



Chatbot Using Machine Learning and Natural Language Processing in Healthcare Predictions

Master Dissertation Report

19037876

MSc DATA SCIENCE

SUPERVISOR: Dr Mahmoud Elbattah

COURSE: Csct masters project (UFCF9Y-60-M)

Thursday, 14th September 2022.

ABSTRACT

While overall human longevity has increased, some disorders are more common than ever, including growing rates of diabetes, hypertension, hypotension, cholesterol imbalance, obesity, and infections like fever. Medical advice on how to lead a healthy lifestyle is being sought out in advance by people all around the globe. They want to reduce their chance of contracting numerous ailments. Early illness detection and treatment may be beneficial. Physical consultation is expensive and time-consuming, especially in remote locations. Health care is crucial to living a long and healthy life. The whole range of medications or signs and symptoms for a particular condition are often unknown to the general public.

Nowadays, it is challenging for many people to acquire a doctor's appointment for any health issues, manage phone calls and users the problem of travelling to the hospital in person for a taxing test. Using healthcare chatbots will solve the problem. A healthcare chatbot application that can diagnose diseases based on user symptoms is described in the suggested solution.

A healthcare chatbot may help keep track of a person's health. The notion is to create a medical chatbot that can give the info, diagnose the disease, and deliver essential information about the condition and when and where to consult a doctor. As a virtual medical assistant, a chatbot for healthcare is computer software that mimics conversations with human users. These problems can be solved using Healthcare/Medicinal Chatbot by providing appropriate guidance regarding healthy living.

By offering relevant advice on leading a healthy lifestyle, healthcare/medical chatbots may help with these kinds of issues. The idea is to develop a medical chatbot that can supply information, diagnose the illness, and give fundamental knowledge about the disease and when to seek medical advice. It assists the user in making a diagnosis of probable diseases based on provided symptoms, monitors the user's health state, and warns the user of potential health risks. A medical chatbot will increase access to medical information while lowering the cost of healthcare. The programme development strategy calls for consumer sentiment analysis.

ACKNOWLEDGEMENT

First of all, I would like to express my most profound appreciation to my supervisor, Professor Mahmoud Elbattah, for the continuous guidance, enthusiasm and support given to me throughout the progress of this project. Professor Mahmoud has given me many variable instructions and encouraged me to learn for myself. Through this encouragement, I understand the knowledge for this project and the skills to solve problems independently.

A special note of thanks to my tutor student for their kind advice and guidance during the project. I have received many benefits from their suggestions and discussions. And I would also like to thank my colleague, who has helped me a lot throughout my studies and project by motivating me and discussing with me whenever I encounter a problem.

Not to forget, I would also like to take this opportunity to thank the University of the West of England and all my lecturers; Dr Paul, Dr Prakash and many more to mention; without their help, I would not be able to pass all my exams and focus on my dissertation.

Finally, I would like to sincerely thank my father, who recently passed away from cancer while I was doing this project and tolerated and supported me all these years. My father and mother have given me so much that they can. Without their support and care, I would not have had the opportunity to come to the UK for further study.

TABLES OF CONTENTS

ABSTRACT.....	2
ACKNOWLEDGEMENT.....	3
LIST OF TABLES.....	7
LIST OF FIGURES.....	7
LIST OF ABBREVIATIONS.....	8
CHAPTER 1 INTRODUCTION.....	9
1.1 Chatbot and Machine Learning.....	9
1.1.1 Chatbot.....	9
1.1.2 Applications of Chatbot.....	9
1.1.3 Importance of Chatbot.....	10
1.1.4 Types of Chatbot.....	10
1.2 Project Background.....	11
1.2.1 Problem Definition.....	11
1.2.2 Problem Overview.....	12
1.3 Scope of the Project.....	13
1.4 Object of the Project.....	14
1.5 Outline of the Report.....	15
1.6 Project Outcomes.....	15
1.7 Project Planning.....	17
CHAPTER 2 LITERATURE SURVEY.....	18
2.1 Brief History of the Information Chatbot System.....	18
2.2 Chatbot System.....	20
2.2.1 Working Principles.....	20
2.2.2 How does Natural Language Processing Work?.....	20
2.3 Related Work and Existing Systems.....	21
2.4 Proposed System.....	22

2.5 How a Chatbot Works?	23
CHAPTER 3 PROPOSED METHODOLOGY OF CHATBOT	24
3.1 Methodology Healthcare Chatbot Using Machine Learning	24
3.2 Data Pre-processing	25
3.3 Natural Language Processing Text Pre-processing	25
3.4 Classification Algorithms	26
CHAPTER 4 DESIGN OF HEALTHCARE CHATBOT SYSTEM	29
4.1 System Architecture of Healthcare Chatbot	29
4.1.1 Tokenisation	31
4.1.2 Stop word removal	31
4.1.3 Stemming and Lemmatization	31
4.1.4 Term Frequency and Inverse Document Frequency	31
4.1.5 N-gram	33
4.1.6 Sentence similarity	34
4.1.7 Retrieve the sentences that were the model	34
4.2 Flow chart of Model Healthcare Chatbot	34
4.3 Design of User Communication	37
4.3.1 Detecting Disease	37
4.3.2 Consultation Process	37
CHAPTER 5 DEVELOPMENT OF HEALTHCARE CHATBOT SYSTEM	39
5.1 Development of Healthcare Chatbot System	39
5.1.1 Description of Programming Language	39
5.1.2 Software Development of the Python Healthcare Chatbot	39
5.2 Program Algorithm	40
5.2.1 JSON file for intents and training data hoping for improved answers	40
5.2.2 Importing Libraries	41
CHAPTER 6 RESULTS AND DISUSSION	45

6.1 Performance Analysis.....	45
6.2 Performance metrics.....	48
6.3 Chatbot Interface.....	51
6.4 Challanges and Limitations.....	54
6.4.1 Advantages.....	54
6.4.2 Disadvantages.....	56
6.5 Discussion.....	57
CHAPTER 7 CONCLUSION AND FUTURE WORKS.....	58
7.1 Conclusion.....	58
7.2 Suggestion of Future Works.....	58
APPENDIX 1 GENERATING CHATBOT IN PYTHON.....	60
APPENDIX 1.1 The figures below show configuring the input Data and Tkinder used for creating login and registration on the application about disease predi- ction.....	60
APPENDIX 1.2 The figures below show configuring the input Data used for training and test disease prediction by Machine Learning and results.....	61
APPENDIX 1.3 The figures below show configuring the input Data used for training to find K- Fold disease prediction.....	65
APPENDIX 1.4 The figures below show configuring the input Data used for training and test to create a conversion interface about disease prediction....	66
APPENDIX 1.5 The figures below show configuring the input Data used for trai- ning and testing to create a conversion interface on the web- browser for disease prediction.....	68
REFERENCES.....	72

LIST OF TABLES

TABLES		PAGES
Table 6.1	Results of K-Fold Cross Validation	45
Table 6.2	Algorithm evaluation metric results	50
Table 6.3	Table 6.3 Conversion window User	54

LIST OF FIGURES

FIGURES		PAGES
Figure 1	Project Planning Healthcare Chatbot	17
Figure 3.1	Design Architecture of Chatbot	28
Figure 4.1	System Architecture	29
Figure 4.2	Shows the system flow	30
Figure 4.3	List of medical illnesses covered by the bot	35
Figure 4.4	Flowchart of Model's Working	36
Figure 5.1	Json file for intents and training data	41
Figure 5.2	Methodology Implementation	43
Figure 6.1	Shows the K-Fold result of the experiment	46
Figure 6.2	Shows the test accuracy result of the experiment	46
Figure 6.3	Shows the test accuracy results of the experiment	47
Figure 6.4	Home Screen	51
Figure 6.5	Registration Screen	52
Figure 6.6	Chatbot answering queries of the user	53

LIST OF ABBREVIATIONS

AI	Artificial intelligence
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PoS	Part of Speech
DT	Decision Tree Classifiers
SVM	Support vector machine
KNN	K-nearest neighbors' algorithm,
ML	Machine learning (ML) i
RF	Random Forest Classifier,
TF-IDF	Term Frequency and Inverse Document Frequency
GUI	Graphical User Interface
CUI	Character User Interface

CHAPTER 1: INTRODUCTION

1.1 Chatbot and Machine Learning

1.1.1 Chatbot

A chatbot is computer software created to start conversations with people. It uses NLP (Natural Language Processing) techniques for this objective. It is used to help individuals by delivering a message via a conversational flow like that of a person or by giving them a more functional interface to carry out a particular activity. [24]

The Chatbot also called a "chatter robot," is software that pretends to converse with a person through text or voice messages. One of Chatbot's first and most important goals has always been to look like an intelligent person and make it hard for others to figure out what they are [2].

Their use has dramatically increased as additional chatbots of diverse architectures and capabilities have developed. These conversational agents are severely constrained in how much they can expand their knowledge base in real-time while still deceiving the users into thinking they are speaking to a person.

A chatbot could benefit doctors, nurses, patients, or their families. The Chatbot use artificial intelligence and machine learning techniques to comprehend user input and provide a relevant answer. Additionally, they communicate with people using natural language in various chatbot applications, including call centres and medical chatbots.

Chatbots may be able to step in and relieve some of the pressure on medical staff by better organising patient data, managing medications, assisting in crises or with first aid, and providing a solution for minor medical difficulties.[18]

1.1.2 Applications of Chatbot

According to [61], A chatbot can assist in the following ways:

- Reducing the requests load by answering frequently asked questions and receiving similar requests.
- Chatbots can effectively collect data from users.

- It can be used to provide recommendations for users.
- Confidential credentials can be stored in the database using the chatbots, such as ATM pin codes.
- Users can perform tasks like seat reservations and online hotel bookings more efficiently by interacting with a human-like conversational interface.
- Chatbots can be used to advertise different products.
- The Chatbot will help in improving services as the Chatbot is accessible 24 hours.

Chatbots such as Siri, Cortana, Bixby, etc., are frequently used on different platforms as personal assistants to perform essential tasks such as reminding and searching in a more efficient and user-friendly way.

1.1.3 Importance of Chatbot

- According to [74], A chatbot has many benefits. Some of the benefits of a chatbot are:
- Chatbot saves users and admins time.
- It provides instant service and improves user experience.
- It saves the cost of hiring a lot of customer service representatives.
- It automatically does most of the work for you (i.e., protects the data provided by the user).
- It automates the chat for you, which is the future of conversation.
- It is a developing field and is expected to grow substantially over the coming years.

1.1.4 Types of Chatbot

According to [58], there are some types of Chatbots in terms of complexity.

- Button-Based Chatbots

These chatbots provide the user with a selection of predetermined buttons. These chatbots employ extremely primitive technology and consume a lot of user time.

- NLP Based Chatbots

These chatbots generate feedback using both artificial intelligence and machine learning. They can recall prior discussions and adjust their responses considering what a user has previously stated by using techniques for natural language processing.

- Hybrid Chatbots

Even some of the most sophisticated NLP-based chatbots have certain buttons for conversational topics. These chatbots combine Natural Language Processing with buttons. Buttons are employed in informal segments when extremely detailed replies are needed, while natural language is used in segments where user inputs are allowed.

- Sales Chatbots

These chatbots are intended to aid users in placing orders. They show the consumer the items, get his information, and make the purchase based on his preferences.

- Marketing Chatbots

These chatbots market a user's good, service, or concept via tailored messages or commercials. The most direct advertising is broadcasting, where each user receives a statement about our marketing materials.

- Support Chatbots

The primary goal of this Chatbot is to help a user in some manner. It might be as easy as assisting a user in finding a particular tab or facility on a website.

1.2 Project Background

1.2.1 Problem Definition

The medical profession is in limited supply in the healthcare business. Physical consultation with medical specialists takes time and money. Consultation with skilled specialists is challenging, especially in rural locations [6]. Rural residents are likewise concerned about affordability. Consequently, many

individuals postpone seeking medical attention, and even minor diseases/ailments become life-threatening, resulting in high death rates.

Nowadays, health care is essential to our survival. People in today's environment are preoccupied with work receptions, office tasks, and their internet addiction. They are uninterested in their health. As a result, people avoid seeing a doctor for a little ailment that might become a significant sickness in the future. Instead of reading through a list of relevant publications on the internet, creating question-answer forums is becoming a more peaceful approach to responding to such enquiries [93]. Chatbots are used in customer service, virtual assistance, online training, online reservations, and general conversations. The suggested Medical Chatbot may communicate with users, giving them the impression that they are conversing with a Medical Professional [89].

1.2.2 Problem Overview

Humans are benefiting from advanced technology. Chatbots based on Artificial Intelligence are one example of these breakthroughs (AI). An AI-based [1] chatbot is a software tool that is mainly used for simulating human conversations and delivering answers to user enquiries. As new companies thrive via the use of cutting-edge technology, the usage of chatbots in customers' everyday lives is fast rising. [84] They may be used for various tasks, such as customer service, request routing, and information retrieval. One example of a chatbot application is in the healthcare industry. As a result, it was determined to develop the system to build human-computer communication using natural language processing (NLP) techniques. As a chatbot is a system that can communicate with people using natural language [41], Three examinations comprehend natural language, namely, identifying the fundamental linguistic linkages that have been completed to split the issue into a phrase. The questions and texts are then explained. [16] To identify the symptoms, Natural Language Processing (NLP) is used to the user input. The symptoms are then sent into a Machine Learning (ML) system that has been taught to detect illnesses based on their appearance. To make a chatbot app, you can make a visual interface to send and receive information. This programme aims to make it sound like a real conversation. It's a programme that works with the user by keeping track of how they interact with it and remembering what they told it to do. The proposed

solution includes text and speech capabilities that can be used in different languages, which are very important in rural areas. The literature review shows that research on healthcare chatbots has mostly been done in English or a single language. The proposed solution can be used in more than just English. In addition to predicting diseases, the solution can answer user questions by comparing sentences. This can help with the first response and lead the person to a specialised health care professional. Healthcare chatbots can be used to support both doctors and patients with their health [14].

1.3 Scope of the Project

For the purpose of improving people's ability to communicate with one another and facilitating illness prediction, researchers have developed a chatbot. The user can also find it easier to get medical advice and learn about different diseases and how to treat them. Chatbots are software that acts like people [33]. They are based on artificial intelligence (AI) and use natural language processing (NLP) to understand what the user says and respond to it. This study uses NLP and machine learning algorithms to develop a chatbot plan for predicting diseases. The Decision tree algorithm is used to make the prediction.

The main goal of this project is to build an engine that lets a chatbot talk. This NLP AI application will use the well-known NLTK text categorisation package. Rule-based chatbots will now be made better with the help of the NLTK library. The Chatbot is one NLP application that uses NLTK, which stands for Natural Language Toolkit. To be clear, the Chatbot is a piece of software that uses AI and NLP to interact with users and answer their questions without any help from a person. [32]

Natural Language Processing (NLP) is required for the project in this article. To manage vast amounts of natural language data, an AI must use NLP. NLP allows computers and algorithms to understand human interactions in many languages. [3] Human language is very complicated and varied. We have access to a lot of unstructured material, yet we often overlook grammar or use misspelt or shortened terminology while writing. Natural language processing allows computers to speak with humans and do other language-related tasks more rapidly and effectively.[78]

Furthermore, knowing this will help us construct the essential component of any conversational chatbot to better AI chatbots and enable them to perceive the various subtleties and undertones of human discussions. [48]

1.4 Objective of the Project

Chatbots have the potential to make certain chores simpler. Chatbots may be used for an engaging health campaign on the Interactive Healthcare Platform. Chatbots are beneficial for completing certain activities simpler.

- To attenuate the time required to unravel the queries.
- To give responses to the user-supported queries.
- To simplify the communication between the human and the machine.

To get beyond these challenges and develop a workable system that can be used to understand human speech, process input from the user, and provide a suitable response, several features must be included in NLP technology.

NLP tasks are responsible for decomposing human text and audio signals from speech data so they can be examined and translated into computer-readable data.[3]

To learn to create a simple chat assistant or Chatbot using Python's NLTK library. The following are some of the tasks that are contained in NLP data:

- In the first stages of text preprocessing, NLP text categorisation - tokenisation
- Removing Stopwords, PoS (Part of Speech), Tagging, Lemmatization and Stemming.
- Learning approaches may be used to create a general function for preprocessing textual data. Then parsing a document using a bag of words.
- Machine learning text categorisation models: preparing data k-nearest neighbors' algorithm, Support vector machine and Decision Tree Classifiers.
- Creating an NLP chatbot engine function that any UI can use in the project.

1.5 Outline of the Report

Chapter 1 Introduction: This chapter presents the introduction, the reason for choosing the topic, and the objective and displays an overview research, limitations and layout of the project.

Chapter 2 Literature Review: This Chapter shows the content Theoretical Basis, Introduction to Chatbots, NLP work, Proposed System and principle of Chatbot.

Chapter 3 Proposed Methodology of Chatbot This Chapter displays the methodology of the Healthcare Chatbot System using Machine learning and NLP.

Chapter 4 Design of Healthcare Chatbot System: This chapter demonstrates each block's Design and Build system block diagram function.

Chapter 5 Development of Healthcare Chatbot: This chapter shows System Construction, Design of Chatbots and writing programs on Python with a data connection, then running the above program, Raspberry.

Chapter 6 Results and Discussion: This chapter demonstrates Results, Comments and Evaluation. And the results achieved when implementing the program, analysing, commenting, and evaluating results with challenges.

Chapter 7 Conclusion and Future Works: This chapter displays Conclusion and Development Direction. And Summary of achievements, limitations and development directions future.

1.6 Project Outcomes

This paper shows how Artificial Intelligence can be used to make a Healthcare Chatbot that can interact with humans to answer basic questions about health parameters before going to a doctor. [11] The real reason for this work is to focus on the user's symptoms and make medical suggestions based on those symptoms, which will save time and money. It is a system that lets people talk to it using Natural Language Processing (NLP). It is also a computer program that lets people talk to it using NLP.

The user input is processed by the Chatbot, which analyses the keywords in the phrase, concludes on how to best answer the user's question, and then responds appropriately. The TF-IDF, Stemming, n-grams, and cosine similarity algorithms are used in the performance of a number of these computations, including the Rank calculation and the sentence similarity analysis.

Natural Language Processing and Machine Learning, two subfields of Artificial Intelligence, provide the methodologies and techniques used by chatbots. Despite this, plenty of obstacles and constraints are associated with their use. Create the central component of a chatbot with this NLP application. [28] The NLTK library will then be used to teach us how to classify text using NLP approaches.

The bot will not be very bright, and as a result, it will not always understand what we are saying. However, if given enough instances, it will be able to do a respectable job of interpreting. [23] Because of recent advancements in natural language processing, artificial intelligence bots can now read and understand human writing and intelligently reply to it (NLP). On the other hand, communication between humans is not an easy task to do. Because there are so many nuances, accents, and advanced forms of language, creating a faultless chatbot or virtual assistant that can understand and respond appropriately to everyone is difficult.

Understanding how natural language processing (NLP) works is critical to understanding these challenges or concerns if we are to convert human speech into something that an algorithm or artificial intelligence can understand. [30] On the other hand, training an AI chatbot is akin to trying to learn a whole new language from scratch. Machines and algorithms have a challenging problem when it comes to comprehending and reacting to the many meanings linked with intonation, context, voice modulation, and other elements.

1.7 Project Planning

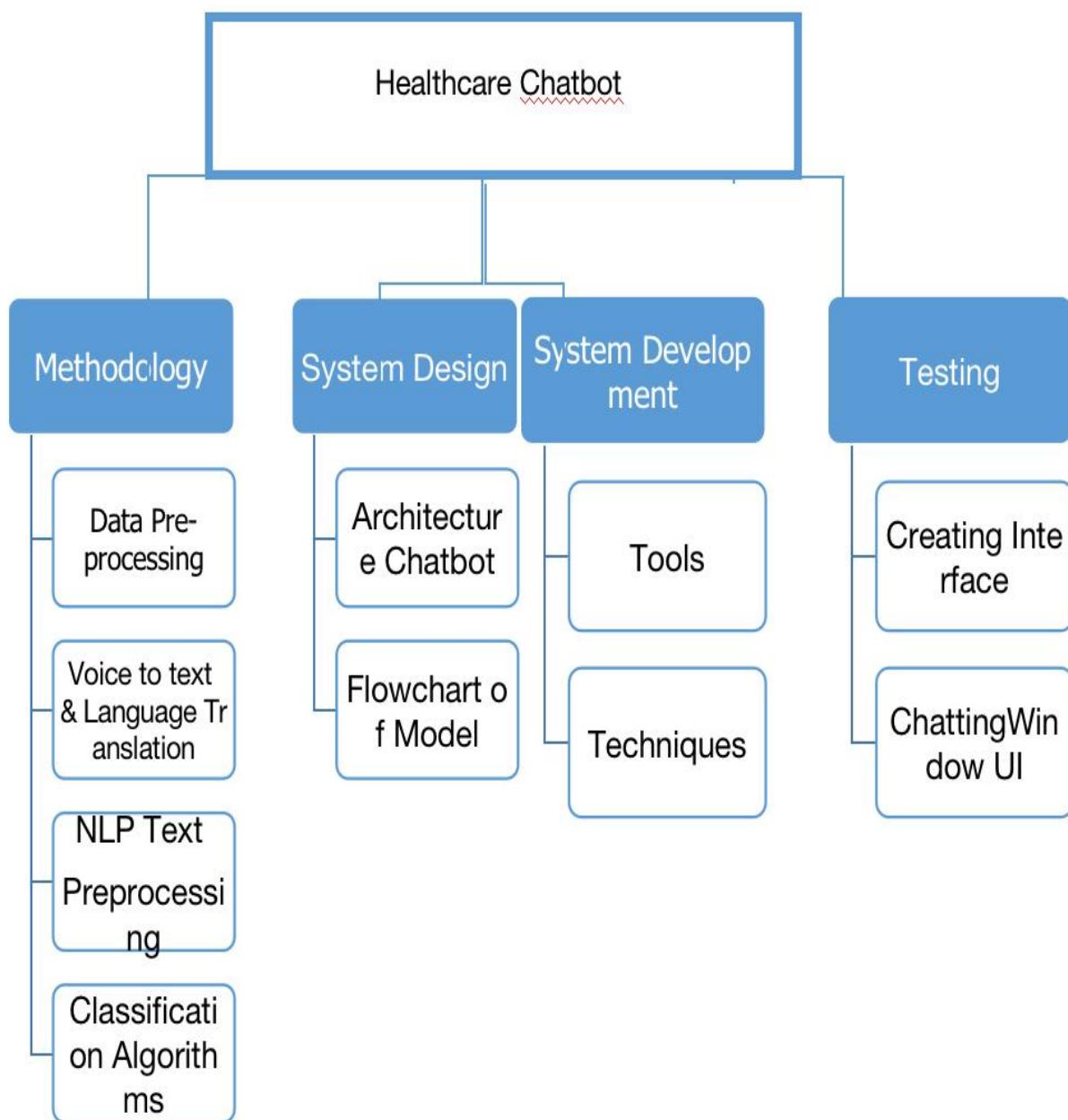


Figure 1. Project Planning Healthcare Chatbot

CHAPTER 2: LITERATURE SURVEY

2.1 Brief History of the Information Chatbot System

The literature survey in this paper was mainly focused on discussing the results from past successful cases. A literature survey is essential for understanding the artificial intelligence domain and the implementation of the working system. [10] Various medicinal chatbot designs have been presented in recent years to supply the user with prognosis and medicine sanctions after detecting the ailment data from the user chats.

A bot may be programmed to reason and respond without looking back. The suggested programmer development strategy takes advantage of health care [52]. First, a chatbot has been developed to assist users in determining the symptoms of their ailments. The programmable bot assists in tracking users who used the processed input from the Chatbot and transformed it into action on the website. "Chatbots will be widely employed to facilitate communication between humans and machines." [51] A Chatbot is a conversational program designed to stimulate conversation with users via auditory or textual methods [42]. The first-ever Chatbot, ELIZA, was developed in 1966 and could interact with humans as a psychotherapist could. There was a consensus that ELIZA was able to pass the Turing Test, a test proposed in the article 'Computing Machinery and Intelligence' (Turing, 1950). The Turing Test aimed to test a computer's ability to display behaviour that could be mistakenly considered human.[63]

When a chatbot is utilised to conduct an online chat discussion through text or text-to-speech instead of directly interacting with a live human agent, it is referred to as a chatterbot. Michael Mauldin (developer of the first Verbot) was initially created in 1994 to characterise these conversational algorithms; the word "ChatterBot" has become widely accepted. Today, practically all businesses rely on chatbots to provide customers with a positive customer care experience.[70]

One relevant field regarding Chatbots is Natural Language Processing. In 1997, a Chatbot named Cleverbot was created by British AI Scientist Rollo Carpenter. It is a web application that integrates Artificial Intelligence and Natural Language

Processing to hold conversations with users. In 2016, Cleverbot saw 69 million sessions from 41.8 million distinct visitors (Existor, 2016).

Over the past few years, chatbots have gained popularity and are now widely accessible. This was pushed due to the rise in popularity of Smartphones, where web designers had faced the issue of collapsing websites and their data to fit into a much smaller screen. [38] Chatbots can now solve this issue, where conversations are an easy task on a smartphone. Facebook has seen this need, introducing their official API for easy Chatbot development for their Messenger service.

According to Gartner forecasts, "By 2022, 70 per cent of white-collar professionals will engage with conversational platforms daily." The issue of human-computer interaction is the subject of human-computer interaction in the discipline of Natural Language Processing, which is a branch of computer science that analyses the interactions between people and computers. According to Wikipedia, this field is located at the crossroads of computer science, artificial intelligence, and computational linguistics [69].

"AI (artificial intelligence) to foresee the sickness based on the symptoms and present a list of readily available therapies for that specific ailment. It may help us identify the issue and find a solution.". This intelligent communication may be created using NLU, natural language processing NLP, and machine learning (ML). Consequently, the Chatbot becomes a visible buddy of the communication user (commonly employed in healthcare counselling). There are several domains in which intelligent chatbots may be used. [53] The purpose of this article is to provide an overview of the chatbot system for the healthcare sector. Additionally, it describes and compares the several NLU, NLG, and ML algorithms employed in the Chatbot. [91]

Machine learning (ML) is a field of computer science that "allows computers to learn without being explicitly programmed." Instead of being predetermined by the programmer, the formula parameters are computed using the input. Regression and Classification are the two most frequent applications of Machine Learning. As of [14], [7] Classification refers to grouping various sorts of data into categories, while regression refers to coming up with a technique to characterise the data.

2.2 Chatbot System

2.2.1 Working Principles

Algorithms of Natural Language Processing is a part of artificial intelligence that helps computers and people talk to each other using natural language. The main goal of Natural Language Processing (NLP) is to read, decipher, understand, and understand human languages meaningfully [54].

A chatbot that is based on natural language processing (NLP) is a computer programmed or artificial intelligence that can talk to a customer through text or sound. [38] Most of the time, these kinds of projects are meant to help customers in person or over the phone. Most of the time, chatbots are used in messaging apps like Slack, Facebook Messenger, and Telegram. (Rodriguez, 2019)

2.2.2 How does Natural Language Processing work?

NLP (Natural Language Processing) aims to transform unstructured language data into a framework that computers can understand by performing computations to identify and distinguish the specific language rules. [68] When the material is provided, the PC will apply mathematics to determine the significance of each phrase and obtain the essential information from them.

Occasionally, the PC may not understand the meaning of a sentence well, which can lead to bad things. In the 1950s, for example, it was funny to see how some words were translated between the English and Russian dialects. Here is the scriptural sentence that is the necessary interpretation:

"The spirit is willing, yet the substance is weak."

Here's what happened when the sentence was meant in English and then changed to Russian:

"The vodka is good; however, the meat is spoiled" (Garbade, 2018).

Text is analysed using natural language processing algorithms, which help robots comprehend human speech. How the computer interacts with the user provides insight into how real-world applications like sentiment analysis, topic extraction, named entity identification, parts-of-speech tagging, relationship extraction,

stemming, and others work. It aids chatbot systems in providing better responses [54].

2.3 Related Work and Existing System

In natural language processing (NLP), the most common research topics are figuring out what the user wants and getting information from the language. In the past, researchers put forward several models. [88] Building self-learning chatbots have significantly benefited from recent advances in artificial intelligence, intense learning and deep neural network models. Wu and AI [42] explored long-discussion answer selection in retrieval-based chatbots. This practise matches a prospective response with the context of a specific topic; the issue is identifying essential context components and applying speech connections [43]. Existing matching approaches may lose context-sensitive data. The Chatbot was created in the Python programming language using natural language processing and pattern matching methods. [25] The poll used to evaluate the Chatbot's effectiveness found that 80% of the replies were accurate, while 20% were unclear or inaccurate.[9] According to the previous results, this Chatbot can be primarily used as a virtual doctor for care and awareness and for teaching medical students.

According to Divya Madhu, artificial intelligence may be used to forecast any ailment and present a list of potential remedies based on provided symptoms. Additionally, if a person's body is frequently analysed, AI may assist in predicting any possible sickness even before any bodily harm has been placed [67].

Moshiur Rahman et al. recommended a healthcare chatbot. Chatbots use Machine Learning. This Chatbot solely talks to users. Six supervised machine learning techniques were examined. They've also worked on Random Forest Classifier, Support Vector Machine (SVM) and (KNN). SVM has the highest accuracy, NB the poorest. [71] They translated their dataset to make predictions in that language. Name, Blood group, and age extracted from unstructured user input. They transformed Bengali text into vectors using TF-IDF and then used Cosine Similarity to compare texts. Their method leverages Cosine Similarity [65].

The paper by R. B. Mathew et al. focuses on a chatbot system for discussing health issues. The system has been educated on illness signs. The algorithm predicts the

illness using NLP and the user's symptom list. [55] The dataset was divided in half and used for testing and training. Applying KNN of Machine Learning to disease symptoms. They used NLTK for Python to prepare text using NLP. It analyses symptoms and leads to therapy and prescription information. [56] describes a medical chatbot system that leverages human interaction and natural language diagnostics to serve customers. Accuracy after training dataset on SVM, KNN, and Naive Bayes algorithm.

According to Lekha, the research has created a chatbot to respond to user questions and provide basic information on illnesses. Tokenisation and stop-word elimination are used to pre-process the user's input. For keyword extraction, N-gram and TF-IDF algorithms are used. A Cosine Similarity algorithm balances the keywords, and the Chatbot's knowledge base is used to deliver the best suitable response [50].

Other research implements a chatbot application for illness prediction and question-and-answer sessions on health-related topics in the medical field. It attempts to mimic human-like speech. The study recommends using word order similarity for creating a vector. [26] It utilises the SVM algorithm to categorise diseases. An overview of chatbot solutions for consumer communication is given in the study report. The system uses NLP and query expansion algorithms to address commonly requested queries. The query expansion module processes the user input before using Cosine Similarity [58] to deliver the most relevant answer to the user inquiry.

According to his article, Urmil Bharti presented a conversational bot called "Aapka Chikitsak" that was created using Dialog Flow and serverless architecture to provide information on the COVID-19 epidemic. This programme was created to interact with patients as virtual physicians. [96] It primarily seeks to increase patient access to information about healthcare via multilingual assistance.

2.4 Proposed System

Chatbots are virtual assistants that can have conversations with users and do so automatically. Chatbots are run by artificial intelligence, which uses techniques like machine learning to understand natural language. The paper's primary goal is to help users find health information. [62] When a person initially accesses the website, they must register before they may ask the bot questions. If the answer is not found in the database, the system employs an expert system to respond to the requests.

The application focuses on the input a user provides while beginning a dialogue. This method recognises the user's potential issues by their earliest signs and symptoms. [75] After the Chatbot has gathered enough words from the initial messages that it recognises, it will start to take the lead in the discussion by asking the user questions and attempting to come up with a limited list of symptoms that the user could be experiencing.

Additionally, the Chatbot identifies the illnesses a patient may have; it currently ranks the diseases a user might have. The Chatbot begins by asking the user how they feel when a list of probable ailments appears [76]. Once it has gathered enough information, it may identify the infection that the person is most likely to have. After determining the user's health, [64] the Chatbot assesses the severity of the illness and responds accordingly by advising herbal treatments or sending the user to a doctor if the dosage exceeds a specified level.

2.5 How a chatbot works?

The most important part of making a chatbot is choosing the right engine for natural language processing (NLP) [92]. For example, if you talk to the Chatbot, it needs a speech recognition engine to turn what you say into text. Programmers also must decide if they want conversations to be structured or not. Chatbots that are made for structured conversations are very scripted. This makes programming more accessible, but it also limits the questions a user can ask.

Because of the application of artificial intelligence (AI) technologies, such as deep learning, natural language processing, and machine learning (ML) algorithms, chatbots have evolved significantly in recent years. [73] To provide correct results, these systems need a significant quantity of data. The more you communicate with the bot, the more accurate it becomes.

CHAPTER 3: PROPOSED METHODOLOGY OF CHATBOT

3.1 Methodology Healthcare Chatbot Using Machine Learning

Machine learning is a subfield of artificial intelligence, broadly defined as a machine's capability to imitate intelligent human behaviour. Artificial intelligence systems are used to perform complex tasks in a way that is like how humans solve problems. [35] Using statistical methods, algorithms are trained to make classifications or predictions and uncover critical insights in data mining projects. Machine learning algorithms, such as TensorFlow, are typically created using frameworks that accelerate solution development [31].

Train a reliable model to test the user's mental health. Provide users with support strategies and medical advice through the Chatbot. The suggested system has a chat interface that is simple to use and allows users to interact with it. The user has two options: he may either input current symptoms or health-related questions. Depending on the user's input, the Chatbot will forecast the sickness or provide helpful information about inquiries [13]. Then, the NLP component first does tokenisation or breaking down the language into individual words, then changes all the uppercase letters to lowercase and finally removes any frequently used stop words. After the stop words have been eliminated, the words are stemmed to return to their simplest form.

The trained machine learning model is employed for this purpose. The system checks to see if the keywords match a health-related question the user asks or the symptoms the user is having. [66] The system diagnoses the disease if the user's symptoms are the keywords. For better disease prediction, there is a threshold of four user symptoms. Because many diseases have similar symptoms, the prediction will be less accurate if the user enters less than four symptoms.

The dataset is used to gather information on the illness description and the precautions to take. If the Chatbot gets a lower number of symptoms than the threshold value, it will request further information from the user. When the illness has been successfully predicted, the result is communicated to the user, with a description of the ailment and any preventative measures or actions the user should take [59].

3.2 Data Pre-processing

The dataset that was received via [40] consisted of raw data with records and distinct illnesses, as well as the mapping of diseases to their associated symptoms. The dataset also explains each ailment and the appropriate safety measures to take. Following the completion of the check for inconsistencies in the dataset, the count vectoriser method was used in order to transform the unstructured categorical data into the structured numerical data format. [79], [86] The completed dataset consists of each column representing a different illness and each row representing a symptom of that condition. If a symptom is associated with a particular ailment, the cell's value corresponds to it as 1, but if it is not, the value is 0. Therefore, for each disease symptom combination, a value of 1 indicates the existence of that specific symptom for that illness. In contrast, a value of 0 shows that the disease does not display the corresponding symptom. This holds regardless of which disease and symptom pair is being considered.

3.3 Natural Language Processing text pre-processing

Natural Language Processing (NLP) provides consumers with a simple and human-like chat experience. To extract symptoms and appropriate keywords from the user input text, it must first be pre-processed. [55] Pre-processing techniques, including tokenisation, stemming, TF-IDF, and cosine similarity, have been utilised to provide a suitable answer for the user question.

a) Tokenisation: Case-folding, often known as text conversion to lowercase, is the first step in tokenisation. Lower case user input is a raw text that undergoes tokenisation to become a collection of words. Doing so will separate the terms that make up the text. [77] Tokenisation may be used to handle and analyse each word separately. The final bag of words is acquired once all punctuation has been eliminated.

b) Stop words removal: The bag of words gathered from the previous pre-processing stage is cleaned up to make room for crucial keywords by removing stop words like "a," "an," "the," etc. Stop words must be eliminated since they consume precious pre-processing time and space.

c) **Stemming:** The bag of words is then iterated, and the root form of each word is made by removing any suffixes or prefixes. In NLP, this is known as stemming. Here, the system uses the Porter Stemmer algorithm, which gives the best results compared to the other stemmers. The tokens made by the stemmer are what it takes in.

d) **TF-IDF:** Term Frequency (TF) shows how many times a term appears in the sentence. [80] Inverse Document Frequency (IDF) measures how unique a word is in a document and, by extension, how important it is. Combining TF and IDF values gives the term weight in the document.

$$W_i = tf * idf$$

(3)

tf = number of times term occurred in a sentence

IDF = inverse document frequency

Using the above formula, the weight of each user-supplied term is computed, and the resulting vector is produced. Similarly, another vector is generated following the exact pre-processing and weight computation of words in the database of question lists.

e) **Sentence similarity:** the distance between two vectors created by those sentences represents their similarity. The distance between the two vectors is calculated using the cosine similarity [49] technique. If the cosine angle between two sentence vectors is higher than 0, the associated text is returned to the user. Otherwise, the user is prompted to submit the appropriate information before receiving a chatbot answer.

3.4 Classification Algorithms

The research compares the effectiveness of classifiers for detecting anxiety and depression, including Decision Tree, K nearest neighbour (KNN), Support Vector Machines (SVM) and Random Forest [94]. The system has compared the following illness categorisation algorithms:

a) **Random Forest Classifier**

Random Forest Classifier is comprised of Decision Trees. [46] A random forest similarly utilises many decision trees. It trains Decision trees using samples of the initial training set. The accuracy of the Random Forest Classifier is improved as a result, while over-fitting is decreased.

b) Decision Tree

A Decision Tree is a tool for Classification that takes the shape of a tree. The training data are continually subdivided based on the characteristics.

c) K-Nearest Neighbours (KNN)

The KNN algorithm is based on the concept of similar things occurring near one another. [45] For the goal of Classification, the KNN algorithm considers how similar the characteristics are to one another. The value of the new data point is determined by comparing it to the training data points and seeing how well it matches up. It is one of the more straightforward classification algorithms that may be implemented.

The Naive Bayes algorithm is a probabilistic approach to learning. [43] The Bayes theorem is founded on the premise that all characteristics are independent of one another. This assumption is used to categorise the provided sample.

e) Support Vector Machine (SVM)

Moreover, the neural network often takes a long time to learn and analyse, which is not conducive to real-time health detection. [83] The primary function of the chat robot in this article is to evaluate health in real time. Therefore, this article uses SVM to train the learning model in Figure 3.1.

The primary purpose of the SVM algorithm is to find a hyperplane in a space with n dimensions. There can be more than one hyperplane for any classification problem.

[63] The chosen hyperplane must have the most significant margin, which means that the distance between the data points of the different classes must be the greatest. The hyperplanes act as boundaries between the other classes.

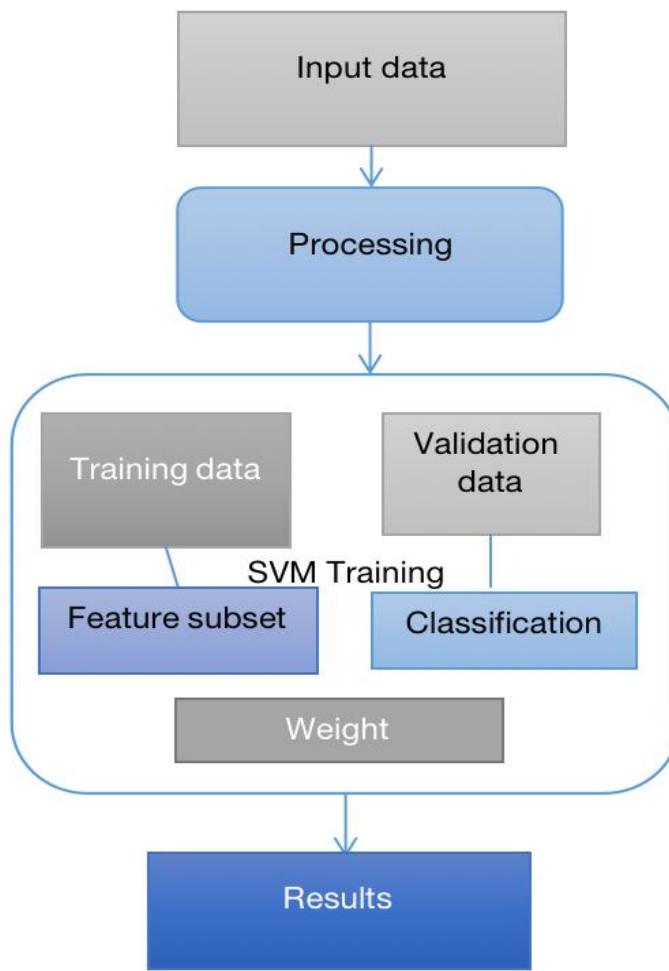


Figure 3.1 Design Architecture of Chatbot

The classifiers mentioned above use supervised learning techniques. [81] Machine learning techniques such as supervised learning include training models using real-world data. In this instance, characteristics that are mapped to labels or target values make up the training set—the algorithm correlates the inputs and outputs to learn from the training data.

In addition, the classification basis will learn some decision-making functions; thus, the functions realised by the neural network are sometimes like the observed samples, which may cause the model to predict incorrectly.

CHAPTER 4: DESIGN OF HEALTHCARE CHATBOT SYSTEM

4.1 System Architecture of Healthcare Chatbot

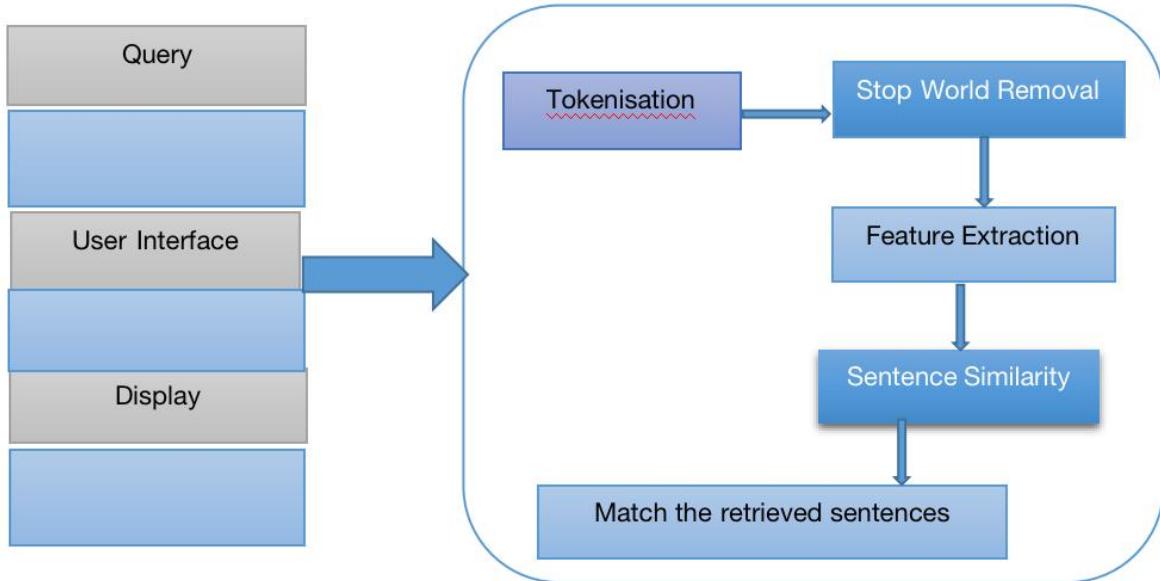


Figure 4.1 System architecture

The architecture design for a chatbot depends on the domain it will inhabit. For instance, you may pose a query to a chatbot, and it will answer. Perhaps you interrupt a chat in the midst and restart it later [5]. Depending on the Chatbot you create, the conversation history could or might not be kept. An ideal architecture for limited domains would match patterns. Advanced, cutting-edge neural network topologies, such as Long Short-Term, are appropriate under specific scenarios. Agents for learning with reinforcement and memory are used.

The figure from Reference Figure 4.1 shows the chatbot healthcare application's system architecture. The client enters the text-based queries into the user interface. The chatbot editor receives the user inquiry once it has been approved by the user interface [90]. In a chatbot application, the pre-processing of the text includes the creation of tokens, with suspended words being erased. The queries and replies are saved on the website for information. For more efficient processing, texts or sentences were divided into individual words. Additionally, it divides the text into words when coupled with one of the shown letters. Words are separated, and

redundant punctuation is eliminated. [21] This suggests the subsequent actions. To enhance analysis, words or phrases were separated word for word. When coupled with the displayed characters, it splits text into words. Punctuation and disagreements are separated from sentences. This suggests the following actions: Stop words are deleted to extract keywords from sentences. It is often used to remove unneeded material, such as words that repeatedly occur in texts. [60] In addition, it may be used to get rid of superfluous or unimportant words, such as "a" or words with no value. This section reduces the time it takes to dispense medication or simplifies problems with computational complexity.

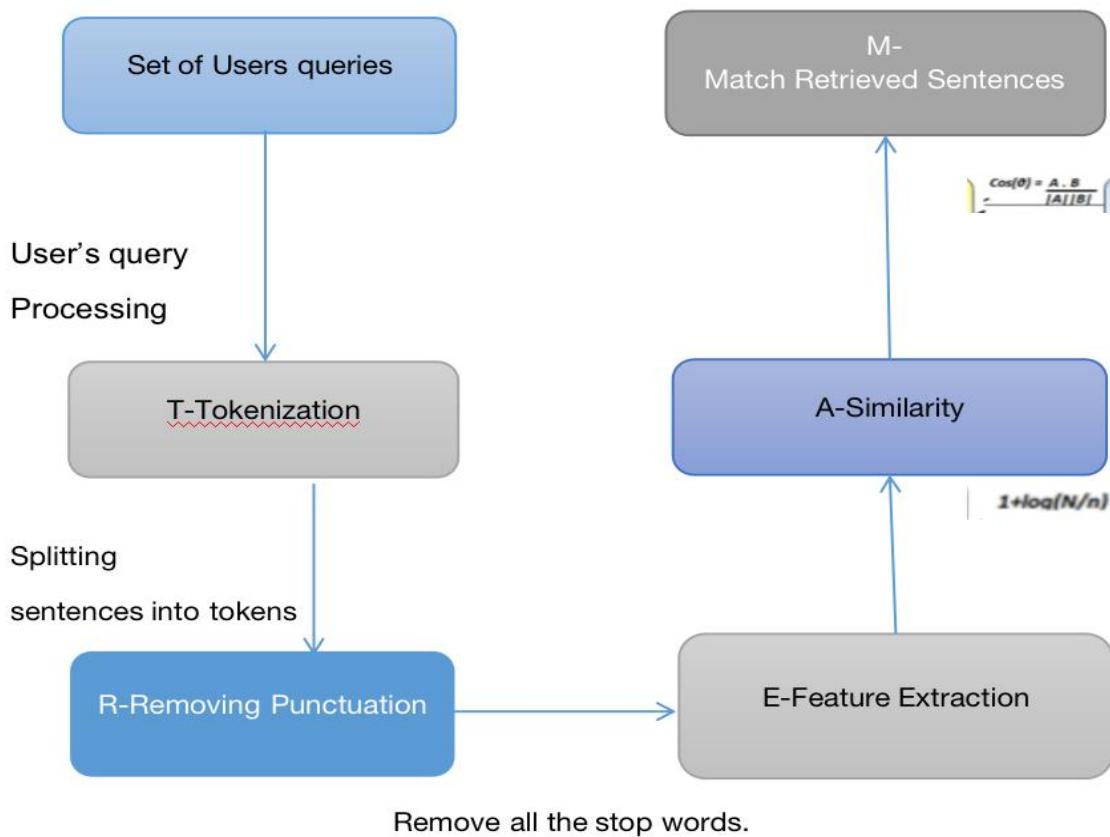


Figure 4.2 shows the system flow.

The overview of the system architecture for the chatbot healthcare application is shown in Figure 4.2. The customer enters the inquiry into the user interface (UI) as the text. The user interface collects the query from the user and then transmits it to the chatbot programme.

Tokenisation is one of the literary experiences pre-processing procedures that are included in the chatbot application. Once the words have been tokenised, stop words are eliminated, and feature extraction is carried out using n-gram, TF-IDF, and cosine likeness. [22] The responses to the questions are saved in the knowledge database so that the answer may later be retrieved and recovered.

4.1.1 Tokenisation

Tokenisation refers to breaking down larger blocks of text into smaller, more manageable chunks called tokens. Whether they're words, letters, or subwords, tokens are nothing more than words. The primary goal of doing this is to speed up the processing. It is widely agreed that these tokens are essential to any language model used for natural language processing. [8] It separates text when it meets a character rundown. Words or sentences are separated to aid understanding. All words and punctuation are removed from phrases. It implies additional action.

4.1.2 Stop word removal

The stop words are taken out of the sentences to find the critical keyword.[37] This step is used to cut down on processing time or make computations more manageable. It is mainly used to eliminate unnecessary things, like words used too often in a sentence. It is also used to eliminate words that are unimportant or have no specific meaning, such as "an", "a", or "the".

4.1.3 Stemming and Lemmatization

Stemming is the process of reducing a word to its stem word, its root word. [23]. The word from which another term is formed is the root word. For instance, Game is a root word, but Gaming and gamer are derivative terms. Stemming and Lemmatization are almost equivalent. Lemmatization provides more accuracy than stemming, which is the only distinction between the two. Thus, stemming minimises index size and improves retrieval precision.

4.1.4 Term Frequency and Inverse Document Frequency (TFIDF)

The following features may be extracted based on the N-gram and TFIDF:

The process of reducing the number of characteristics in a document is called feature extraction. [19] Feature extraction also ranks the attributes according to the requirements of the paper. Performing this step increases both the speed at which the document is processed and its level of appropriateness. It is used to extract the list of keywords and the frequency of the keywords found in the paper.[20]

With this assistance, it is possible to determine the importance of each word in the text and the corpus.

TF-IDF is Term Frequency and Inverse Document Frequency, abbreviated as TFIDF.

$$\text{tf}(t,d) = \frac{\text{count of } t \text{ in } d}{\text{number of words in } d}$$

(4.1)

To determine how much weight each phrase should be given in the sentence, a technique known as Term Frequency and Inverse Document Frequency (TF-IDF) is used.

The following formula may be used to determine a phrase's frequency inside a sentence by counting the number of times the term appears using the term "frequency."

$$\text{tf} = \frac{\text{tf}_i}{\text{N}}$$

(4.2)

Note: tf stands for term frequency.

IDF calculated the weight of unusual terms across all the reports included in the text. The IDF score for the terms that occur in the manuscript occasionally is relatively high. It is deduced from the condition that is shown below.

IDF - IDF is an abbreviation for "inverse document frequency," and it is used to quantify the informativeness of a phrase. Document frequency is simply the number of times a term appears in a document and is determined by, among other things, the following:

$$\text{idf}(t) = \frac{N}{df}$$

(4.3)

IDF is very low for stop words. in some cases, there will be a large corpus, so we take the log of IDF value.

$$\text{Idf} = \log\left(\frac{N}{df}\right)$$

(4.4)

Note: IDF stands for inverse document frequency.

When combined, the tf and IDF values determine the importance of the phrase or word across the page. When calculating the significance of each phrase in the document, the TF and IDF values are multiplied together.

Therefore, the weight of each term in the document can be calculated as,

$$Wi = \text{tf}_i * \log\left(\frac{N}{df}\right)$$

(4.5)

4.1.5 N-gram

A series of n objects (words, characters, or symbols) located next to one another in a text is referred to as an n-gram. When these n-grams are used, the probabilities of the occurrences of precise words in exact sequences can then be computed, which, in the end, improves the projection of completion systems [12].

The objective of the N-gram is to expand N-gram models with variable length configurations. A sequence might be a cluster of words, a word class, a grammatical characteristic, or anything else that the modeller deems to have important information about language structure [4]. In this system, N-gram is utilised for text compression or to minimise the document's data space to extract relevant keywords from the database.

4.1.6 Sentence similarity:

The cosine similarity measure is used to determine how similar two sentences are. The amount of query weights directly affects the degree to which the query and the document share similarities. Since the word frequency cannot take a negative value, the computation used to determine the documents' degree of similarity varies from 0 to 1. [27]. The following is the formula for determining the cosine similarity:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

(4.6)

Where,

$\|\mathbf{A}\|$ is the Euclidean norm of vector A and

$\|\mathbf{B}\|$ Euclidean norm of B.

The measure attempts to determine the cosine angle between vectors A and B. The cosine similarity is a value that ranges from 0 to 1 and is most often used for detecting instances of plagiarism.

4.1.7 Retrieve the sentence that was matched

After completing the steps outlined above, the user interface will then be updated to include the returned responses to the query that was run.

4.2 Flowchart of Model Chatbot

Our chatbot covers as many disease domains as possible. This chatbot diagnoses ordinary health conditions users face daily. It gives patients a simple predictive diagnosis that may help them identify their illnesses. [79] As technology improves, these technologies will help the healthcare industry satisfy basic customer requests.

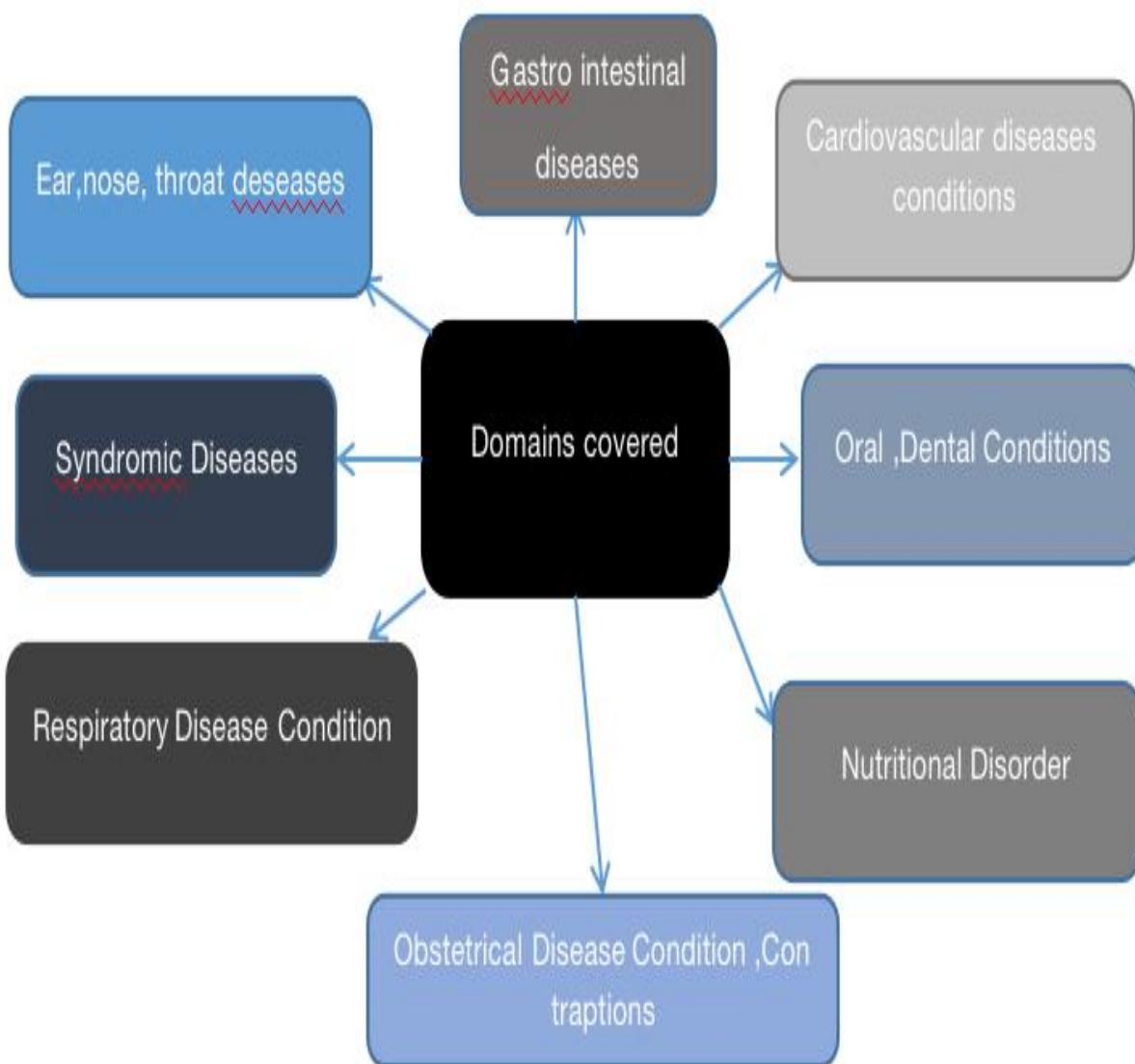


Figure 4.3 List of medical illness covered a bot

After completing the steps outlined above, the user interface will be updated to include the retrieved answers to the questions. Within this context, the application is intended to provide input soon. Giving feedback to the user directly via the expert system relieves the load normally placed on the feedback provider. When seeking a health solution from a doctor or expert, the user will spend fewer time thanks to this project, which was built specifically for the user's benefit.

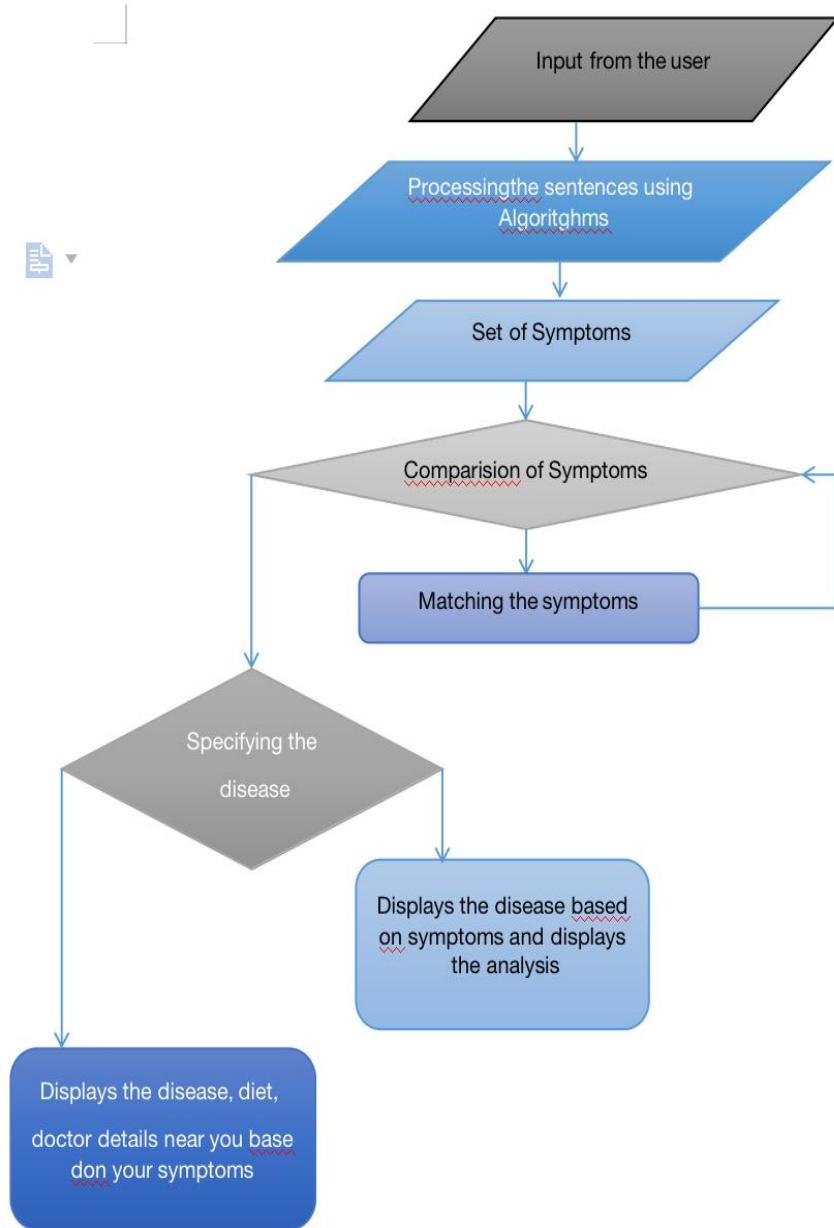


Figure 4.4 Flowchart of Model's Working

The system's processing in Figure 4 is shown here using a flow diagram, as is evident to us. The use of directed arrays illustrates the flow of the procedure, which is broken down into steps and examples.

4.3 Design of User Communication

4.3.1 Detecting Disease

Considering the user's first symptoms, the engine may have narrowed down the list of possible diseases. Now, the engine has to figure out what disease the user most likely has so it can give the user suggestions for how to treat it. The engine does this by going through the list of diseases in order and asking the user a question about each one [29].

This is how it's done:

- The Chatbot asks the user the first inquiry for each condition. If the user acknowledges the symptom, the engine adds 1 to the disease's score.
- Similarly, the engine performs the same for the last two questions about all disorders. The Chatbot has now asked the user the top three questions for each illness on the shortlist.
- The engine now chooses the disease with the most significant score.
- If the engine can locate a single condition with the highest score effectively, then it is quite probable that this ailment is the one the user is suffering from.

4.3.2 Consultation Process

After the engine has successfully detected the ailment that the user may be suffering from [6], our Chatbot has to speak to the user to gather the difficulties he is suffering from. Then it needs to propose a remedy that is based on the problems that the user is experiencing. The following is the procedure that is followed for the consultation process:

a) Questioning Symptoms

When a user clicks on a disease in the Chatbot's database, it asks them about their symptoms. For each symptom that matches the user's issue, the engine adds the symptom's severity score to the disease's severity score [10].

b. Threshold Checking

Occasionally, the topic is too complicated for the Chatbot to resolve. If this were the case, the Chatbot would connect the patient with the physician and provide the doctor access to the patient's discussion records. The customer gets the first-aid solution before the doctor's availability for online consultation. For this treatment to commence, the severity score must be at or above the threshold level [15].

Formula to check Threshold Level

$$H = \Sigma (\text{Score}) / \text{Threshold Value}$$

H = It is the decision parameter to check if the threshold level was hit.

$\Sigma (\text{Score})$ = It is the total score of all the symptoms the user claims to experience.

Threshold Value = The upper limit value until the Chatbot can handle it.

Condition for triggering= if $H \geq 1$, the Chatbot connects the user to the doctor.

CHAPTER 5: DEVELOPMENT OF HEALTHCARE CHATBOT SYSTEM.

5.1 Development of Healthcare Chatbot System

5.1.1 Description of Programming Languages

- a) **Python:** It is a general-purpose coding language which means that, unlike HTML, CSS, and JavaScript, it can be used for other programming and software development besides web development.
- b) **Machine Learning:** Machine learning (ML) studies computer algorithms that improve automatically through experience and data. [36] Machine learning algorithms build a model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed.
- c) **NLP (Natural Language Processing):** Natural language processing (NLP) refers to the branch of computer science—and more specifically, the extension of artificial intelligence or AI—concerned with giving computers the ability to understand the text and spoken words in much the same way human beings can [58].
- d) **Natural Language Toolkit** or NLTK is one of the most potent NLP libraries, this library contains packages that helps machines in figuring out human language and reply to it with an appropriate response. Character count, Tokenisation, Punctuation, Lemmatization, Stemming, and Word count are some of its features. To install this type of pip, install nltk in Command Prompt.

5.1.2 Software Development of the Python Healthcare Chatbot

a) Description of Libraries use

According to [2],

- **NumPy:** NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multi-dimensional array objects and a collection of routines for processing of array. Using NumPy, mathematical and logical operations on displays can be performed.

- **Pandas:** Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another

package named Numpy, which supports multi-dimensional arrays. As one of the most popular data wrangling packages.

- **Matplotlib:** Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object-oriented API to embed plots in Python applications.

b) Building a chatbot using machine learning techniques in this Python project with source code.

The Chatbot will be trained on the dataset containing categories (intents), patterns, and responses. After classifying the user's message's kind using a specific recurrent neural network, choose a random response from the available options. Chatbot implementation's main challenges are:

- Classify user input to recognise intent (this can be solved with Machine Learning and Keras with TensorFlow backend).
- Keep context: This part is programming, and there is nothing many machines Learning-related here. This is the list of Python libraries which are used in the implementation. Keras library is used to build a classification model. Keras runs training on top of the TensorFlow backend. Let's create a retrieval-based chatbot using NLTK, Keras, Python, etc.

5.2 Program Algorithm

This Chatbot seeks to identify common healthcare challenges in our everyday lives. It delivers simple predictive diagnoses to assist patients in understanding what condition they may have. The Chatbot will cover as many illness domains as possible.

5.2.1 JSON file for intents and training data hoping for improved answers

The JSON file utilised in this study contains several instances that will aid in training the model for our purposes. In this file, intentions will be expressed by tags, and

inside these tags, patterns will be the kind of input the user will write, even if the patterns. Word for word, the trained model will automatically determine the precise purpose and provide replies that are closest to that pattern (highest probability). Users have also used context sets, which will aid in grouping similar types of intents. For example, suppose a bot is building a conversation with a user about their disease. In that case, the programmed will group all the symptom tags related to that disease, assisting in determining which disease the patient requires diagnosis.

```
In [1]: import json

with open('intents_short.json', 'r') as f:
    intents = json.load(f)

intents

Out[1]: {'intents': [{}{'tag': 'abdominal_pain',
  'patterns': ['abdominal pain',
   'abdomen hurts',
   'pain in stomach',
   'pain in abdomen']},
 {}{'tag': 'abnormal_menstruation',
  'patterns': ['abnormal menstruation',
   'heavy period',
   'Heavy flow',
   'Period lasts longer',
   'period painful',
   'strong menstrual pain',
   'Menstrual cramps strong']},
 {}{'tag': 'acidity',
  'patterns': ['acid reflux', 'acidity problems', 'heartburn']},
 {}{'tag': 'acute_liver_failure',
  'patterns': ['acute liver failure',
   'liver hurts',
   'pain around liver']}]
```

Figure 5.1 JSON file for intents and training data

5.2.2 Importing Libraries

Important packages like Numpy, NLTK, TF Learn, and TensorFlow have previously been addressed. Another set of packages required for this role is listed below:

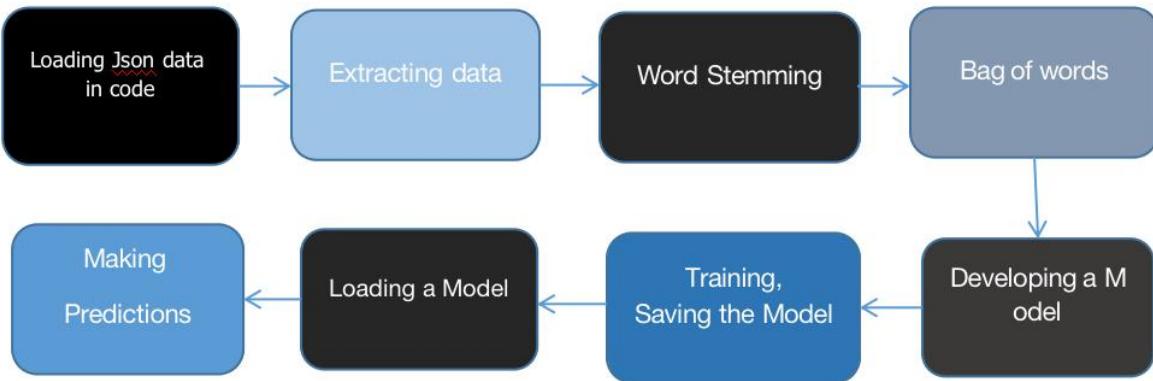
Random: The random module helps with a variety of tasks. This random package provides unexpected answers each time we run the code; these responses are related to the learnt pattern's tag.

JSON: JSON is a compact data-exchange format that is also referred to as JavaScript Object Notation. The main benefit of using JSON is it's easy to understand and use. JSON data is completely linguistically neutral and is readily generated by machines. JSON is a great choice for a programming language due to its data interchangeability features. This package may be imported to help read training data and extract pertinent intents.

Pickle: Pickling is a Python library for serialising and deserialising object structures. The pickling process may turn any Python object, such as a list, dictionary, or other data structure, into a character stream. Pickling facilitates the storage of pickled data for later use.

Lancaster Stemmer: The rationale for stemming is that it lessens the stress on processes and allows the software to function quicker and more efficiently. The stemmed words may not always have the same morphological meaning, but they may aid in the reduction of their form.

Following the import of all the necessary data from the JSON file, a step called pickling saves the information so that the model will not need to be trained each time the program is performed. Then there will be tokenisation, which means that the provided phrase will be chopped into smaller sections (token). And put this data in arrays. This program will utilise two variables, one to store the tags and another to hold the proper replies. The code will then cycle over our JSON data and retrieve the info required for implementation. Each pattern was turned into a list of words using the nltk. word tokeniser instead of being stored as a string. Then there will be stemming implementation, an effort to determine the root meaning of a given the word; for example, the stem word for cooling is "cool." This procedure will aid in the reduction of our model's vocabulary, allowing it to train more effectively and identify more broad meanings behind the statement.

**Figure 5.2 Methodology Implementation**

After importing the data and constructing a stemmed vocabulary, the next stage will be to use Natural Language Processing techniques such as 'Bag of Words.' Machine learning techniques and neural networks now need numerical inputs. As a result, our string list sufficed. A bag of words will assist us in representing our phrases numerically. They will express each phrase with a list of the length and number of terms in our vocabulary. Every place on the list will most likely reflect a vocabulary word. If the location in the list stores a 1, the word occurs in our phrase, but if it holds a 0, it means that the term does not exist. The sequence in which the words occur in the phrase is lost in the sentence, and we just know the existence of terms in our lexicon.

Our output also needed to be structured so that the neural network could interpret it. Therefore preparation of our input was required. Like a bag of words, our output lists needed to be as long as the total number of labels/tags in the dataset. Each point in the list will represent a unique label/tag; for example, one may be in any of those locations and will aid in recognising the related label/tag if present. We were ready to develop and train a model after pre-processing all our data. They utilised a typical feed-forward neural network with two hidden layers for this Chatbot. The trained dataset network will examine a collection of words and provide a class to which it belongs (one of our tags from the JSON file). The next stage was to define the model's architecture.

The focus of our project activity will be the development and training models. More layers may be added to aid in the optimisation of bigger training models.

The output layers include functions, which aid in returning the neuron with the most significant probability, hence determining the correct answer. After that, store the training model, so you don't have to retrain the bot every time it runs [17]. The model should answer all user-entered statements. This requires a bag of words input rather than string input. Furthermore, rather than spitting out phrases, our model returns a list of probabilities for each class. The steps taken were as follows:

- ❖ Obtaining user inputs
- ❖ Translating or converting it into a bag of words
- ❖ Predicting the model
- ❖ Finding class with needed probabilities
- ❖ Picking responses at random from the class

The bag of words function will use our previously built words list to convert our string input into a bag of words. The chat function will investigate handling model predictions and picking a suitable answer from a JSON file.

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Performance Analysis

The healthcare Chat Bot System must be implemented in Python. The machine learning model experiments were performed using jupyter on Python 3.9 as the core language and sci-kit learn Machine Learning library. The results for the disease classification algorithm have been obtained by comparing five different classification algorithms mentioned in the proposed methodology. The dataset used has records. Training and testing split has been performed on the dataset.

The K-Fold Cross Validation has validated the experiment's different classifiers. The approach involves testing the different algorithms on the dataset and evaluating the results using the K-fold Cross Validation score. The results obtained from the 10-fold Cross Validation are presented in Table 6.1. The Chatbot accurately analyses these symptoms and gives the user the appropriate disease diagnosis.

Table 6.1 shows the results of K-fold Cross Validation for each classifier, and they all perform better than any other classification techniques tried on this dataset. Three different measures for measuring algorithm performance have been used in this assessment.

Table 6.1. Results of K-Fold Cross Validation

Classification Algorithm	K-Fold Cross Validation Score
Decision Tree	1.00
KNN	1.00
SVM	1.00
Gaussian Naive Bayes	1.00
Logistic Regression Model	1.00
Random Forest	1.00

```
=====
SVC
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
Gaussian NB
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
Random Forest
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
KNN
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
```

Figure 6.1 shows the k-fold result of the experiment.

And Figure 6.1 depicts a graph comparing the k-fold of the different algorithms trained and tested on the same train and test datasets. By training and evaluating some of the most popular classification algorithms, the best one is selected based on the score achieved by a K-Fold. The scores following are calculated, saving a fourth of the dataset for testing afterwards and using the rest to get these results. This process can be automated through a grid search that optimises the best parameters.

Support Vector Machines, such as the SVM algorithm, achieve higher performance. Using 3/4 of the dataset as training data and the rest as testing data, svm accomplishes 1.00 accuracy. The scores shown in Figure 6.1 are obtained by performing a K-Fold and calculating the mean of the scores.

Print accuracy of trained model

```
print("Accuracy")
print(accuracy_score(clf.predict(x_test), y_test))
```

```
Accuracy
1.0
```

Figure 6.2 shows the test accuracy result of the experiment.

Figure 6.3 shows the test accuracy result of the experiment graphically. As we can see, classification reports for each classifier, knn, SVM and logistic regression's success score is 100%, while decision tree and random forest's accuracies are approximately 98%. The experimental results for the algorithm evaluation have been summarised in this figure.

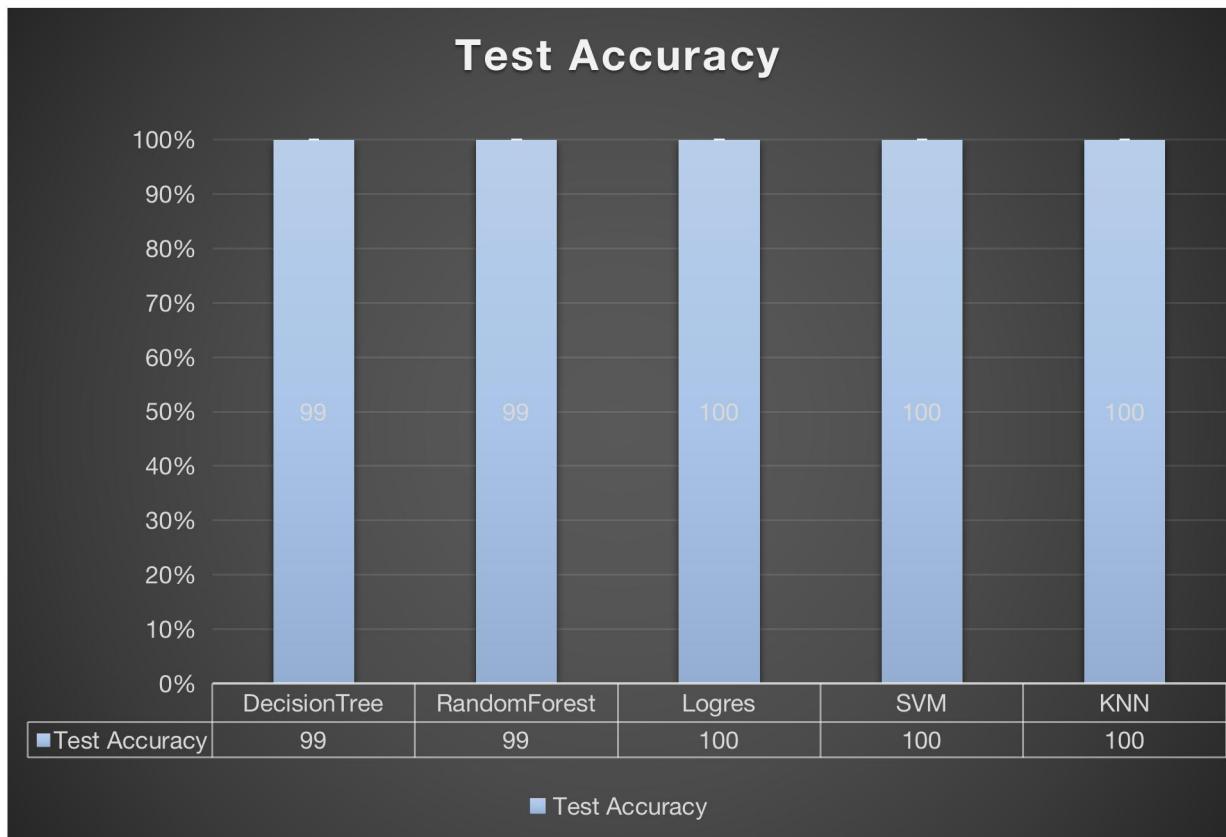


Figure 6.3 shows the test accuracy result of the experiment

Decision Tree and random forest's accuracy is approximately 98% and performance for the system. The experimental results indicate an accuracy of 100% for the Logistic Regression, SVM, and KNN.

6.2 Performance metrics

However, since the test and training dataset has been used, the tested processing of the model using that dataset is too. The three performances have used evaluation metrics, i.e., Accuracy, Precision (weighted avg.), and F1-Score (weighted avg.) in our experiment. The precision and F1-Score of each class are weighted by the number of samples from that class in our test set. The metrics can be defined as follows:

a) Decision Tree, KNN, SVM, RF Classifier - Accuracy

The accuracy reaches around 0.95 to 1, which is measured using the following data given below.

```
y_pred=classifier.predict(X_test)  
confidence_level=metrics.accuracy_score(y_test, y_pred)
```

$$\text{Accuracy} = \frac{(TN + TP)}{(TN+TP+FN+FP)}$$

(6.1)

b) Confusion Matrix

It is a table that is often used to measure the performance scale of a classifier model on a set of test data from which the true values are noted. It is relatively easy to understand, but the related terms could be misleading.

c) Precision

Based on the field that relevant document refers to symptoms supplied and retrieved documents related to symptoms provided by the patient, precision is the proportion of symptoms that are pertinent to the query.

$$\text{Precision} = \frac{TP}{TP + FP}$$

(6.2)

d) Recall

It is based on the field that relevant document refers to symptoms given, and retrieved documents refer to symptoms provided by the patient, so recall is the fraction of the symptoms given that are successfully retrieved.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(6.3)

e) F-score: After collecting the values for precision and recall, the F score is used to calculate accuracy.

Support

Support is the number of data values provided in each set of test accuracy.

$$F1\ Score = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

(6.4)

Where:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

Determine the accuracy and the macro and weighted average in the table supplied.

Table 6.2 Algorithm evaluation metric results

Classification Algorithm	Accuracy	Precision	F1-Score
Decision Tree	0.99	0.99	0.98
KNN	1.00	1.00	1.00
SVM	1.00	1.00	1.00
Random Forest	0.99	0.99	0.98
Logistic Regression	1.00	1.00	1.00

SVM, KNN and Logistic Regression give the highest accuracy for the system. Thus, the SVM has been used as the primary classification algorithm for the system.

Although all the classifiers have shown outstanding performances with high precision, table 6.2 demonstrates in our experiment that KNN, SVM and Logistic Regression show the best performance with the highest precision of 100 per cent. In addition, Logistic Regression, SVM and KNN show the best performance with an F1-score rate of 100 per cent, whereas Decision Tree and Random Forest offers the worst performance in this case with a 98 per cent of F1-score rate. This should be the core classifier of the system.

6.3 Chatbot Interface

Conversational interfaces are platforms that mimic a conversation with a real human. The models have been designed with a Conversational User Interface (CUI) that focuses on linguistic cues to make the conversation interactive, user-friendly, and more intuitive. [91] These are some vital elements that are considered while designing the CUI for our conversational agent, like the look and feel of the Chatbot, notifications to engage its users, ease in getting responses, good NLP support, keeping conversation brief, and reducing users' monotony.

The TKINTER package for Python is used to develop a more efficient and user-friendly GUI for health chat. Tkinter often comes with Python integration, utilises Tk, and is a standard Python GUI framework. It is renowned for its simplicity and user-friendliness. It is open-source, accessible, and distributed under a Python licence [21].

a) Login to System:

Following Figure 6.4 shows the UI Home Screen of the Healthcare Chatbot Application.

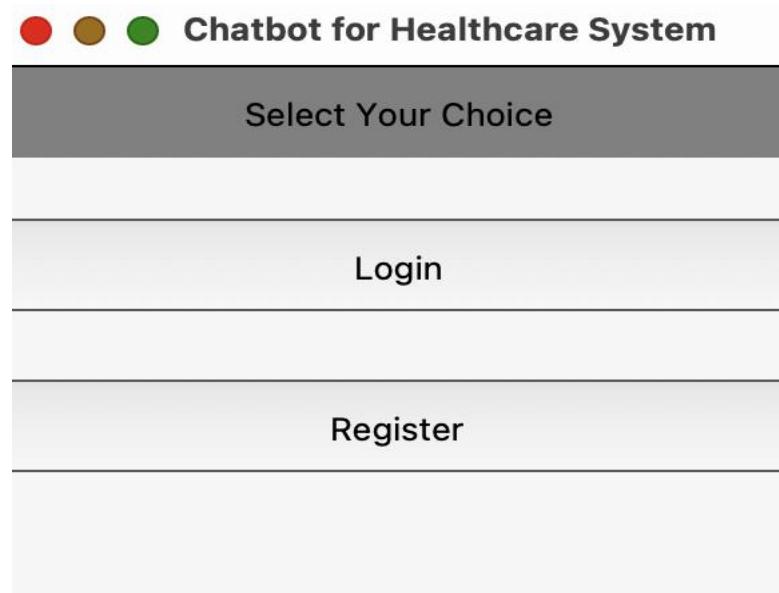


Figure 6.4 Home Screen

b) Consumer registrations

The operation of chatbot devices requires consumer registrations. Over time, these insights were used to help patients get more individualised treatment.

The user inputs the symptoms into an input area, and the input text is pre-processed to conform to the program's specifications. The phrases given in the document are used to identify and convey the indications for clarifying purposes.

The application consists of Register and Login page:



The image shows a registration form titled "Register". At the top left are three colored circles: red, orange, and green. Below the title is a instruction message: "Please enter the details below". There are two text input fields: one for "Username *" and one for "Password *". Both fields have placeholder text and are preceded by a small icon. At the bottom is a large blue "Register" button.

Figure 6.5 Registration Screen

If the user is new, they must enter details to register Ref. Figure 6.5, and after registration, they are directed to the login window; on successful login, the user is directed to a window where they can start a conversation with the BOT.

c) Review of the question-and-answer program.

Once the user is finished and the system has determined that all questions have been satisfactorily answered, the system sends out a list of possible illnesses based on the provided symptoms [34]. Since this is a probabilistic system, we provide a list of the most likely ailments and their respective probabilities.

The implementation of the multilingual healthcare chatbot system is shown in Figure 6.6. This figure shows the user communicating with the chatbot system, which analyses the user's symptoms and predicts the disease using the Support Vector Machine (SVM). In the figure, the user enters symptoms corresponding to jaundice.

```
-----HealthCare ChatBot-----
Your Name?          ->19037876
Hello, 19037876

Enter the symptom you are experiencing      ->shivering,chills
Enter valid symptom.

Enter the symptom you are experiencing      ->3
Enter valid symptom.

Enter the symptom you are experiencing      ->vomiting
searches related to input:
0 ) vomiting
Okay. From how many days ? : 2
Are you experiencing any
ok to scroll output; double click to hide >S
stiff_neck ? : yes
swelling_joints ? : no
movement_stiffness ? : yes
painful_walking ? : no
It might not be that bad but you should take precautions.
You may have Arthritis
Arthritis is the swelling and tenderness of one or more of your joints. The main symptoms of arthritis are joint pain and stiffness, which typically worsen with age. The most common types of arthritis are osteoarthritis and rheumatoid arthritis.
Take following measures :
1 ) exercise
2 ) use hot and cold therapy
3 ) try acupuncture
4 ) massage
```

Figure 6.6 Chatbot answering queries of the user

Fig 6.6 shows the expected result. After processing all the above operations, the final answer to the user's query is obtained. In simple words, it catches keywords from the user's queries, starts performing with the system flow and generates the final result on the chatbot interface.

The model has been built and is working fine; adding more intents will help in assessing the user in a much better manner. Our Chatbot is trained with multiple responses so that whenever someone reuses the bot, they get new responses, not just the old ones, which will be accurate to the query. Also, multiple user inputs have been coded in the program to help with appropriate responses. Everyone is aware that chatbots are programmed to respond to any inquiry posed by the user. As a result, anytime a user writes a query to which the bot has an answer, a suitable response should be sent. This is done so that the Chatbot will return any wrong answer near to any tag but unrelated to the user's inquiry [95].

Table 6.3 Chat sample of a conversation with Chatbot

Question	Answer
Please reply Yes or No for the following symptoms runny_nose?	yes
[You may have Common Cold] symptoms present ['runny_nose'] symptoms given ['continuous_sneezing', 'chills', 'fatigue', 'cough', 'high_fever', 'headache', 'swelled_lymph_nodes', 'malaise', 'phlegm', 'throat_irritation', 'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'congestion', 'chest_pain', 'loss_of_smell', 'muscle_pain'] a confidence level is 0.058823529411764705	

6.4 Benefits, Challenges and Limitations

Chatbots are still in the early phases of development and heavily rely on artificial intelligence and machine learning, so there are still some limitations or challenges they have faced in recent years [3].

6.4.1 Advantages

a) Accessible anytime:

They replace live chat, emails, and phone calls. They'll operate year-round without breaks. This improves UX and boosts industry rankings. You may also construct this fast answer to retain your image and brand.

b) Handling Capacity:

Chatbots, in contrast to humans, who can only discuss with one other person at a time, can simultaneously converse with thousands of individuals. It does not matter what time of day it is or how many individuals are trying to get in touch with you; each one of their inquiries will be handled without delay. [97] Using a chatbot would eliminate this issue, allow for better service to every customer, and guarantee that no orders are forgotten.

c) Flexible attribute:

One of the main advantages of chatbots is how simply they can be used in any industry. To make a bot work in a different domain, you must teach it how to have conversations in a new way. In contrast to other products, such as those requiring extensive testing and development before release, chatbots may be easily ported to several platforms.

d) Customer Satisfaction:

Emotions are unable to be controlled by humans. On the other hand, chatbots are bound by certain norms and follow them for as long as they are trained to do so. No matter how rude or offensive a consumer may be or how foul language they use, they will always handle the customer appropriately [96]. Customers like this kind of easy conversation and want every one of their dealings to be as simple as possible.

e) Cost Effective:

Because of the inherent limitations of human nature, a single human being can only effectively juggle the needs of one or two other persons simultaneously. Any more than that would be very difficult for the employee to handle. Chatbots could assist in finding a solution to this age-old dilemma. Because one Chatbot is equivalent to many workers, it can readily interact with thousands upon thousands of clients simultaneously.[82]

f) Faster Onboarding:

Every level of the organisational structure that an employee will go through includes a component of ongoing instruction. Additionally, there will be a lot of changes in the personnel; some will remain, some will be dismissed, some new employees will join, and so on.[98] That is to say that it is inevitable that some employees will go, and

others will be hired. Chatbots can reduce this wait time drastically, but only if they adhere to a clear and straightforward conversational format. There will, of course, be adjustments to make here as well, but they should take just a tiny fraction of the time that employees would.

6.4.2 Disadvantages

a) Too many functions

Most programmers want to create a chatbot that can communicate with any user interface and act as a trustworthy personal assistant. However, functioning bots end up not being able to handle most inquiries. [39] They have a few drawbacks, including a tendency to misunderstand the user, forget what they were taught only five minutes before, etc. Even in this scenario, it is vital to update and enhance such applications while they are in use.

b) Primitive algorithms

There are two categories of bots: those based on artificial intelligence and capable of learning during conversation and those built for certain behaviour situations. Chatbots powered by artificial intelligence are favoured because of their adaptability to different scenarios. However, to achieve this goal, sophisticated algorithms must be created. As a result, it would be preferable for regular businesses to concentrate on the second kind of bots since they are more dependable and need less effort to operate. Because they do not have intellect, they will not be able to engage in impolite modes of communication or break free from their creators' authority.

c) Complex interface

Conversation with a bot takes place in a chat environment, which requires the user to produce a significant amount of text. And if the bot cannot comprehend the user's request, the user will be required to write even more. In the future, virtual assistants may become even more helpful thanks to the addition of a speech recognition feature. However, their functional capabilities are limited at the moment, and there are only a select number of commercial domains in which they may be of any real

value. Therefore, conversing with a chatbot is not likely to result in time savings in most circumstances.

6.5 Discussion

In this study, chatbot health problems were predicted. The reason natural language processing (NLP) is utilised is so that the machine can speak the user's language. Thus, the SVM algorithm and the sickness indicators system may predict outbreaks.**[24]** The user may go to the appropriate displayed response and submit it for review. Chatbots have the advantage of providing invisible medical aid and advice.

For building chatbot health apps and a collaborative user-centric design strategy. While chatbots have been discussed in health [2], [3] theory-based development approaches are not commonly used. Developing chatbots may be challenging, requiring training and meeting user demands [4]. The recommended prototype enables stakeholders to be involved early in the process and to give training data resulting in a "trained" chatbot that can be utilised in regular healthcare practice. By responding to various user inquiries, HealthBot advances our comprehension of user queries in the context of connected data.

CHAPTER 7: CONCLUSION

7.1 Conclusion

Advances in natural language processing have helped fuel the rise of chatbots, which are becoming as common as voice assistants. This is the most excellent choice for those who have demanding work schedules. They do not need to wait in line for hours to see a doctor every time; instead, they may communicate with the bot. Our medical chatbot assists patients with common disorders such as fever, cold, typhoid, malaria, jaundice, etc.

This paper's suggested system is straightforward. This technique is effective when a speedy diagnosis is required since the most common disorders are readily cured; it relieves the load on medical personnel to concentrate on more difficult-to-solve issues. Because patients are typically not medically trained and do not know the terminology, the NLP (natural language processing) system is sophisticated enough to grasp what they are trying to say, which improves the program's reach and helps a lot more people. Furthermore, [56] In this research, Healthcare Chatbot uses machine learning and natural language understanding technologies, such as named entity recognition, factoids, repeated queries, and conversation management. Usability testing reveals that the suggested Healthcare Chatbot has enhanced the end-to-end user experience regarding interactive question responding and performance. [87] It is more convenient for information retrieval, acquisition, intent classification, query interpretation, and continuous learning.

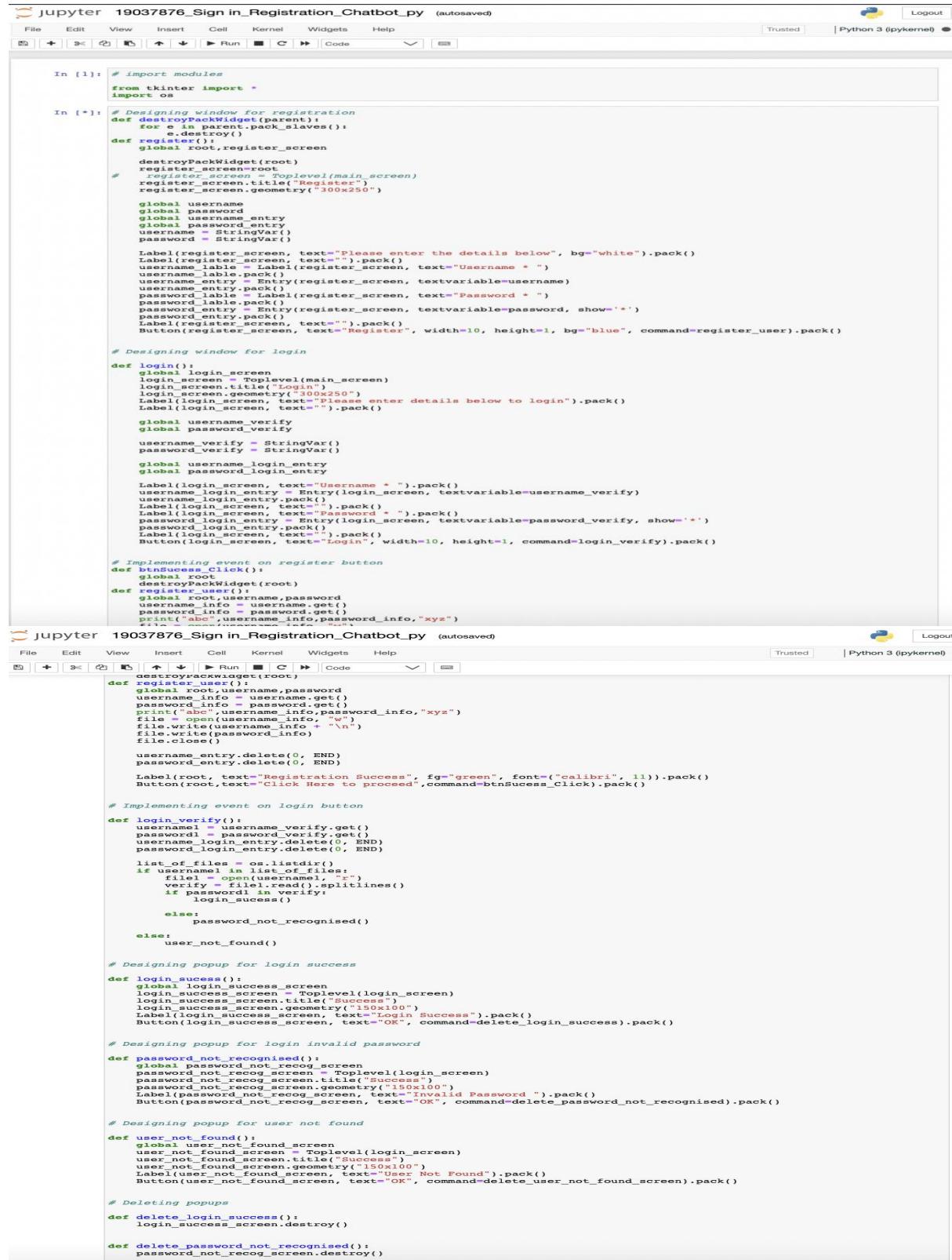
7.2 Suggestions for Future Works

Healthcare Chatbot is a terrific chat solution for patients who require immediate, hassle-free support. The technology identifies API releases and drug names. There are several possible options for future studies, such as a medical centre or clinic system to help consumers by inquiring about medicinal objectives. [85] Further study in this area may reveal new bot and automated software properties that may be used to identify and prevent malware. People must adapt to comprehend the information as the Web's intricacies grow. The Metaverse, in particular, will combine AI and conversational agents to encourage and replicate realistic learning situations while running in tandem with cloud computing, edge computing, 5G, and computer vision.

According to a study [8], roughly 5.19 billion unique smartphone users, 4.54 billion internet users through smartphones, and 3.80 billion people actively utilising social media. Every internet user spends six to seven hours every day online on average. With the increasing number of consumers on online platforms, servers provide an incredible platform for companies to engage with their customers. Medical information is compassionate and should not be disclosed to anybody. No one should access the user's medical records [57]. Just encoded data should be shown if someone enters a website without providing helpful information. To avoid the hassle of texting, the concept of a voice-based chatbot system may be enhanced so that anybody can ask their inquiry and obtain an answer on the chatbot.

APPENDIX 1: GENERATING CHATBOT IN PYTHON

APPENDIX 1.1 The figures below show configuring the input Data and Tkinder used for create login and registration on the application about disease prediction



```

  In [1]: # import modules
    from tkinter import *
    import os

  In [2]: # Designing window for registration
    def destroyPreviousWidget(parent):
        e = parent.pack_slaves()
        e.destroy()
    def register():
        global root,register_screen
        destroyPackWidget(root)
        register_screen = Toplevel(main_screen)
        register_screen.title("Register")
        register_screen.geometry("300x250")
        global username
        global password
        global username_entry
        global password_entry
        username = StringVar()
        password = StringVar()

        Label(register_screen, text="Please enter the details below", bg="white").pack()
        username_label = Label(register_screen, text="Username * ")
        username_label.pack()
        username_entry = Entry(register_screen, textvariable=username)
        username_entry.pack()
        password_label = Label(register_screen, text="Password * ")
        password_label.pack()
        password_entry = Entry(register_screen, textvariable=password, show='*')
        password_entry.pack()
        Label(register_screen, text="").pack()
        Button(register_screen, text="Register", width=10, height=1, bg="blue", command=register_user).pack()

    # Designing window for login
    def login():
        global login_screen
        login_screen = Toplevel(main_screen)
        login_screen.title("Login")
        login_screen.geometry("300x250")
        Label(login_screen, text="Please enter details below to login").pack()
        Label(login_screen, text="").pack()
        global username_verify
        global password_verify
        username_verify = StringVar()
        password_verify = StringVar()

        global username_login_entry
        global password_login_entry

        Label(login_screen, text="Username * ").pack()
        username_login_entry = Entry(login_screen, textvariable=username_verify)
        username_login_entry.pack()
        Label(login_screen, text="Password * ").pack()
        password_login_entry = Entry(login_screen, textvariable=password_verify, show='*')
        password_login_entry.pack()
        Label(login_screen, text="").pack()
        Button(login_screen, text="Login", width=10, height=1, command=login_verify).pack()

    # Implementing event on register button
    def btnSucess_Click():
        global root
        destroyPackWidget(root)
    def register_user():
        global root,username,password
        username_info = username.get()
        password_info = password.get()
        print("abc",username_info,password_info,"xyz")
        file = open(username_info, "w")
        file.write(username_info + "\n")
        file.write(password_info)
        file.close()

        username_entry.delete(0, END)
        password_entry.delete(0, END)
        Button(root, text="Registration Success", fg="green", font=("calibri", 11)).pack()
        Button(root, text="Click Here to proceed", command=btnSucess_Click).pack()

    # Implementing event on login button
    def login_verify():
        username_ = username_verify.get()
        password_ = password_verify.get()
        username_.login_entry.delete(0, END)
        password_.login_entry.delete(0, END)

        list_of_files = os.listdir()
        if username_ in list_of_files:
            f = open(username_, "r")
            verify = f.readline().splitlines()
            if password_ in verify:
                login_sucess()
            else:
                password_not_recognised()
        else:
            user_not_found()

    # Designing popup for login success
    def login_sucess():
        global login_success_screen
        login_success_screen = Toplevel(login_screen)
        login_success_screen.title("Success")
        login_success_screen.geometry("100x100")
        Label(login_success_screen, text="Login Success").pack()
        Button(login_success_screen, text="OK", command=delete_login_success).pack()

    # Designing popup for login invalid password
    def password_not_recognised():
        global password_not_recog_screen
        password_not_recog_screen = Toplevel(login_screen)
        password_not_recog_screen.title("Success")
        password_not_recog_screen.geometry("100x100")
        Label(password_not_recog_screen, text="Invalid Password").pack()
        Button(password_not_recog_screen, text="OK", command=delete_password_not_recognised).pack()

    # Designing popup for user not found
    def user_not_found():
        global user_not_found_screen
        user_not_found_screen = Toplevel(login_screen)
        user_not_found_screen.title("Success")
        user_not_found_screen.geometry("150x100")
        Label(user_not_found_screen, text="User Not Found").pack()
        Button(user_not_found_screen, text="OK", command=delete_user_not_found_screen).pack()

    # Deleting popups
    def delete_login_success():
        login_success_screen.destroy()

    def delete_password_not_recognised():
        password_not_recog_screen.destroy()

    def delete_user_not_found():
        user_not_found_screen.destroy()

```

```

def delete_user_not_found_screen():
    user_not_found_screen.destroy()

# Designing Main(first) window

def main_account_screen(frmmain):
    main_screen=frmmain

    main_screen.geometry("300x250")
    main_screen.title("Chatbot for Healthcare System")
    Label(main_screen,text="Select Your Choice", bg="grey", width="300", height="2", font=("Calibri", 13)).pack()
    Label(main_screen,text="").pack()
    Button(main_screen,text="Login", height="2", width="30", command=login).pack()
    Label(main_screen,text="").pack()
    Button(main_screen,text="Register", height="2", width="30", command=register).pack()

root = Tk()
main_account_screen(root)

root.mainloop()

```

APPENDIX 1.2 The figures below show configuring the input Data used for training and testing to disease prediction by Machine Learning and results

The screenshot shows a Jupyter Notebook interface with the following content:

Task Highlights :

- Perform Supervised Machine Learning on Disease Prediction dataset <https://www.kaggle.com/datasets/tntdoan/19037876-ml-deseaseprediction>
- Perform data visualization
- Make use of different algorithms to predict target label
- Show 3 different algorithms' accuracies with the help of graphs

The main problems that this data science task is supposed to solve have been correctly framed in terms of the client's goals, background information, and the task's purpose. This makes sure that the task is understood and looked into so that the decision-making process can be better informed about the different ways to solve the problems.

This work will harvest case study data that is relevant, representative, and adequate from a credible and dependable web source. To provide trustworthy and acceptable results and outputs, appropriate preprocessing modifications and data exploration will be conducted on the data. The popular classifier models of Decision Tree, Random Forest, Logistic Regression, K Nearest Neighbor and SVM will be fitted, analysed, and evaluated in terms of the performance metrics of accuracy, precision, recall, and F1-score in predicting disease classifications during the data mining and modelling process. All relevant interpretations and observations will be recorded and taken into account for future enhancements.

Keeping in mind the goal of this task, the main focus will be on disease-related factors, which are a set of symptoms that can be represented by a number between 0 and 1. By looking at these, we can find areas of concern and predict how the disease will progress.

Data Collection

The first step of a data science task is to obtain, gather, and measure the necessary and targeted data from available internal or external data sources, and then compiled into an established system. In this case, version 1 of Disease Prediction dataset by 19037876 in Kaggle is used. The Excel csv file that was extracted as at 1 April 2021 for the purpose of this task is available at <https://www.kaggle.com/datasets/tntdoan/19037876-ml-deseaseprediction>

```

In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

/kaggle/input/disease-prediction-using-machine-learning/Training.csv
/kaggle/input/disease-prediction-using-machine-learning/Testing.csv

```

Content

- Import Data
- Data Cleaning

Project Title: Chatbot Using Machine Learning and NLP in Healthcare Predictions

Student ID: 19037876- UFCF9Y-60-M

- Data Cleaning
 - KNN
 - Model Complexity
 - Cross Validation
- Decision Tree Classifier
- Random Forest Classifier
- Support Vector Classifier
- Logistic Regression
- Metrics

Import Data

- First look at data
- We must read our csv files from directory

```
In [2]: data_dir = "../input/disease-prediction-using-machine-learning"
train_df = pd.read_csv(data_dir + "/Training.csv")
test_df = pd.read_csv(data_dir + "/Testing.csv")
train_df.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	skin_peeling
0	1	1		1	0	0	0	0	0	0	0	0	0
1	0	1		1	0	0	0	0	0	0	0	0	0
2	1	0		1	0	0	0	0	0	0	0	0	0
3	1	1		0	0	0	0	0	0	0	0	0	0
4	1	1		1	0	0	0	0	0	0	0	0	0

5 rows × 134 columns

The dataframe format type will facilitate the use of a wider variety of syntax and methods for data analysis, including describe() and info().

Regarding the attributes included in Disease Prediction dataset, there are 132 different disease symptoms, the first 3 recorded being 'itching', 'skin rash', and 'nodal skin eruptions'. The target class label is 'prognosis'.

Data Preprocessing

Data preprocessing is a type of data mining that turns raw data into a format that is easy to understand. This process has four main steps: cleaning the data, putting the data together, changing the data, and getting rid of some of the data.

Data cleaning will filter, find, and deal with bad data to make sure that both the data and the analysis results are good. In this case, there may be noises of impossible and extreme values, outliers, and missing values. There might be data that isn't right or attributes and data that are repeated.

As the first step, the dataset's null values will be found and, if possible, replaced with correct values.

```
In [3]: train_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 134 entries, itching to Unnamed: 133
```

Data Modelling

The training set and the test set are the two sets that make up the two datasets. They both have the same attributes, but their values are different. The classification models are trained and built with the help of the training set. The test set is used to predict the classifications of new, unbiased data that wasn't used to train the model. Based on the accuracy, precision, recall, and F1-score of those classifications, the model's performance is then judged.

The number of target class labels in each dataset is almost the same. It is assumed that the training and test sets are fair and accurate representations of the target classes. For example, they may have already been split randomly using a list of random numbers starting from a randomly chosen position. There are 4920 times in the training subset, but only 43 times in the test subset.

For the Logistic Regression, K Nearest Neighbor and SVM, information patterns are fed into the model through the input units. These patterns then trigger the many layers of hidden units, which send the information to the output units. The output is the values that were worked out for the nodes in the output layer.

The Decision tree is built using the parameters of the best split strategy and the entropy criterion, which uses information gain to choose the next node based on which features are more important. This improves the quality of the splits. There can't be more than 1 leaves at most. The decision tree is used to get the rules for classifying things. These are decided by the order in which the flow goes from the root node and its branches to the internal or decision nodes. The flow stops when it gets to the leaf node, which is the class label.

Random Forest classifier will combine multiple base models of Decision Trees during its training period, using the strategy of ensemble machine learning methods. This will construct a single optimal predictive model, with the criterion of entropy which utilises information gain to iteratively select the next node according to higher feature importance to optimise the quality of splits. The minimum number of leaves are restricted to 1. The final output may be the mode class or the mean prediction of individual trees.

```
i]: # Modelling
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 3) # k = 5
x_train, y_train = train_df.loc[:,train_df.columns != "prognosis"], train_df.loc[:,["prognosis"]]
x_test, y_test = test_df.loc[:,train_df.columns != "prognosis"], test_df.loc[:,["prognosis"]]
knn.fit(x_train, y_train)
prediction = knn.predict(x_test)
print("Prediction list: {}".format(prediction[0:20]))
print("With KNN (K=5) accuracy is: ",knn.score(x_test, y_test))

Prediction list: ['Fungal infection' 'Allergy' 'GERD' 'Chronic cholestasis' 'Drug Reaction'
 'Peptic ulcer disease' 'AIDS' 'Diabetes' 'Gastroenteritis'
 'Bronchial Asthma' 'Hypertension' 'Migraine' 'Cervical spondylosis'
 'Paralysis (brain hemorrhage)' 'Jaundice' 'Malaria' 'Chicken pox'
 'Dengue' 'Typhoid' 'hepatitis A']
With KNN (K=5) accuracy is: 1.0
```

Model Evaluation

The model performance is evaluated and validated by using the test set of 43 records to predict the classifications of these new unbiased data that were not used to train the model. The confusion matrix is then used to determine the performance metrics of accuracy, precision, recall, and F1-score, based on those classifications. The supports are 1 instance for each target class label of prognosis. This process ensures that the models are useful by being generalisable even when the specific training data used is extended to include new test data, or in technical terms 'avoid over fitting'.

Model Complexity

In fact, this dataset doesn't require tuning, but I want to try basics of complex modelling

```
i]: neighbors = np.arange(1,25) # for k tuning
train_accuracy = []
test_accuracy = []
# Loop for different k values
for i, k in enumerate(neighbors):
    # k nearest neighbor classifier
    knn = KNeighborsClassifier(n_neighbors=k)
```

Project Title: Chatbot Using Machine Learning and NLP in Healthcare Predictions

Student ID: 19037876- UFCF9Y-60-M

jupyter 19037876_Supervised Machine Learning on Disease Prediction Last Checkpoint: 38 phút trước (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

Model Complexity

In fact, this dataset doesn't require tuning, but I want to try basics of complex modelling

```
In [15]: neighbors = np.arange(1,25) # for k tuning
train_accuracy = []
test_accuracy = []
# Loop for different k values
for i, k in enumerate(neighbors):
    k -= 1 # to exclude
    knn = KNeighborsClassifier(n_neighbors = k)
    # fit the knn
    knn.fit(x_train, y_train)
    # train accuracy
    train_accuracy.append(knn.score(x_train, y_train))
    # test accuracy
    test_accuracy.append(knn.score(x_test, y_test))
```

```
In [16]: # Plot the accuracies
plt.figure(figsize = (13,8))
plt.plot(neighbors, train_accuracy, label = "Training accuracy")
plt.plot(neighbors, test_accuracy, label = "Testing accuracy")
plt.legend()
plt.title("Accuracy for both train and test data")
plt.xlabel("Number of Neighbors (k)")
plt.ylabel("Accuracy")
plt.xticks(neighbors)
plt.show()
```

Accuracy for both train and test data

Number of Neighbors (k)

Training accuracy
Testing accuracy

Cross Validation

```
In [17]: from sklearn.model_selection import cross_val_score
```

jupyter 19037876_Supervised Machine Learning on Disease Prediction Last Checkpoint: 38 phút trước (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

Cross Validation

```
In [17]: from sklearn.model_selection import cross_val_score
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
k = 5 # for k-fold cross validation
cv_result = cross_val_score(knn, x_train, y_train, cv = k) # uses R^2 score
print("CV scores: ", cv_result)
print("CV scores average: ", np.sum(cv_result)/len(cv_result))
```

CV scores: [1. 1. 1. 1. 1.]
CV scores average: 1.0

Decision Tree Classifier

```
In [18]: from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(random_state = 42)
dt.fit(x_train, y_train)
dt.predict(x_test)
dt.score(x_test, y_test)
```

Out[18]: 0.9761904761904762

Random Forest Classifier

```
In [19]: from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state = 42, n_estimators = 100)
rfc.fit(x_train, y_train)
rfc.predict(x_test)
rfc.score(x_test, y_test)
```

Out[19]: 0.9761904761904762

Support Vector Classifier

```
In [20]: from sklearn.svm import SVC
svc = SVC(C = "auto", kernel = "rbf" )
svc.fit(x_train, y_train)
svc.predict(x_test)
svc.score(x_test, y_test)
```

Out[20]: 1.0

Logistic Regression Classifier

```
In [21]: from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state = 42)
logreg.fit(x_train, y_train)
logreg.predict(x_test)
logreg.score(x_test, y_test)
```

Out[21]: 1.0

Project Title: Chatbot Using Machine Learning and NLP in Healthcare Predictions

Student ID: 19037876- UFCF9Y-60-M

```

Metrics

In [22]: y_predictions = {"KNN": knn.predict(x_test),
                 "SVC": svc.predict(x_test),
                 "DT": dt.predict(x_test),
                 "RFC": rfc.predict(x_test),
                 "LOGREG": logreg.predict(x_test)}

from sklearn.metrics import classification_report, confusion_matrix

for classifier, y_pred in y_predictions.items():
    cm = confusion_matrix(y_test, y_pred)
    print(classifier, 'Confusion matrix: \n', cm)
    print("-----")
    print(classifier, 'Classification report: \n', classification_report(y_test, y_pred))

KNN Confusion matrix:
[[1 0 0 ... 0 0 0]
[0 1 0 ... 0 0 0]
[0 0 1 ... 0 0 0]
...
[0 0 0 ... 1 0 0]
[0 0 0 ... 0 1 0]
[0 0 0 ... 0 0 1]]

KNN Classification report:
              precision    recall   f1-score   support
(vertigo) Paroxysmal Positional Vertigo    1.00     1.00     1.00      1
          AIDS           1.00     1.00     1.00      1
          Acne            1.00     1.00     1.00      1
Alcoholic hepatitis    1.00     1.00     1.00      1
          Allergy          1.00     1.00     1.00      1
          Arthritis         1.00     1.00     1.00      1
Bronchial Asthma       1.00     1.00     1.00      1
          ...             1.00     1.00     1.00      1

```

	precision	recall	f1-score	support
(vertigo) Paroxysmal Positional Vertigo	1.00	1.00	1.00	1
AIDS	1.00	1.00	1.00	1
Acne	1.00	1.00	1.00	1
Alcoholic hepatitis	1.00	1.00	1.00	1
Allergy	1.00	1.00	1.00	1
Arthritis	1.00	1.00	1.00	1
Bronchial Asthma	1.00	1.00	1.00	1
Cervical spondylosis	1.00	1.00	1.00	1
Chicken pox	1.00	1.00	1.00	1
Chronic cholestasis	1.00	1.00	1.00	1
Common Cold	1.00	1.00	1.00	1
Dengue	1.00	1.00	1.00	1
Diabetes	1.00	1.00	1.00	1
Dimorphic hemorrhoids(piles)	1.00	1.00	1.00	1
Drug Reaction	1.00	1.00	1.00	1
Fungal infection	1.00	1.00	1.00	2
GERD	1.00	1.00	1.00	1
Gastroenteritis	1.00	1.00	1.00	1
Heart attack	1.00	1.00	1.00	1

As we can see classification reports for each classifier, knn,svm and logistic regression's success score is 100% while decision tree and random forest's accuracies are approximately 98%.

The SVM has the best performance metrics of 100% for all four metrics of test accuracy, precision, recall, and F1-score. The KNN and Logistic Regression model also achieved 100% for all, but this is not preferred as the time complexity will usually be larger. On the other hand, the Decision Tree and Random Forest model have the precision of 99%, and 98% for all three of accuracy, recall, and F1-score. For all chosen models, all the prognosis are almost perfectly classified and predicted.

Although there are only small differences between the train and test accuracy and thus no overfitting in this sense, the 100% train accuracy still indicates that the learnt rules are specific for the train set and do not generalize well beyond the train set to the test set. This will be taken care of later on, when the best model is chosen.

Accuracy indicates the overall proportion of correct predictions for all the three classes. The train accuracy is measured based on examples that the model was constructed on, while the test accuracy is based on those it has yet to see. DT achieved 99% for both train accuracy and test accuracy.

However, it will be misleading to solely base decisions on this, as the dataset used is relatively small and biased. Recall and precision metrics are thus also considered to measure model performance.

Recall indicates the proportion of correct predictions for each individual class, out of the corresponding actual class. In other words, the proportion of all actual classes that were predicted correctly. It was found that 98% of all actual classes were predicted correctly by DT.

Precision indicates the proportion of correct predictions for each individual class, out of the corresponding predicted class. In other words, the proportion of all predicted classes were actually predicted correctly. It was found that 99% of all predicted classes were actually predicted correctly by DT.

However, recall and precision have an inverse relationship. In order to make them comparable for cases where they are both important, F1 score is introduced. F1-score, also known as F-score or F-measure, is used to make precision and recall comparable in cases where they are both important, by measuring their harmonic mean. This allows it to consider both metrics and punish extreme values more heavily. Therefore, F-score will compute the overall quality of translations produced by the chosen machine learning engine, which is 100% by the SVM model.

APPENDIX 1.3 The figures below show configuring the input Data used for training to find K-Fold disease prediction

```

jupyter 19037876_K-Fold_Desease Prediction Last Checkpoint: một giờ trước (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) Logout

In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python
# For example, here's several helpful packages to load https://www.kaggle.com/datasets/tna2doan
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under
import os
for dirname, _, filenames in os.walk('../kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

In [1]: # Importing libraries
import numpy as np
import pandas as pd
from scipy import stats
from sklearn import model_selection
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
%matplotlib inline

In [2]: # Reading the train.csv by removing the
# last column since it's an empty column
DATA_PATH = '../input/disease-prediction-using-machine-learning/Training.csv'
data = pd.read_csv(DATA_PATH).dropna(axis = 1)

# Checking whether the dataset is balanced or not
disease_counts = data['prognosis'].value_counts()
print(disease_counts)

temp_df = pd.DataFrame({'Disease': disease_counts.index, 'Counts': disease_counts.values})
plt.figure(figsize = (18,8))
sns.barplot(x = "Disease", y = "Counts", data = temp_df)
plt.xticks(rotation=90)
plt.show()

Fungal infection          120
Hepatitis C               120
Hepatitis E               120
Alcoholic hepatitis       120
Tuberculosis              120
Common Cold                120
Pneumonia                  120
Dimorphic hemimeloidiasis 120

```

```

jupyter 19037876_K-Fold_Desease Prediction Last Checkpoint: một giờ trước (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) Logout

In [3]: # Encoding the target value into numerical
# value using LabelEncoder
encoder = LabelEncoder()
encoder.fit(data['prognosis'])
data['prognosis'] = encoder.transform(data['prognosis'])

In [4]: x = data.iloc[:, :-1]
y = data.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 24)
print(f"Train: {X_train.shape}, {y_train.shape}")
print(f"Test: {X_test.shape}, {y_test.shape}")

Train: (3936, 132), (3936, 1)
Test: (984, 132), (984, 1)

In [9]: # Defining scoring metric for k-fold cross validation
def cv_scoring(estimator, X, y):
    return accuracy_score(y, estimator.predict(X))

# Initializing Models
models = {'SVC':SVC(), 'Gaussian NB':GaussianNB(), 'Random Forest':RandomForestClassifier(random_
# Producing cross validation score for the models
for model_name in models:
    model = models[model_name]
    scores = cross_val_score(model, X, y, cv = 10, n_jobs = -1, scoring = cv_scoring)
    print("====")
    print(model_name)
    print(f"Scores: {scores}")
    print(f"Mean Score: {np.mean(scores)}")

SVC
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
Gaussian NB
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
Random Forest
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
=====
KNN
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0

In [10]: # Training and testing SVM Classifier
svm_model = SVC()
svm_model.fit(X_train, y_train)
preds = svm_model.predict(X_test)

print(f"Accuracy on train data by SVM Classifier"
: (accuracy_score(y_train, svm_model.predict(X_train))*100))
```

APPENDIX 1.4 The figures below show configuring the input Data used for training and testing to create conversion interface about disease prediction

Jupyter 19037876_SVM_chatbot.py Last Checkpoint: 08/09/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) 0

In [2]:

```
import re
import pandas as pd
import pytsxs3
from sklearn import preprocessing
from sklearn import tree
from sklearn import model_selection
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.svm import SVC
import csv
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

/Users/alex_le/opt/anaconda3/lib/python3.9/site-packages/scipy/_init_.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Scipy (detected version 1.23.2)
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [3]:

```
training = pd.read_csv('Data/Training.csv')
testing = pd.read_csv('Data/Testing.csv')
cols = training.columns
cols = cols[:-1]
x = training[cols]
y = training['prognosis']
y1 = y
```

In [4]:

```
reduced_data = training.groupby(training['prognosis']).max()

#mapping strings to numbers
le = preprocessing.LabelEncoder()
le.fit(y)
y = le.transform(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
testx = testing[cols]
testy = testing['prognosis']
testy = le.transform(testy)

clf1 = DecisionTreeClassifier()
clf1 = clf1.fit(x_train,y_train)
# print(clf1.score(x_train,y_train))
# print ("cross result====")
scores = cross_val_score(clf1, x_test, y_test, cv=3)
# print (scores)
print (scores.mean())

model=SVC()
model.fit(x_train,y_train)
print("for svm:")
print(model.score(x_test,y_test))

importances = clf1.feature_importances_
indices = np.argsort(importances)[-1:-11:-1]
```

jupyter 19037876_SVM_chatbot_py Last Checkpoint: 08/09/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) O

Run C Code

```

# readn(f"You may have {present_disease[0]}")
# readn(f"{description_list[present_disease[0]]}")

else:
    print("You may have ", present_disease[0], "or ", second_prediction[0])
    print(description_list[present_disease[0]])
    print(description_list[second_prediction[0]])

# print(description_list[present_disease[0]])
precution_list=precautionDictionary[present_disease[0]]
print("Take following measures : ")
for i,j in enumerate(precution_list):
    print(i+1,".",j)

# confidence_level = (1.0*len(symptoms_present))/len(symptoms_given)
# print("confidence level is " + str(confidence_level))

    recurse(0, 1)
getSeverityDict()
getDescription()
getprecautionDict()
 getInfo()
tree_to_code(clf,cols)
print("-----HealthCare ChatBot-----")

```

Your Name? ->19037876
Hello, 19037876

Enter the symptom you are experiencing ->shivering,chills
Enter valid symptom.

Enter the symptom you are experiencing ->3
Enter valid symptom.

Enter the symptom you are experiencing ->vomiting
searches related to input:
0) vomiting
Okay. From how many days ? : 2
Are you experiencing any
muscle_weakness ? : yes
stiff_neck ? : yes
swelling_joints ? : no
movement_stiffness ? : yes
painful_walking ? : no
It might not be that bad but you should take precautions.
You may have Arthritis
Arthritis is the swelling and tenderness of one or more of your joints. The main symptoms of arthritis are joint pain and stiffness, which typically worsen with age. The most common types of arthritis are osteoarthritis and rheumatoid arthritis.
Take following measures :
1) exercise
2) use hot and cold therapy
3) try acupuncture
4) massage

APPENDIX 1.5

The figures below show configuring the input Data used for training and test to create conversion interface on the web-browser about disease prediction

```

In [1]: # Importing the libraries
from tkinter import *
from tkinter import messagebox
import os
import webbrowser

import numpy as np
import pandas as pd

In [2]: class HyperlinkManager:
    def __init__(self, text):
        self.text = text
        self.text.tag_config("hyper", foreground="blue", underline=1)
        self.text.tag_bind("hyper", "<Enter>", self._enter)
        self.text.tag_bind("hyper", "<Leave>", self._leave)
        self.text.tag_bind("hyper", "<Button-1>", self._click)

    def reset(self):
        self.links = {}

    def add(self, action):
        # add an action to the manager. returns tags to use in
        # associated text widget
        tag = "hyper-%d" % len(self.links)
        self.links[tag] = action
        return "hyper", tag

    def _enter(self, event):
        self.text.config(cursor="hand2")

    def _leave(self, event):
        self.text.config(cursor="")

    def _click(self, event):
        for tag in self.text.tag_names(CURRENT):
            if tag[6] == "hyper":
                self.links[tag]()
                return

# Importing the dataset
training_dataset = pd.read_csv('Training.csv')
test_dataset = pd.read_csv('Testing.csv')

# Slicing and Dicing the dataset to separate features from predictions
X = training_dataset.iloc[:, 0:132].values
Y = training_dataset.iloc[:, -1].values

# Dimensionality Reduction for removing redundancies
dimensionality_reduction = training_dataset.groupby(training_dataset['prognosis']).max()

# Encoding String values to integer constants
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

```

```

In [3]: class SignUp:
    def __init__(self, master):
        master.title("Register")
        master.geometry("300x250")
        self.createWidget()

    def createWidget(self):
        self.lblMsg = Label(self, text="Please enter details below", bg="grey")
        self.username_label = Label(self, text="Username * ")
        self.username_label.pack()
        self.username = StringVar()
        self.username_entry = Entry(self, textvariable=self.username)
        self.username_entry.pack()

        self.password_label = Label(self, text="Password * ")
        self.password_label.pack()
        self.password = StringVar()
        self.password_entry = Entry(self, textvariable=self.password, show='*')
        self.password_entry.pack()
        self.btnRegister = Button(self, text="Register", width=10, height=1, bg="grey", command=self.register_user)
        self.btnRegister.pack()

    def register_user(self):
        file = open(self.username_entry.get(), "w")
        file.write(self.username_entry.get() + "\n")
        file.write(self.password_entry.get())
        file.close()

        self.destroyPackWidget(SignUp.main_Root)
        self.lblSucess = Label(root, text="Registration Success", fg="green", font=("calibri", 11))
        self.lblSucess.pack()

        self.btnSucess = Button(root, text="Click Here to proceed", command=self.btnSucess_Click)
        self.btnSucess.pack()

    def btnSucess_Click(self):
        self.destroyPackWidget(SignUp.main_Root)
        frmQuestion = QuestionDigionosis(SignUp.main_Root)
        frmQuestion.pack()

    root = Tk()
    frmMainForm = MainForm(root)
    frmMainForm.pack()
    root.mainloop()

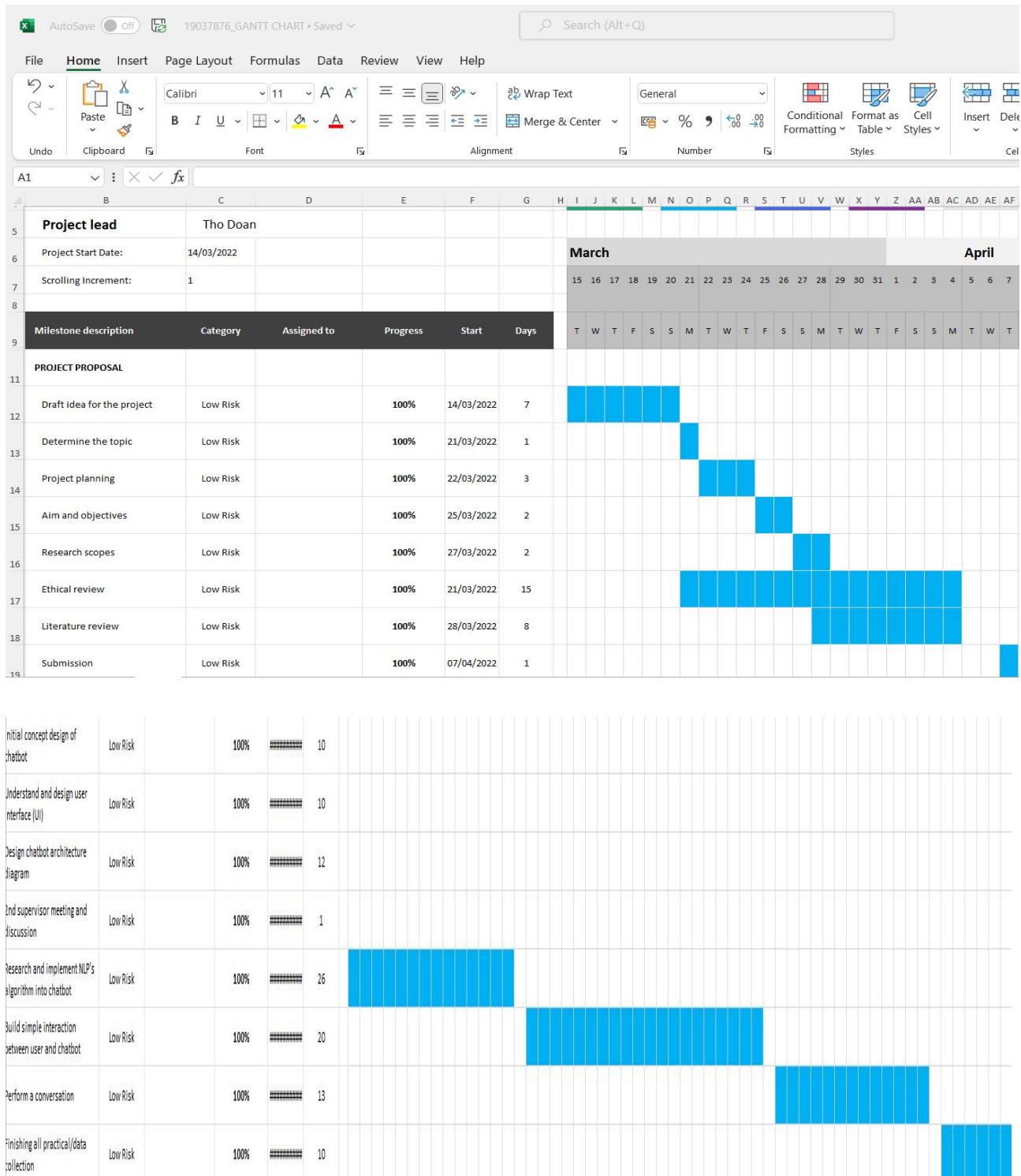
/Users/alex_le/opt/anaconda3/lib/python3.9/site-packages/scipy/_init_.py:146: UserWarning:
A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version
1.23.2
    warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")


SignUp Class

```

The diagram illustrates a mobile application interface for a Health Care Chatbot. At the top, there is a header section with three colored circles (red, brown, and green) followed by the text "Account Login". Below this is a large orange rectangular area containing the text "Health Care Chatbot". Underneath is a light gray rectangular area with the word "Login". Below that is another light gray rectangular area with the word "Register". A dark blue horizontal bar follows, with the text "Made by:" above a lighter blue bar containing the name "Tho Doan". The main body of the screen is a large white area. At the bottom, there is a light blue footer section. On the left side of this footer, there are two orange rectangular buttons labeled "Question" and "Diganosis". On the right side, there are two white rectangular buttons labeled "Yes" and "Start". At the very bottom center, there is a small white button labeled "No" and a white rectangular button labeled "Clear".

APPENDIX 2: PROJECT GANTT CHART



Project Title: Chatbot Using Machine Learning and NLP in Healthcare Predictions

Student ID: 19037876- UFCF9Y-60-M

Second stage						
Analysing costs and resources	Low Risk		100%	11/04/2022	3	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Investigating background knowledge of Natural Language Processing (NLP)	Low Risk		100%	14/04/2022	10	<div style="width: 90%;"><div style="width: 100%;"> </div></div>
1st supervisor meeting and discussion	Low Risk		100%	25/04/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Initial concept design of chatbot	Low Risk		100%	26/04/2022	10	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Understand and design user interface (UI)	Low Risk		100%	06/05/2022	10	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Design chatbot architecture diagram	Low Risk		100%	17/05/2022	12	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
2nd supervisor meeting and discussion	Low Risk		100%	30/05/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Research and implement NLP's algorithm into chatbot	Low Risk		100%	31/05/2022	26	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Build simple interaction between user and chatbot	Low Risk		100%	27/06/2022	20	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Perform a conversation	Low Risk		100%	18/07/2022	13	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Finishing all practical/data collection	Low Risk		100%	01/08/2022	10	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Final stage						

Build simple interaction between user and chatbot	Low Risk		100%	27/06/2022	20	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Perform a conversation	Low Risk		100%	18/07/2022	13	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Finishing all practical/data collection	Low Risk		100%	01/08/2022	10	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Final stage						
Completing testing chatbot	Low Risk		100%	12/08/2022	20	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Write evaluation and conclusion	Low Risk		100%	02/09/2022	4	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Update and compile appendices	Low Risk		100%	07/09/2022	3	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Final review	Low Risk		100%	11/09/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Plagiarism checking	Low Risk		100%	12/09/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Record video summary	Low Risk		100%	13/09/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Hand-in report and video	Goal		100%	14/09/2022	1	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
VIVA	Milestone		0%			<div style="width: 10%;"><div style="width: 100%;"> </div></div>

REFFENCES

[1] About: Chatbot available at

<https://dbpedia.org/page/Chatbot>

[2] Agrawal, R. Build a simple Chatbot using NLTK Library in Python (2021) available at

<https://www.analyticsvidhya.com/blog/2021/07/build-a-simple-chatbot-using-python-and-nltk/>

[3] Ahmed, I. and Singh, S., (2015), "*AIML Based Voice Enabled Artificial Intelligent Chatterbot*", in International Journal of u-and e-Service, Science and Technology Vol.8, No.2.

[4] Amato, F., and Marrone, S., (2018), "*Chatbots meet eHealth: automating healthcare*", Issue May.

[5] Athota, L., Shukla, V. K., Pandey, N. and Rana, A., (2020), "*Chatbot for Healthcare System Using Artificial Intelligence*," in 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 619-622, doi: 10.1109/ICRITO48877.2020.9197833.

[6] Athota, L., Shukla, V. K., Pandey, N., and Rana, A., (2020), "*Chatbot for Healthcare System Using Artificial Intelligence*" in 8th International Conference on Reliability, Infocom Technologies and Optimization Amity University.

[7] Badlani, S., Aditya, T., Dave, M., and Chaudhari, S., (2021), "*Multilingual Healthcare Chatbot Using Machine Learning*", in Emerging Technology (INCET)2021 2nd International Conference for, pp. 1-6.

[8] Bali, M., Mohanty, S., Chatterjee, S., Sarma, M., and Puravankara, R., (2019), "*Diabot: A predictive medical chatbot using ensemble learning* ", in International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8.

[9] Bates, M., (2019), "*Health Care Chatbots Are Here to Help*," in IEEE Pulse, vol. 10, no. 3, pp. 12-14, May-June 2019, doi: 10.1109/MPULS.2019.2911816.

[10] Bharadwaj, H., K., Agarwal, A., Chamola, V., Lakkaniga, N. R., Hassija, V., Guizani, M. and Sikdar, B., "A Review on the Role of Machine Learning in Enabling IoT Based Healthcare Applications", in Access IEEE, vol. 9, pp. 38859-38890, 2021.

[11] Bharti, U., Bajaj, D., Batra, H., Lalit, S., Lalit S. and Gangwani, A., (2020), "Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19," in 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 870-875, doi: 10.1109/ICCES48766.2020.9137944.

[12] Bishop, C. M., (1995)," *Neural Networks for Pattern Recognition.*.", in New York: Oxford University Press.

[13] BORKAN, "www.blog.casper.com," 21 september 2016. [Online].

Available: [https://blog.casper.com/meet insomnobot-3000/](https://blog.casper.com/meet-insomnobot-3000/).

[14] Bérubé, C., Schachner, T., Keller, R., Fleisch, E., Wangenheim, F., Barata, F. and Kowatsch, T." *Voice-Based Conversational Agents for the Prevention and Management of Chronic and Mental Health Conditions: systematic Literature Review.*", in Journal of Medical Internet Research 2021;23(3): e25933

[15] Cohen, P.R., (1995)," Empirical Methods in Artificial Intelligence. Cambridge", in MA: MIT Press.

[16] Comendador, B. E., Francisco, B. M., Medinilla, J. S., Nacion, S. M., & Serac, T. B. (2015), "Pharmabot: A Pediatric Generic Medicine Consultant Chatbot", in Journal of Automation and Control Engineering, 3(2), 137-140.doi:10.12720/joace.3.2.137-140.

[17] Dahiya, M., (2017), (30 May 2017), "A Tool of Conversation: Chatbot", in Dept. of Computer Science, Maharaja Surajmal Institute, Janakpuri, India, JCSE International Journal of Computer Sciences and Engineering.

[18] DataRobot," *What is Natural Language Processing? Introduction to NLP*", (2016) general at

<https://www.datarobot.com/blog/what-is-natural-language-processing-introduction-to-nlp/>

[19] Devi, S., Merchant, Z. A. M., Siddiqui, M. S., and Lobo, M., (1_4, 2019), "Artificial intelligence based personal assistant," in *Asian J. Converg. Technol.*, vol. 5, no. 1, pp.

[20] Dharwadkar, R., and Deshpande, A. N., (2018), "A Medical ChatBot". in International Journal of Computer Trends and Technology (IJCTT) V60(1):41-45 June 2018. ISSN:2231-2803.

[21] Dhyani, M., Kumar, R., and Bajaj, G. L., (16 May 2020), "An intelligent Chatbot using deep learning with Bidirectional RNN and attention model" Elsevier, in Institute of Technology and Management, Greater Noida, Uttar Pradesh, India.

[22] Divya, S., Indumathi, V., Ishwarya, S., Priyasankari, M., and Devi, K. S., (2018),"A Self-Diagnosis Medical Chatbot Using Artificial Intelligence", in *J. Web Dev. Web Des.*, vol. 3, no. 1, pp. 1-7.

[23] D. S., Hormansyah, E. L., Amalia, I., Aulia, (1 December 2018), "N-gram Accuracy Analysis in the Method of Chatbot Response.", In Computer Science International Journal of Engineering & Technology.

[24] DK, S. *Chatbot using the NLTK library| Build Chatbot in Python using NLTK* (2020) available at

<https://datapeaker.com/big-data/oltp-frente-a-olap-diferencia-entre-oltp-y-olap/>

[25] Fernandes, S., Gawas, R., Alvares, P., Femandes, M., Kale, D. and Aswale, S., (2020),"Survey on Various Conversational Systems," in International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1- 8, doi: 10.1109/ic-ETITE47903.2020.126.

[26] Fitzpatrick, K. K., Darcy, A., and Vierhile, M. (2017), "Delivering cognitive behavior therapy to Young adults with symptoms of depression and anxiety using a

fully automated conversational agent (Woebot): A randomized controlled trial." in JMIR Mental Health, 4(2), e19.

[27] Frieder A, Fersh M, Hainline R, Deligiannidis, K. M., (2019)," Pharmacotherapy of Postpartum Depression: Current Approaches and Novel Drug Development." in CNS Drugs; 33(3):265-282.

[28] Gentner, T., Neitzel, T., Schulze, J. and Buettner, R. (2020), "A Systematic Literature Review of Medical Chatbot Research from a Behavior Change Perspective," IEEE 44th Annual Computers, Software, and Applications Conference, pp.735-740, doi:10.1109/COMPSAC48688.2020.0-

[29] Gujar, D., Biyani, R., Bramhane, T., Bhosale, S. and Vaidya, T. P., (2018)," Disease prediction and doctor recommendation system,"

Int. Res. J. Eng. Technol., Tamilnadu, India, vol. 5, no. 3, pp. 3207_3209, issue Mar.

[30] Gupta, J., Singh, V. and Kumar, I. (2021), "Florence- A Health Care Chatbot," in 7th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 504-508, doi: 10.1109/ICACCS51430.2021.9442006.

[31] Hajare, A., Bhosale, P., Nanaware, R., and Hiremath, G.," Chatbot for education system", in ISSN: 2454-132X, Impact factor: 4.295, Volume 4, Issue 3, IJARIIT.

[32] Haller, E., and Rebedea, T. (2013), "Designing a Chat-bot that Simulates a Historical Figure", in Faculty of Automatic Control and Computers university Politehnica of Bucharest, 978-0-7695-4980-4/13, IEEE.

[33] Hameedullah, K., Chowdhry, S. B., and Zeesha, M. (2012), "Med ChatBot: An UMLS based Chatbot for Medical Students, in International Journal of Computer Applications.", 55. 1-5. 10.5120/8844-2886.

[34] Hidayatin L. and Rahutomo, F., (2018), "Query Expansion Evaluation for Chatbot Application," in International Conference on Applied Information Technology and Innovation (ICAITI), Padang, Indonesia, pp. 92-95, doi: 10.1109/ICAITI.2018.8686762.

[35] Higashinaka, R., (2014) et al. Towards an open domain conversational system fully based on natural language processing. In COLING'14, pages 928-939.

[36] Ho, T. K., (1995), "Random decision forests," in Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.

[37] Hoy, M. B., (Jan. 2018)," Alexa, *siri*, *cortana*, and more: An introduction to voice assistants".

[38] Huang, C., Yang, M. C., and Huang, C. Y., (2018), "A chatbot-supported Smart Wireless Interactive Healthcare System for Weight Control and Health Promotion", in proceeding of the IEEE, issue April.

[39] https://censusindia.gov.in/2011provresults/paper2/data_files/india/Rural_urban_2011.pdf

[40] <https://www.kaggle.com/tna2doan/datasets>

[41] IBM Cloud Educational. *Natural Language Processing (NLP)* (2020) available at <https://www.ibm.com/cloud/learn/natural-language-processing>

[42] Ina *the History of Chatbots – From ELIZA to ALEXA* (2021) available at <https://onlim.com/en/the-history-of-chatbots/>

[43] Jiang, L., Cai, Z., Wang D. and Jiang, S, (2007), "Survey of Improving KNearest-Neighbor for Classification," in Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Haikou, China, pp. 679-683, doi: 10.1109/FSKD.2007.552.

[44] Jovanović, M., Baez, M. and Casati, F. (June 2021), "Chatbots as Conversational Healthcare Services," in *IEEE Internet Computing*, vol. 25, no. 3, pp. 44-51, 1 doi: 10.1109/MIC.2020.3037151.

[45] Kalla, D., and Samaah, F., (Jan - Feb 2020), "Chatbot for Medical Treatment using NLTK Lib", IOSR Journal of Computer Engineering.

[46] Kavitha, B. R., Chethana, R., M., (2019), "Chatbot for Healthcare System using Artificial Intelligence" IJARIIT.

[47] Kazi, H., and S. A. N. H., (2019), "Effect of Chatbot Systems on Student's Learning Outcomes," in SYLWAN.

[48] Kumar, V. M., (3 March 2016), "*Sanative Chatbot for Health Seekers*", JECS Volume 05 Issue Page No.16022-16025.

[49] Lee, C. B., and H. N., (2017), "*Chatbots and Conversational Agents: A Bibliometric Analysis*", in IEEE Int. Conf. Ind. Eng. Eng. Manag., pp. 215-219.

[50] Lenin, M., Gerritsen, C. and Bosse, T., (2019), "Towards humanlike chatbots helping users cope with stressful situations."In International Conference on Computational Collective Intelligence, pp. 232-243. Springer, Cham.

[51] Lin, J., Joseph, T., Belinkie, P. J., Mandel, A., Schumacher, R., Neumann, K., Scalise, L., Gaulton, J., Christ, L., Leitner, K., and Rosin, R., (2021)," Development of a practical training method for a healthcare artificial intelligence (AI) chatbot.", BMJ Innovations;7(2):441.

[52] Louise, K. B., (2019)," Measuring the postnatal health outcomes for mothers and fathers during the first twelve months after the birth of a baby", PhD thesis, University of Sheffield.

[53] Madhu, D. and C. J. N. J. S. S., (2017), "*A novel approach for medical assistance using trained chatbot*," in International Conference on Inventive Communication and Computational Technologies (ICICCT).

[54] Mathew, R. B., Varghese, S., Joy, S. E. and Alex, S. S., (2019) "*Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning*," in 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 851-856, doi: 10.1109/ICOEI.2019.8862707.

[55] Mathew, R. B., Varghese, S., Joy, S. E. and Alex, S. S., (2019), "*Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning*," in 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 851-856, doi: 10.1109/ICOEI.2019.8862707.

[56] Meshram, S., Naik, N., VR, M. More, T., and Kharche, S.," (2021)," *Conversational AI: Chatbots*", in Intelligent Technologies (CONIT) International Conference on, pp. 1-6.

[57] Mesko, D. B., (2020), "*The Medical Futurist*," in The Medical Futurist

Institute [Online].

Available: <https://medicalfuturist.com/magazine/>.

[58] Meyer," *P. Natural Language Processing Tasks* ", (2021) available at

<https://towardsdatascience.com/natural-language-processing-tasks-3278907702f3>

[59] Mishra, S. K., Bharti, D., and Mishra, N., (3,2017), " *Dr. Vdoc: A Medical Chatbot that Acts as a virtual Doctor*", in Journal of Medical Science and Technology, Volume: 6.

[60] Mohanta, B., and Patnaik, S., (2019), " *Paradigm shift in digital healthcare system using Artificial Intelligence, IOT and 5G Communication*" in Department of Computer Science & Engineering S'O'A University Bhubaneswar, India, Dr. Priti Das Department of Pharmacology SCB Medical College and Hospital Cuttack, India, (ICAML).

[61] Mondal, " *A. Complete Guide to build your AI Chatbot with NLP in Python*", (2021) available at

<https://www.analyticsvidhya.com/blog/2021/10/complete-guide-to-build-your-ai-chatbot-with-nlp-in-python/>

[62] Nadarzynski, T., Miles, O., Cowie, A., and Ridge, D, (2019)," *Acceptability of artificial intelligence (AI)-led chatbot services in healthcare: A mixed-methods study.*" in Digital Health, 5, 1–12.

[63] Paranjape, K., Schinkel M. and P., Nanayakkara, (2020), " *Short Keynote Paper: Mainstreaming Personalized Healthcare— Transforming Healthcare Through New Era of Artificial Intelligence*", in *Biomedical and Health Informatics IEEE Journal of*, vol. 24, no. 7, pp. 1860-1863,

[64] Polignano, M., Narducci, F., Iovine, A., Musto, C., Gemmis, M. D. and Semeraro, G., (2020), " *A Personal Health Assistant for the Italian Language*" in IEEE.

[65] Pouria K., and Sunita, D., " *Short Survey on Naive Bayes Algorithm,*" in International Journal of Advance Research in Computer Science and Management 04, 2017.

[66] Przegalinska, A., Ciechanowski, L., Stroz, A., Gloor, P. and Mazurek, G., (2019), "*In bot, we trust A new methodology of chatbot performance measures*"

[67] Qurashi, A. W., Holmes, V. and Johnson, A. P., (2020) "*Document Processing: Methods for Semantic Text Similarity Analysis,*" in International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Novi Sad, Serbia, pp. 1-6, doi: 10.1109/INISTA49547.2020.9194665.

R

[68] Rahman, M. M., Amin, R., Liton, M. N. K. and Hossain, N., (2019), "*Disha: An Implementation of Machine Learning Based Bangla Healthcare Chatbot,*" in 22nd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICCIT48885.2019.9038579.

[69] Reddy, J. E. P. and Bhuwaneshwar, C. N. and Palakurthi, S. and Chavan, A., (2020), "*AI-IoT based Healthcare Prognosis Interactive System*", Innovation in Technology (INOCON) 2020 IEEE International Conference for, pp. 1-5.

[70] Reddy, J. E. P. and Chavan, A., 2020, "*AI-IoT based Smart Pill Expert System*", in Trends in Electronics and Informatics (ICOEI) (48184) 2020 4th International Conference on, pp. 407-414.

[71] Safavian, S. R., and Landgrebe, D., (1991), "*A survey of decision tree classifier methodology,*" in IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, May-June 1991, doi: 10.1109/21.97458.

[72] Sachiko B., (2020), et al. "*Impact of intention and feeling toward being pregnant on postpartum depression: the Japan Environment and Children's Study (JECS).*". Archives of women's mental health 23.1: 131-137.

[73] Sameera, A. K., and Woods, J., (2015), "*Survey on Chatbot Design Techniques in Speech Conversation Systems*", in (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 6, no. 7.

[74] SAS" Natural Language Processing (NLP) What it is and why it matters "available at

<https://www.sas.com/dk/insights/analytics/what-is-natural-language-processing-nlp.html>

[75] Schwank, S., Lindgren, H., Wickberg, B., Fu, S.-C., Yan, D., and Andersson, E., (2020)," *When a New Mother Becomes Mentally Unhealthy, It Is Everyone's Problem: Shanghai Women's Perceptions of Perinatal Mental Health Problems*", in *Women's Reproductive Health*, 7:3, 190-204.

[76] Schwank, S., Andersson, E. Wickberg, B., Fu, S-C., Ding, Y and Lindgren, H, (2019), "Care-seeking behavior and disclosure on mental health among women in urban Shanghai, China." in *Health Psychology Open*.

[77] Setiaji, B., and Wibowo, F. W., (2016), "Chatbot Human-to- Machine Conversation Modeling", in Intelligent Systems Modelling and Simulation (ISMS) 7th International Conference.

[78] Setiaji, B., Wibowo, F. W. (2016), "Chatbot Using a Knowledge in Database-Human-to-Machine Conversation Modeling", in Department of Informatics Engineering STMIK AMIKOM Yogyakarta, Indonesia, IEEE.

[79] Setiaji, B., Wibowo, F. W., (2016),"Chatbot Using a Knowledge in Database", in 7th International Conference on Intelligent Systems Modelling and Simulation.

[80] Shahzad, Q. and Ramsha, A., (2018), "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," International Journal of Computer Applications.

[81] Shawar, B. A., and Atwell, E., (2007),"Chatbots: are they really useful?", in LDV Forum.

[82] Shabariram, C.P., Srinath, V., Indhuja, C.S., and Ratatta, V., (2017)," Chatbot Application Using Expert System", inIJAR in Computer Science and Software Engineering.

[83] Shangrapawar, A., Ravekar, A., Kale, S., Kumari, N., Shende, A.and Taklikar, P., (2020), "Artificial Intelligence based Healthcare Chatbot System" in International

Research Journal of Engineering and Technology, e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 07.

[84] Shein, E. "Why conversational AI will become a c-suite priority in 2020".

[85] Shukla V.K, and Verma, A., (2019), "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language", in International Conference on Automation Computational and Technology Management (ICACTM).

[86] Softić, A., Husić, J. B., Softić A., Baraković, S., (2021), "Health Chatbot: Design Implementation Acceptance and Usage Motivation", INFOTEH-JAHORINA (INFOTEH) 2021 20th International Symposium, pp. 1-6.

[87] Srivastava, P. and Singh, N. (2020), "Automatized Medical Chatbot (Medibot)," in International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), pp. 351-354, doi: 10.1109/PARC49193.2020.236624.

[88] Srivastava, P. and Singh, N. (2020), "Automatized Medical Chatbot (Medibot)," in International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2020, pp. 351-354, doi: 10.1109/PARC49193.2020.236624.

[89] Stormon, A. 5 Chatbot Challenges and How to Overcome Them (2016) available at

<https://chatbotsmagazine.com/5-chatbot-challenges-and-how-to-overcome-them-caccc3a26d7c>

[90] Sutoyoa, R., Chowandaa, A., Kurniatia, A., and Wongsoa R., (19 Sept 2019), "Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States" in Elsevier, ScienceDirect.

[91] Swathi, S., Saranya, E., Prabakaran, R. M., Kumar, M., Bairavel, S., (2021), "Virtual Health Assistant", in System Computation Automation and Networking (ICSCAN) International Conference on, pp. 1-4.

[92] The British Journal of Psychiatry. (1987); 150(6):782-786. BC Reproductive Mental Health Program and Perinatal Services BC. (2014), Best Practice Guidelines for Mental Health Disorders in the Perinatal Period.

[93] *Types of Bots: An Overview* general at

<http://botnerds.com/types-of-bots/>

<https://www.techrepublic.com/article/why-conversational-ai-will-become-a-c-suite-priority-in-2020/>

[94] Ullas and Rahman, T., (2019), et. al," *A machine learning approach to detect depression and anxiety using supervised learning.*" in Diss. Brac University.

[95] W. Astuti, D. P. I. Putri, A. P. Wibawa, Y. Salim, Purnawansyah and A. Ghosh, (2021), "Predicting Frequently Asked Questions (FAQs) on the COVID-19 Chatbot using the DIET Classifier", 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), pp. 25-29, doi: 10.1109/EIConCIT50028.2021.9431913.

[96] Yanting, W., (2020) et al. "Perinatal depressive and anxiety symptoms of pregnant women during the coronavirus disease 2019 outbreak in China." American journal of obstetrics and gynecology vol. 223,2: 240.e1-240.e9.

[97] Zhang, J., Oh, Y., Lange, P., Yu, Z., and Fukuoka, Y., (2020)," *Artificial Intelligence Chatbot Behavior Change Model for Designing Artificial Intelligence Chatbots to Promote Physical Activity and a Healthy Diet: Viewpoint*", in Journal of Medical Internet Research.