

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

----------

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP**

*Tên đề tài: Ứng dụng học sâu xây dựng chatbot*

*chẩn đoán bệnh thông qua triệu chứng*

**Giảng viên hướng dẫn:** Trần Đình Quế

**Sinh viên thực hiện: Nguyễn Hoàng Dũng**

**Mã sinh viên: B18DCAT034**

**Lớp: E18CQCN02-B**

**Hà Nội, 2022**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

----------

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP**

*Tên đề tài: Ứng dụng học sâu xây dựng chatbot*

*chẩn đoán bệnh thông qua triệu chứng*

**Giảng viên hướng dẫn:** Trần Đình Quế

**Sinh viên thực hiện: Nguyễn Hoàng Dũng**

**Mã sinh viên: B18DCAT034**

**Lớp: E18CQCN02-B**

**Hà Nội, 2022**

# LỜI CẢM ƠN

Lời đầu tiên để có thể hoàn thành đồ án tốt nghiệp này em xin chân thành cảm ơn tới thầy Trần Đình Quế đã tận tình hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn tới thầy cô trong Học viện và khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức rất quan trọng. Đó là nền tảng để em áp dụng vào thực hiện đồ án cũng như công việc sau khi ra trường.

Lời cảm ơn cuối cùng em muốn gửi tới gia đình, bạn bè đã giúp đỡ, quan tâm em trong suốt quá trình học tập và làm đồ án tốt nghiệp.

Do kiến thức và kinh nghiệm còn hạn chế nên mặc dù đã cố gắng. đồ án của em vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của thầy cô để đồ án hoàn thiện hơn

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng 12 năm 2022* |
|  | **Sinh viên** |
|  |  |
|  | **Nguyễn Hoàng Dũng** |

**Nhận xét, đánh giá, cho điểm**

**(Của giảng viên hướng dẫn)**

………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….Điểm: …………………………………………..(bằng chữ: …………………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ đồ án trước hội đồng chấm tốt nghiệp

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 12 năm 2022 |
|  | **Giảng viên hướng dẫn** |
|  |  |
|  | **PGS.TS. Trần Đình Quế** |

**Nhận xét, đánh giá, cho điểm**

**(Của giảng viên phản biện)**

………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….Điểm: …………………………………………..(bằng chữ: …………………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ đồ án trước hội đồng chấm tốt nghiệp

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 12 năm 2022 |
|  | **Giảng viên phản biện** |

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc117893764)

[MỤC LỤC iv](#_Toc117893765)

[DANH MỤC TỪ VIẾT TẮT vi](#_Toc117893766)

[DANH MỤC HÌNH ẢNH vii](#_Toc117893767)

[MỞ ĐẦU viii](#_Toc117893768)

[CHƯƠNG 1. TỔNG QUAN VỀ CHATBOT 1](#_Toc117893769)

[CHƯƠNG 2. CÁC KỸ THUẬT SỬ DỤNG TRONG CHATBOT 2](#_Toc117893770)

[2.1. Xử lý ngôn ngữ tự nhiên (Natural Language Processing) 2](#_Toc117893771)

[2.1.1. Khái niệm 2](#_Toc117893772)

[2.1.2. Các bước xử lý ngôn ngữ tự nhiên 2](#_Toc117893773)

[Phân tích hình vị: 2](#_Toc117893774)

[Phân tích cú pháp: 2](#_Toc117893775)

[Phân tích ngữ nghĩa: 2](#_Toc117893776)

[Phân tích diễn ngôn 2](#_Toc117893777)

[2.1.3. Các thuật toán NLP được sử dụng 2](#_Toc117893778)

[Word Tokenization 2](#_Toc117893779)

[Stemming và Lemmatization 3](#_Toc117893780)

[Mô hình túi từ (Bag of words) 4](#_Toc117893781)

[2.2. Mạng nơ-ron nhân tạo (Neural Network) 5](#_Toc117893782)

[2.2.1. Kiến trúc mạng nơ-ron 5](#_Toc117893783)

[2.2.2. Các hàm kích hoạt 5](#_Toc117893784)

[Hàm sigmoid 6](#_Toc117893785)

[Hàm tanh 6](#_Toc117893786)

[Hàm reLU 6](#_Toc117893787)

[2.3. K láng giềng gần nhất (K nearest neighbor) 5](#_Toc117893788)

[2.4. Hồi quy logistic (Logistic Regression) 5](#_Toc117893789)

[2.5. Support Vector Machine (SVM) 5](#_Toc117893790)

[2.6. Essemble learning 5](#_Toc117893791)

[CHƯƠNG 3. THỬ NGHIỆM – ĐÁNH GIÁ – DỰNG HỆ THỐNG 6](#_Toc117893792)

[3.1. Tập dữ liệu 6](#_Toc117893793)

[3.2. Xây dựng và huấn luyện mô hình 10](#_Toc117893794)

[3.2.1. Tiền xử lý dữ liệu 10](#_Toc117893795)

[3.2.2. Huấn luyện mô hình nhận diện triệu chứng 15](#_Toc117893796)

[3.2.3. Huấn luyện mô hình chẩn đoán bệnh 16](#_Toc117893797)

[3.2.4. Dựng giao diện chatbot và chạy thử 20](#_Toc117893798)

[CHƯƠNG 4. KẾT LUẬN 24](#_Toc117893799)

[4.1. Kết luận 24](#_Toc117893800)

[4.2. Hướng phát triển 24](#_Toc117893801)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** | **Giải nghĩa** |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| lr | Learning rate | Tốc độ học |
| ASGD | Gradient Descent |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# DANH MỤC HÌNH ẢNH

[Hình 2.1. Phương pháp Tokenization 3](#_Toc117893802)

[Hình 2.2 Kỹ thuật Stemming 3](#_Toc117893803)

[Hình 2.3. Kỹ thuật Lemmatization 4](#_Toc117893804)

[Hình 2.4. Kiến trúc một mạng nơ-ron 5](#_Toc117893805)

[Hình 2.5 Đồ thị hàm sigmoid 6](#_Toc117893806)

[Hình 2.6. Đồ thị hàm tanh 6](#_Toc117893807)

[Hình 2.7. Đồ thị hàm reLU 6](#_Toc117893808)

[Hình 3.1. Minh họa dữ liệu trong file dataset.csv 6](#_Toc117893809)

[Hình 3.2.Minh họa dữ liệu trong file symptom\_description.csv 7](#_Toc117893810)

[Hình 3.3. Minh họa dữ liệu trong file symptom\_precaution.csv 8](#_Toc117893811)

[Hình 3.4. Minh họa dữ liệu trong file symptom-severity.csv 9](#_Toc117893812)

[Hình 3.5. Dữ liệu đã gán nhãn trong file intents.json 10](#_Toc117893813)

[Hình 3.6. Load dữ liệu từ file intents.json 11](#_Toc117893814)

[Hình 3.7. Áp dụng phương pháp Tokenization để tách từ 12](#_Toc117893815)

[Hình 3.8. Áp dụng Stemming 13](#_Toc117893816)

[Hình 3.9. Dữ liệu để chạy thử nghiệm 14](#_Toc117893817)

[Hình 3.10. Áp dụng phương pháp túi từ 14](#_Toc117893818)

[Hình 3.11. Huấn luyện mô hình nhận diện triệu chứng 15](#_Toc117893819)

[Hình 3.12. Thử nghiệm mô hình 16](#_Toc117893820)

[Hình 3.13. Trích xuất triệu chứng bệnh và tạo bộ test cho model 17](#_Toc117893821)

[Hình 3.14. Hàm kiểm chứng chéo (cross-validation) 18](#_Toc117893822)

[Hình 3.15.Xây dựng model sử dụng kNN 19](#_Toc117893823)

[Hình 3.16. Xây dựng model sử dụng Decision Tree 19](#_Toc117893824)

[Hình 3.17. Essemble Learning 20](#_Toc117893825)

[Hình 3.18. Lưu model chẩn đoán bệnh 20](#_Toc117893826)

[Hình 3.19. Load các model để dựng hệ thống chatbot 21](#_Toc117893827)

[Hình 3.20. Giao diện chatbot 23](#_Toc117893828)

# MỞ ĐẦU

# TỔNG QUAN VỀ CHATBOT

# CÁC KỸ THUẬT SỬ DỤNG TRONG CHATBOT

## Xử lý ngôn ngữ tự nhiên (Natural Language Processing)

### Khái niệm

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

### Các bước xử lý ngôn ngữ tự nhiên

#### Phân tích hình vị:

Là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại, v.v.

Hai bài toán điển hình trong phần này là tách từ (word segmentation) và gán nhãn từ loại (part-of-speech tagging).

#### Phân tích cú pháp:

Là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức

Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.

#### Phân tích ngữ nghĩa:

Là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ.

Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.

#### Phân tích diễn ngôn

Là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use). Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.

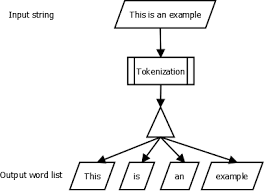
### Các thuật toán NLP được sử dụng

#### Word Tokenization

- Tách từ (Tokenization) là một trong những bước quan trọng nhất trong quá trình tiền xử lý văn bản.

- Tokenization là quá trình tách một câu, cụm từ, đoạn văn thành các đơn vị nhỏ hơn. Đầu vào của thuật toán là một câu hoàn chỉnh. Đầu ra sẽ là một danh sách các từ riêng lẻ đã được tách ra từ câu đầu vào. Mỗi từ riêng lẻ được gọi là các tokens

VD: Cho một câu “This is an example” là câu đầu vào. Sau khi Tokenization đầu ra sẽ là một danh sách các từ riêng lẻ chứa trong câu đầu vào trên, đó là [“this”, “is”, “an”, “example”]



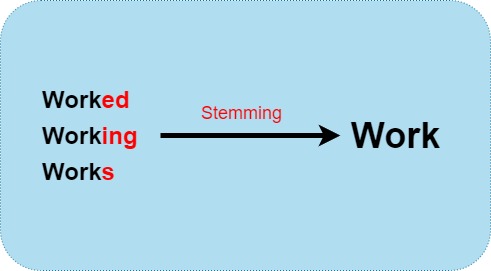
Hình 2.1. Phương pháp Tokenization

Tokenization dễ sử dụng với các câu tiếng Anh: ta sẽ tách các từ bất kỳ khi nào có khoảng cách giữa chúng.

#### Stemming và Lemmatization

**- Stemming** là kỹ thuật dùng để biến đổi một từ về dạng gốc (được gọi là stem hoặc root form) bằng cách rất đơn giản – loại bỏ một số ký tự hậu tố có khả năng là biến thể của từ

VD: Các động từ **worked**, **working**, **works** chỉ khác nhau ở ký tự nằm ở cuối cùng của từ. Bằng việc bỏ đi các hậu tố **-ed**, **-ing**, **-s**, kết quá đều được từ nguyên thể là **work.**



Hình 2.2 Kỹ thuật Stemming

Ưu điểm: Tốc độ xử lý nhanh

Nhược điểm: Cho kết quả không như mong muốn (VD: các từ ở dạng bất quy tắc như went, spoke thì kỹ thuật Stemming không thể đưa về dạng gốc)

- **Lemmatization** là kỹ thuật xử lý thông minh hơn bằng một bộ từ điển hoặc một bộ ontology nào đó. Điều này sẽ đảm bảo rằng các từ goes, went và go sẽ trả về kết quả giống nhau là go.



Hình 2.3. Kỹ thuật Lemmatization

Ưu điểm: kết quả trả về có độ chính xác cao

Nhược điểm: Tốc độ chậm

<https://viblo.asia/p/xu-ly-ngon-ngu-tu-nhien-phan-1-OeVKB8eQlkW>

<https://vinbigdata.com/top_thuat_toan_nlp/>

#### Mô hình túi từ (Bag of words)

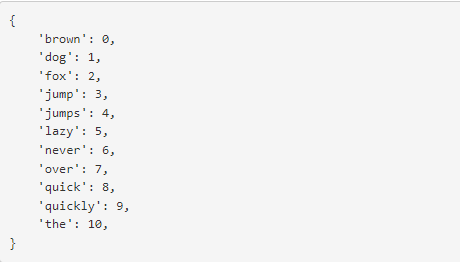
- Mô hình túi từ (bag of words) là một biểu diễn đơn giản hóa được sử dụng trong NLP và truy vấn thông tin (IR). Trong mô hình này, một văn bản được thể hiện dưới dạng túi chứa các từ của nó, không quan tâm đến ngữ pháp và trật tự từ nhưng vẫn giữ tính đa dạng

- Phương pháp này yêu cầu một tập hợp các từ cho trước, gọi là túi đựng từ. [Đặc trưng](https://trituenhantao.io/tu-dien-thuat-ngu/dac-trung/" \t "_blank) của một văn bản sẽ được thể hiện thông qua sự xuất hiện của các từ thuộc văn bản đó trong túi đựng từ.

VD: Cho 2 câu sau:

The quick brown fox jumps over the lazy dog and  
Never jump over the lazy dog quickly

Từ 2 câu trên tạo từ điển chứa các từ xuất hiện trong từng câu



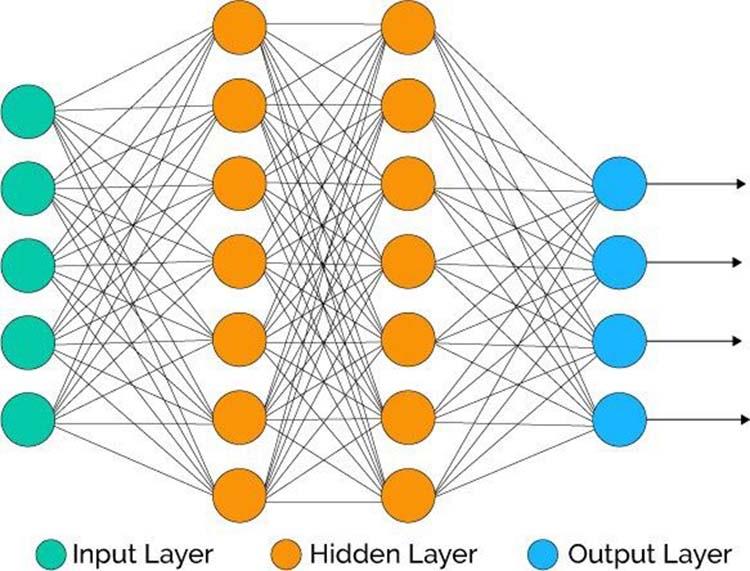
Dựa vào từ điển vừa tạo, tiến hành tạo vector lưu trữ số lần xuất hiện của từ trong từ điển ứng với mỗi câu

Kết quả ta được:



## Mạng nơ-ron nhân tạo (Neural Network)

### Kiến trúc mạng nơ-ron



Hình 2.4. Kiến trúc một mạng nơ-ron

Một mạng Nơ-ron gồm 3 lớp chính:

* Lớp đầu vào (Input layer): lớp bên trái ngoài cùng, thể hiện cho các đầu vào của mạng
* Lớp đầu ra (Output layer): lớp bên phải ngoài cùng, thể hiện cho các đầu ra của mạng
* Lớp ẩn (Hidden layer): là lớp nằm giữa hai lớp đầu vào và lớp đầu ra, tại đây thể hiện quá trình suy luận logic của mạng

Các node ở các lớp hidden layer và output layer có đặc điểm:

* Đều liên kết với các node ở layer trước đó với các hệ số w riêng
* Mỗi node đều có một hệ số bias riêng (b)
* Diễn ra 2 bước: Tính tổng linear và áp dụng hàm kích hoạt (Activation function)

### Các hàm kích hoạt

- Hàm kích hoạt là các hàm định nghĩa đầu ra của node đó cho bởi đầu vào hay một tập đầu vào. Các hàm này là hàm phi tuyến tính được áp dụng trên mỗi lớp ẩn (hidden layer) của mạng nơ-ron

- Sự kết hợp các hàm kích hoạt giữa lớp ẩn giúp mô hình học được các quan hệ phi tuyến tính tiềm ẩn trong dữ liệu. Tùy vào từng bài toán và dữ liệu sẽ sử dụng hàm kích hoạt khác nhau

#### Hàm sigmoid

Sigmoid nhận đầu vào và thực hiện những việc sau:

* Đối với hầu hết các đầu vào âm, sigmoid sẽ biến đổi đầu vào thành một số rất gần với 0.
* Đối với hầu hết các đầu vào dương, sigmoid sẽ chuyển đầu vào thành một số rất gần với 1.
* Đối với các đầu vào tương đối gần 0, sigmoid sẽ chuyển đầu vào thành một số nào đó từ 0 đến 1.

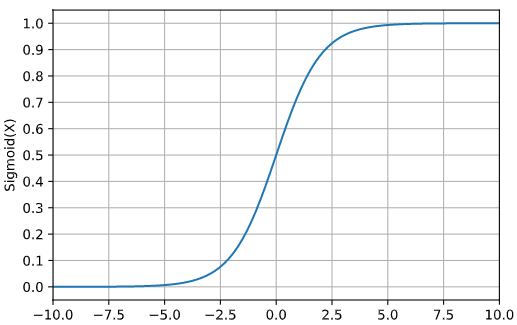
Công thức:



Trong đó:

là đầu ra từ 0 đến 1 (ước tính xác suất)

Đồ thị:



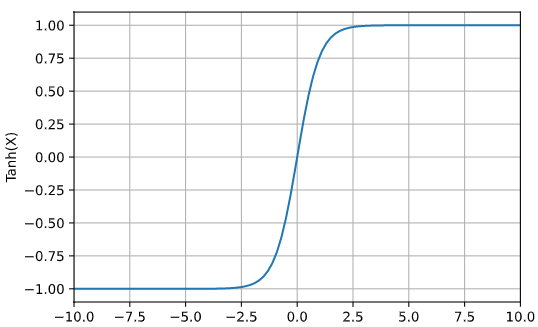
Hình 2.5 Đồ thị hàm sigmoid

#### Hàm tanh

Công thức:



Đồ thị:



Hình 2.6. Đồ thị hàm tanh

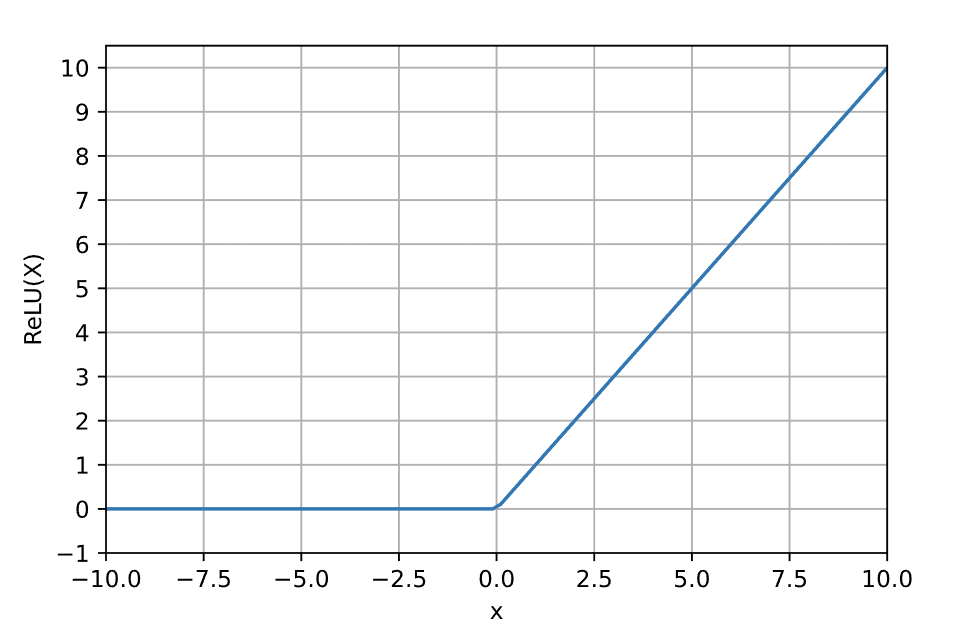
Đầu vào của hàm tanh là một số thực và đầu ra là giá trị nằm trong khoảng (0,1). Hàm này cũng có nhược điểm giống với hàm sigmoid ở chỗ nếu đầu vào có giá trị tuyệt đối lớn thì gradient sẽ rất nhỏ tiến dần tới giá trị bão hòa.

#### Hàm reLU

Công thức:



Đồ thị:



Hình 2.7. Đồ thị hàm reLU

Gradient của hàm reLU được tính cực nhanh với gradient bằng 1 nếu đầu vào lớn hơn 0, bằng 0 nếu đầu vào nhỏ hơn 0. Đặc biệt hàm này không có đạo hàm tại điểm s = 0. Nhưng người ta vẫn thường định nghĩa reLU’(s) = 0 và khẳng định thêm rằng xác suất để input của một unit bằng 0 là rất nhỏ.

Tham khảo tại:

<https://aicurious.io/posts/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks/>

https://tek4.vn/ham-kich-hoat-trong-mang-neural

## K láng giềng gần nhất (K-nearest neighbor)

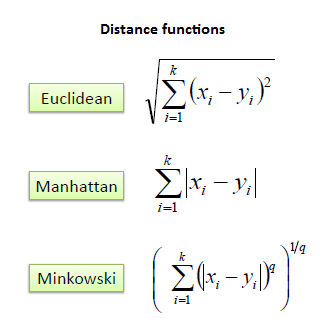
Khái niệm

- K láng giềng gần nhất (k-nearest neighbor hay KNN) là thuật toán học máy có giám sát, đơn giản và dễ triển khai. KNN thường được dùng trong các bài toán phân loại và hồi quy.

Ý tưởng:

Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó việc của ta sẽ là tìm ra k điểm gần với dữ liệu cần kiểm tra nhất.

Đây là 3 cách cơ bản để tính khoảng cách giữa hai điểm dữ liệu x,y có k thuộc tính:



## Hồi quy logistic (Logistic Regression)

## Support Vector Machine (SVM)

**SVM** là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

## Cây quyết định (Decision Tree)

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như nhị phân, định danh, thứ tự, số lượng trong khi đó thuộc tính phân cấp phải có kiểu nhị phân hoặc thứ tự.

## Essemble learning

# THỬ NGHIỆM – ĐÁNH GIÁ – DỰNG HỆ THỐNG

## Tập dữ liệu

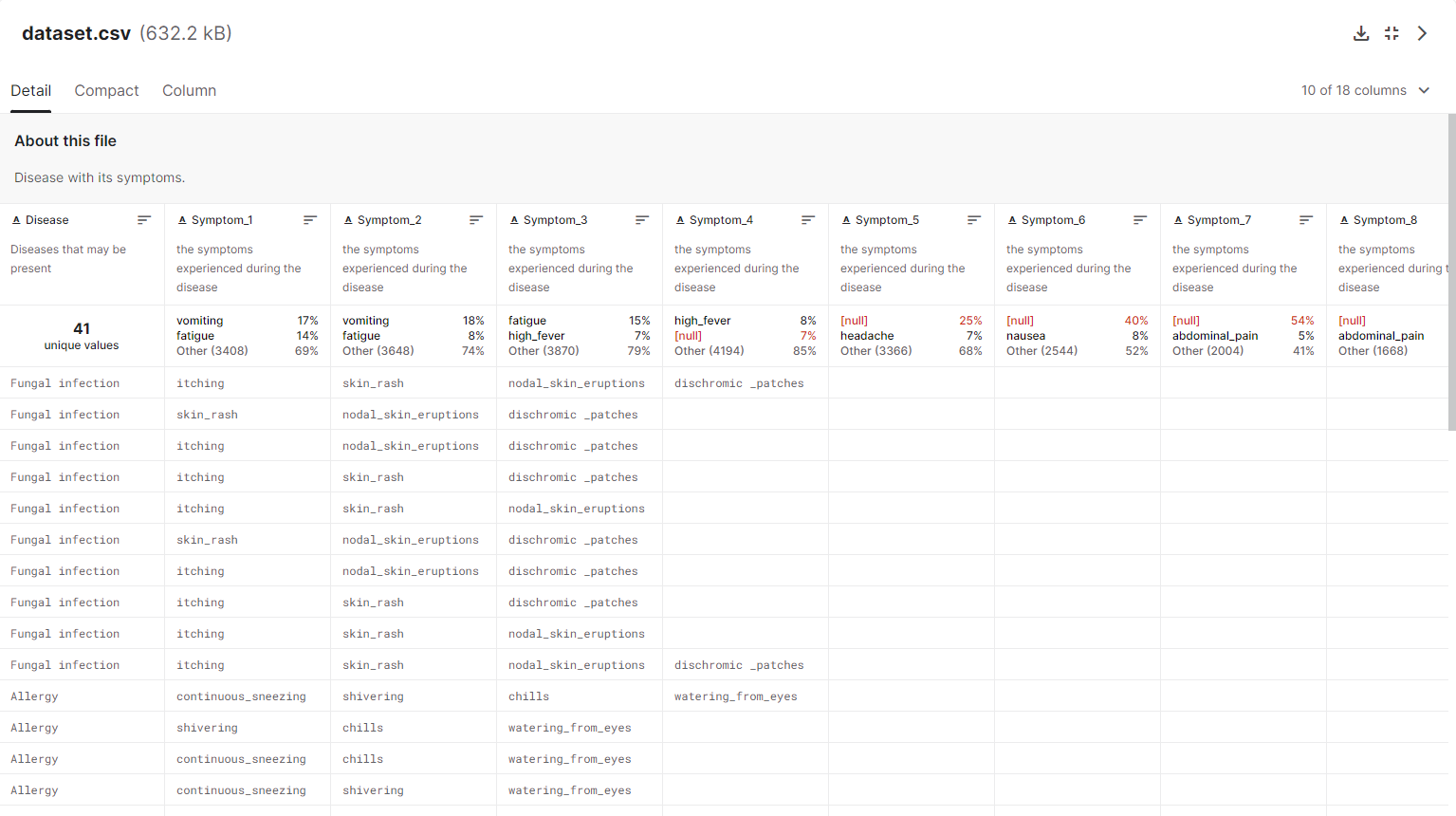
- Tập dữ liệu sử dụng cho hệ thống lấy từ *Kaggle:*

<https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset>

- Tập dữ liệu này gồm 4 file:

* Symptom-severity.csv
* Dataset.csv
* Symptom\_Description.csv
* Symptom\_precaution.csv

- File dataset.csv dùng để huấn luyện model. Cột một gồm tên của bệnh, các cột còn lại là các triệu chứng của bệnh đó



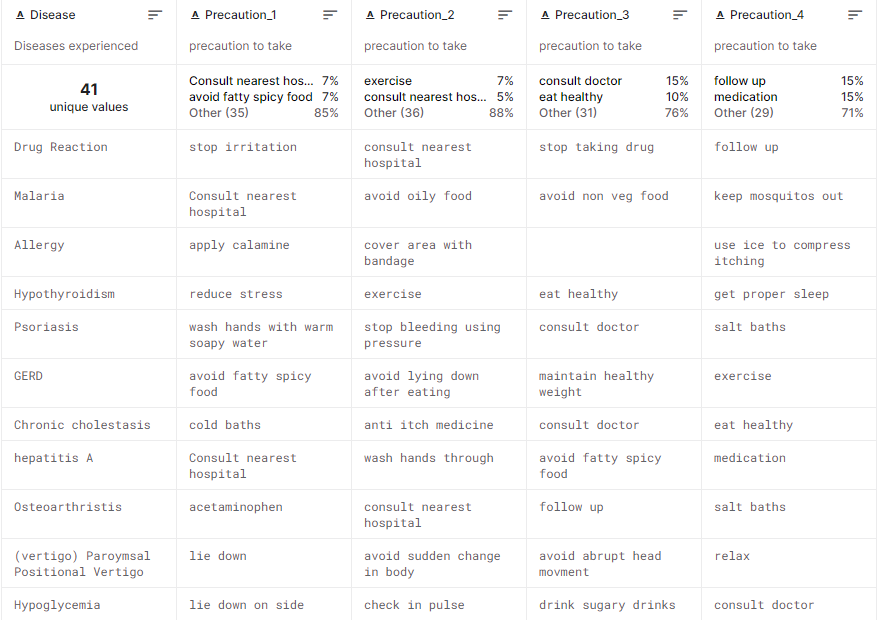
Hình 3.1. Minh họa dữ liệu trong file dataset.csv

- File symptom\_Description.csv gồm tên bệnh và mô tả của bệnh đó



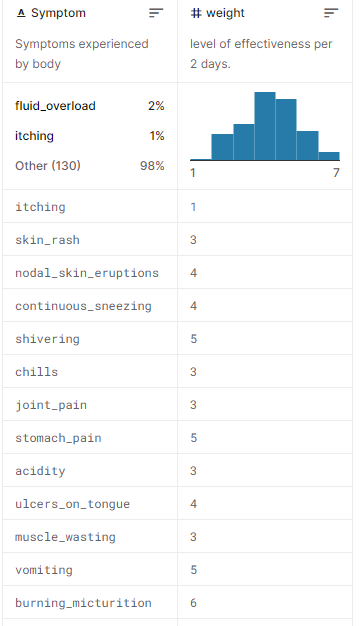
Hình 3.2.Minh họa dữ liệu trong file symptom\_description.csv

- File symptom\_precaution.csv bao gồm: cột một là tên bệnh, các cột còn lại là các cách đề phòng bệnh



Hình 3.3. Minh họa dữ liệu trong file symptom\_precaution.csv

- File symptoms-severity.csv gồm 2 cột: cột đầu là tên các triệu chứng, cột còn lại là trọng số thể hiện mức độ nghiêm trọng của triệu chứng (từ 1 đến 7)



Hình 3.4. Minh họa dữ liệu trong file symptom-severity.csv

- Ngoài ra còn có dữ liệu để nhận diện triệu chứng bệnh là file intents.json có cấu trúc như sau:



Hình 3.5. Dữ liệu đã gán nhãn trong file intents.json

- Ta sẽ có 2 sử dụng hai mô hình

+ Nhận diện triệu chứng: đầu vào là một câu mô tả triệu chứng do người dùng nhập, đầu ra sẽ là tên triệu chứng nhận diện từ câu mô tả của người dùng.

+ Chẩn đoán bệnh: đầu vào là các triệu chứng đã nhận diện từ mô hình trước đó, đầu ra là tên bệnh dự đoán từ các triệu chứng, đồng thời đưa ra mô tả bệnh, cách đề phòng bệnh.

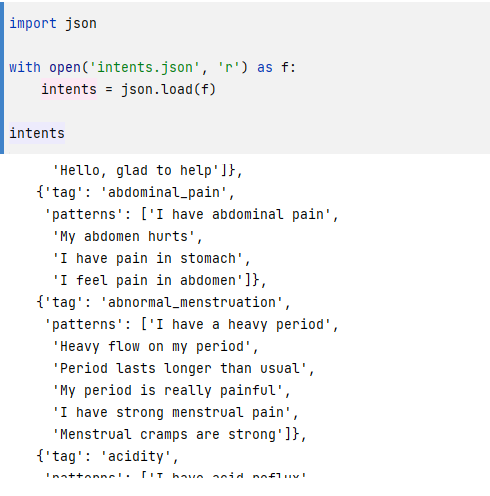
## Xây dựng và huấn luyện mô hình

### Tiền xử lý dữ liệu

- Ta có dữ liệu về triệu chứng bệnh:

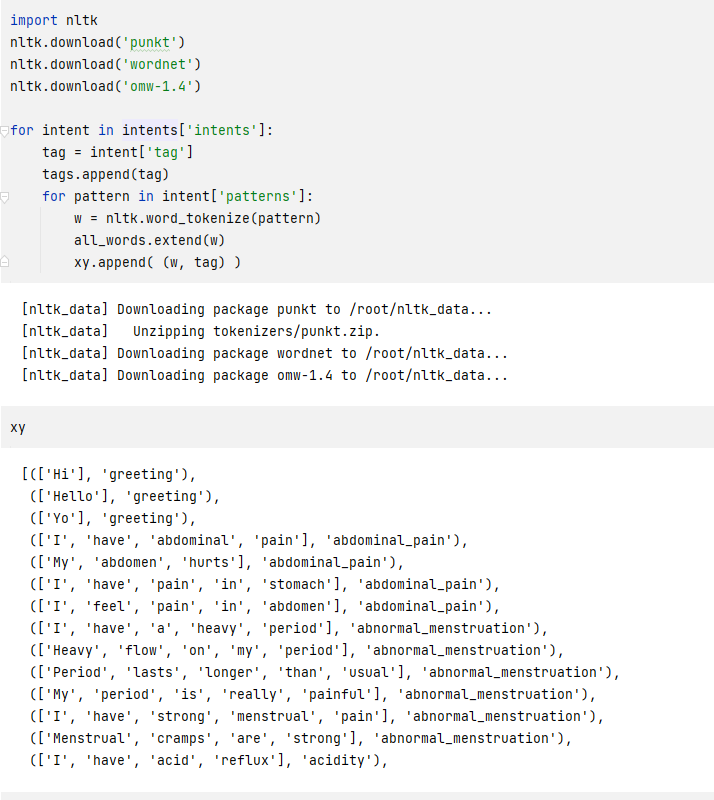
* Tag là tên triệu chứng (nhãn)
* Pattern là các câu mô tả triệu chứng sử dụng để huấn luyện

- Load dữ liệu từ file intents.json để tiến hành tiền xử lý



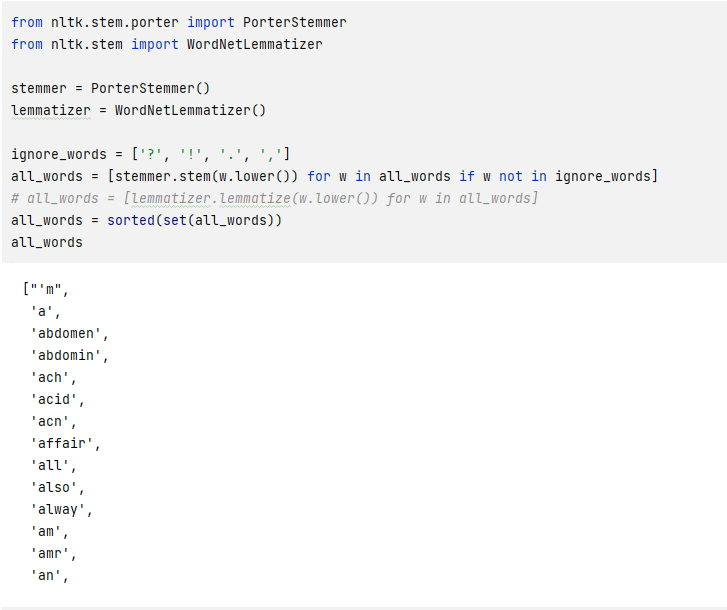
Hình 3.6. Load dữ liệu từ file intents.json

- Sử dụng phương pháp Tokenization để tách câu mẫu thành các từ riêng lẻ



Hình 3.7. Áp dụng phương pháp Tokenization để tách từ

- Sử dụng phương pháp Stemming để chuyển các từ về dạng nguyên thể



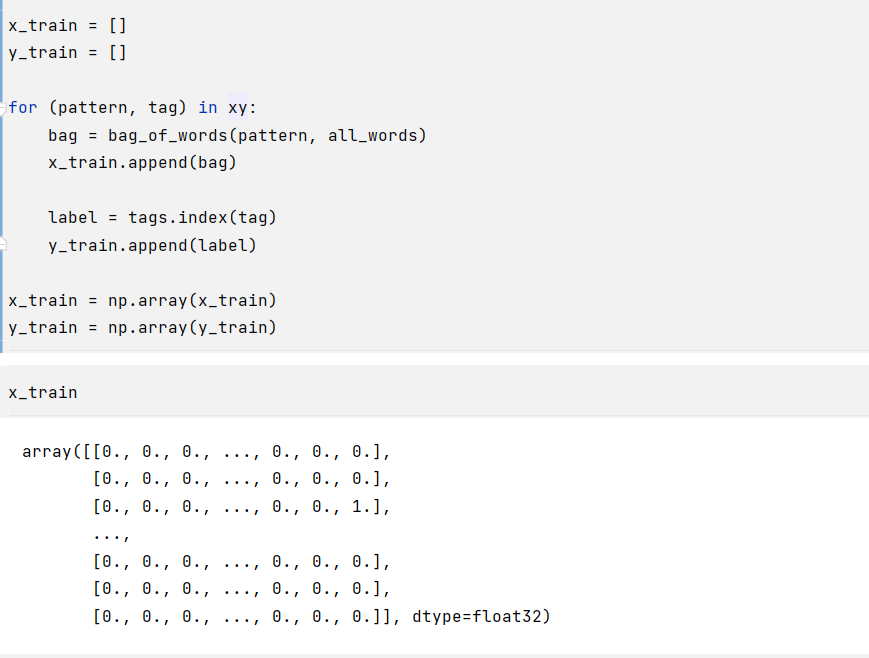
Hình 3.8. Áp dụng Stemming

- Chuẩn bị bộ dữ liệu test để huấn luyện mô hình



Hình 3.9. Dữ liệu để chạy thử nghiệm

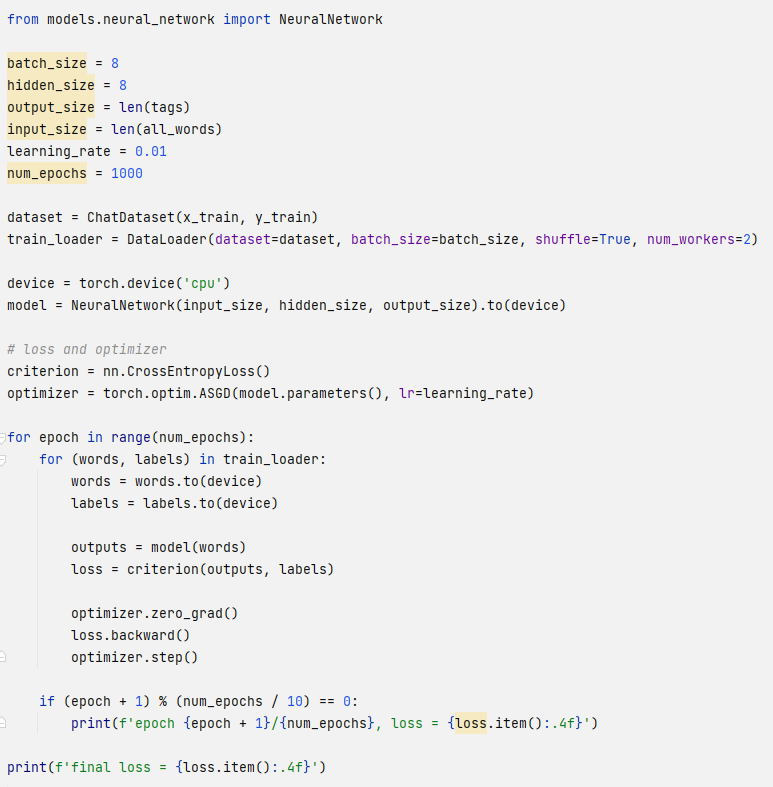
- Áp dụng phương pháp túi từ



Hình 3.10. Áp dụng phương pháp túi từ

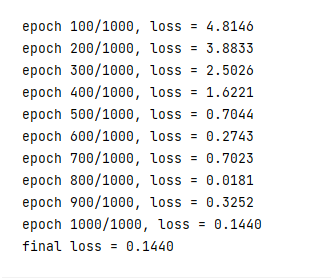
### Huấn luyện mô hình nhận diện triệu chứng

- Huấn luyện mô hình nhận diện dựa trên dữ liệu đã tiền xử lý

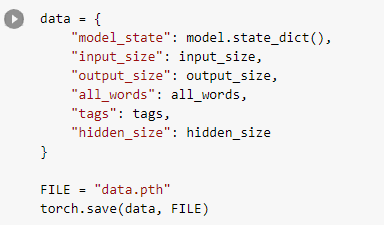


Hình 3.11. Huấn luyện mô hình nhận diện triệu chứng

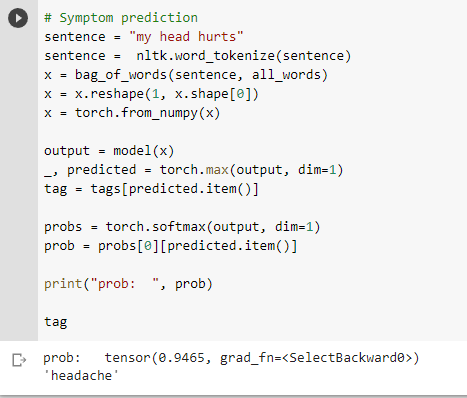
Kết quả thu được:



- Lưu lại các thông số cần thiết



- Thử nghiệm mô hình nhận diện triệu chứng



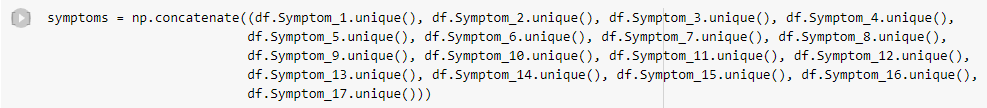
Hình 3.12. Thử nghiệm mô hình

Đánh giá: Kết quả nhận diện triệu chứng của mô hình đúng với kết quả dự kiến. Tuy nhiên khi nhập một câu phức tạp thì mô hình nhận diện chưa chính xác. Cần thêm nhiều câu mô tả cho patterns để huấn luyện ra mô hình nhận diện tốt

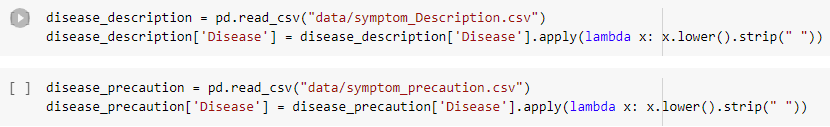
### Huấn luyện mô hình chẩn đoán bệnh

- Tiền xử lý dữ liệu bệnh





- Load dữ liệu mô tả bệnh, phòng chống bệnh

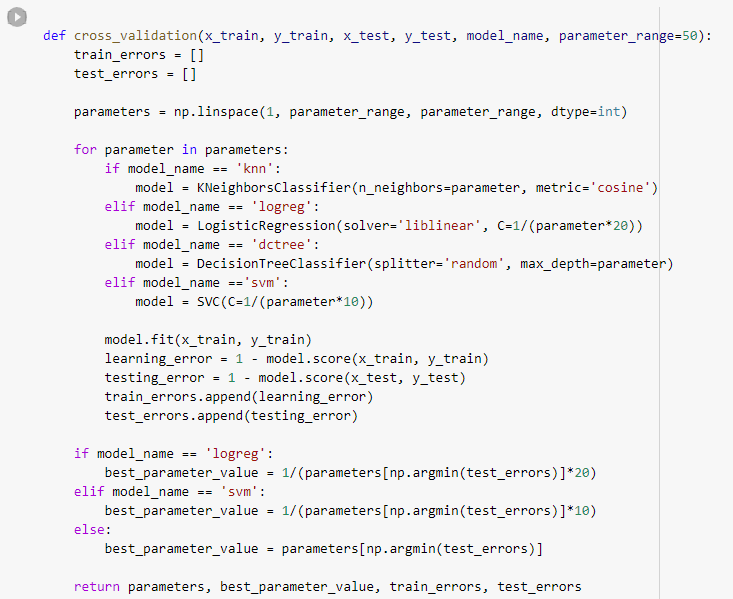


- Trích xuất danh sách triệu chứng bệnh sử dụng pickle và tạo bộ test để thử nghiệm model



Hình 3.13. Trích xuất triệu chứng bệnh và tạo bộ test cho model

- Xây dựng hàm kiểm chứng chéo cho model



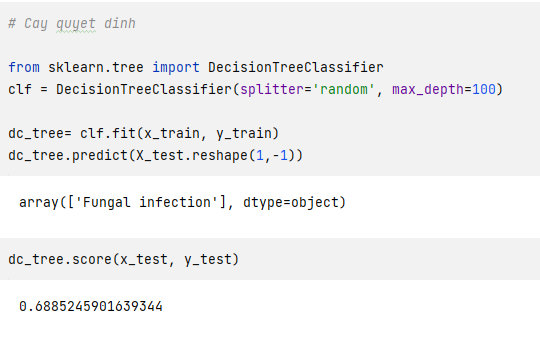
Hình 3.14. Hàm kiểm chứng chéo (cross-validation)

- Sử dụng K láng giềng gần nhất (kNN) để xây dựng model dự đoán bệnh



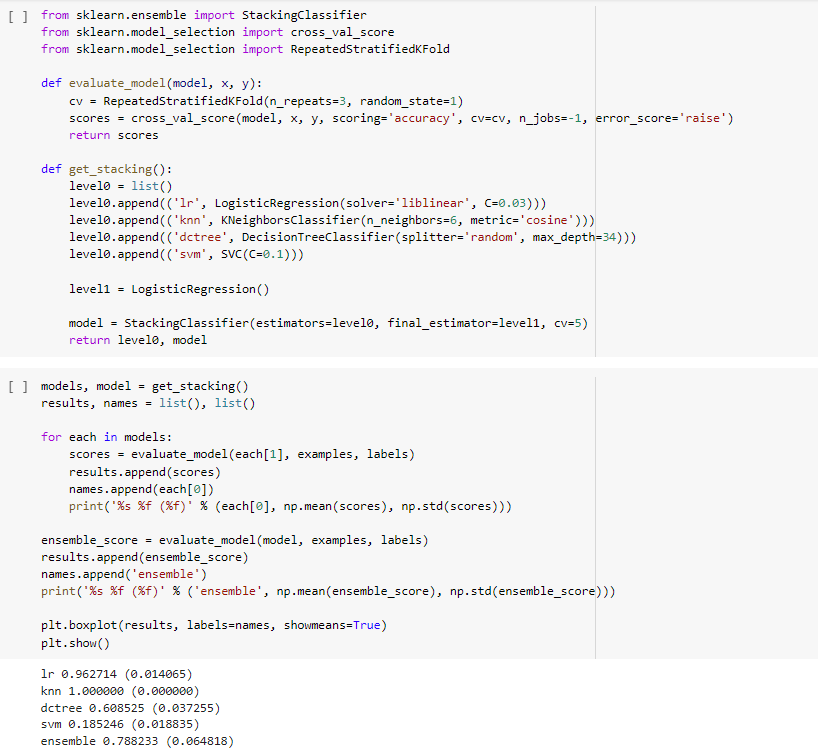
Hình 3.15.Xây dựng model sử dụng kNN

- Xây dựng model sử dụng Cây quyết định (Decision Tree)



Hình 3.16. Xây dựng model sử dụng Decision Tree

- Sử dụng Essemble Learning kết hợp các model đã xây dựng:



Hình 3.17. Essemble Learning

- Lưu model cuối cùng bằng pickle



Hình 3.18. Lưu model chẩn đoán bệnh

### Dựng giao diện chatbot và chạy thử

Ứng dụng được xây dựng trên thư viện Flask sẵn có của Python

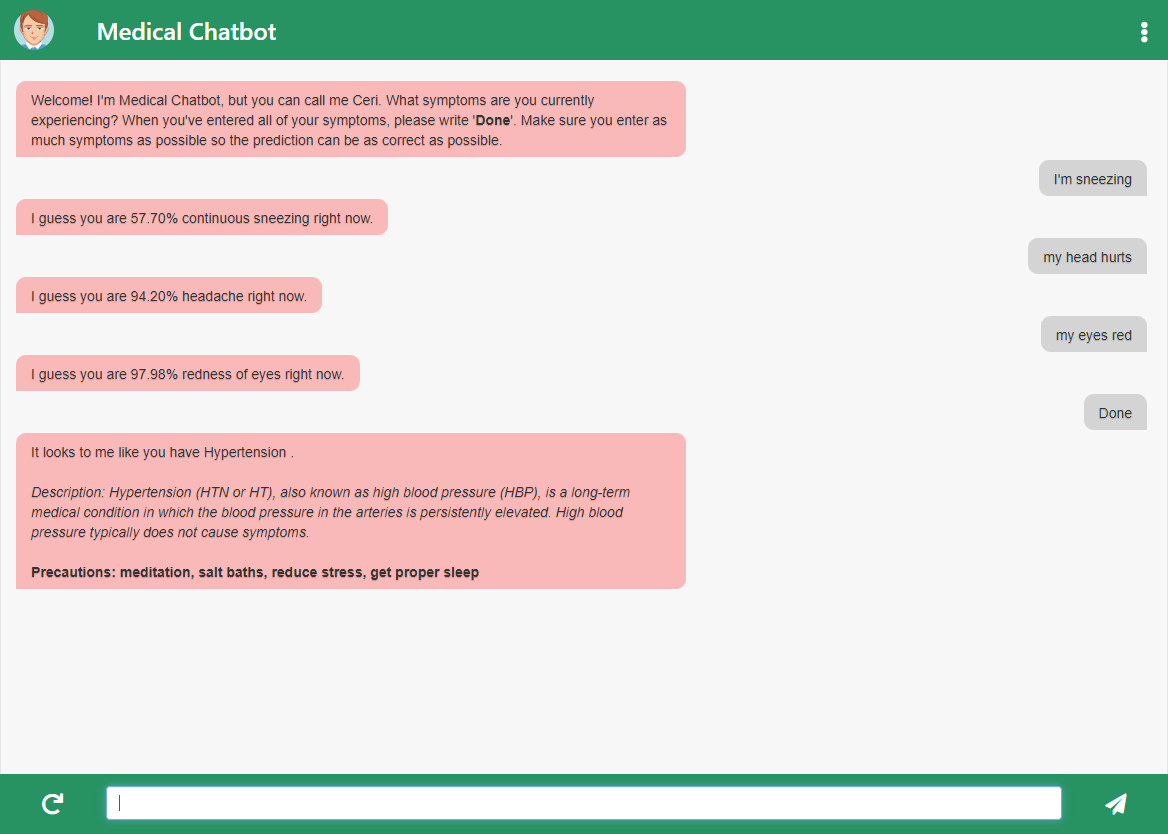


Hình 3.19. Load các model để dựng hệ thống chatbot

- Xây dựng hàm dự đoán bệnh

@app.route('/symptom', methods=['GET', 'POST'])  
def predict\_symptom():  
 print("Request json:", request.json)  
 sentence = request.json['sentence']  
 if sentence.replace(".", "").replace("!", "").lower().strip() == "done":  
  
 if not user\_symptoms:  
 response\_sentence = random.choice(  
 ["I can't know what disease you may have if you don't enter any symptoms :)",  
 "Ceri can't know the disease if there are no symptoms...",  
 "You first have to enter some symptoms!"])  
 else:  
 x\_test = []  
  
 for each in symptoms\_list:  
 if each in user\_symptoms:  
 x\_test.append(1)  
 else:  
 x\_test.append(0)  
  
 x\_test = np.asarray(x\_test)  
 disease = prediction\_model.predict(x\_test.reshape(1, -1))[0]  
 print(disease)  
  
 description = \  
 diseases\_description.loc[diseases\_description['Disease'] == disease.strip(" ").lower(), 'Description'].iloc[  
 0]  
 precaution = disease\_precaution[disease\_precaution['Disease'] == disease.strip(" ").lower()]  
 precautions = 'Precautions: ' + precaution.Precaution\_1.iloc[0] + ", " + precaution.Precaution\_2.iloc[  
 0] + ", " + precaution.Precaution\_3.iloc[0] + ", " + precaution.Precaution\_4.iloc[0]  
 response\_sentence = "It looks to me like you have " + disease + ". <br><br> <i>Description: " + description + "</i>" + "<br><br><b>" + precautions + "</b>"  
  
 severity = []  
  
 for each in user\_symptoms:  
 severity.append(symptom\_severity.loc[symptom\_severity['Symptom'] == each.lower().strip(" ").replace(" ", ""), 'weight'].iloc[0])  
  
 if np.mean(severity) > 4 or np.max(severity) > 5:  
 response\_sentence = response\_sentence + "<br><br>Considering your symptoms are severe, and Ceri " \  
 "isn't a real doctor, you should consider talking to one. :) "  
 user\_symptoms.clear()  
 severity.clear()  
  
 else:  
 symptom, prob = get\_symptom(sentence)  
 print("Symptom:", symptom, ", prob:", prob)  
 if symptom == "greeting":  
 response\_sentence = "Hello"  
 elif prob > .5:  
 responses = [f"Hmm, I'm {(prob \* 100):.2f}% sure this is " + symptom + ".",  
 f"Perhaps you are {(prob \* 100):.2f}% " +symptom + ".",  
 f"I guess you are {(prob \* 100):.2f}% " +symptom + " right now."]  
 response\_sentence = random.choice(responses)  
 user\_symptoms.add(symptom)  
 else:  
 response\_sentence = "I'm sorry, but I don't understand you."  
  
 print("User symptoms:", user\_symptoms)  
  
 return jsonify(response\_sentence.replace("\_", " "))

- Giao diện chatbot và chạy thử



Hình 3.20. Giao diện chatbot

# KẾT LUẬN

## Kết luận

## Hướng phát triển

- Về chức năng của hệ thống

- Về mô hình nhận diện triệu chứng và chuẩn đoán bệnh