didn't understand that. Can you please repeat?")

if __name__ == "__main__":
   main()
```

REFERENCES

[1] Ayushi Trivedi,Navya Pant, Pinal Shah,Simran Sonik, Supriya Agrawal. (2018) "Speech to text and text to speech recognition systems-Areview", *IOSR Journal of Computer Engineering*.

[2] Divya S, Indumathi V, Ishwarya S, Priyasankari M, Kalpana Devi S. (2018) "A Self-Diagnosis Medical Chatbot Using Artificial Intelligence", *Journal of Web Development and Web Designing Volume 3 Issue 1*.

[3] Dong Jin Park, Min Woo Park, Homin Lee, Young-Jin Kim, Yeongsic Kim & Young Hoon Park. (2021) "Development of machine learning model for diagnostic disease prediction based on laboratory tests", *Nature Portfolio.*

[4] Dr. Meera Gandhi, Vishal Kumar Singh, Vivek Kumar. (2019) "IntelliDoctor – AI based Medical Assistant", *Fifth International Conference on Science Technology Engineering and Mathematics*.

[5] Nicholas A. I. Omoregbe, Israel O. Ndaman, Sanjay Misra, Olusola O. Abayomi-Alli, Robertas Damasevicius. (2020) "Text Messaging Based Medical Diagnosis Using Natural Processing and Fuzzy Logic", *Hindawi Journal of Healthcare Engineering*.

[6] Rayan Alanazi. (2022) "Identification and Prediction of Chronic Diseases Using Machine Learning Approach", *Hindawi Journal of Healthcare Engineering*.

[7] Shahadat Uddin1, Arif Khan, Md Ekramul Hossain and Mohammad Ali Moni. (2019) "Comparing different supervised machine learning algorithms for disease prediction", *BMC Medical Informatics and Decision Making*.

[8] Xu Tan∗, Jiawei Chen∗, Haohe Liu∗, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang Yichong Leng, Yuanhao Yi, Lei He, Frank Soong Tao Qin, Sheng Zhao, Tie-Yan Liu. (2018) "NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality", *arXiv preprint arXiv:2205.04421*.

CERTIFICATE

PRESIDENCY UNIVERSITY

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi | Approved By BCI
Bengaluru

45 YEARS OF ACADEMIC WISDOM

Certificate of Presentation

This is to certify that Mr./Ms. Smitha Reddy S Under the Supervision of Dr./Mr./Ms. Swati Sharma from from PRESIDENCY UNIVERSITY, BANGALORE has successfully PRESENTED the paper at the National Conference on Recent Advancements and Challenges in Information Technology [NCRACIT-23] bearing the paper title Voice Assistant For Disease Diagnosis Using Machine Learning And Natural Language Processing and paper ID 225 held during 28th April 2023 - 29th April 2023.

Dr. Gopal K Shyam	Dr. Manujakshi B C	Dr. C Kalaiarasan	Dr. Md. Sameeruddin Khan
Conference Chair, Prof. and Head, Dept. Of CSE	Conference Chair, Associate Prof., Dept. Of CSE	General Co-Chair, Associate Dean – SOCS&IS	General Chair, Dean – SOCS&IS

Voice Assistant for Disease Diagnosis Using Machine Learning and Natural Language Processing

ORIGINALITY REPORT

23%	18%	9%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to M S Ramaiah University of Applied Sciences Student Paper	5%
2	irejournals.com Internet Source	3%
3	Submitted to University of Essex Student Paper	3%
4	www.geeksforgeeks.org Internet Source	1%
5	Submitted to University of Computer Studies Student Paper	1%
6	www.hindawi.com Internet Source	1%
7	www.researchgate.net Internet Source	1%
8	ijariie.com Internet Source	1%
	www.slideshare.net	