

**BỘ THÔNG TIN TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**BÀI TẬP NHÓM 5**

**Môn học: Nhập môn Trí tuệ nhân tạo**

**Đề tài: Chatbot chăm sóc sức khỏe**

**Đoàn Thị Trà My**

**MSV: B21DCAT134**

**Nguyễn Thảo Linh**

**MSV: B21DCCN485**

**Triệu Trung Kiên**

**MSV: B21DCAT010**

**Nguyễn Thu Hà**

**MSV: B21DCCN041**

**Hà Nội – 2024**

## MỤC LỤC

<b>I. Giới thiệu .....</b>	<b>3</b>
<b>1.1. Giới thiệu đề tài .....</b>	<b>3</b>
<b>1.2. Ngôn ngữ sử dụng .....</b>	<b>3</b>
1.2.1. Python .....	3
1.2.2. HTML .....	3
1.2.3. CSS.....	3
1.2.4. JavaScript .....	4
<b>II. Nội dung.....</b>	<b>4</b>
<b>1. Xây dựng và huấn luyện Chatbot để phân loại, dự đoán và trả lời câu hỏi .....</b>	<b>4</b>
1.1. Thu thập và xử lý dữ liệu .....	4
1.2. Sử dụng dữ liệu để huấn luyện cho mô hình học máy.....	10
<b>2. Gọi các file để chạy Chatbot bằng hàm main.....</b>	<b>12</b>
2.1. Đọc dữ liệu.....	12
2.2. Dự đoán câu trả lời và phản hồi câu hỏi .....	12
2.3. Đưa ra phản hồi cho câu hỏi .....	13
<b>3. Giao diện của Chatbot và Website .....</b>	<b>14</b>
3.1. Sử dụng HTML & CSS tạo giao diện cho Chatbot và Website .....	14
3.2. Sử dụng JavaScript để tạo giao diện, tạo hiệu ứng và xử lý dữ liệu .....	23
3.3. Kết quả .....	25
<b>III. Kết luận .....</b>	<b>26</b>
<b>IV. Tài liệu tham khảo: .....</b>	<b>26</b>

## **I. Giới thiệu**

### **1.1. Giới thiệu đề tài**

Chatbot là một chủ đề nổi tiếng và phổ biến trong lĩnh vực Trí tuệ nhân tạo (Artificial Intelligence) và Xử lý ngôn ngữ tự nhiên (Natural Language Processing). Chatbot là một ứng dụng máy tính có khả năng tương tác và trò chuyện với con người theo cách tự động và tự nhiên, giống như việc trò chuyện với một người thật.

Mục đích chính của Chatbot hỗ trợ sức khỏe là cung cấp hỗ trợ và thông tin y tế cho người dùng một cách nhanh chóng, thuận tiện và chính xác. Chatbot y tế có thể giúp người dùng tiết kiệm thời gian và tăng cường sự tự chăm sóc sức khỏe. Đồng thời, Chatbot cũng có thể là một công cụ hữu ích cho các nhà cung cấp dịch vụ y tế để tăng cường tương tác và hỗ trợ khách hàng.

### **1.2. Ngôn ngữ sử dụng**

#### **1.2.1. Python**

Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở và đa nền tảng. Python được Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm là dễ đọc, dễ học và dễ nhớ.

Một số thư viện được sử dụng trong bài tập lớn: sklearn, numpy, pandas, pyvi, re, pickle, nltk, pyopenxl.

Web Framework: Flask.

#### **1.2.2. HTML**

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng và hiển thị các trang web trên Internet. Nó định nghĩa cấu trúc và ý nghĩa của nội dung trên trang web bằng cách sử dụng các thẻ và các thuộc tính.

#### **1.2.3. CSS**

CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để mô tả cách mà các phần tử HTML được hiển thị trên trình duyệt. Nó cho phép lập trình viên điều chỉnh màu sắc, font chữ, khoảng cách, kích thước và nhiều thuộc tính khác của các phần tử trong trang web để tạo ra giao diện người dùng hấp dẫn và thú vị. CSS giúp tách biệt phần nội dung của trang web và cách mà nó được hiển thị, giúp cho việc thiết kế và bảo trì trang web trở nên dễ dàng hơn.

## 1.2.4. JavaScript

JavaScript là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu trong việc phát triển ứng dụng web. Được tích hợp trực tiếp vào trình duyệt web, JavaScript cho phép tương tác động và thay đổi nội dung của trang web một cách linh hoạt và đáp ứng. Nó được sử dụng để thêm các tính năng động như xử lý sự kiện, thao tác với DOM (Document Object Model), tạo và điều khiển hiệu ứng, gửi và nhận dữ liệu từ máy chủ, và nhiều nhiệm vụ khác. JavaScript là một phần không thể thiếu trong phát triển web hiện đại, giúp tạo ra các trang web có trải nghiệm người dùng tốt hơn và tương tác hơn.

## II. Nội dung

### 1. Xây dựng và huấn luyện Chatbot để phân loại, dự đoán và trả lời câu hỏi

#### 1.1. Thu thập và xử lý dữ liệu

```
from sklearn import preprocessing
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from pyvi import ViTokenizer
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
import pickle
import nltk
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
```

- **from sklearn import preprocessing:** import module preprocessing từ thư viện sklearn cho tiền xử lý dữ liệu trước khi huấn luyện mô hình.
- **import numpy as np:** import thư viện numpy để sử dụng cho các phép toán số học và ma trận.
- **import pandas as pd:** import thư viện pandas để làm việc với dữ liệu dạng bảng (DataFrame) và dữ liệu chuỗi thời gian.
- **from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer:** import CountVectorizer và TfidfVectorizer để biểu diễn văn bản thành ma trận số học.
- **from pyvi import ViTokenizer:** import ViTokenizer để tách từ cho tiếng Việt.
- **import re:** import thư viện re để thực hiện các phép biến đổi và tìm kiếm chuỗi sử dụng các biểu thức chính quy (regular expressions).

- **from sklearn.model\_selection import train\_test\_split:** import module train\_test\_split để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
- **from sklearn.naive\_bayes import MultinomialNB:** import mô hình Naive Bayes dạng Multinomial. Đây là một mô hình học máy được sử dụng cho phân loại văn bản.
- **import pickle:** import thư viện pickle để thực hiện nén và giải nén đối tượng Python.
- **import nltk:** import thư viện nltk cho xử lý ngôn ngữ tự nhiên (NLP).
- **from sklearn.model\_selection import Kfold:** import KFold để thực hiện phân chia dữ liệu thành các tập con cho cross-validation.
- **from sklearn.model\_selection import cross\_val\_score:** import module cross\_val\_score để đánh giá hiệu suất của mô hình qua cross-validation.
- **from sklearn.metrics import accuracy\_score:** import module accuracy\_score để tính toán chỉ số độ chính xác của mô hình phân loại.

Ở phần đọc dữ liệu này, bọn em sử dụng đường dẫn của file excel đã được tạo sẵn chứa câu hỏi và câu trả lời muốn đào tạo cho Chatbot.

```
data = pd.read_excel('E:\\Data of TuTran\\A06- Kỳ 2 năm 3\\Nhập môn AI\\BTL TTNT\\HealthCare-Chatbot\\data\\DATACHATBOT.xlsx')
X= data["question"]
Y= data["answer"]
print("thuoc tinh dieu kien")
print(X)
print("thuoc tinh can du doan")
print(Y)
```

File data của chatbot như sau, với hai cột bên trái là question còn bên phải là answer, gán cột question cho X, cột answer cho Y.

460	sung do mat	hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ
461	giảm sung do mat	hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ
462	giảm mắt đỏ sung	hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ
463	cải thiện mắt đỏ và sung	hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ
464	giảm tình trạng mắt đỏ sung	hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ
465	Có cách nào để giảm nguy cơ mắc bệnh tim mạch?	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
466	giảm nguy cơ mắc bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
467	hạn chế bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
468	cách phòng ngừa bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
469	phòng tránh bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
470	ngăn ngừa bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
471	tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
472	bệnh tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
473	tim mạch bệnh	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
474	duy trì tim mạch khỏe mạnh	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
475	giữ ổn định tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
476	cách giúp tim mạch khỏe mạnh	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
477	ổn định tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
478	làm sao để đảm bảo sức khỏe tim mạch	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
479	tim mạch khỏe mạnh	hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu
480	Làm thế nào để duy trì sức khỏe răng miệng	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
481	cách nào để răng miệng khỏe	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
482	để răng miệng khỏe cần làm gì	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
483	răng miệng khỏe mạnh	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
484	răng miệng tốt	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
485	làm sao để răng miệng luôn đảm bảo	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
486	sức khỏe răng miệng	hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ
487	làm sao để giảm nguy cơ sâu răng	hãy đánh răng đúng cách, sử dụng chỉ floss, hạn chế thức ăn có đường, tránh thức ăn và đồ uống chua, và kiểm tra định kỳ với nha sĩ.



In ra màn hình X và Y:

```
thuoc tinh dieu kien
0      Tạm biệt bot
1      Goodbye
2      Tạm biệt mà y nhé bot
3      Chào tạm biệt
4      Ngày mai gặp lại nhé bot
...
555    mệt mỏi
556    hết mệt mỏi
557    cách giảm bớt mệt mỏi
558    điều trị mệt mỏi
559    mệt mỏi nên làm gì
Name: question, Length: 560, dtype: object

thuoc tinh can du doan
0      bye
1      bye
2      bye
3      bye
4      bye
...
555    hãy nghỉ ngơi đầy đủ , ăn uống đầy đủ , tập th...
556    hãy nghỉ ngơi đầy đủ , ăn uống đầy đủ , tập th...
557    hãy nghỉ ngơi đầy đủ , ăn uống đầy đủ , tập th...
558    hãy nghỉ ngơi đầy đủ , ăn uống đầy đủ , tập th...
559    hãy nghỉ ngơi đầy đủ , ăn uống đầy đủ , tập th...
Name: answer, Length: 560, dtype: object
```

Mã hóa Y:

```
le = preprocessing.LabelEncoder()#tạo một đối tượng LabelEncoder từ module preprocessing trong scikit-learn.
#Đối tượng này được sử dụng để mã hóa dữ liệu chuỗi thành số nguyên.
le.fit(Y)#sử dụng đối tượng LabelEncoder để "học" từ dữ liệu trong biến Y.
#Điều này có nghĩa rằng nó xác định tất cả các giá trị duy nhất trong Y và gán một số nguyên với mỗi giá trị duy nhất.
list_label = list(le.classes_)#tạo một danh sách list_label chứa tất cả các giá trị duy nhất trong Y.
#Điều này là hữu ích để theo dõi các giá trị đã được gán số nguyên tương ứng.
print(list_label)
print(len(list_label))
label = le.transform(Y)#sử dụng LabelEncoder để chuyển đổi dữ liệu trong Y thành dãy số nguyên tương ứng. Kết quả được lưu trong biến label,
#và dãy số này thường được sử dụng trong các mô hình học máy vì chúng là các giá trị số thay vì chuỗi.
print(label)
```

Kết quả của câu lệnh in ra trong code trên lần lượt là:

[ 'Bạn có thể nghỉ ngơi, thực hiện các biện pháp thư giãn, uống nước đủ, và sử dụng thuốc chống đau theo hướng dẫn.', 'Để tăng cường miễn dịch, bạn nên ăn đủ thức ăn có chứa vitamin C và D, duy trì giấc ngủ đủ giờ, tập thể dục đều đặn, và hạn chế tiếp xúc với người bệnh.', 'Bạn nên duy trì một chế độ ăn uống và lối sống lành mạnh, tránh căng thẳng quá mức, đảm bảo đủ giấc ngủ, và kiểm tra liệu có thức ăn hoặc thức uống gây ra đau đầu cho bạn.', 'Bạn nên nghỉ ngơi một chút', 'Có, một số phương pháp tự nhiên như thư giãn, massage, thiền, và thay đổi chế độ ăn uống có thể giúp giảm đau đầu', 'Duy trì chế độ ăn uống cân đối và giàu chất xơ, vitamin và khoáng chất, sử dụng sản phẩm chăm sóc tóc và da đầu phù hợp với loại tóc và da đầu của bạn', 'Khí đau bụng, bạn nên nghỉ ngơi, uống nước, sử dụng thuốc kháng đau nhẹ. Nếu đau kéo dài hoặc nghiêm trọng, hãy thăm bác sĩ.', 'Mắt ngủ có thể do căng thẳng, lo âu, sử dụng thiết bị điện tử trước khi ngủ, thay đổi môi trường ngủ, sử dụng thuốc, hoặc vấn đề sức khỏe.', 'Nguyên nhân gây đau bụng bao gồm tiêu hóa kém, nhiễm trùng, stress, dị ứng thức ăn, tổn thương, và nhiều bệnh lý khác.', 'Nguyên nhân đau đầu có thể đa dạng, bao gồm căng thẳng, thiếu ngủ, môi trường xung quanh, tiền sử gia đình, sử dụng quá nhiều caffeine, hoặc vấn đề y tế khác.', 'Nếu mắt bị đỏ và sưng, bạn nên tránh chà mắt, giữ mắt sạch sẽ, và thăm bác sĩ để kiểm tra và xác định nguyên nhân.', 'Rửa mũi bằng nước muối sinh lý và sử dụng máy tạo ẩm để giảm triệu chứng sổ mũi, tránh tiếp xúc với chất gây dị ứng và sử dụng thuốc antihistamine nếu cần thiết', 'Sổ mũi có thể do nhiều nguyên nhân khác nhau, bao gồm cảm lạnh, viêm nhiễm, dị ứng, viêm xoang, viêm amidan, và tiền sử bệnh lý.', 'Tiêu thụ đủ lượng canxi và vitamin D, tập thể dục để tăng cường sức mạnh xương và linh hoạt khớp, hạn chế tiêu thụ cồn và thực phẩm chua.', 'Tránh tiếp xúc với âm thanh quá lớn và sử dụng bảo vệ tai khi cần thiết, thăm bác sĩ tai mũi họng định kỳ để kiểm tra tai và thính giác của bạn', 'Tập thể dục đều đặn, ăn đủ protein và chế độ ăn uống cân đối, thực hiện bài tập sức mạnh, nghỉ ngơi và phục hồi, hạn chế căng thẳng, uống đủ nước', 'Tắt điện thoại, thở sâu, tập thể dục ngắn, nghỉ ngơi, thư giãn âm nhạc, uống nước, và cười.', 'Tự chăm sóc mắt khi bị đau mắt bao gồm việc giữ mắt sạch sẽ, tránh chà mắt, không sử dụng kính áp tròng khi mắt bị viêm nhiễm, và sử dụng nước muối sinh lý để rửa mắt.', 'bye', 'bạn có thể thử sử dụng các sản phẩm chứa acid salicylic hoặc benzoyl peroxide. Đảm bảo làn da được làm sạch đúng cách và tránh chà xát mạnh', 'bạn cần duy trì chế độ ăn uống cân đối và vận động thường xuyên. Hãy cân nhắc giới hạn khẩu phần và ăn nhiều rau và trái cây hơn.', 'bạn nên duy trì một chế độ ăn uống và lối sống lành mạnh, tránh căng thẳng quá mức, đảm bảo đủ giấc ngủ, và kiểm tra liệu có thức ăn hoặc thức uống gây ra đau đầu cho bạn.', 'cảm ơn bạn', 'cố lên', 'duy trì sự cân đối trong chế độ ăn uống và cung cấp đủ nước cho cơ thể để tránh tiếp xúc quá mức với ánh nắng mặt trời', 'giữ mắt sạch sẽ, tránh chà mắt, không sử dụng kính áp tròng khi mắt bị viêm nhiễm, và sử dụng nước muối sinh lý để rửa mắt, đến khám ở các cơ quan y tế', 'giữ mắt sạch sẽ, tránh chà mắt, không sử dụng kính áp tròng khi mắt bị viêm nhiễm, và sử dụng nước muối sinh lý để rửa mắt.', 'hãy duy trì chế độ ăn uống cân đối, tập thể dục đều đặn, và tìm kiếm sự hỗ trợ từ một chuyên gia dinh dưỡng hoặc huấn luyện viên', 'hãy duy trì thói quen ngủ đều đặn, tạo môi trường ngủ thoải mái, hạn chế sử dụng điện thoại và máy tính trước khi đi ngủ', 'hãy nghỉ ngơi đầy đủ, ăn uống đầy đủ, tập thể dục đều đặn, thư giãn và giải trí', 'hãy sử dụng kem dưỡng ẩm hợp lý, uống nhiều nước, và tránh tắm nước quá nhiệt để giữ da mềm mại', 'hãy thực hiện kiểm tra mắt định kỳ, duy trì chế độ ăn uống giàu chất bảo vệ mắt như beta-carotene và vitamin A, và hạn chế tiếp xúc quá nhiều với ánh nắng mặt trời, xanh', 'hãy thực hiện thiền, tập luyện thể dục, quản lý thời gian, ăn cân đối, và ngủ đủ giấc.', 'hãy thực hiện tạm ngừng mắt, điều chỉnh độ sáng màn hình, và thực hiện khoảng nghỉ ngắn', 'hãy tránh tiếp xúc với hút thuốc và khói, sử dụng máy tạo ẩm trong phòng nếu cần, và thường xuyên rửa mắt để duy trì sự ẩm cho mắt', 'hãy tránh tiếp xúc với tác nhân gây dị ứng, sử dụng thuốc dị ứng được chỉ định bởi bác sĩ, và giữ môi trường sạch sẽ', 'hãy ăn ít chất béo bão hòa và đường, tập thể dục đều đặn, duy trì trọng lượng lý tưởng, và kiểm soát áp lực máu', 'hãy đánh răng ít nhất hai lần mỗi ngày, sử dụng chỉ floss để làm sạch giữa răng, và thực hiện kiểm tra răng định kỳ với nha sĩ', 'hãy đánh răng đúng cách, sử dụng chỉ floss, hạn chế thức ăn có đường, tránh thức ăn và đồ uống chua, và kiểm tra định kỳ với nha sĩ.', 'hi hi ...', 'noanswer', 'thật', 'tôi không biết', 'tôi được tạo bởi các thành viên trong nhóm 16 môn mã nguồn mở', 'xin chào bạn, chúng tôi có thể giúp được gì cho bạn', 'xin lỗi bạn', 'Ăn nhiều rau xanh và thực phẩm giàu chất xơ, uống đủ nước và tránh tiêu thụ quá nhiều thức ăn chua hoặc có đường', 'Đau mắt có thể do nhiều nguyên nhân khác nhau, bao gồm viêm nhiễm, tổn thương, dị ứng, tiền sử bệnh lý, và căng thẳng mắt.', 'Để giảm mất ngủ, quản lý stress, thiết lập thói quen ngủ tốt, và duy trì môi trường ngủ thoải mái. Nếu u mắt ngủ kéo dài, bạn nên thăm bác sĩ để tìm hiểu phương pháp điều trị.', 'Để giảm nguy cơ mất ngủ sau khi bơi, hãy đảm bảo sử dụng kính bơi để bảo vệ mắt khỏi hóa chất trong nước bể và nước biển', 'Để điều trị đau đầu căng thẳng, bạn có thể nghỉ ngơi, thực hiện các biện pháp thư giãn, uống nước đủ, và sử dụng thuốc chống đau theo hướng dẫn.']

[illegible]

Xử lý dữ liệu của X:

```
def tienxuly(document):  
    document = ViTokenizer.tokenize(document)  
    # đưa về lower  
    document = document.lower()  
    # xóa các ký tự không cần thiết  
    document = re.sub(r'  
[^\s\wáàâãäåääääăǻāāāāāēèěẽêëēēēōòóôõöőűüíîïìúûũůűŭřŕſşýÿıđ_]', ' ',  
document)  
    # xóa khoảng trắng thừa  
    document = re.sub(r'\s+', ' ', document).strip()  
    return document  
  
for i in range(0,X.count()):  
    X[i] = tienxuly(X[i])
```

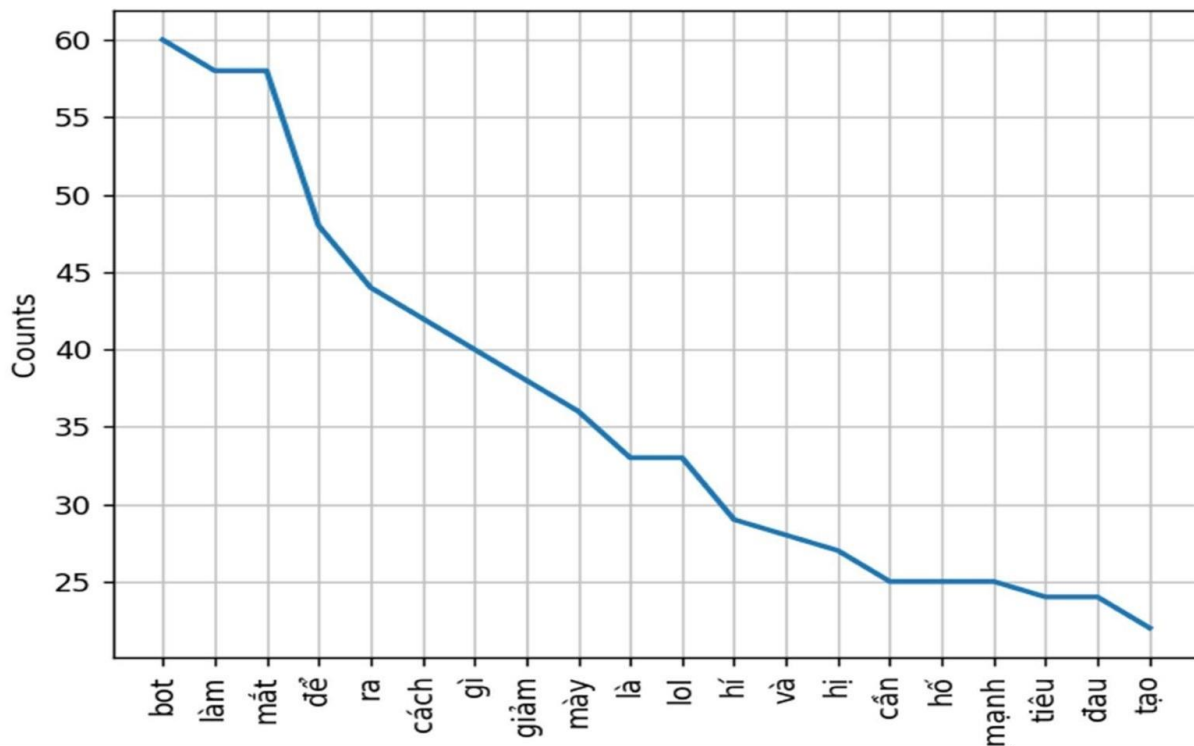
Khi chạy xong đoạn code trên thì dữ liệu của X là:

```
0          tạm_biệt bot
1          goodbye
2      tạm_biệt mày nhé bot
3          chào tạm_biệt
4      ngày_mai gặp lại nhé bot
    ...
555          mệt_mỏi
556          hết mệt_mỏi
557      cách giảm bớt mệt_mỏi
558          điều_trị mệt_mỏi
559          mệt_mỏi nên làm gì
Name: question, Length: 560, dtype: object
```

Tiếp theo đó là tìm những từ có tần suất xuất hiện nhiều nhất (stopword) trong phần dữ liệu của X:

[illegible]

Biểu đồ tần suất xuất hiện sẽ như sau:





Như vậy là đã biết được một số từ xuất hiện thường xuyên trong các câu hỏi và nó sẽ ảnh hưởng đến mô hình dự đoán. Vì vậy chúng em sẽ loại bỏ chúng để model có độ chính xác cao hơn.

```
stopword = ["bot", "ra"]

def remove_stopwords(line):
    words = []
    for word in line.strip().split():
        if word not in stopwords:
            words.append(word)
    return ' '.join(words)

for i in range(0, X.count()):
    X[i] = remove_stopwords(X[i])
```

Xây dựng bộ từ điển cho mô hình học máy.

```
#buoc tiep theo chung ta se xay dung bo tu dien cho may hoc
vectorizer = CountVectorizer()#Tạo một đối tượng CountVectorizer.
#Đây là một công cụ trong Scikit-learn dùng để chuyển đổi dữ liệu văn bản thành ma trận đếm từ (count matrix).

def transform(data):#Hàm này được tạo ra để chuyển đổi dữ liệu văn bản thành ma trận đếm từ.
    data= list(data)
    return vectorizer.fit_transform(data).todense()#Sử dụng đối tượng CountVectorizer để học từ điển từ dữ liệu data
#và chuyển đổi dữ liệu văn bản thành ma trận đếm từ.
#Mỗi hàng trong ma trận này tương ứng với một đoạn văn bản và mỗi cột là một từ từ từ điển.
#Giá trị tại mỗi phần tử của ma trận là số lần xuất hiện của từ tương ứng trong đoạn văn bản.

data1 = transform(X)#biến đổi nó thành ma trận đếm từ và gán kết quả vào biến data1.
print(data1)
```

Kết quả của data 1 sẽ như sau:

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

## 1.2. Sử dụng dữ liệu để huấn luyện cho mô hình học máy

```
#chia du lieu ra lam 2 phan
X_train, X_test, Y_train, Y_test = train_test_split(data1, label, test_size=0.1, random_state=0)
print(X_train)
print(X_test)
print(Y_train)
print(Y_test)
```

Mã trên sử dụng thư viện Scikit-learn để chia dữ liệu thành tập huấn luyện (training set) và tập kiểm tra (testing set) bằng cách sử dụng hàm “train\_test\_split” trong đó:

- “data1” là ma trận đếm từ (ma trận đặc trưng) chứa dữ liệu văn bản được biến đổi từ “X”.
- “label” là dãy nhãn (labels) hoặc kết quả tương ứng với từng mẫu dữ liệu trong “data1”.
- “test\_size=0.1” xác định tỷ lệ của dữ liệu được sử dụng cho tập kiểm tra. Trong trường hợp này, 10% của dữ liệu sẽ được dành cho tập kiểm tra và 90% còn lại dành cho tập huấn luyện.
- “random\_state=0” là một số nguyên được sử dụng để khởi tạo máy tạo số ngẫu nhiên. Nếu thiết lập cùng một “random\_state”, kết quả sẽ đảm bảo được tái lập mỗi khi chạy chương trình.

Kết quả sau khi chạy lệnh print:

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 1]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
...
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]]
[ 3 45 13 39 36 44 34 14 35 40 40 45 40 22 32 44 43 42 44 22 22 10 40 40
 22 49 10 22 40 29 40 22 20 1 48 39 43 4 2 30 34 14 39 45 15 0 22 24
 46 40 18 12 40 18 12 43 6 35 40 22 40 22 26 19 45 42 48 29 30 43 44 31
 33 49 12 22 40 46 40 22 27 43 30 40 5 40 33 43 7 18 32 14 40 15 33 22
 22 40 40 22 44 29 44 22 40 39 14 23 22 43 22 18 18 15 24 40 37 28 12 44
 3 16 40 45 39 40 41 38 43 40 36 22 13 38 5 29 36 40 40 22 40 40 30 40
 40 44 24 44 8 44 15 44 22 44 40 22 40 18 46 40 10 8 14 36 18 43 28 40
 43 14 43 37 44 22 40 44 30 28 18 40 11 1 39 40 14 22 5 28 43 22 31 6
 45 3 32 15 40 32 40 49 39 41 40 40 22 7 30 40 22 20 22 22 24 5 36 47
 22 13 11 37 40 43 43 44 14 10 39 3 38 18 40 5 43 23 40 25 27 22 42 36
 40 36 40 7 34 48 11 43 15 39 11 3 37 31 39 40 43 44 22 23 44 22 40 10
 32 16 13 40 41 42 22 23 44 7 44 34 33 33 50 5 40 44 30 48 40 36 33 43
 40 22 31 49 31 40 37 40 40 14 40 40 39 22 44 40 42 40 43 36 22 40 39 46
 44 45 36 22 20 20 25 18 23 26 43 40 40 31 40 22 22 6 20 44 22 43 43 18
 27 47 22 35 31 43 40 39 11 34 3 47 13 40 30 37 44 25 23 36 39 48 40 37
 18 33 43 45 23 22 40 8 36 32 45 23 5 40 44 22 3 39 18 48 35 22 39 44
 40 22 44 39 30 44 40 46 46 14 14 43 40 27 22 29 16 43 40 44 36 46 44 8
 20 18 24 40 19 8 35 14 22 1 11 41 27 23 40 8 39 22 40 22 15 40 40 22
 41 44 12 6 28 40 45 14 40 43 43 47 5 40 7 35 43 46 43 44 49 39 17 40
 22 28 22 22 33 22 23 22 3 39 34 44 1 10 45 38 44 5 43 44 40 35 13 43
 40 39 23 27 43 40 22 49 33 22 22 5 40 22 38 3 28 36 22 45 18 7 40 29]
[13 19 22 10 40 31 46 15 7 13 40 29 22 44 22 32 40 19 9 40 40 14 13 5
 40 40 22 20 22 40 12 18 38 3 35 14 49 36 40 22 40 21 40 28 5 27 40 47
 19 40 18 13 38 23 16 18]
```

```
X_test.shape#cho biết số hàng và số cột của X_test
# tiếp theo là chuyển dữ liệu từ dạng matrix sang array
X_train1=np.asarray(X_train)
X_test1=np.asarray(X_test)
Y_train1=np.asarray(Y_train)
Y_test1=np.asarray(Y_test)
```

Sử dụng mô hình học máy Multinomial Naive Bayes (MultinomialNB) để phân loại dữ liệu và đánh giá hiệu suất. Mô hình này đặc biệt phù hợp cho các dữ liệu đặc trưng rời rạc như văn bản hoặc dữ liệu đếm.

```
clf= MultinomialNB()#một mô hình Multinomial Naive Bayes được khởi tạo và lưu trong biến clf
scores = cross_val_score(clf, X_train1, Y_train1, cv=5) #sử dụng kiểm tra chéo (cross-validation) để đánh giá hiệu suất của mô hình clf
#Tham số cv=5 chỉ định rằng kiểm tra chéo sẽ được thực hiện với 5 fold (phân chia dữ liệu thành 5 phần và thực hiện kiểm tra trên mỗi phần)
print("%.2f accuracy with a standard deviation of %.2f" % (scores.mean()*100, scores.std()))
```

Hiệu suất sẽ nhận được bằng:

```
75.39 accuracy with a standard deviation of 0.03
```

Đưa dữ liệu vào cho model học bằng câu lệnh:

```
clf.fit(X_train,Y_train)
```

Sau khi nạp dữ liệu cho model học chúng em sẽ kiểm tra độ chính xác giữa dữ liệu dự đoán và dữ liệu test.

```
#kiem tra voi du lieu testing xem do chinh xac cua no la bao nhieu
prediction = clf.predict(X_test1)
print("Độ chính xác so với dữ liệu dự đoán và dữ liệu test là: ")
accuracy = accuracy_score(Y_test1, prediction)
precision = precision_score(Y_test1, prediction, average='weighted')
recall = recall_score(Y_test1, prediction, average='weighted')
f1 = f1_score(Y_test1, prediction, average='weighted')

print(f'Accuracy: {accuracy:.2f}')
print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
print(f'F1 Score: {f1:.2f}')
```

Kết quả thu được:

```
Accuracy: 0.71
Precision: 0.60
Recall: 0.71
F1 Score: 0.64
```

Như vậy chatbot có độ chính xác là hơn 70%. Sau đó dữ liệu sẽ được ghi vào 3 file có đuôi .pkl như hình dưới:

```
pickle.dump(vectorizer.vocabulary_, open('vocab.pkl', 'wb')) #lưu lại
pickle.dump(clf, open('NB_ChatBot_model.pkl', 'wb')) #lưu model lại
#lưu cách mã hóa của nhan lại
pickle.dump(le, open('decode_label.pkl', 'wb'))
```

## 2. Gọi các file để chạy Chatbot bằng hàm main

### 2.1. Đọc dữ liệu

```
clf = pickle.load(open('E:\\Data of TuTran\\A06- Kỳ 2 năm 3\\Nhập môn AI\\BTL TTNT\\HealthCare-Chatbot\\colab\\NB_ChatBot_model.pkl', 'rb'))
vocabulary_to_load = pickle.load(open('E:\\Data of TuTran\\A06- Kỳ 2 năm 3\\Nhập môn AI\\BTL TTNT\\HealthCare-Chatbot\\colab\\vocab.pkl', 'rb'))
le = pickle.load(open('E:\\Data of TuTran\\A06- Kỳ 2 năm 3\\Nhập môn AI\\BTL TTNT\\HealthCare-Chatbot\\colab\\decode_label.pkl', 'rb'))
```

Giải nén 3 file .pkl đã được tạo ra từ file xulytiengviet.py và gán vào các biến để xử lý dữ liệu.

### 2.2. Dự đoán câu trả lời và phản hồi câu hỏi

```
def prediction(input):
    ngram_size = 1
    loaded_vectorizer = CountVectorizer(ngram_range=(ngram_size, ngram_size), min_df=1, vocabulary=vocabulary_to_load)
    loaded_vectorizer._validate_vocabulary()
    a = tienxuly(input)

    input1 = remove_stopwords(a)
    vect = loaded_vectorizer.transform([input1]).toarray()
    predict = clf.predict(vect)
    predict = le.inverse_transform(predict)[0]

    if predict=="noanswer":
        predict= "xin lỗi bạn, câu này tôi không biết trả lời như thế nào. Bạn vui lòng liên hệ theo số điện thoại 12345"

    return predict
```

a. ngram\_size = 1: Đặt kích thước n-gram là 1. N-gram là một mô hình ngôn ngữ trong xử lý ngôn ngữ tự nhiên, nó được sử dụng để dự đoán từ tiếp theo trong một chuỗi văn bản dựa trên n từ trước đó.



- b. Câu lệnh `loaded_vectorizer = CountVectorizer(ngram_range = (ngram_size, ngram_size), min_df = 1, vocabulary = vocabulary_to_load)` để khởi tạo một đối tượng `CountVectorizer` với kích thước n-gram (được đặt là 1), `min_df = 1` (tức Tổ 5 nhóm 02 Nhập môn trí tuệ nhân tạo là từ phải xuất hiện ít nhất 1 lần trong tập dữ liệu để được coi là hợp lệ) và từ điển được tải từ .
- c. `loaded_vectorizer._validate_vocabulary()`: Kiểm tra từ điển đã tải có hợp lệ hay không.
- d. `a = tienxuly(input)`: Tiền xử lý đầu vào bằng cách gọi hàm `tienxuly()`, hàm này thực hiện các tác vụ như tách từ, chuyển đổi văn bản thành chữ thường, loại bỏ các ký tự không cần thiết và loại bỏ khoảng trắng thừa.
- e. `input1 = remove_stopwords(a)`: Loại bỏ các từ dừng (những từ xuất hiện nhiều trong câu hỏi) từ văn bản đã được tiền xử lý.
- f. `vect = loaded_vectorizer.transform([input1]).toarray()`: Chuyển đổi văn bản đã được tiền xử lý và loại bỏ từ dừng thành một vector đặc trưng sử dụng `CountVectorizer`.
- g. `predict = clf.predict(vect)`: Dự đoán nhãn của văn bản dựa trên vector đặc trưng bằng cách sử dụng mô hình học máy `clf`.
- h. `predict = le.inverse_transform(predict)[0]`: Chuyển đổi nhãn dự đoán từ dạng số về dạng văn bản gốc sử dụng `le` (một đối tượng của `LabelEncoder`).
- i. `if predict == "noanswer": predict= "xin lỗi bạn,..."`: Nếu mô hình không thể dự đoán một câu trả lời, thì trả về một thông báo mặc định.
- j. `return predict`: Trả về câu trả lời dự đoán.

### 2.3. Đưa ra phản hồi cho câu hỏi

```
def chatbot_response():  
    if request.method == "POST":  
        message = request.form.get("msg")  
        ok = prediction(message)  
    return ok
```



- Tạo một hàm “chatbot\_response()” để xử lý yêu cầu từ phía người dùng. Trong hàm này, nếu phương thức yêu cầu là “POST”, chatbot sẽ lấy nội dung của trường “msg” từ yêu cầu và gọi hàm “prediction()” để dự đoán câu trả lời cho tin nhắn.
- Sau đó, giá trị trả về từ hàm “prediction()” được gán cho biến “ok”. Cuối cùng, hàm “chatbot\_response()” trả về giá trị của biến “ok”.

### 3. Giao diện của Chatbot và Website

#### 3.1. Sử dụng HTML & CSS tạo giao diện cho Chatbot và Website

Trang web sẽ được chia làm 2 phần, phần 1 là giao diện Website làm background và phần 2 là cửa sổ Chatbox.

Khởi tạo 1 file HTML, đặt tên là index.html, lý do đặt tên là index vì khi đăng tải trang Web lên các Hosting, đa phần chúng sẽ tự động truy cập vào file có tên là index.html mà không cần nhập tên tệp, từ đó giảm thiểu việc gặp lỗi 404 Not Found.

**Thẻ <head></head>:**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- Tiêu đề website -->
  <title>Healthcare Chatbot</title>

  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='chat.css')}}" />
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css')}}" />

  <script src="https://kit.fontawesome.com/dcbd7d5889.js" crossorigin="anonymous"></script>

  <link rel="icon" href="https://images.ctfassets.net/mrop88jh71hl/55rrbZfwMaURHZKAUc5oOW/9e5Fe805eb03135b82e962e92169ce6d/pyt" />

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <!-- Kết nối jquery online -->
</head>
```

- Đặt ngôn ngữ mặc định là Tiếng Anh, mã hóa kí tự Unicode bằng UTF-8 và cài đặt chế độ hiển thị 100% trên máy tính.
- Kết nối 2 file .css với index.html bao gồm: chat.css (cho giao diện boxchat) và style.css (chỉnh sửa giao diện trang web) bằng lệnh link.
- Kết nối với 1 số trang web online để sử dụng thêm 1 số chức năng như Fontawesome hay thư viện jQuery bằng lệnh link.

## Website-background

### Thẻ <body></body>

```
<body>
  <!-- WEB-BACKGROUND -->

  <!-- Header -->
  <div class="header">
    <!-- logo -->
    <div class="header-logo">
      
    </div>

    <!-- Name -->
    <div class="header-name">
      <p class="header-tv">HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG</p>
      <p class="header-ta">Posts and Telecommunications Institute of Technology</p>
      <p class="header-gr">
        <i class="fa-brands fa-python fa-bounce"></i>
        Tổ 05 Nhóm 02 Nhập môn trí tuệ nhân tạo</p>
      <p class="header-topic">
        Chủ đề: Chatbot
      </p>
    </div>
  </div>
```

Header được chia làm 3 cột: header-logo để lưu logo PTIT, header-name để lưu tên trường và nhóm, chủ đề bài tập lớn và header-options để hiển thị các tùy chọn khác như thanh tìm kiếm, thay đổi ngôn ngữ...

- Thẻ <p> dùng để lưu nội dung văn bản, thẻ <i> để hiển thị icon.
- Thẻ <img> để hiển thị ảnh lên website.
- Thẻ <a> là nơi lưu trữ đường dẫn (url).

## CSS - Header

```
/* Xây dựng bố cục cho mục header, chia header thành 3 phần */
.header{
  padding: 0 72px;
  display: flex;
  height: 160px;
  background-image: linear-gradient(90deg, #ADA996, #F2F2F2, #DBDBDB, #EAEAEA);
}

/* Xây dựng header-logo */
.header-logo{
  width: 250px;
  padding-left: 20px;
  padding-top: 32px;
}

.header-logo .logo{
  width: 100%;
}

/* Xây dựng header-name */
.header-name{
  text-align: center;
  padding: 32px 20px;
  width: 650px;
  line-height: 30px;
}
```

```

.header-name .header-tv{
  font-size: 28px;
  font-weight: 550;
  color: red;
}

.header-name .header-ta{
  font-size: 24px;
  color: blue;
  letter-spacing: 0.5px;
  text-shadow: 0.1px 0.1px 2px gray;
}

.header-name .header-gr,
.header-name .header-topic{
  font-size: 20px;
}

.header-name .header-topic{
  text-decoration: underline;
}

/* Xây dựng header-options */
.header-options{
  width: 150px;
  padding-top: 32px;
  padding-left: 30px;
}

.header-options .help-line{
  font-size: 20px;
  color: blue;
}

/* Tạo hiệu ứng khi di chuột vào nút Trợ giúp */
.header-options .help-line:hover{
  text-decoration: underline;
  opacity: 0.7;
  color: red;
}

```

## HTML – Navigator

Dùng unordered list để tạo ra một danh mục các lựa chọn trên thanh điều hướng.

```

<!-- Tạo 1 navigavtor bar (thanh điều hướng)-->
<div class="nav-bar">
  <ul class="nav">
    <li><a href="#">Trang chủ</a></li>
    <li><a href="#">Giới Thiệu</a></li>
    <li><a href="#">Giáo dục và đào tạo</a></li>
    <li><a href="#">Khoa học và công nghệ</a></li>
    <li><a href="#">Hợp tác quốc tế</a></li>
    <li><a href="#">Tuyển sinh</a></li>
    <li class="chosen" ><a href="#">Healthcare Chatbot</a></li>
    <li><a href="#">Thư viện</a></li>
    <li><a href="#">Thư điện tử</a></li>
  </ul>
</div>

```

## CSS - Navigator

```
/* Xây dựng thanh điều hướng */
.nav-bar{
  width: 100%;
  text-align: center;
  background-image: linear-gradient(90deg, #ADA996, #F2F2F2, #00D800, #EAEAEA);
  border-bottom: 3px solid red;
  font-size: 20px;
}

/* Kéo nó trong thanh điều hướng thành 1 hàng bằng inline-block */
.nav li{
  display: inline-block;
  padding: 16px 8px;
}

.nav li a{
  color: black;
}

.nav li: hover{
  background-color: white;
  color: red;
}
```

```
/* tạo hiệu ứng khi trỏ chuột vào các mục*/
.nav li a: hover{
  opacity: 0.9;
  text-decoration: underline;
  color: rgb(209, 34, 34);
  transition: 0.3s ease;
}

.nav .chosen{
  background-color: white;
}

.nav .chosen a{
  opacity: 0.9;
  color: rgb(209, 34, 34);
}
```

## HTML - Content

```
<!-- Content -->
<div class="contentx">
  <div>
    
  </div>
</div>
```

Dùng thẻ <img> để hiển thị ảnh lên màn hình.

## CSS - Căn chỉnh độ lớn và mờ đục của ảnh cũng như cách hiển thị

```
/* Xây dựng phần nd chỉ bằng 1 bức ảnh */
.contentx .contentx-pic{
    width: 1349px;
    opacity: 0.8;
    display: block;
}
```

## HTML - Footer

```
<!-- Footer bao gồm 3 thẻ con: name-copy-right, icon-footer và author-->
<div class="footer">
    <!-- -->
    <div class="name-copyright">
        Healthcare Chatbot
    </div>
    <!-- -->
    <div class="icon-footer">
        <i style="color: blue;" class="fa-brands fa-facebook fa-beat"></i>
        <i style="color: dodgerblue;" class="fa-brands fa-twitter fa-beat"></i>
        <i class="fa-brands fa-github fa-beat"></i>
        <i class="fa-brands fa-pinterest fa-beat" style="color: #c70a0a;"></i>
        <i class="fa-brands fa-square-whatsapp fa-beat" style="color: #4db71f;"></i>
    <!-- -->
    <div class="author">
        <i class="fa-solid fa-copyright"></i>
        Powered by <span>
            <a style="color: blue;" target="_blank" href="#">
                Tổ 05 Nhóm 02 TTNT
            </a>
        </span>
    </div>
</div>
```



## CSS Footer

```
/* xây dựng footer */
.footer{
  text-align: center;
  line-height: 40px;
  padding-top: 40px;
  padding-bottom: 40px;
  background-image: linear-gradient(-45deg, ■ #ADA996, ■ #F2F2F2, ■ #DBDBDB, ■ #EAEAEA);
}

/* chỉnh name-copyright */
.footer .name-copyright{
  line-height: 64px;
  font-weight: bold;
  font-size: 28px;
  font-family: monospace;
}

/* chỉnh icon */
.footer .icon-footer{
  font-size: 24px;
  letter-spacing: 4px;
}

/* chỉnh tên bản quyền tác giả */
.footer .author{
  font-size: 20px;
  letter-spacing: 0px;
}

.author a:hover{
  text-decoration: underline;
  opacity: 0.7;
  color: ■ red;
}
```

## HTML - Chatbox

```
<!-- CHAT BAR BLOCK -->
<form>
  <!-- Khung chat -->
  <div class="chat-bar-collapsible">

    <!-- Header của khung chat -->
    <button id="chat-button" type="button" class="collapsible">Tư vấn
      <i id="chat-icon" style="color: ■ #fff;" class="fa fa-fw fa-comments-o"></i>
    </button>

    <div class="content">
      <div class="full-chat-block">
        <!-- Message Container -->
        <div class="outer-container">
          <div class="chat-container">
            <!-- Messages -->
            <div id="chatbox">
              <h5 id="chat-timestamp"></h5>
              <p id="botStarterMessage" class="botText"></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</form>
```

```
<!-- User inputbox gồm 2 thẻ con là userInput và chat-bar-icon -->
<div class="chat-bar-input-block">
  <div id="userInput">
    <input id="text" class="input-box" type="text" name="msg"
      placeholder="Tap 'Enter' to send a message">
    <p></p>
  </div>

  <div class="chat-bar-icons">
    <i id="chat-icon" onclick="heartButton()"></i>
    <button type="submit" id="send">
      <i style="font-size:24px; color:■ #84495F;" class="fa-sharp fa-solid fa-paper-plane"></i>
    </button>
  </div>
</div>
</div>
</div>
```

## CSS – Xây dựng khung hiển thị, cửa sổ nhập input và icon gửi tin nhắn trong boxchat

```
/* Xây dựng ô chatbox */
.chat-bar-collapsible {
  position: fixed;
  bottom: 0;
  right: 25px;
  box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2);
}

/* Chính header của boxchat*/
.collapsible {
  background-color: #84495F;
  color: #fff;
  cursor: pointer;
  padding: 18px;
  width: 350px;
  text-align: left;
  outline: none;
  font-size: 18px;
  border-radius: 16px 16px 0px 0px;
  border: 3px solid white;
  border-bottom: none;
  font-weight: 600;
}
```

```
/* Xây dựng phần nội dung của boxchat */
.content {
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.2s ease-out;
  background-color: #f1f1f1;
}

.full-chat-block {
  width: 350px;
  background: white;
  text-align: center;
  overflow: auto;
  height: max-content;
  transition: max-height 0.2s ease-out;
}

.outer-container {
  min-height: 500px;
  position: relative;
}
```

```
/* Nội dung trong phần khung chat */
.chat-container {
  position: absolute;
  max-height: 500px;
  width: 100%;
  bottom: 0;
  left: 0;
  scroll-behavior: smooth;
}

/* Ẩn thanh scrollbar khi chưa có nhiều nội dung */
.chat-container::-webkit-scrollbar {
  display: none;
}

/* xây dựng bố cục phần nhập câu hỏi */
.chat-bar-input-block {
  display: flex;
  float: left;
  box-sizing: border-box;
  width: 100%;
  align-items: center;
  background-color: #d9d9d9;
  padding: 10px 0px 10px 10px;
  margin-top: 16px;
}
```

```
.chat-bar-icons {
  display: flex;
  justify-content: space-evenly;
  box-sizing: border-box;
  width: 25%;
}

#userInput {
  width: 75%;
}

/* căn chỉnh box nhập input */
.input-box {
  float: left;
  border: 1px solid #808080;
  width: 100%;
  border-radius: 16px;
  padding: 10px;
  font-size: 16px;
  color: #000;
  background-color: white;
  outline: none;
}
```

## CSS - Căn chỉnh tin nhắn trong boxchat

```
/* căn chỉnh chữ trong bóng chat */
.userText span {
  text-align: left;
  line-height: 1.5em;
  display: inline-block;
  background: #2a8af8;
  padding: 10px;
  border-radius: 12px;
  border-bottom-right-radius: 2px;
  max-width: 80%;
  margin: 10px;
  animation: floatup 0.75s
}

.botText {
  color: #000;
  font-size: 16px;
  text-align: left;
}
```

```
.botText {
  color: #000;
  font-size: 16px;
  text-align: left;
}

.botText span {
  line-height: 1.5em;
  display: inline-block;
  background: #d4d4d4;
  padding: 10px;
  border-radius: 8px;
  border-bottom-left-radius: 2px;
  max-width: 80%;
  margin-left: 10px;
  animation: floatup 2.5s;
}
```

Sử dụng thuộc tính Keyframe: là một CSS animation, nó định nghĩa hiệu ứng sẽ thay đổi và có kiểu dáng như thế nào tại mọi thời điểm trong thời gian của hiệu ứng, ở đây @keyframe dùng để tạo hiệu ứng trượt lên của bong bóng chat từ điểm dưới vị trí hiện tại 14px lên vị trí hiện tại.

### 3.2. Sử dụng JavaScript để tạo giao diện, tạo hiệu ứng và xử lý dữ liệu

#### JavaScript - Xử lý giao diện chatbox và gửi dữ liệu lên server

```
<script>
$(document).ready(function() {
    // sử dụng để chạy mã JavaScript khi DOM của trang web được tải hoàn tất
    // ngăn mã JavaScript cố gắng truy cập các phần tử HTML trước khi chúng được tạo
    $("form").on("submit", function(event) {
        // gắn trình xử lý sự kiện vào biểu mẫu
        var rawText = $("#text").val(); //biến rawText lưu dữ liệu của thẻ có id = text
        var userHtml = '<p class="userText"><span>' + rawText + "</span></p>";
        $("#text").val(""); //xóa giá trị của trường đầu vào có id = text, ngăn người dùng gửi nhiều tin nhắn 1 lúc
        $("#chatbox").append(userHtml);
        //nối phần tử userHtml vào phần tử chatbox => thêm tin nhắn của người dùng vào danh sách tin nhắn trong boxchat
        $.ajax({
            data: {
                msg: rawText,
            },
            type: "POST",
            url: "/get",
            //gửi dữ liệu từ rawText lên server
        }).done(function(data) {
            var botHtml = '<p class="botText"><span>' + data + "</span></p>";
            $("#chatbox").append($.parseHTML(botHtml));
            document.getElementById("userInput").scrollIntoView(true);
            //cuộn trang về đầu userInput khi nhận được câu trả lời => thấy phần hồi của chat bot ngay lập tức
        });
        event.preventDefault();
    });
});
</script>
```

Một số hàm trong thư viện jQuery được dùng:

- `$(document).ready()` là một hàm được sử dụng để chạy mã JavaScript khi DOM của trang web được tải hoàn tất. DOM (Document Object Model) là một đại diện của trang web dưới dạng một cây các đối tượng ng.
- `$("#form").on("submit", function(event))`: là hàm được sử dụng để gắn trình xử lý sự kiện vào biểu mẫu, hàm sẽ chạy khi nhận được tín hiệu nộp biểu mẫu từ user.
- `$.ajax.done(function())`: là hàm để gửi một yêu cầu AJAX đến máy chủ. Khi máy chủ trả về phản hồi, hàm “done()” sẽ được thực thi.
- Các biến như `rawText`, `userHTML` hay `botHTML` dùng để lưu dữ liệu nhập vào của user hoặc câu trả lời của chatbot, các biến này sẽ được hàm `$(id).append()` đẩy vào trong thẻ HTML có id tương ứng để hiển thị lên trên cửa sổ chat.



## JavaScript - Hàm bật tắt cửa sổ box chat

```
var coll = document.getElementsByClassName("collapsible");

for (let i = 0; i < coll.length; i++) {
  coll[i].addEventListener("click", function () {
    this.classList.toggle("active");
    // thêm hoặc xóa active vào collapsible
    var content = this.nextElementSibling;

    if (content.style.maxHeight) {
      content.style.maxHeight = null;
    } else {
      content.style.maxHeight = content.scrollHeight + "px";
    }
  });
}
```

## JavaScript - Hàm xử lý thời gian trong cửa sổ boxchat

```
function getTime() {
  let today = new Date();
  hours = today.getHours();
  minutes = today.getMinutes();

  if (hours < 10) {
    hours = "0" + hours;
  }

  if (minutes < 10) {
    minutes = "0" + minutes;
  }

  let time = hours + ":" + minutes;
  return time;
}
```

- Hàm getHours() và getMinutes() để lấy thời gian thực trong máy tính.
- Nếu giờ < 10 thì thêm số 0 vào trước.
- Nếu phút < 10 thì thêm số 0 vào trước.

- Trả về giá trị xâu có dạng hh:mm.

## JavaScript - Hàm gọi ra khung thời gian và dòng tin nhắn đầu tiên của box chat

```
// Gets the first message
function firstBotMessage() {
    let firstMessage = "Chào bạn, mình là Henlen - AI tư vấn sức khỏe cho sinh viên PTIT. Hi vọng sẽ giúp đỡ được cho bạn!"
    document.getElementById("botStarterMessage").innerHTML = '<p class="botText"><span>' + firstMessage + '</span></p>';

    let time = getTime();

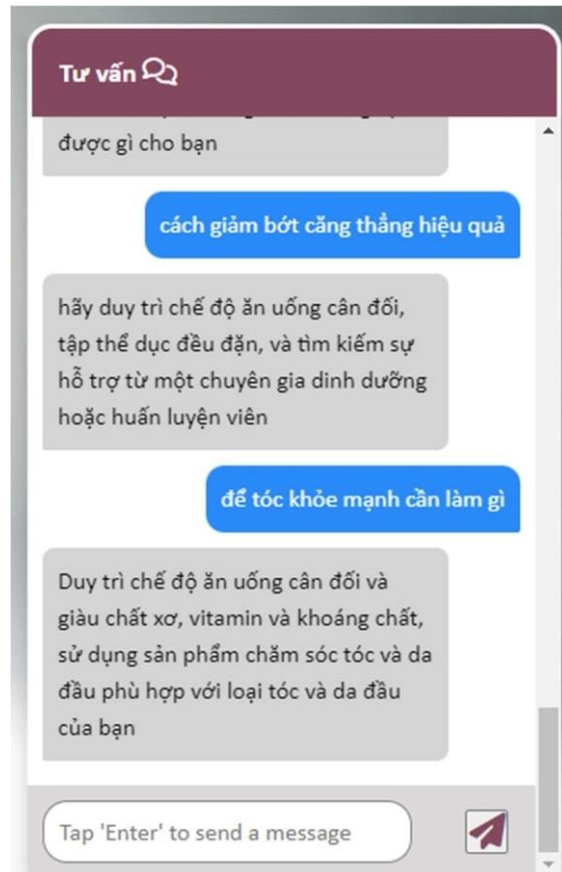
    $("#chat-timestamp").append(time);
    document.getElementById("userInput").scrollIntoView(true);
}

firstBotMessage();
```

- Gán giá trị xâu cho biến firstMessage và time.
- Đẩy giá trị biến time lên cửa sổ chat sau đó đến firstMessage.

### 3.3. Kết quả





### III. Kết luận

Nhóm đã nắm được các bước cơ bản trong việc xử lý dữ liệu, biết sử dụng một số thư viện có sẵn trong Python, huấn luyện một chatbot sử dụng mô hình học máy Naive Bayes. Chatbot có các chức năng cơ bản như: cung cấp thông tin y tế về các loại bệnh thông thường, tư vấn sức khỏe dựa trên bộ dữ liệu đã được chuẩn hóa.

Ngoài ra Chatbot vẫn còn một số nhược điểm:

- Hạn chế trong việc trả lời những câu hỏi phức tạp.
- Khả năng phân loại dữ liệu còn chưa tối ưu.
- Độ chính xác đạt được chưa cao.

### IV. Tài liệu tham khảo:

- <https://www.w3schools.com>
- <https://github.com/trungvuong55555/Chabot-AI-/tree/master>
- <https://fullstack.edu.vn>