

**A  
Project Report  
on  
HEALTH NUTRITION CHATBOT**

Project report submitted in partial fulfilment of the requirement for the award  
of the Degree of  
**BACHELOR OF TECHNOLOGY**

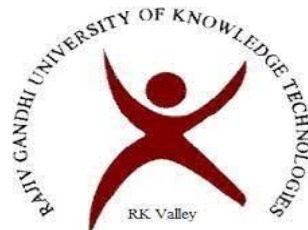
**In  
COMPUTER SCIENCE & ENGINEERING**  
Submitted by

**S. KRISHNAVENI – R180893**

**P. DIVYA – R180897**

Under the guidance of

**M.MUNI BABU  
M. Tech, (Ph.D), Assistant Professor**



**Rajiv Gandhi University of Knowledge Technologies (AP IIIT)**

**RK Valley, Kadapa (Dist.), Andhra Pradesh, 516330**

**Department of Computer Science and Engineering**

**2020-2024**



# **Rajiv Gandhi University of Knowledge Technologies**

**RK Valley, Kadapa (Dist.), Andhra Pradesh, 516330**

**Department of Computer Science and Engineering**

**2020 -2024**

---

## **CERTIFICATE**

This is to certify that the project work titled “**HEALTH NUTRITION CHATBOT**” is a Bonafede project work submitted by **S. KRISHNAVENI (R180893)**, **P. DIVYA (R180897)**, in the department of **COMPUTER SCIENCE AND ENGINEERING** in partial fulfilment of requirements for the award of degree of Bachelor of Technology in **Computer Science and Engineering** for the year 2023- 2024 carried out the work under the supervision.

**Signature of Project Guide**

GUIDE  
M. MUNI BABU  
M. Tech, (Ph. D)  
Assistant Professor.

**Signature of HOD**

HEAD OF THE DEPARTMENT  
Dr. P. Ravi Kumar  
Ph.D (University of AIZU, Japan)  
M.E (IISc Bangalore)

**Signature of External Guide**

## **DECLARATION**

We hereby declare that the project report entitled “**HEALTH NUTRITION CHATBOT**” submitted to the **Department of COMPUTER SCIENCE AND ENGINEERING** in partial fulfilment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY**. This project is the result of our own effort and that it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

**WITH SINCERE REGARDS**

**S. KRISHNAVENI (R180893)**

**P. DIVYA (R180897)**

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, **Prof. A V S S KUMARA SWAMI GUPTA** for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department **Dr. P. Ravi Kumar** for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college **Mr. M. MUNI BABU** for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

## **Table of Contents**

<b>S.NO</b>	<b>INDEX</b>	<b>PAGE NO</b>
	Abstract	ii
	List of Figures	iii
I.	Introduction <ul style="list-style-type: none"><li>• Problem Statement</li></ul>	01-02 02
II.	Literature Review <ul style="list-style-type: none"><li>• Motivation</li><li>• Contribution</li></ul>	03 03 04-11
III.	Modules Module-1: Install and import Python libraries Module-2: Load training and testing data <ul style="list-style-type: none"><li>• Module-2.1: Load symptom description</li><li>• Module-2.2: check symptom pattern</li></ul> Module-3: Predict disease and get description	12 13-19 15-18 19 20-22
IV.	Results and Discussion	23-25
V.	Conclusion and Future Enhancements	26
	References	27

## **Abstract:**

To start a good life, healthcare is more important. Today's people are busy with their works at home, office works and more addicted to Internet. They are not concerned about their health. So they avoid to go hospitals for small problems. It may become a major problem. So we create a Health Nutrition Chatbot using machine learning that can find the disease before consulting a doctor. The chatbot provides text you can communicate with bot like user friendly. The chatbot will clarify the users symptoms by asking questions, here symptom confirmation will be done.

Bot will provides which type of disease you have based on user symptoms. The disease will be categorized as minor and major disease. Chatbot will reply whether it is a major or minor disease. For major disease it appeared consult a doctor and for minor disease it provides precautions like food suggestion that means which type of food you have to take, especially vegetables and fruits and its uses to increase nutrition levels. Health Nutrition chatbot reduce the healthcare cost and time of the users as it is not possible for the users to visit the doctors or experts when immediately needed.

Mainly by using N-gram algorithm, Term frequency-inverse data frequency (TF-IDF) and Cosine similarity algorithms we are implementing Health Nutrition chatbot.

## LIST OF FIGURES

Figure No.	Title	Page No.
<b>Figure 1.1</b>	Health Nutrition chatbot	01
<b>Figure 1.2</b>	Complete process of Health Nutrition Chatbot	02
<b>Figure 2.1</b>	Data flow diagram	06
<b>Figure 3.1.1</b>	Import python libraries	12
<b>Figure 3.2.1</b>	Training and testing data	13
<b>Figure 3.2.2</b>	Code for training and testing data	14
<b>Figure 3.2.1.1</b>	Load the symptom description, precaution and severity	15
<b>Figure 3.2.1.2</b>	Code for symptom_description	15
<b>Figure 3.2.1.3</b>	Symptom_description data	16
<b>Figure 3.2.1.4</b>	Code for symptom_severity	16
<b>Figure 3.2.1.5</b>	Symptom_severity data	17
<b>Figure 3.2.1.6</b>	Code for symptom_precautions	17
<b>Figure 3.2.1.7</b>	Symptom_precaution data	18
<b>Figure 3.2.2.1</b>	Code for symptom pattern	19
<b>Figure 3.3</b>	Code for predict the disease	20
<b>Figure 3.3.1</b>	Code for getting the details of precautions, suggest fruits and vegetables and its uses	22
<b>Figure 4.1</b>	Run the majorproject.py	23
<b>Figure 4.2</b>	By run the majorproject.py this will open	23
<b>Figure 4.3</b>	Entering the inputs	24
<b>Figure 4.4</b>	Precautions, suggest fruits and vegetables and its uses	25

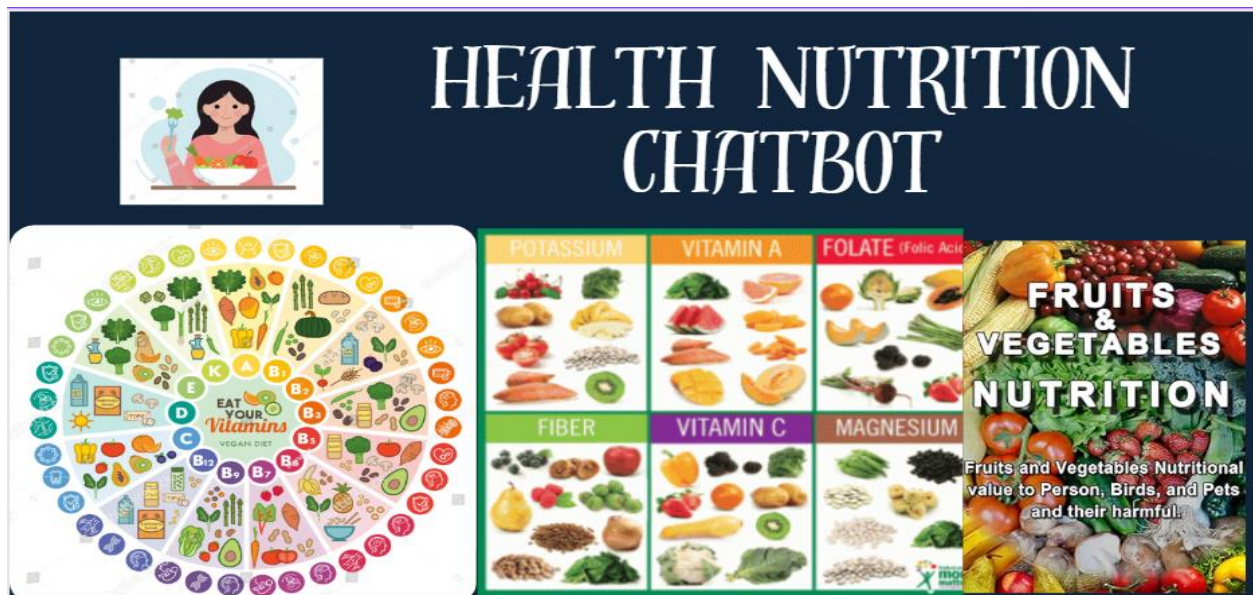
## LIST OF TABLES

Table No	Title	Page No
1	Literature Review	3
2	Calculation of cosine similarity algorithm	10

# I. INTRODUCTION

Now a days, health care is very important in our life. Today's people are busy with their works at home, office works and more addicted to internet. They are not concerned about their health. So, they avoid to go in hospitals for small problems. It may become a major problem. So, we create a Health Nutrition chatbot using machine learning that can find the disease before consulting a doctor. The chatbot provides text you can communicate with bot like user friendly. The chatbot will clarify the user's symptoms by asking questions here symptom confirmation will be done. Chatbot will provides which type of disease you have based on user symptoms. The disease will be categorized as minor and major disease. Chatbot will reply whether it is a major or minor disease.

For major disease it appeared consult a doctor and for minor disease it provides precautions like food suggestion that means which type of food you have to take, suggest fruits and vegetables and its uses to increase nutrition levels. Health Nutrition chatbot reduce the healthcare cost and time of the users as it is not possible for the users to visit the doctors or experts when immediately needed.

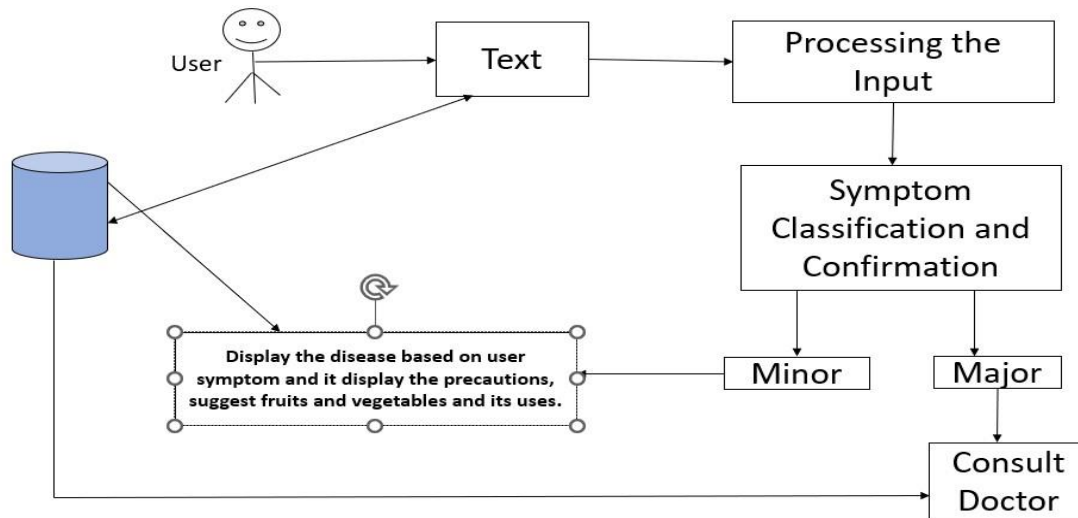


**Figure 1.1: Health Nutrition chatbot**

As shown in figure 1.1, Health Nutrition chatbot contains precautions like food suggestions, suggest fruits and vegetables and its uses for our healthy life.



## System Architecture



**Figure 1.2:** Complete process of Health Nutrition chatbot

As shown in figure 1.2, architecture of Health Nutrition chatbot. It explains about the working of the project form beginning to ending. Firstly, user text his or her name and enter the symptom then processing the input and confirm whether it is major or minor disease. If it is major consult a doctor otherwise display the precautions, suggest fruits and vegetables and its uses for minor diseases.

### **Problem Statement:**

We have live chats through text and it may charge amount to perform live chat or telephony communication. So, Health Nutrition chatbot is help to user can chat with the bot regarding the health query through text, to reduce the health care cost and time of the users. Now a days people show interest on interface like chatbot because it gives friendly response that's why we are using Health Nutrition chatbot. User can also view the food suggestion that means which type of food you have to take, suggest fruits and vegetables and its uses.

## II. LITERATURE REVIEW

**Table 2.1:** Literature Review

Year	Author	Method	Outcome	Drawback
2023	Al-Hajri et al.	NLP (Natural Language Processing)	Satisfactory user satisfaction and engagement.	Limited data, user trust.
2022	Ali et al.	Support Vector Machine	High accuracy in symptom classification and disease prediction.	Biased data, limited technical capabilities.
2021	Kadam et al.	Deep Learning	Improved diagnosis support through medical image analysis.	Complex model development, limited data.
2020	Pradeep et al.	Reinforcement Learning	Enhanced chatbot performance with increased user interaction.	Black-box nature of the algorithm, privacy concerns.
2019	Verma et al.	Multilingual NLP	Effective communication with diverse populations.	Language accuracy limitations, user acceptance.

### Motivation:

The motivation for creating Health Nutrition Chatbot is when we are in hostel, we cannot able to consult doctor for minor diseases and we have seen some members at hostel get more depressed for small problems and in that situation there will no parents at hostel to suggest and support, so by this Health Nutrition Chatbot we clear our health queries and follow the suggestion that means which type of food you have to take, suggest fruits and vegetables and its uses. It provides the services at any time. By taking fruits and vegetables to increase our nutrition levels, especially students in hostel and workers in organization. Now a days people show interest on interface like chatbot because it gives friendly response that's why we are using Health nutrition Chatbot.

## **Contribution:**

Health Nutrition Chatbot is used to communicate with users through textual method. It is a health care provider and help to improve their performance by interacting with users in a human-like way. It helps the people before consulting a doctor to know their health conditions. In order to make communications more effective we are implementing a chatbot for health suggestions. Chatbot is based on Machine Learning and Users Natural Language Process (NLP) to interpret and accordingly respond to the user. N-gram algorithm, Term Frequency (TF) is used to build a model with the goal of understanding text and stop words being removed. Inverse Document Frequency (IDF) determines the weight of rare words. Cosine Similarity algorithm is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

## **Technologies Used:**

### **Python:**

Python is an excellent programming language for building software application like chatbots due to its simplicity and versatility, and the availability of various libraries and frameworks. Python extensive libraries, and robust community support make it an ideal choice for building chatbots with advanced features like natural language processing, machine learning. Python is a computer programming language often used to build websites and software automated tasks and conduct data analysis. python is a general-purpose language Meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. this versatility along with its beginner's friendliness has made it one of the most used programming languages today.

### **Uses of python:**

python is commonly used for developing websites and software task automation data analysis and data visualization. Since it's relatively easy to learn python has been adopted by many non-programmers such as accountants and scientists for a variety of everyday tasks like organizing finance.

- data analysis and machine learning

## **Python Libraries:**

### **NumPy:**

NumPy is a python library used for working with arrays it also has functions for working in domain of linear algebra flourier transform and matrices. NumPy stands for numerical python. In python we have lists that serve the purpose of arrays but they are slow to process. NumPy aims to provide an array object that is up to 50 times faster than traditional python list the array object in NumPy is called Nd array. It provides a lot of supporting functions that make working with Nd array very easy.

### **Pandas:**

Pandas is a python library used for working with datasets. It has functions for analysing cleaning exploring and manipulating data. The name pandas have reference to both panel data and python data analysis and was created by wes mckinney in 2008.

### **Regular Expression(re):**

Regular expressions are special sequence of characters that uses a search pattern to find a string or set of strings it can detect the presence or absence of a text by matching it with a particular pattern and also can split a pattern into one or more sub patterns. Python provides are module that supports the use of regex in python its primary function is to offer a search.

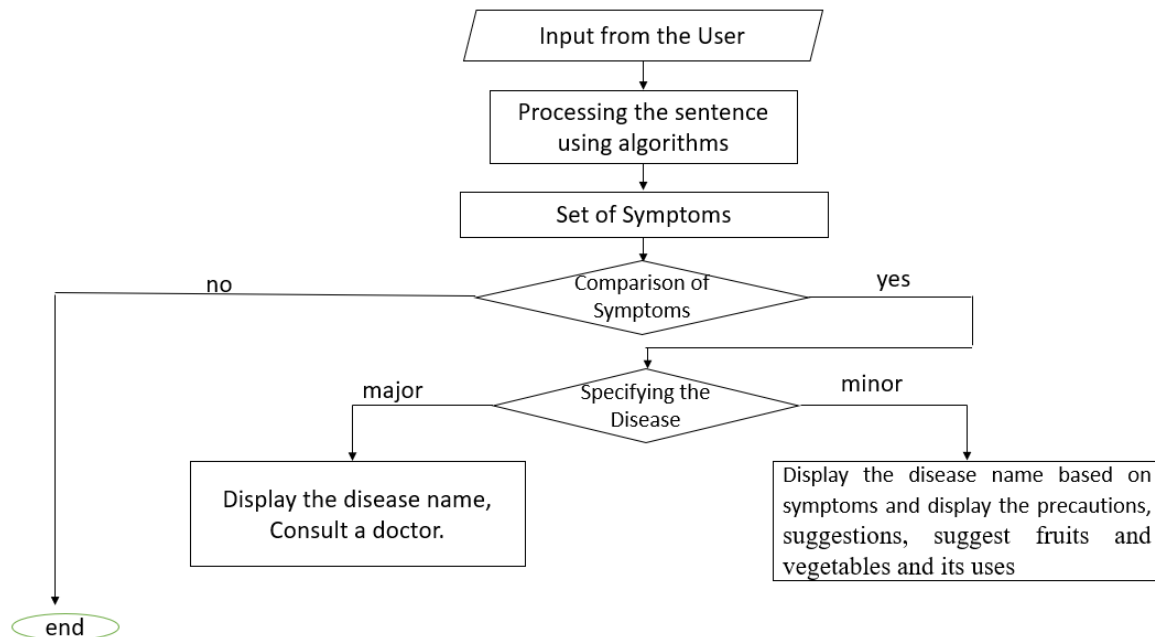
### **Scikit-Learn:**

Scikit-learn is an open-source python library that implements a range of machine learning preprocessing cross validation and visualization algorithms using a unified interface. Important features of scikit-learn: simple and efficient tools for data mining and data analysis.

### **CSV:**

Common separated values (csv) module was incorporated in python's standard library as a result of PEP 305. CSV is a preferred export data format by Microsoft's Excel spreadsheet software.

## Data Flow Diagram:



**Figure 2.1:** Data Flow Diagram

As shown in figure 2.1, data flow diagram explains about the process of Health Nutrition Chatbot. Firstly, take the input from the user such as name and symptom. After that processing the input chatbot will confirm the disease by asking questions. Based on the user response chatbot specify the disease whether it is major or minor. If it is major disease display the disease name and consult a doctor. If it is minor disease display the disease name and precautions like suggest fruits and vegetables and its uses.

## Algorithms:

We are using three algorithms to implement Health Nutrition Chatbot.

1. N-gram Algorithm
2. Term Frequency - Inverse Data Frequency (TF-IDF)
3. Cosine similarity algorithm

### 1. N-gram Algorithm:

N-Grams are way to help machines understand a word in the content to get a better understanding of word. N-gram is a neighbouring sequence of n-items from a given sample of text. N-items means we can have two items, three items and so on. So, it is a contiguous sequence of some items.it helped to predicting the next words in a sentence. Things will be characters, words, sentences. When n is 2 then we can call it as bigrams and n is 3 then we can call it as trigrams. Based on sentence we can change the value of 'n'.

## Advantages:

1. **Simple and Efficient:** N-gram models are relatively simple and computationally efficient compared to more complex language models. This simplicity allows for quick implementation and processing.
2. **Captures Local Context:** They capture the local context of words in a sequence. For example, a trigram model (3-gram) considers the probability of a word based on the two preceding words, which can help in tasks like text prediction or completion.
3. **Useful in Language Generation:** N-grams are effective in generating text, such as in machine translation or autocomplete features, by predicting the next words based on the preceding sequence.
4. **Feature Extraction:** In some machine learning applications, N-grams serve as useful features for text classification tasks like sentiment analysis or spam detection. They capture word combinations that might be indicative of specific classes or categories.
5. **Scalability:** They can be adapted to different sizes (unigram, bigram, trigram, etc.) depending on the level of context needed, making them flexible for various NLP tasks.

## Disadvantages:

1. **Limited Context:** N-gram models have a limited context window based on the 'n' value chosen. This limitation can lead to a lack of understanding of broader semantics or dependencies beyond the chosen 'n' words.
2. **Fixed Length:** The length of the sequence is fixed based on the chosen 'n' value, potentially leading to difficulties in handling variable-length sequences or capturing long-range dependencies.
3. **Overfitting:** Higher 'n'-gram models are prone to overfitting, especially when dealing with limited training data. This is due to the model memorizing specific sequences rather than generalizing well to unseen data.

## 2. Term Frequency-Inverse Data Frequency (TF-IDF):

### Term Frequency (TF):

when building a model with the goal of understanding text, you'll see all of stop words being removed. Another approach is to score the relative importance of words using TF-IDF.

The number of times a word seems during a document divided by the total number of words in the document. Each document has its own term frequency.

$$tf_{i,j} = n_{i,j} / \sum_k n_{i,j} \quad \text{—————} \quad (i)$$

➔  $n_{i,j}$  = the number of times a word appears in a document.

➔  $\sum_k n_{i,j}$  = total number of words in the document.

Every document has its own term frequency.

Example1: I am suffering from fever.

Example2: I am suffering from headache.

➤  $TF = \text{Number of occurrences} / \text{Number of words in document.}$

Example1:

$$TF_1 = 1/5 = 0.2$$

$$TF_2 = 1/5 = 0.2$$

$$TF_3 = 1/5 = 0.2$$

$$TF_4 = 1/5 = 0.2$$

$$TF_5 = 1/5 = 0.2$$

Example2:

$$TF_1 = 1/5 = 0.2$$

$$TF_2 = 1/5 = 0.2$$

$$TF_3 = 1/5 = 0.2$$

$$TF_4 = 1/5 = 0.2$$

$$TF_5 = 1/5 = 0.2$$

## Inverse Data Frequency(IDF):

The log of the number of documents divided by the number of documents that contain the word **w**. Inverse data frequency determines the weight of unique words across all documents in the corpus.

$$\text{idf}(w) = \log (N / \text{df}_t) \quad \text{———— (ii)}$$

➔  $N$  = Number of documents.

➔  $\text{df}_t$  = Number of documents that contain words.

Example1: I am suffering from fever.

Example2: I am suffering from headache.

$\text{Idf} = \log (\text{number of documents} / \text{number of documents contain a word})$

$$\text{IDF}(\text{I}) = \log (2/2) = 0$$

$$\text{IDF}(\text{am}) = \log (2/2) = 0$$

$$\text{IDF}(\text{suffering}) = \log (2/2) = 0$$

$$\text{IDF}(\text{from}) = \log (2/2) = 0$$

$$\text{IDF}(\text{fever}) = \log (2/1) = 0.301$$

$$\text{IDF}(\text{headache}) = \log (2/1) = 0.301$$

### Advantages:

**1. Term Importance:** TF-IDF helps identify the importance of a term in a document relative to a collection of documents. Terms that appear frequently in a document but rarely in other documents are considered more important.

**2. Reduction of Common Terms:** Commonly occurring words like "the," "is," etc., tend to have lower TF-IDF scores due to their prevalence across documents. This helps in identifying and prioritizing more meaningful terms.

**3. Flexible and Simple:** It's a relatively simple algorithm to implement, yet it offers flexibility in capturing term importance based on frequency within a document and rarity across the entire document collection.

**4. Normalization:** TF-IDF normalizes term frequencies, reducing the impact of document length on the importance of terms. This normalization allows for fair comparisons between documents of different sizes.



**5. Useful in Text Classification:** It's valuable for text classification tasks by highlighting discriminative terms that are more relevant to particular classes or categories.

### Disadvantages:

**1. Ignores Word Order and Semantics:** TF-IDF solely considers term frequency and document frequency but disregards word order or semantics. Hence, it might not capture the context or meaning of phrases or multi-word expressions.

**2. Sensitive to Term Variations:** It treats each term independently, leading to challenges with variations of words (e.g., singular vs. plural, different verb tenses, synonyms), potentially overlooking their semantic similarity.

### 3. Cosine Similarity Algorithm:

Cosine similarity algorithm is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The technique is also used to measure cohesion within clusters in the field of data mining.

$$TF * IDF \quad \text{---(iii)}$$

Documents	I	am	suffering	from	fever	headache
Document 1	$0.2 * 0 = 0$	$0.2 * 0 = 0$	$0.2 * 0 = 0$	$0.2 * 0 = 0$	$0.2 * 0.301 = 0.012$	0
Document 2	$0.2 * 0 = 0$	$0.2 * 0 = 0$	$0.2 * 0 = 0$	$0.2 * 0 = 0$	0	$0.2 * 0.301 = 0.012$

**Table 2.2:** Calculation of cosine similarity algorithm

### Advantages:

**1. Scale Invariant:** Cosine similarity is unaffected by the magnitude of the vectors but rather focuses on the direction, making it robust when dealing with documents of varying lengths or scales.

**2. Ignores Magnitude:** It considers the angle between vectors, not their magnitude, which is beneficial when comparing documents or text irrespective of their sizes or word counts.

**3. Efficiency:** It's computationally efficient, especially when dealing with high-dimensional spaces, making it suitable for large-scale document comparisons or similarity-based tasks.

**4. *Effective in Text Mining*:** Cosine similarity is effective in text mining tasks such as document clustering, information retrieval, and content-based recommendation systems, as it helps quantify the similarity between documents or items based on their content.

**5. *Normalization*:** It implicitly normalizes the vectors, making comparisons fair and independent of the overall frequency of terms within documents.

### **Disadvantages:**

**1. *Limited to Vector Space Models*:** Cosine similarity requires data representation as vectors in a vector space model. Hence, it might not capture more complex relationships or non-linear structures present in the data.

**2. *Lack of Semantic Understanding*:** It doesn't consider semantics or meaning, treating each term or dimension independently. This might lead to mismatches in similarity when comparing documents with similar meanings but different wording.

### III. MODULES

#### MODULE-1: Install and Import Python Libraries:

Here, we install python libraries like regular expression(re), pandas, NumPy, warnings, sklearn and its models in pycharm.

We use below commands to install python libraries.

- ➔ pip install re
- ➔ pip install pandas
- ➔ pip install numpy
- ➔ pip install sklearn.tree
- ➔ pip install sklearn.model selection
- ➔ pip install warnings
- ➔ pip install sklearn.svm
- ➔ pip install streamlit

```
1 import re
2 import pandas as pd
3 from sklearn.tree import DecisionTreeClassifier, _tree
4 import numpy as np
5 from sklearn.model_selection import train_test_split
6 from sklearn.svm import SVC
7 import csv
8 import warnings
9 import streamlit as st
10
11 warnings.filterwarnings(action="ignore", category=DeprecationWarning)
```

**Figure 3.1.1:** Import Python Libraries

As shown in figure 3.1.1, after successful installation we can import the above libraries in pycharm. Make sure that installation of all dependent libraries must before importing otherwise it will be showing error.

## MODULE-2: Load Training and Testing Data:

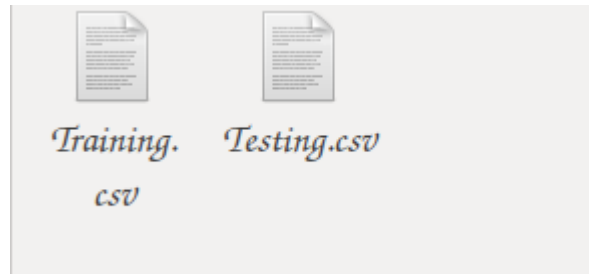


Figure 3.2.1: Training and testing data

As shown in figure 3.2.1, we can load training and testing data using training.csv as well as testing.csv files. The data can be trained and tested after pre-processing the data to give better results.

```
16 # Load training and testing data with specified encoding
17 training = pd.read_csv(filepath_or_buffer: 'Training.csv', encoding='latin1') # Specify encoding
18 testing = pd.read_csv(filepath_or_buffer: 'Testing.csv', encoding='latin1') # Specify encoding
19
20 # Extract columns and features
21 cols = training.columns
22 cols = cols[:-1]
23 x = training[cols]
24 y = training['prognosis']
25 y1 = y
26
27 # Group data by prognosis
28 reduced_data = training.groupby(training['prognosis']).max()
29
30 # Encode labels
31 from sklearn.preprocessing import LabelEncoder
32 le = LabelEncoder()
33 le.fit(y)
34 y = le.transform(y)
35
36 # Split data into training and testing sets
37 x_train, x_test, y_train, y_test = train_test_split(*arrays: x, y, test_size=0.33, random_state=42)
38 testx = testing[cols]
39 testy = testing['prognosis']
40 testy = le.transform(testy)
41
42 # Initialize decision tree classifier and fit the model
43 clf1 = DecisionTreeClassifier()
44 clf = clf1.fit(x_train, y_train)
45
```

```

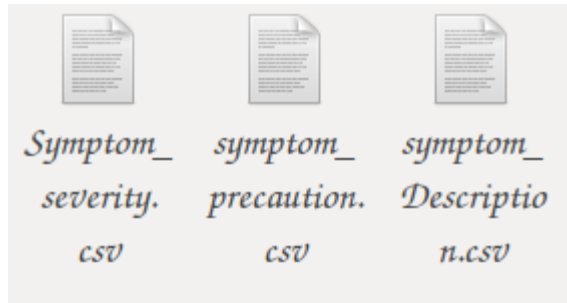
46 # Initialize SVM classifier and fit the model
47 model = SVC()
48 model.fit(x_train, y_train)
49 svm_score = model.score(x_test, y_test)
50
51 # Extract feature importances
52 importances = clf.feature_importances_
53 indices = np.argsort(importances)[::-1]
54 features = cols
55
56 # Create dictionaries for symptom severity, description, and precautions
57 severityDictionary = dict()
58 description_list = dict()
59 precautionDictionary = dict()
60
61 # Create a dictionary to map symptoms to their indices
62 symptoms_dict = {}
63
64 # Populate symptoms_dict with symptom-index mappings
65 for index, symptom in enumerate(x.columns):
66     symptoms_dict[symptom] = index
67
68 # Calculate the condition based on symptoms and duration
69 1 usage
70 def calc_condition(symptoms, days):
71     sum = 0
72     for item in symptoms:
73         sum += severityDictionary[item]
74     if ((sum * days) / (len(symptoms))) > 10:
75         return "You should seek consultation."
76     else:
77         return "It might not be that bad, but you should take precautions."

```

**Figure 3.2.2:** Code for training and testing data

As shown in figure 3.2.2, the code is for training and testing data using training.csv, testing.csv files. Here comma separated value(csv) files are used because of these files have a simple structure, human readable, storing the data can be easily imported or exported. So, csv files are used for training and testing the data.

## Module-2.1: Load symptom description



**Figure 3.2.1.1:** Load the symptom description, precaution and severity

As shown in figure 3.2.1.1, we can load the symptom description, precaution and severity files using Symptom\_severity.csv, symptom\_precaution.csv and symptom\_Description.csv files.

```
78 # Load symptom descriptions from a CSV file
79 1 usage
79 def getDescription():
80     global description_list
81     with open('symptom_Description.csv', encoding='latin1') as csv_file: # Specify encoding
82         csv_reader = csv.reader(csv_file, delimiter=',')
83         for row in csv_reader:
84             _description = {row[0]: row[1]}
85             description_list.update(_description)
86
87 # Load symptom severity information from a CSV file
```

**Figure 3.2.1.2:** Code for symptom\_Description

As shown in figure 3.2.1.2, the above code is for symptom description. Using symptom\_Description.csv file gives the correct solution to the user about the disease. Here csv\_reader used to read the csv file in which type of cases. For example, the above code consists of delimiter as ','.

## Symptom\_Description.csv file:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		An adverse drug reaction (ADR) is an injury caused by taking medication. ADRs may occur following a single dose or prolonged administration of a drug or result from the combination of two or more drugs.	Uses: Incorporate blueberries into your diet by adding them to smoothies, yogurt, oatmeal, or enjoying them as a standalone snack.											
2	Drug Reaction	2.Spinach: Spinach is rich in vitamins A, C, and K, as well as minerals like magnesium and iron. These nutrients support overall health and may help mitigate certain drug reactions.	Uses: Include spinach in salads, stir-fries, omelets, or blend it into green smoothies.											
		3.Broccoli: Broccoli contains sulforaphane, a compound with potent antioxidant and anti-inflammatory properties. It may help protect against drug-induced toxicity and support liver health, with certain medications.	Uses: Steam, roast, or stir-fry broccoli as a side dish or add it to pasta, soups, or salads.											
		4.Oranges: Oranges are abundant in vitamin C, which acts as a powerful antioxidant and supports immune function. Vitamin C may help reduce oxidative stress caused by certain medications.	Uses: Enjoy oranges as a refreshing snack, or squeeze fresh orange juice to drink or use in marinades and dressings.											
		5.Avocado: Avocado is a nutrient-dense fruit rich in healthy fats, vitamins E and K, potassium, and fiber. Its anti-inflammatory properties may help mitigate some drug reactions.	Uses: Spread avocado on toast, add it to salads, sandwiches, or smoothies, or make guacamole as a dip.											
		An infectious disease caused by protozoan parasites from the Plasmodium family that can be transmitted by the bite of the Anopheles mosquito or by a contaminated needle or transfusion.	Uses: Enjoy ripe papaya as a fruit or blend it into smoothies. Some people also consume papaya leaf extract as a supplement, but it's essential to consult with a healthcare professional before using it.											
		2.Citrus fruits: Citrus fruits like oranges, lemons, and grapefruits are rich in vitamin C, which plays a crucial role in supporting the immune system and combating infections.	Uses: Consume citrus fruits fresh as a snack or juice, or incorporate them into salads, marinades, or desserts.											
		3.Dark leafy greens (e.g., spinach, kale): Dark leafy greens are packed with nutrients like vitamins A, C, and K, as well as minerals like iron and magnesium. These nutrients support overall health and may help mitigate certain drug reactions.	Uses: Include dark leafy greens in salads, soups, stir-fries, or smoothies.											
		4.Ginger: Ginger contains gingerol, a bioactive compound with antioxidant and anti-inflammatory properties. It may help alleviate symptoms associated with malaria, such as fever and inflammation.	Uses: Add fresh ginger to teas, soups, stir-fries, or smoothies for its flavor and potential health benefits.											
		5.Garlic: Garlic contains allicin, a compound with antimicrobial properties that may help combat infections. It also supports immune function and has been traditionally used for its medicinal properties.	Uses: Incorporate fresh garlic into various cuisines, such as soups, stews, stir-fries, or roasted vegetables.											

Figure 3.2.1.3: symptom\_Description data

As shown in figure 3.2.1.3, the data is about symptom descriptions. It consists of disease name and its description that means the information about the disease in few lines. As well as it contains information about the fruits and vegetable its uses in detailed.

```

60 # Load symptom severity information from a CSV file
    usage
61 def getSeverityDict():
62     global severityDictionary
63     with open('Symptom_severity.csv') as csv_file:
64         csv_reader = csv.reader(csv_file, delimiter=',')
65         for row in csv_reader:
66             if len(row) >= 2:
67                 _diction = {row[0]: int(row[1])}
68                 severityDictionary[_diction: dict = {row[0]: int(row[1])} :

```

Figure 3.2.1.4: Code for symptom\_severity

As shown in figure 3.2.1.4, the above code is for symptom severity. Using symptom\_severity.csv file chatbot predict about the condition of the disease.

## Symptom\_severity.csv file:

	$\mathcal{A}$	$\mathcal{B}$
1	itching	1
2	skin_rash	3
3	nodal_skin_eruptions	4
4	continuous_sneezing	4
5	shivering	5
6	chills	3
7	joint_pain	3
8	stomach_pain	5
9	acidity	3
10	ulcers_on_tongue	4
11	muscle_wasting	3
12	vomiting	5
13	burning_micturition	6
14	spotting_urination	6
15	fatigue	4
16	weight_gain	3
17	anxiety	4
18	cold_hands_and_feets	5
19	mood_swings	3
20	weight_loss	3
21	restlessness	5
22	lethargy	2
23	patches_in_throat	6
24	irregular_sugar_level	5

**Figure 3.2.1.5.** symptom\_severity data

As shown in figure 3.2.1.5, the data is about symptom severity. It consists of severity of symptoms. The severity of symptoms decides whether it is major or minor disease.

```

1 usage
71 def getprecautionDict():
72     global precautionDictionary
73     with open('symptom_precaution.csv') as csv_file:
74         csv_reader = csv.reader(csv_file, delimiter=',')
75         for row in csv_reader:
76             _prec = {row[0]: [row[1], row[2], row[3], row[4]]}
77             precautionDictionary.update(_prec)
78

```

**Figure 3.2.1.6:** Code for symptom\_precautions

As shown in figure 3.2.1.6, the above code is for symptom precautions. Using symptom\_precautions.csv file Health Nutrition chatbot will give precautions such as which type of food we take and also suggest fruits and vegetables and its uses.



## Symptom\_precautions.csv file:

AZ		Tx 2 Malaria													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1		<p>An adverse drug reaction (ADR) is an injury caused by taking medication. ADRs may occur following a single dose or prolonged administration of a drug or result from the combination of two drugs.</p> <p>Uses: Incorporate blueberries into your diet by adding them to smoothies, yogurt, oatmeal, or enjoying them as a standalone snack.</p> <p>2. Spinach: Spinach is rich in vitamins A, C, and K, as well as minerals like magnesium and iron. These nutrients support overall health and may help mitigate certain drug reactions.</p> <p>Uses: Include spinach in salads, stir-fries, omelets, or blend it into green smoothies.</p> <p>3. Broccoli: Broccoli contains sulforaphane, a compound with potent antioxidant and anti-inflammatory properties. It may help protect against drug-induced toxicity and support liver health, which is important for drug metabolism.</p> <p>Uses: Steam, roast, or stir-fry broccoli as a side dish or add it to pasta, soups, or salads.</p> <p>4. Oranges: Oranges are abundant in vitamin C, which acts as a powerful antioxidant and supports immune function. Vitamin C may help reduce oxidative stress caused by certain medications.</p> <p>Uses: Enjoy oranges as a refreshing snack, or squeeze fresh orange juice to drink or use in marinades and dressings.</p> <p>5. Avocado: Avocado is a nutrient-dense fruit rich in healthy fats, vitamins E and K, potassium, and fiber. Its anti-inflammatory properties may help mitigate some drug reactions.</p> <p>Uses: Spread avocado on toast, add it to salads, sandwiches, or smoothies, or make guacamole as a dip.</p>													
	Drug Reaction	<p>An infectious disease caused by protozoan parasites from the Plasmodium family that can be transmitted by the bite of the Anopheles mosquito or by a contaminated needle or transfusion.</p> <p>Uses: Enjoy ripe papaya as a fruit or blend it into smoothies. Some people also consume papaya leaf extract as a supplement, but it's essential to consult with a healthcare professional before using it.</p> <p>2. Citrus fruits: Citrus fruits like oranges, lemons, and grapefruits are rich in vitamin C, which plays a crucial role in supporting the immune system and combating infections.</p> <p>Uses: Consume citrus fruits fresh as a snack or juice, or incorporate them into salads, marinades, or desserts.</p> <p>3. Dark leafy greens (e.g., spinach, kale): Dark leafy greens are packed with nutrients like vitamins A, C, and K, as well as minerals like iron and magnesium. These nutrients support overall health and may help mitigate certain drug reactions.</p> <p>Uses: Include dark leafy greens in salads, soups, stir-fries, or smoothies.</p> <p>4. Ginger: Ginger contains gingerol, a bioactive compound with antioxidant and anti-inflammatory properties. It may help alleviate symptoms associated with malaria, such as fever and inflammation.</p> <p>Uses: Add fresh ginger to teas, soups, stir-fries, or smoothies for its flavor and potential health benefits.</p> <p>5. Garlic: Garlic contains allicin, a compound with antimicrobial properties that may help combat infections. It also supports immune function and has been traditionally used for its medicinal properties.</p> <p>Uses: Incorporate fresh garlic into various cuisine dishes, such as soups, sauces, stir-fries, or roasted vegetables.</p>													
2															

Figure 3.2.1.7: symptom\_precaution data

As shown in figure 3.2.1.7, the data is about symptom precautions. It consists of precautions and suggest fruits and vegetables and its uses. Through that precautions user will know about fruits and vegetables and its uses.

## Module-2.2: Check symptom pattern:

```
106 # Get user's name
107 usage
108 def get_user_name():
109     return st.text_input("Hello! What's your name?")
110
111 # Check if the input matches any symptom pattern
112 usage
113 def check_pattern(dis_list, inp):
114     pred_list = []
115     inp = inp.replace(' ', '_')
116     patt = f"{inp}"
117     regexp = re.compile(patt)
118     pred_list = [item for item in dis_list if regexp.search(item)]
119     if len(pred_list) > 0:
120         return 1, pred_list
121     else:
122         return 0, []
```

**Figure 3.2.2.1:** Code for symptom pattern

As shown in figure 3.2.2, the code is for symptom pattern. After entering the symptom, it checks if the input matches any symptom in the pre-processing data files or csv files.

### Module-3: Predict disease and get description

```
106 # Get user's name
107 usage
108 def get_user_name():
109     return st.text_input("Hello! What's your name?")
110
111 # Check if the input matches any symptom pattern
112 usage
113 def check_pattern(dis_list, inp):
114     pred_list = []
115     inp = inp.replace(' ', '_')
116     patt = f"{inp}"
117     regexp = re.compile(patt)
118     pred_list = [item for item in dis_list if regexp.search(item)]
119     if len(pred_list) > 0:
120         return 1, pred_list
121     else:
122         return 0, []
```

**Figure 3.3:** Code for predict the disease

As shown in figure 3.3, the code is for disease prediction. Based upon symptom pattern matching chatbot will ask questions. Based upon our response it will predict the disease name.

```
146 def tree_to_code(tree, feature_names):
147     tree_ = tree.tree_
148     feature_name = [
149         feature_names[i] if i != _tree.TREE_UNDEFINED else "undefined!"
150         for i in tree_.feature
151     ]
152
153     chk_dis = ",".join(feature_names).split(",")
154     symptoms_present = []
155
156     while True:
157         disease_input = st.text_input("Enter your symptom:")
158         conf, cnf_dis = check_pattern(chk_dis, disease_input)
159         if conf == 1:
160             st.write("Searches related to input:")
161             for num, it in enumerate(cnf_dis):
162                 st.write(num, " ", it)
163             if num != 0:
164                 selected_index = st.selectbox(f"Select the one you meant (0 - {num}): ", options=list(range(num + 1)))
165                 disease_input = cnf_dis[selected_index]
166
167             break
168         else:
169             st.write("Enter a valid symptom.")
170
```

```

170
171     while True:
172         try:
173             num_days = st.number_input("From how many days have you experienced this symptom? (Enter a count):")
174             break
175         except:
176             st.write("Enter a valid number.")
177

```

```

178     def recurse(node, depth):
179         if tree_.feature[node] != _tree.TREE_UNDEFINED:
180             name = feature_name[node]
181             threshold = tree_.threshold[node]
182
183             if name == disease_input:
184                 val = 1
185             else:
186                 val = 0
187             if val <= threshold:
188                 recurse(tree_.children_left[node], depth + 1)
189             else:
190                 symptoms_present.append(name)
191                 recurse(tree_.children_right[node], depth + 1)
192         else:
193             present_disease = print_disease(tree_.value[node])
194
195             red_cols = reduced_data.columns
196             symptoms_given = red_cols[reduced_data.loc[present_disease].values[0].nonzero()]
197
198             st.write("Are you experiencing any of the following symptoms?")
199             symptoms_exp = []

```

```

200         for syms in list(symptoms_given):
201             inp = st.radio(f"{syms}? (yes/no):", options=["yes", "no"])
202             while inp not in ["yes", "no"]:
203                 inp = st.radio(f"Please provide a valid answer for {syms} (yes/no):", options=["yes", "no"])
204             if inp == "yes":
205                 symptoms_exp.append(syms)
206
207         second_prediction = sec_predict(symptoms_exp)
208
209         condition = calc_condition(symptoms_exp, num_days)
210         if present_disease[0] == second_prediction[0]:
211             st.write("You may have", present_disease[0])
212             st.write(description_list[present_disease[0]])
213         else:
214             st.write("You may have", present_disease[0], "or", second_prediction[0])
215             st.write(description_list[present_disease[0]])
216             st.write(description_list[second_prediction[0]])
217
218         precaution_list = precautionDictionary[present_disease[0]]
219         st.write("Take the following measures:")
220         for i, j in enumerate(precaution_list):
221             st.write(i + 1, ")", j)
222
223         recurse(node=0, depth=1)

```

```

200     for syms in list(symptoms_given):
201         inp = st.radio(f"{syms}? (yes/no):", options=["yes", "no"])
202         while inp not in ["yes", "no"]:
203             inp = st.radio(f"Please provide a valid answer for {syms} (yes/no):", options=["yes", "no"])
204         if inp == "yes":
205             symptoms_exp.append(syms)
206
207     second_prediction = sec_predict(symptoms_exp)
208
209     condition = calc_condition(symptoms_exp, num_days)
210     if present_disease[0] == second_prediction[0]:
211         st.write("You may have", present_disease[0])
212         st.write(description_list[present_disease[0]])
213     else:
214         st.write("You may have", present_disease[0], "or", second_prediction[0])
215         st.write(description_list[present_disease[0]])
216         st.write(description_list[second_prediction[0]])
217
218     precaution_list = precautionDictionary[present_disease[0]]
219     st.write("Take the following measures:")
220     for i, j in enumerate(precaution_list):
221         st.write(i + 1, ", ", j)
222
223     recurse( node=0, depth=1)

```

**Figure 3.3.1:** Code for getting the details of precautions, suggest fruits and vegetables and its uses.

As shown in figure 3.3.1, the code is for getting the details of precautions, suggest fruits and vegetables and its uses. In this code we use `getseverityDict()`, `getdescription()` and `getprecautionDict()` methods for getting the data that means solution to the particular disease.

## IV. RESULTS AND DISCUSSION

As shown in the below figures, we run the `majorproject.py` in pycharm. By running `majorproject.py` file then only the chatbot starts.

### Health Nutrition Chatbot

Hello! What's your name?

Krishnaveni

Hello, Krishnaveni!

**Figure 4.1:** Run the `majorproject.py` file

As shown in figure 4.1, we run the `majorproject.py` code in pycharm.

### Health Nutrition Chatbot

Hello! What's your name?

Krishnaveni

Hello, Krishnaveni!

**Figure 4.2:** By run the `majorproject.py` this will open

As shown in figure 4.2, By running the `majorproject.py` in pycharm it will show like enter a message that means enter your name.

# Health Nutrition Chatbot

Hello! What's your name?

Krishnaveni

Hello, Krishnaveni!

Please provide your symptoms for diagnosis:

Enter your symptom:

headache

Searches related to input:

🔍 ) headache

From how many days have you experienced this symptom? (Enter a count):

1.94

- +

Are you experiencing any of the following symptoms?

back\_pain? (yes/no):

☒ yes  
☐ no

weakness\_in\_limbs? (yes/no):

☒ yes  
☐ no

neck\_pain? (yes/no):

☒ yes  
☐ no

dizziness? (yes/no):

☒ yes  
☐ no

loss\_of\_balance? (yes/no):

☒ yes  
☐ no

**Figure 4.3:** Entering the inputs

As shown in figure 4.3, enter user name, symptom and count that means suffering from how many days. Based on entering the details Health Nutrition Chatbot asks a questions. User respond and give answers to the chatbot like say yes or no.

You may have Cervical spondylosis

Cervical spondylosis is a general term for age-related wear and tear affecting the spinal disks in your neck. As the disks dehydrate and shrink, signs of osteoarthritis develop, including bony projections along the edges of bones (bone spurs). 1. Pineapple:

Pineapple contains bromelain, an enzyme with anti-inflammatory properties that may help reduce pain and inflammation associated with cervical spondylosis. Bromelain may also aid in digestion and improve nutrient absorption. Uses: Enjoy fresh pineapple as a snack or dessert, or incorporate it into smoothies or salads. 2. Leafy greens (such as spinach, kale, and Swiss chard):

Leafy greens are rich in vitamins, minerals, and antioxidants that support overall bone and joint health. They contain vitamin K, which plays a role in bone metabolism and may help maintain bone density. Uses: Add leafy greens to salads, soups, sandwiches, or smoothies. 3. Berries (such as strawberries, blueberries, and raspberries):

Berries are packed with antioxidants, particularly anthocyanins, which have anti-inflammatory properties. These antioxidants may help reduce inflammation and oxidative stress in the body. Uses: Enjoy berries fresh as a snack, add them to yogurt, oatmeal, or smoothies. 4. Broccoli:

Broccoli is rich in vitamins C and K, as well as sulforaphane, a compound with anti-inflammatory properties. Sulforaphane may help reduce inflammation and protect against cartilage degeneration. Uses: Steam, roast, or stir-fry broccoli as a side dish, or add it to salads, soups, or pasta dishes. 5. Turmeric:

Turmeric contains curcumin, a compound known for its potent anti-inflammatory and antioxidant properties. Curcumin may help alleviate pain and inflammation associated with cervical spondylosis. Uses: Incorporate turmeric into curries, soups, stews, or brew turmeric tea.

Take the following measures:

- 1 ) use heating pad or cold pack & consult doctor
- 2 ) Exercise & take otc pain reliver
- 3 ) Maintain Good Posture
- 4 ) Pineapple Leafy greens Berries Broccoli Turmeric

Thank you for using the Healthy Nutrition ChatBot!

**Figure 4.4:** Precautions, suggest fruits and vegetables and its uses

As shown in figure 4.4, by entering the details and respond to the chatbot questions it will predict and confirm whether it is major or minor disease. If it is a major disease give suggestion like consult a doctor. If it is a minor disease give safety measures like precautions such as which type of food you have to take and suggest fruits and vegetables and its uses to increase nutrition levels.



## V. CONCLUSION AND FUTURE ENHANCEMENTS

### Conclusion:

Health Nutrition Chatbot provide precautions like food suggestions, suggest fruits and vegetables and its uses for the people by asking questions based on symptoms related to their health problem. This chatbot is very helpful to users as they can easily get precautions without paying and get a quick response from the chatbot which means without any delay it gives accurate results to the user.

### Future Enhancements:

- 1. Advanced Natural Language Processing (NLP):** Improving the chatbot's understanding and generation of natural language to better comprehend nuanced medical queries and provide accurate responses.
- 2. Emergency Response Integration:** Equipping the chatbot to handle emergency situations by providing first aid guidance or connecting users to emergency services when necessary.
- 3. Voice and Visual Interaction:** Progressing from text-based interactions to support voice recognition and possibly even visual analysis for physical symptoms.
- 4. Real-time Monitoring and Alerts:** Developing capabilities for real-time monitoring of user health data (with proper consent) to provide immediate alerts or suggestions in case of any abnormalities.
- 5. Personalized Recommendations:** Integrating machine learning algorithms to analyse user data (like medical history, lifestyle, genetic information) to offer personalized health recommendations and advice.

The future scope of this project is to enable machine to communicate with people in text format. So that people also easily communicate with machine. Also need to implement good interface. Try to improve accuracy of prediction by using sufficient data.

## REFERENCES

- [1] A. Al-Hajri, M. Al-Khalil, and A. Al-Masri, "Development of a Medical Chatbot Using AI," *International Journal of Recent Performance and Practices*, vol. 3, no. 9, 2022.
- [2] M. A. Ali, A. El-Sappagh, M. A. Al-Ghamdi, and S. A. Salem, "Machine Learning Based Healthcare Chatbot," in *2020 IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, vol. 4, no. 9, 2020.
- [3] S. R. Kadam, G. S. Gadekar, and S. R. Kadam, "A Medical Chatbot using Machine Learning and Natural Language Understanding." *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 1249-1271, 2022.
- [4] A. S. Pradeep, V. N. A. Srinivas, and K. P. Reddy, "Building a Medical Chatbot using Support Vector Machine Learning Algorithm," *Journal of Physics: Conference Series*, vol. 1716, no. 1, p. 012059, 2021.
- [5] S. Verma, A. Singh, U. Sharma, and S. Singh, "Multilingual Healthcare Chatbot Using Machine Learning," in *2020 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, vol. 4, no. 1, pp. 012058, 2020.