**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

***----------------------***

******

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN KHAI PHÁ DỮ LIỆU VÀ MÁY HỌC**

**ĐỀ TÀI: Số 08 – DỰ ĐOÁN KHẢ NĂNG BỊ BỆNH TIM DỰA TRÊN TRIỆU CHỨNG VÀ CÁC CHỈ SỐ**

*Giảng viên hướng dẫn : PGS. TS. Nguyễn Quang Hoan*

*Nhóm sinh viên thực hiện : Vũ Quang Được - 2110A02*

*Ngọ Quốc Minh - 2110A02*

*Bùi Thành Trung - 2110A03*

*Nguyễn Sỹ Khải - 2110A01*

***Hà Nội - 2024***

Mục lục

[I. **TỔNG QUAN MÔN HỌC** 5](#_Toc165536563)

[1. Tổng quan về học máy 5](#_Toc165536564)

[2. Chu trình học máy 5](#_Toc165536565)

[3. Machine Learning 6](#_Toc165536566)

[II. **Bài tập nghiên cứu nhóm** 7](#_Toc165536567)

[1. Sử dụng thuật toán J48 đánh giá độ chính xác của bộ dữ liệu Iris 7](#_Toc165536568)

[**1.1. Các Bước Thực Hiện** 7](#_Toc165536569)

[**1.3. Cây quyết định** 12](#_Toc165536570)

[2. Dùng thuật toán K-Trung bình phân 3 cụm. 12](#_Toc165536571)

[**2.1. Giới thiệu về thuật toán K-Mean** 12](#_Toc165536572)

[**2.2. Định nghĩa K-Mean là gì?** 13](#_Toc165536573)

[**2.3. Ý tưởng của thuật toán K-Mean** 13](#_Toc165536574)

[**2.4. Bài làm** 13](#_Toc165536575)

[3. Sử dụng mạng Neural học sâu để dự báo 17](#_Toc165536576)

[**3.1. Chọn đối tượng và cơ sở dữ liệu** 17](#_Toc165536577)

[**3.2. Tiền xử lý dữ liệu.** 18](#_Toc165536578)

[**3.3. Xây dựng mô hình mạng nơ ron** 25](#_Toc165536579)

[**3.4. Đánh giá mô hình học máy** 35](#_Toc165536580)

[**3.5. Tích hợp mô hình máy học vào ứng dụng** 38](#_Toc165536581)

[**3.6. Hiệu suất** 42](#_Toc165536582)

[**3.7. Kết luận** 43](#_Toc165536583)

[III. **Tài Liệu Tham Khảo** 43](#_Toc165536584)

**Đề 08**

**Câu 1.** Cho tệp dữ liệu iris (có thể tìm trong gói thư viện mlbench hoặc<http://www.ics.uci.edu/~mlearn/MLRepository.html>). Sử dụng ngôn ngữ weka, in kết quả phân lớp dùng thuật toán J48; đánh giá hiệu quả của giải thuật: theo các tham số: Accuracy, Recall, Precision?

**Câu 2.** Cho tập dữ liệu về hoa iris (Hoa Diên Vĩ hoặc Hoa Loa kèn) với các tham số theo thứ tự: chiều dài đài hoa, chiều rộng đài hoa, chiều dài cánh hoa, chiều rộng cánh hoa, loại hoa.

5.4,3.9,1.3,0.4,Iris

5.1,3.5,1.4,0.3,Iris

5.7,3.8,1.7,0.3,Iris

4.9,2.4,3.3,1.0,Iris

6.6,2.9,4.6,1.3,Iris

5.2,2.7,3.9,1.4,Iris

5.0,2.0,3.5,1.0,Iris

5.1,3.8,1.5,0.3,Iris

5.4,3.4,1.7,0.2,Iris

5.1,3.7,1.5,0.4,Iris

4.6,3.6,1.0,0.2,Iris

5.1,3.3,1.7,0.5,Iris

7.6,3.0,6.6,2.1,Iris

4.9,2.5,4.5,1.7,Iris

6.1,2.9,4.7,1.4,Iris

7.3,2.9,6.3,1.8,Iris

6.7,2.5,5.8,1.8,Iris

7.2,3.6,6.1,2.5,Iris

Với: C1: 5.1,3.7,1.5,0.4,Iris-setosa

C2: 6.1,2.9,4.7,1.4,Iris-versicolor

C3: 6.7,2.5,5.8,1.8,Iris-virginica

**Dùng thuật toán K-Trung bình phân 3 cụm.**

**Câu 3.** Chung cho mỗi nhóm sinh viên như sau (có thể phát triển cho đồ án TN sau này)

1. Hãy nghĩ một đầu bài liên quan tới công việc: xây dựng cơ sở dữ liệu hoặc lấy dữ liệu từ các thư viện trên mạng hoặc trong Python
2. Sử dụng một trong các mạng nơ ron học sâu

để nhận dạng hoặc phân lớp đối tượng. Đối tượng tùy chọn (ví dụ phân loại hoa quả, nhận dạng biển số xe, nhận dạng mặt người…)

1. Đánh giá độ chính xác

**Bảng Phân Công**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Công việc** | **Nguyễn Sỹ Khải** | **Ngọ Quốc Minh** | **Vũ Quang Được** | **Bùi Thành Trung** |
| **Tổng quan môn Học** | Giới thiệu học máy | Chu trình học máy | Phân loại | Giới thiệu học máy |
| **Câu 1:** [**Sử dụng thuật toán J48 đánh giá độ chính xác của bộ dữ liệu Iris**](#_Sử_dụng_thuật) | Sử dụng thuật toán J48 để dự báo | Chọn cơ sở dữ liệu | Đánh giá mô hình | Sử dụng thuật toán J48 để dự báo |
| **Câu 2:** [**Dùng thuật toán K-Trung bình phân 3 cụm.**](#_Dùng_thuật_toán) | Giới thiệu về thuật toán K-Mean | Áp dụng thuật toán vào bài làm | Áp dụng thuật toán vào bài làm | Định nghĩa thuật toán K-Mean |
| **Câu 3:** [**Sử dụng mạng Neural học sâu để dự báo**](#_Sử_dụng_mạng) | Xác định vấn đề | Chọn đối tượng và cơ sở dữ liệu | Chọn đối tượng và cơ sở dữ liệu | Đặc điểm dữ liệu |
| Tiền xử lý dữ liệu(Xóa trùng lặp dữ liệu) | Tiền xử lý dữ liệu(Điền bản ghi còn thiếu)  Xây dựng mô hình mạng neural | Tiền xử lý dữ liệu(Tích hợp dữ liệu)  Xây dựng mô hình mạng neural | Tiền xử lý dữ liệu(Giảm chiều dữ liệu) |
| Chọn mô hình học máy phù hợp, Đánh giá mô hình học máy | [Sử dụng mô hình mạng nơron MLP, CNN](#_3.3._Xây_dựng) | [Sử dụng mô hình mạng nơron MLP, CNN](#_3.3._Xây_dựng) | Chọn mô hình học máy phù hợp, Đánh giá mô hình học máy |
| Kết luận mô hình học máy vào ứng dụng | Triển khai mô hình học máy vào ứng dụng | Triển khai mô hình học máy vào ứng dụng | Mô tả mô hình học máy vào ứng dụng |

1. **TỔNG QUAN MÔN HỌC**

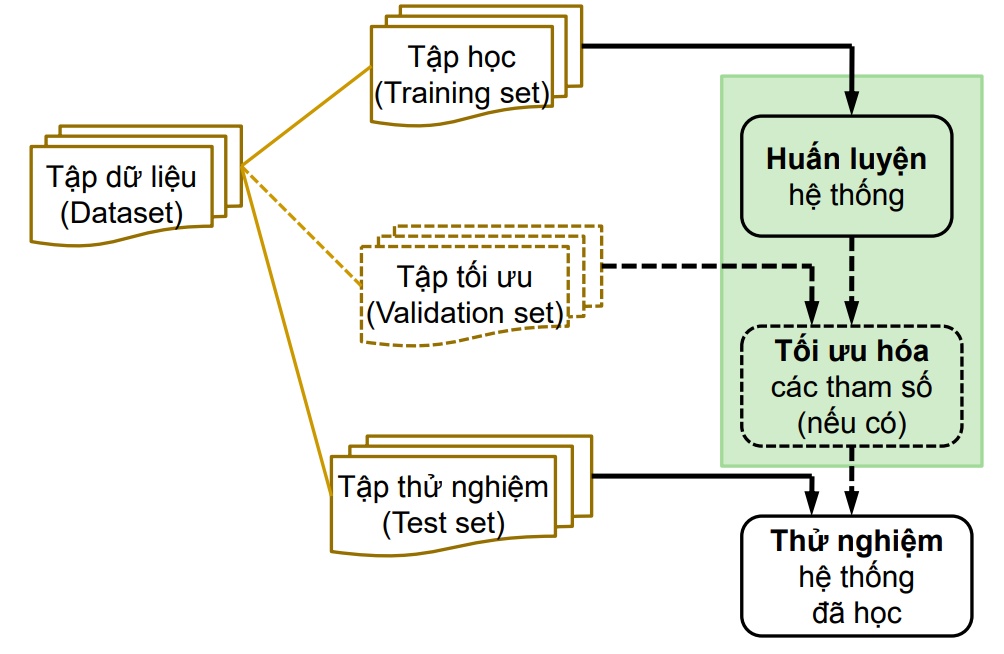
## Tổng quan về học máy

Học máy (machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể.

Một cách tổng quát, trong cuốn sách Machine Learning của tác giả Tom Mitchell xuất bản năm 1997, học máy được định nghĩa như sau: “A computer program is said to learn to perform a task T from experience E, if its performance at task T, as measured by a performance metric P, improves with experience E over time” (một chương trình máy tính được cho là học để thực hiện một nhiệm vụ T từ kinh nghiệm E, nếu hiệu suất thực hiện công việc T của nó được đo bởi chỉ số hiệu suất P và được cải thiện bởi kinh nghiệm E theo thời gian).

Như vậy, học máy nói đến một chương trình giúp cho máy tính có thể giải một bài toán cụ thể bằng những “kinh nghiệm” mà nó đã được “học” từ dữ liệu.

## Chu trình học máy



- **Giai đoạn 1:** Tiền xử lý dữ liệu

Giai đoạn này, tập trung ở việc thu thập dữ liệu, sau đó tiến hành làm sạch dữ liệu(điền các giá trị còn thiếu, xác định các ngoại lệ như dữ liệu bị trùng lặp, dữ liệu không nhất quán và làm mịn dữ liệu nhiễu), tích hợp và chuyển đổi dữ liệu(tích hợp nhiều dataset, chuẩn hóa và tổng hợp dữ liệu), giảm dữ liệu(giảm bớt về các thuộc tính của dữ liệu, giảm bớt kích thước dữ liệu – nhưng vẫn đảm bảo thu được các kết quả khai phá dữ liệu tương đương) và rời rạc hóa.

- **Giai đoạn 2:** Xử lý dữ liệu

Trong quá trình này, chúng ta cũng cần phải thực hiện phân chia dữ liệu thành các tập khác nhau bao gồm: 1 tập huấn luyện (training set), tập tối ưu (validation set) và tập kiểm thử (test set) với mục đích đo đạc, chọn ra mô hình với thông số phù hợp và kiểm tra độ chính xác của mô hình trong thực tế.

- **Giai đoạn 3:** Xây dựng mô hình và điều chỉnh tham số

Mục đích của bước này là xây dựng các mô hình cho việc huấn luyện. Các tham số của các mô hình sẽ được điều chỉnh phù hợp dựa trên kết quả học của tập huấn luyện và tập tối ưu.

- **Giai đoạn 4:** Kiểm thử trên tập thử nghiệm

Sau đó, các mô hình tốt nhất đại diện cho mỗi thuật toán sẽ được đưa vào đánh giá hiệu quả trên tập kiểm thử.

- **Giai đoạn 5:** Áp dụng mô hình học máy vào ứng dụng

Cuối cùng, sau khi chọn được mô hình, chúng ta sẽ áp dụng mô hình học máy vào ứng dụng để người sử dụng có thể nhập các thông tin.

## Machine Learning

* Học có giám sát- Supervised Learning

- Học có giám sát là trường hợp dữ liệu huấn luyện được cho một cách tường minh dưới dạng đầu vào và đầu ra của hàm đích, ví dụ, cho trước tập các mẫu cùng nhãn phân loại tương ứng.

- Hàm y = f(x) từ tập huấn luyện cho trước {x1, x2,…, xn; y1, y2,…, yn} sao cho yi ≅ f(xi) với mọi i

- Học máy khi đó được gọi là có giám sát để thể hiện việc thuật toán nhận được chỉ dẫn trực tiếp về lời giải cho từng trường hợp.

- Việc học máy có giám sát có thể được nhóm lại thành các vấn đề về phân loại và hồi quy.

+ Phân loại (Classification): Việc phân loại diễn ra khi biến đầu ra là một thể loại nào đó, chẳng hạn như “đỏ” hoặc “xanh” hoặc “bệnh” và “không có bệnh”.

+ Hồi quy (Regression): Việc hồi quy xảy ra là khi biến đầu ra là một giá trị thực, chẳng hạn như “đô la” hay “trọng lượng”.

* Học không giám sát -Unsupervised Learning

- Học không giám sát xử lý vấn đề khi chúng ta không biết kết quả đầu ra mong muốn và không có nhãn dữ liệu. Mục tiêu của học không giám sát là khám phá cấu trúc ẩn trong dữ liệu và tìm ra một cách biểu diễn dữ liệu phù hợp.

- Thuật toán dựa trên những tính chất gồm: Mối quan hệ tương tự, xác suất đồng xuất hiện của các đối tượng và các phép biến đổi ma trận để trích xuất đặc trưng

- Học không giám sát chủ yếu được chia thành hai nhóm:

Phân cụm (Clustering) Nhóm các đối tượng tương tự lại với nhau:

+ Partitioning: K Means, K-Medoids

+ Hierarchical: Bottom-up, Top-down

+ Density-based: DBSCan, Optics

Giảm chiều dữ liệu (Dimensionality Reduction): Giảm số lượng biến trong dữ liệu, giữ lại các thông tin quan trong nhất.

* **Học tăng cường**

- **Học tăng cường (RL)** là kỹ thuật máy học giúp đào tạo phần mềm đưa ra quyết định nhằm thu về kết quả tối ưu nhất. Kỹ thuật này bắt chước quy trình học thử và sai mà con người sử dụng để đạt được mục tiêu đã đặt ra. RL giúp phần mềm tăng cường các hành động hướng tới mục tiêu, đồng thời bỏ qua các hành động làm xao lãng mục tiêu.

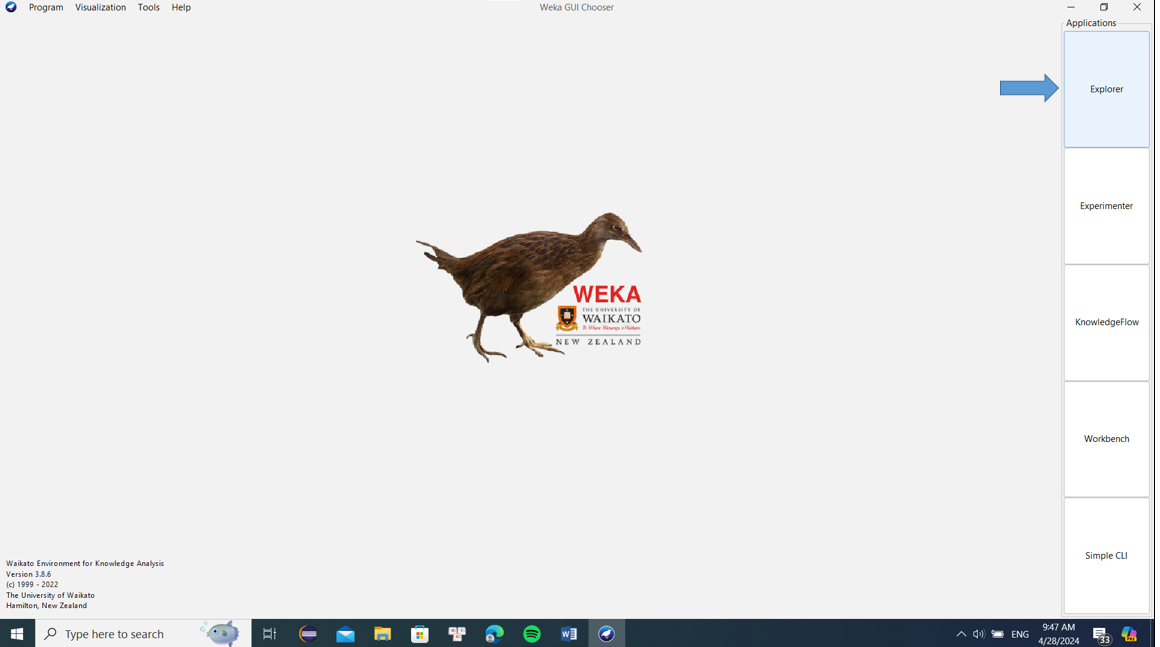
* Deep Learning Neural Network
* **Deep Learning**là một tập hợp con của học máy, về cơ bản là một mạng nơ-ron có ba lớp trở lên. Những mạng lưới thần kinh này cố gắng mô phỏng hành vi của não người cho phép deep learning “học” từ một lượng lớn dữ liệu. Trong khi mạng nơ-ron có một lớp vẫn có thể đưa ra các dự đoán gần đúng, các lớp ẩn bổ sung có thể giúp tối ưu hóa và tinh chỉnh để có độ chính xác.

# **Bài tập nghiên cứu nhóm**

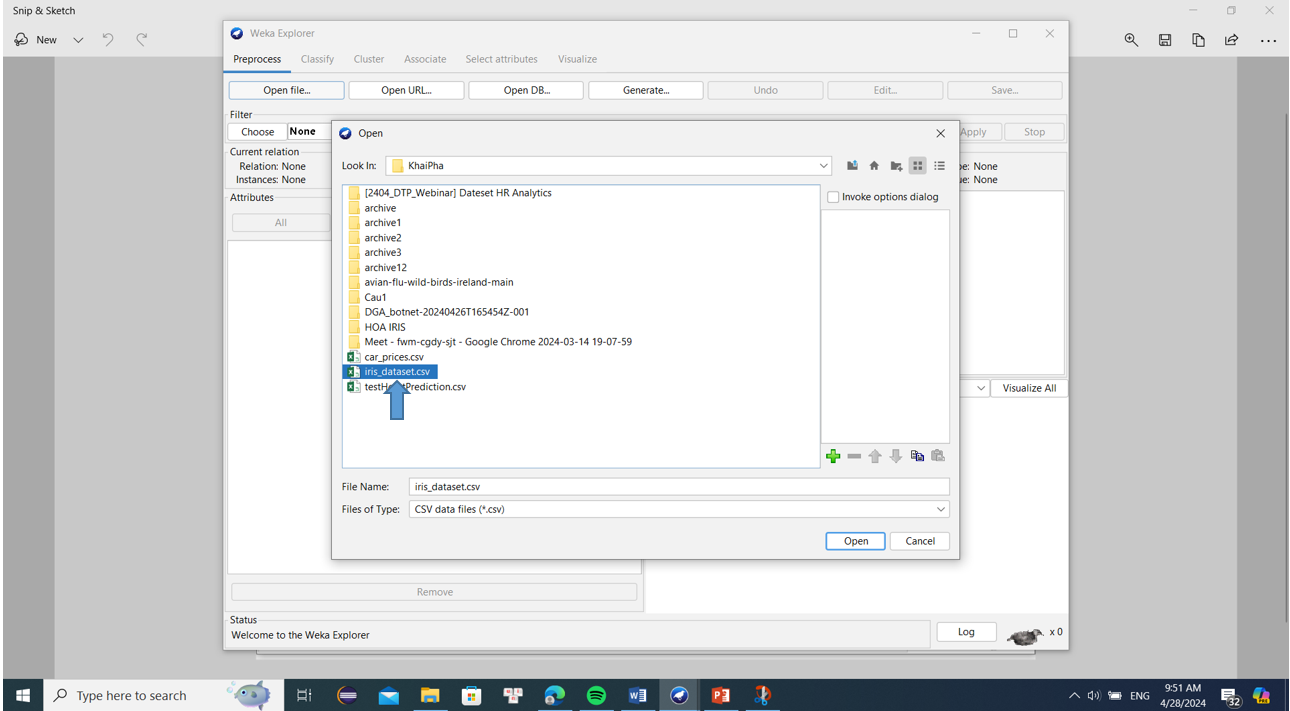
## Sử dụng thuật toán J48 đánh giá độ chính xác của bộ dữ liệu Iris

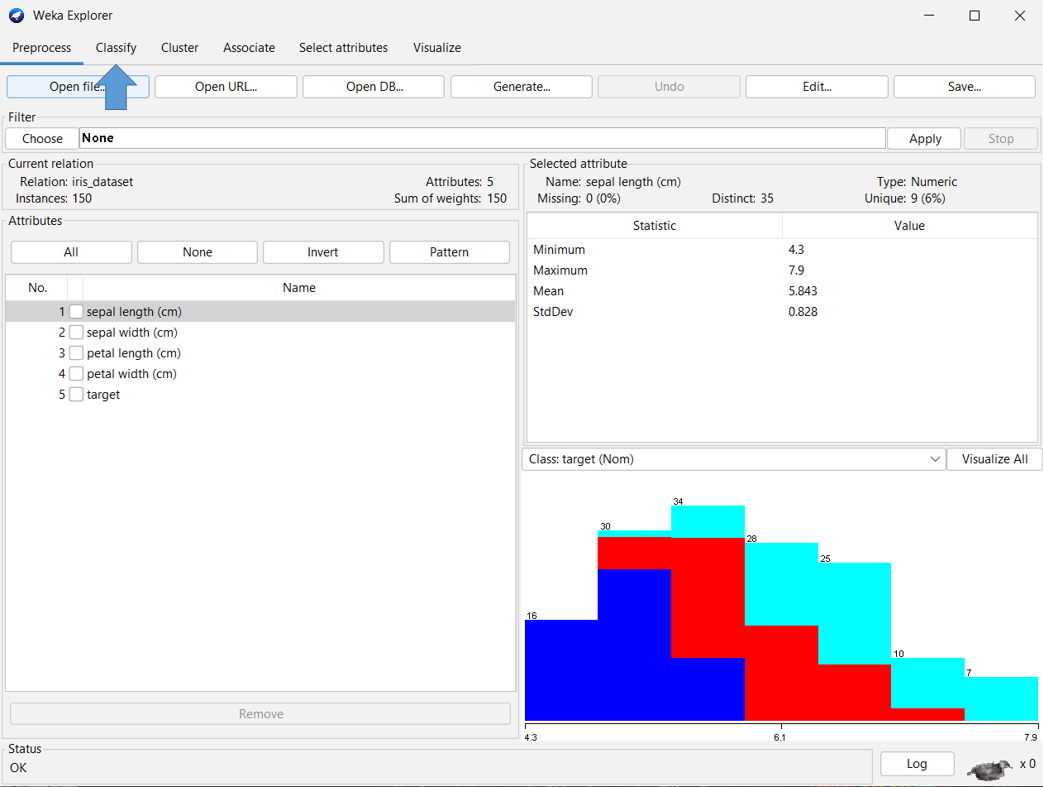
### **1.1. Các Bước Thực Hiện**

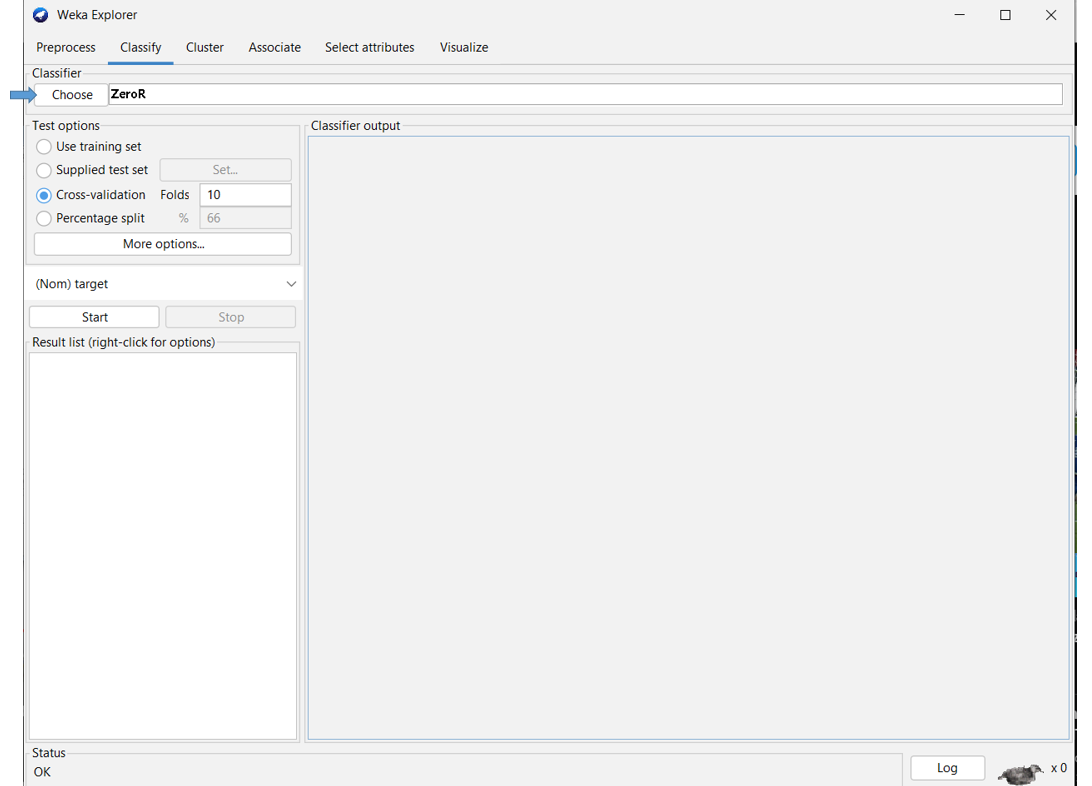
**Bước 1**: Sau khi mở phần mềm weka lên, chúng ta click vào mục Explorer.

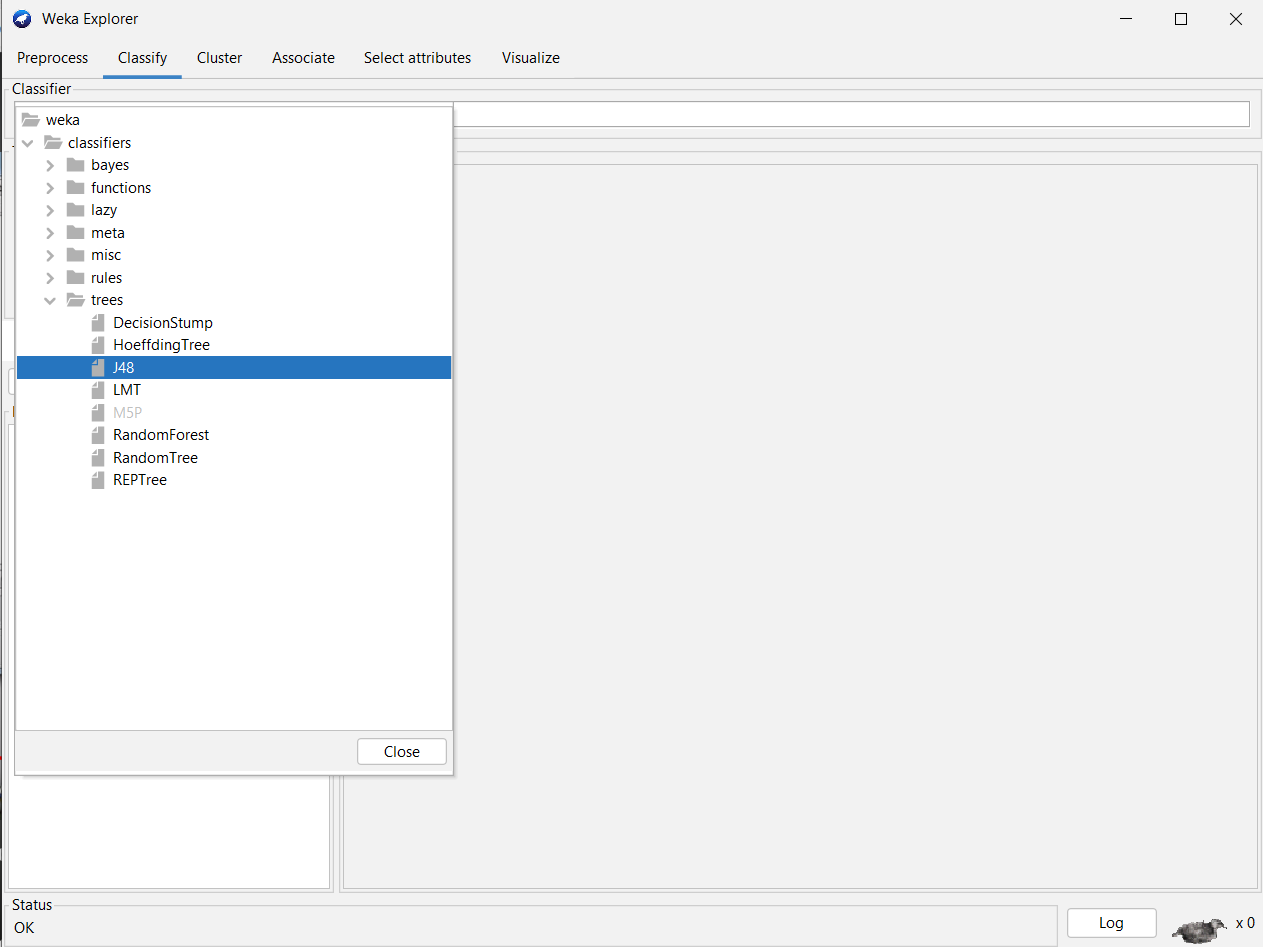
****

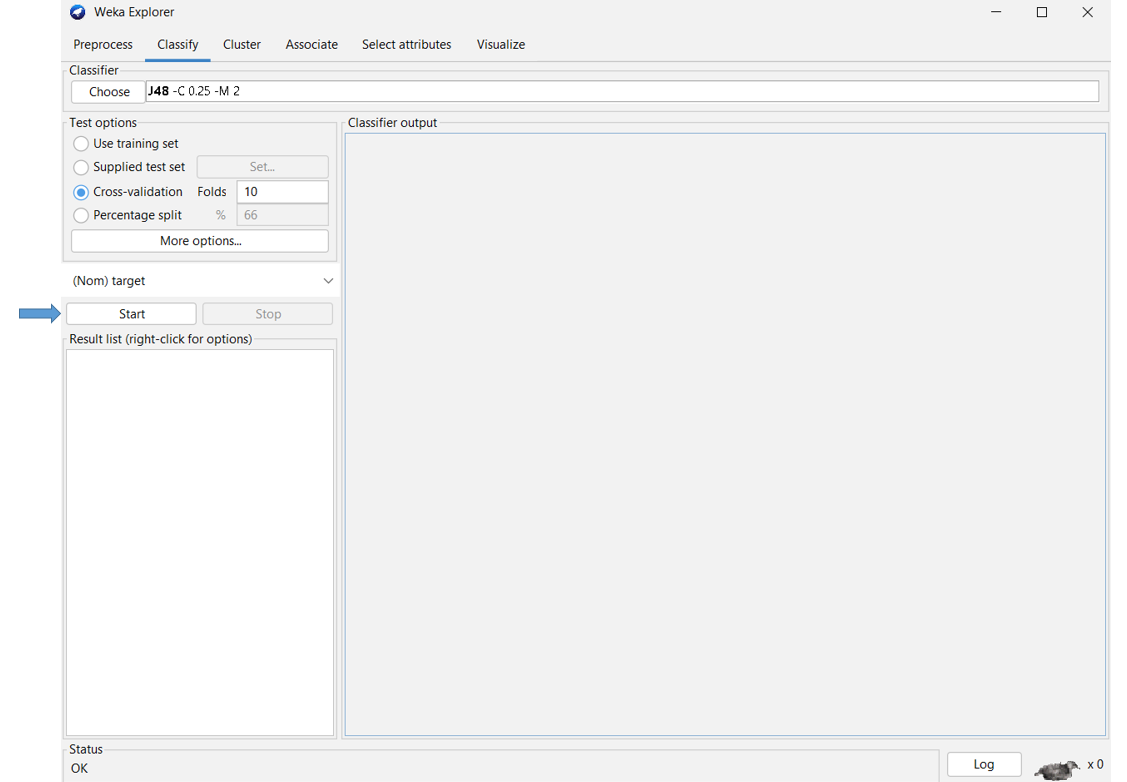
**Bước 2:** Sau đó chúng ta vào phần Open File để mở file dữ liệu.

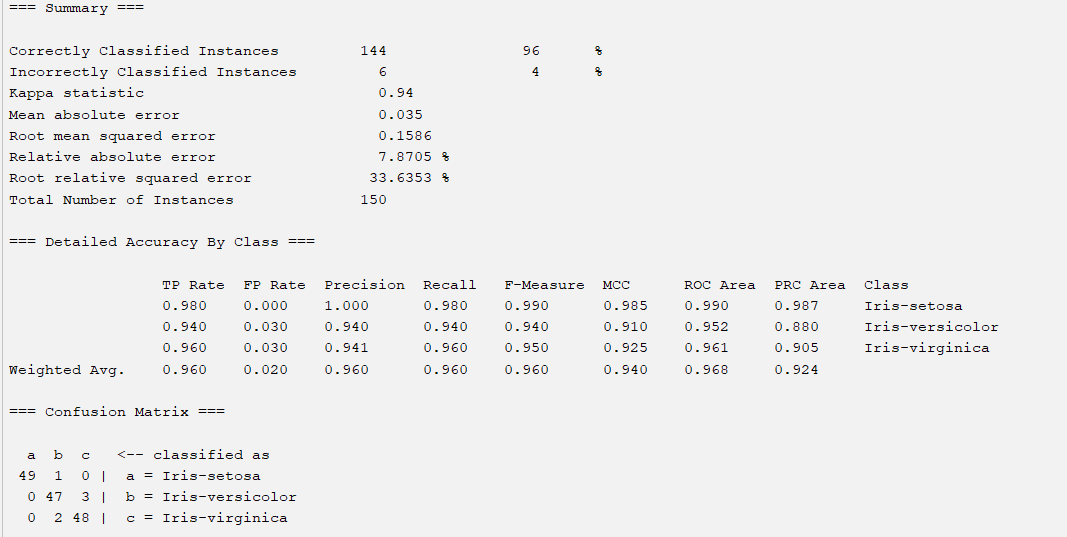
**Bước 3:** Chúng ta chọn file chứa dữ liệu cần sử dụng.

**Bước 4:** Sau khi chọn được dữ liệu, dữ liệu sẽ hiển thị tổng quan và chúng ta sẽ chọn Classify để tiến đánh giá độ chính xác

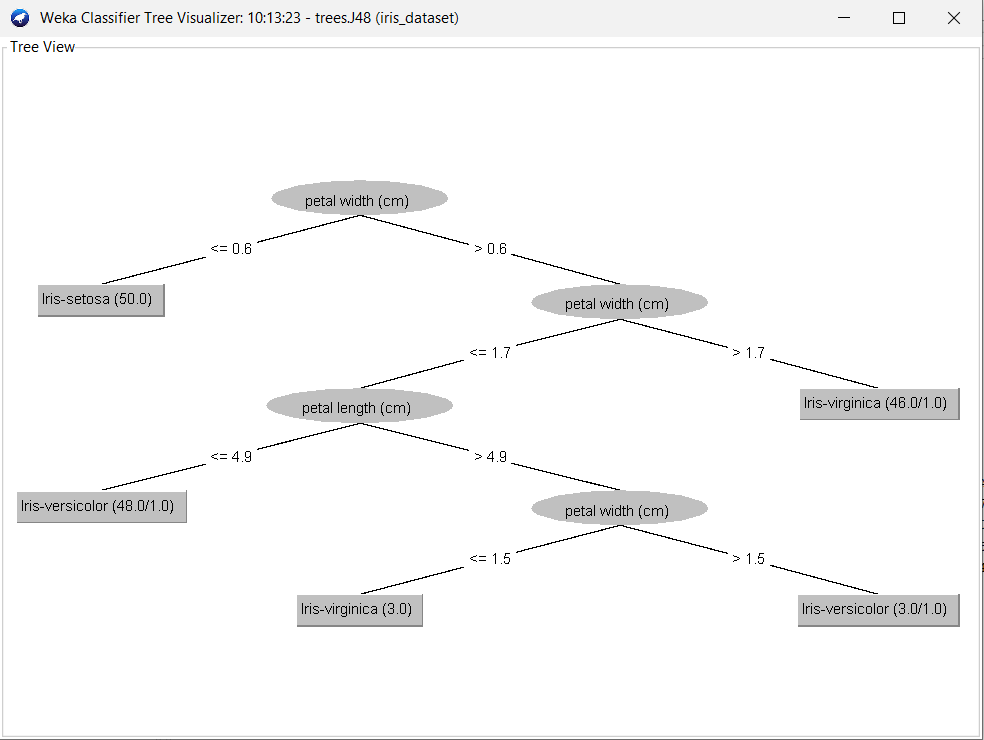
**Bước 5:** Chọn Choose để lựa chọn thuật toán.

**Bước 6:** Tại đây ta chọn thuật toán theo đề bài là J48

**Bước 7:** Chọn start để bắt đầu học.

**1.2. Kết quả**

### **1.3. Cây quyết định**



## 

## Dùng thuật toán K-Trung bình phân 3 cụm.

### **2.1. Giới thiệu về thuật toán K-Mean**

K-means là một thuật toán phân cụm đơn giản thuộc loại học không giám sát(tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm k-means là phân chia 1 bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k. Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng 1 cụm thì phải có cùng 1 số tính chất nhất định. Tức là giữa các điểm trong cùng 1 cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong 1 cụm đó sẽ là các điểm dữ liệu gần nhau.

Thuật toán phân cụm k-means thường được sử dụng trong các ứng dụng cỗ máy tìm kiếm, phân đoạn khách hàng, thống kê dữ liệu,…

### **2.2. Định nghĩa K-Mean là gì?**

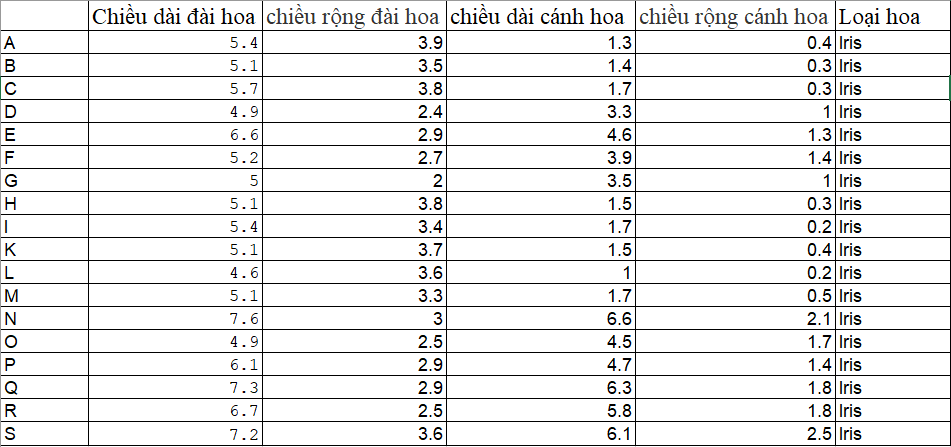
Thuật toán phân cụm k-means là một phương pháp được sử dụng trong phân tích tính chất cụm của dữ liệu. Nó đặc biệt được sử dụng nhiều trong khai phá dữ liệu và thống kê. Nó phân vùng dữ liệu thành k cụm khác nhau. Giải thuật này giúp chúng ta xác định được dữ liệu của chúng ta nó thực sự thuộc về nhóm nào.

### **2.3. Ý tưởng của thuật toán K-Mean**

* Khởi tạo K điểm dữ liệu trong bộ dữ liệu và tạm thời coi nó là tâm của các cụm dữ liệu của chúng ta.
* Với mỗi điểm dữ liệu trong bộ dữ liệu, tâm cụm của nó sẽ được xác định là 1 trong K tâm cụm gần nó nhất.
* Sau khi tất cả các điểm dữ liệu đã có tâm, tính toán lại vị trí của tâm cụm để đảm bảo tâm của cụm nằm ở chính giữa cụm.
* Bước 2 và bước 3 sẽ được lặp đi lặp lại cho tới khi vị trí của tâm cụm không thay đổi hoặc tâm của tất cả các điểm dữ liệu không thay đổi.

### **2.4. Bài làm**

##### **2.4.1. Dataset sử dụng**



Ba cụm lần lượt là :

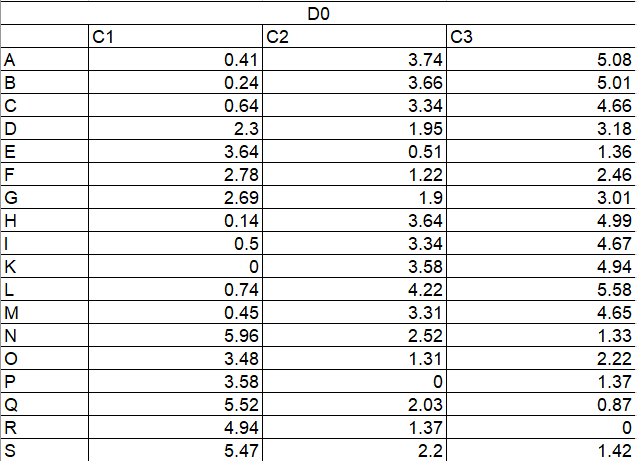
**C1:5.1,3.7,1.5,0.4,Iris-setosa**

**C2: 6.1,2.9,4.7,1.4,Iris-versicolor**

**C3: 6.7,2.5,5.8,1.8,Iris-virginica**

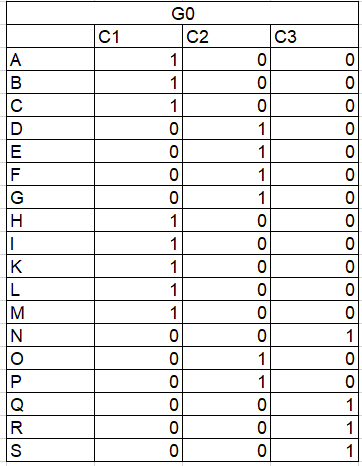
##### **2.4.2. Các bước thực hiện**

Bước 1: Tính khoảng cách từ các điểm tới các tâm dựa vào công thức tính khoảng cách  với p(p1,p2) và q(q1,q2) ta được kết quả :



Bước 2: Từ kết quả, phân nhóm theo khoảng cách gần nhất .

Gọi 0 là điểm không thuộc nhóm và 1 là điểm thuộc nhóm.

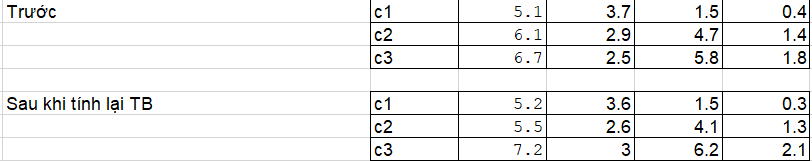


⇒ Ta có A, B, C, H, I, K, L, M thuộc C1

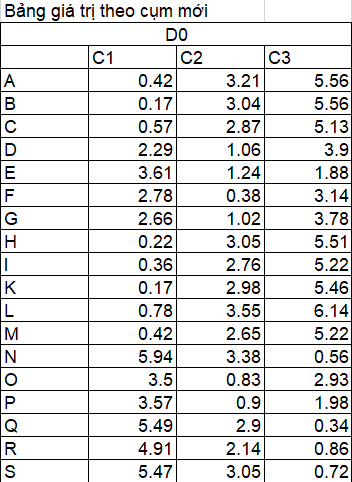
D, E, F, G, O, P thuộc C2

N, Q, R, S thuộc C3

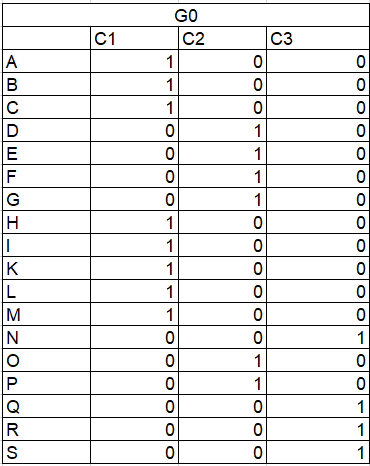
Bước 3: Tính là điểm trung bình của C1, C2, C3



Bước 4: Tính lại khoảng cách theo giá trị C1,C2,C3 mới

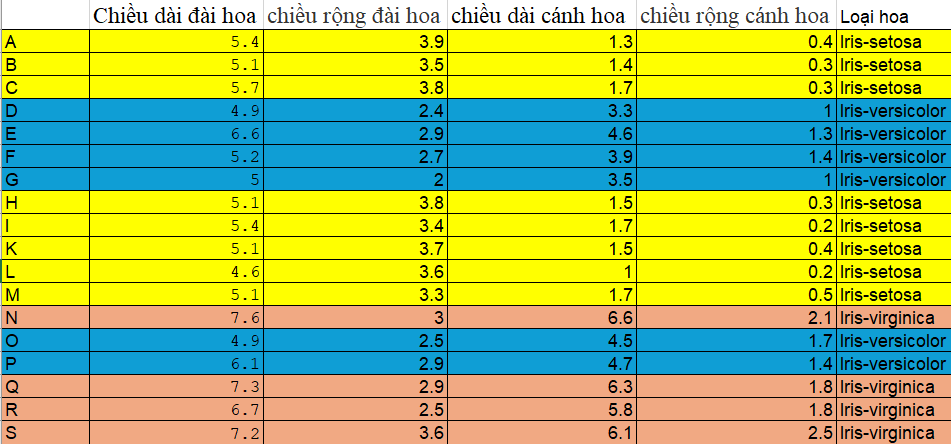


Bước 5: Phân nhóm lại khoảng cách theo khoảng cách mới



Bước 6 :Nhận thấy kết quả phân nhóm không thay đổi ⇒ dừng thuật toán

##### **2.4.3. Kết quả**

****

## Sử dụng mạng Neural học sâu để dự báo

Sử dụng mạng neural học sâu để nhận dạng hoặc phân lớp đối tượng. Đánh giá độ chính xác.

### **3.1. Chọn đối tượng và cơ sở dữ liệu**

* Trong lĩnh vực y tế, các mô hình học máy để dự đoán các căn bệnh nói chung và dự đoán dịch bệnh nói riêng luôn nhận được sự quan tâm của cộng đồng nghiên cứu, và luôn dành được sự ưu tiên trong việc đóng góp ngăn chặn sự bùng phát và lây lan của dịch bệnh.
* Ngày này môi trường ngày càng ô nhiễm, con người có nguy cơ mắc nhiều loại bệnh nguy hiểm khác nhau.Một trong số đó là căn bệnh tim.
* Bệnh tim là 1 loại bệnh vô cùng nguy hiểm.Tuy nhiên nếu người bệnh có thể phát hiện ra bệnh sớm thì có thể điều trị hiệu quả.

⇒ Từ những lý do trên, chúng em quyết định lựa chọn cơ sở dữ liệu “**Dự đoán bệnh tim dựa trên phương pháp học sâu**”.

### **3.2. Tiền xử lý dữ liệu.**

#### **3.2.1. Xác định vấn đề.**

* Giảm thiểu áp lực cho hệ thống y tế.
* Giúp người sử dụng đánh giá được tình trạng bệnh hiện tại của bản thân qua đó lựa chọn các cách phù hợp để điều trị bệnh của mình.
* Góp phần vào việc phát hiện bệnh để được điều trị sớm nhất.

#### **3.2.2. Xác định mục đích**

* Xác định các yếu tố ảnh hưởng:
* Mức độ nghiêm trọng của triệu chứng.
* Xác định mục đích:
* Phát triển một mô hình dự đoán có thể xác định với độ chính xác cao khoảng 90% nguy cơ của người mắc bệnh tiểu đường dựa trên mức độ nghiêm trọng của các yếu tố mà người bệnh có.
* Tăng tỷ lệ phát hiện bệnh.

#### **3.2.3. Đặc điểm dữ liệu.**

* Các đặc trưng đầu vào của dữ liệu gốc:

Age - Tuổi

Sex - Giới tính

Chest pain type - Kiểu đau ngực

Resting bp s - Huyết áp nghỉ ngơi (mm Hg).

Cholesterol: Cholesterol huyết thanh (mg/dl)

Fasting blood sugar: Đường huyết nhanh sau 12 giờ nhanh (1 nếu > 120 mg/dl, 0 nếu không)

Resting ecg: Kết quả điện tâm đồ nghỉ ngơi, được mã hóa theo các giá trị từ 0 đến 2

Max heart rate: Tần số tim tối đa đạt được

Exercise angina: Đau ngực do tập thể dục (1 nếu có, 0 nếu không)

Oldpeak: Giảm đỉnh ST gây ra bởi tập thể dục so với nghỉ ngơi

ST slope: Giai đoạn ST của đỉnh tập thể dục, được mã hóa theo các giá trị từ 1 đến 3

Target: Mục tiêu (1 nếu có bệnh tim, 0 nếu không)

* Độ chính xác và tin cậy chính xác, được đảm bởi các nguồn uy tín như Kaggle, thông tin của chính phủ, bộ y tế cung cấp

#### **3.2.4. Phương pháp thu thập dữ liệu**

Phương pháp thu thập dữ liệu là từ nguồn hiện có, cụ thể được ký hiệu trong báo cáo như sau:

**[DATASET] Heart Disease Dataset,** thu thập dữ liệu từ Kaggle

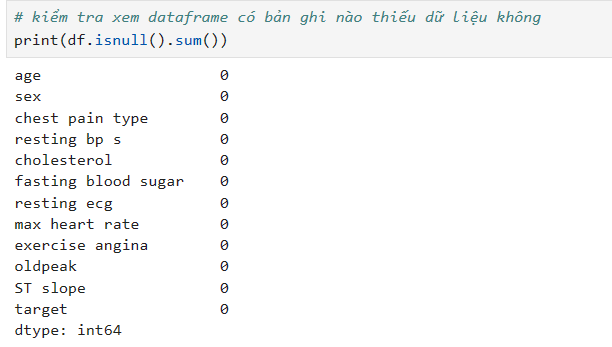
[🫀 Heart Disease Dataset (kaggle.com)](https://www.kaggle.com/datasets/mexwell/heart-disease-dataset?select=documentation.pdf) ,

**[DATASET] Heart Attack Analysis & Prediction Dataset,** thu thập dữ liệu từ Kaggle [Heart Attack Analysis & Prediction Dataset (kaggle.com)](https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset)

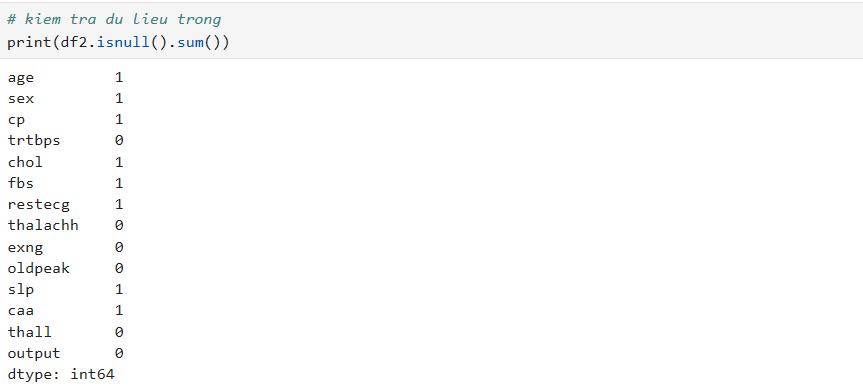
#### **3.2.5. Tiền xử lý dữ liệu**

###### **Điền các bản ghi còn thiếu**

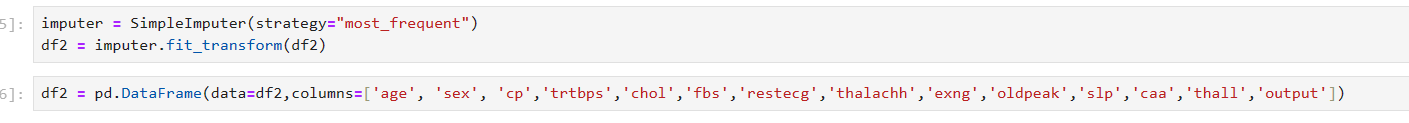
* Phương pháp:
* Kiểm tra xem dataset có bị missing dữ liệu hay không
* Điền các giá trị thường xuyên xuất hiện hoặc giá trị trung bình
* Cách thực hiện:
* Bước 1: Kiểm tra xem dataset có bị thiếu dữ liệu hay không?
  + Dataset 1 : Không có giá trị thiếu



* Dataset 2: Có dữ liệu bị thiếu

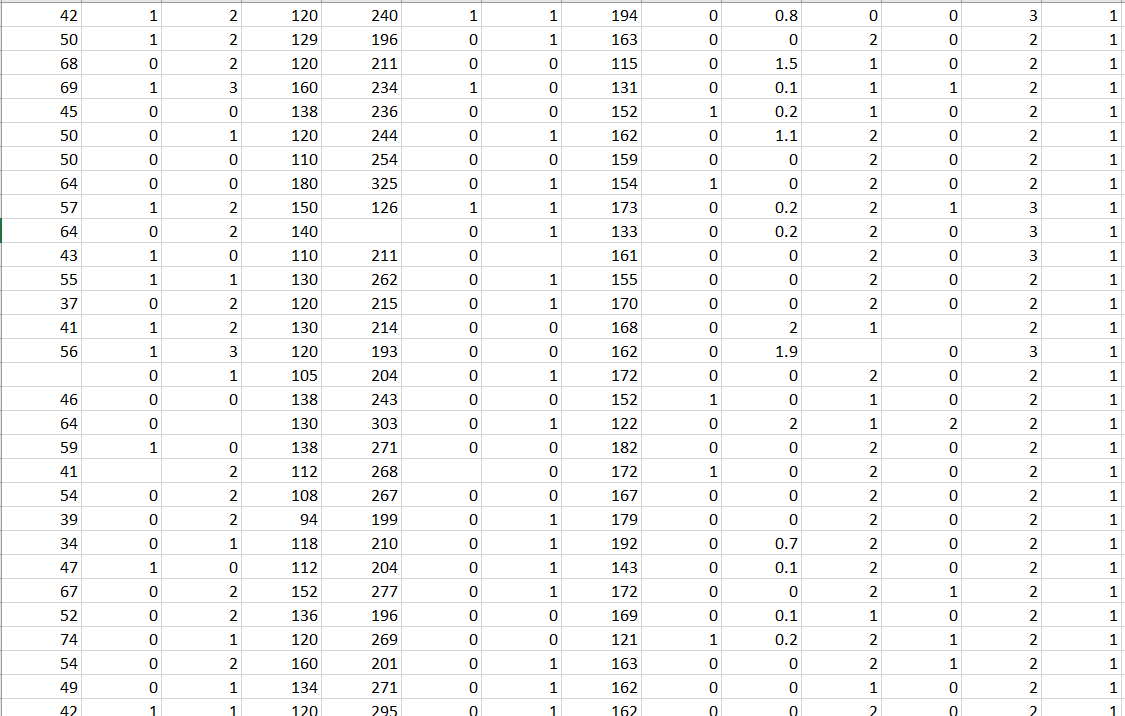


* Bước 2: điền giá trị trung bình hoặc giá trị thường xuyên.

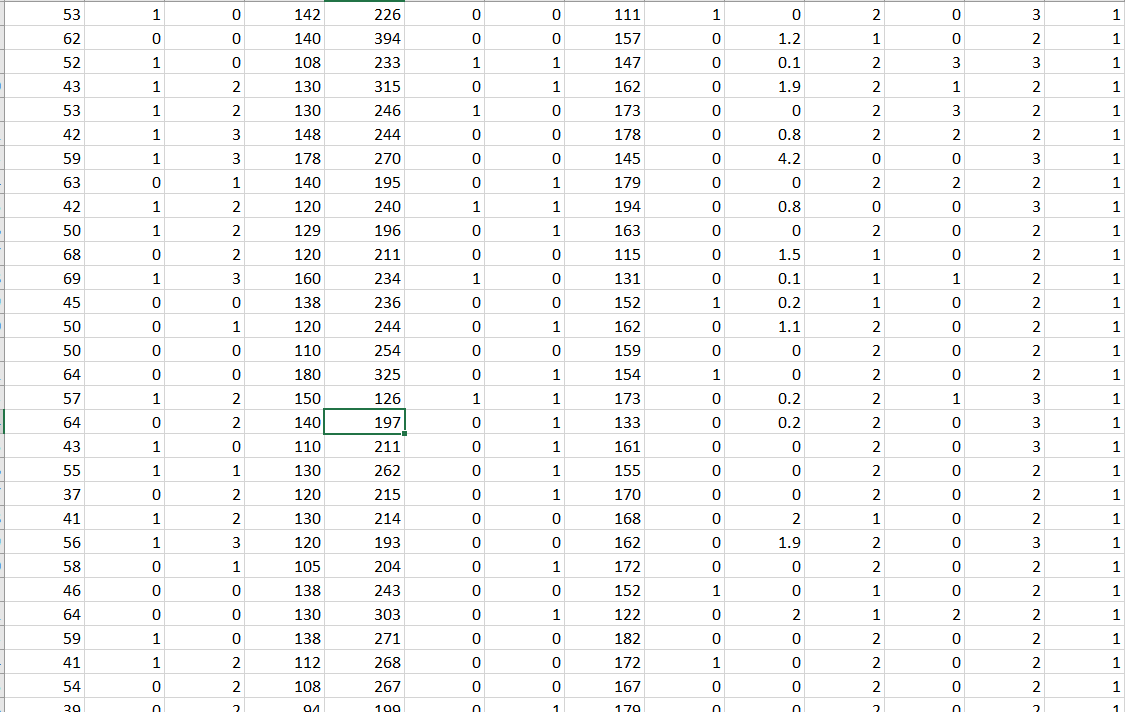


⇒ Dữ liệu sau khi được điền

Trước khi chạy thuật toán:

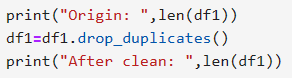


Sau khi chạy thuật toán :



###### **Lọc các bản ghi trùng lặp**

* Sử dụng hàm drop\_duplicates() để xác định các bản ghi trùng lặp và loại bỏ các bản ghi đó
* Hàm drop\_duplicates() sử dụng một thuật toán băm (hashing) để xác định các bản ghi có cùng giá trị cho tất cả các cột được chỉ định. Sau khi đã xác định được các bản ghi trùng lặp thì tiến hành loại bỏ chúng khỏi dataset.
* Code lọc các bản ghi trùng lặp sử dụng hàm drop\_duplicates()



* Kết quả
* Dataset 1



* Dataset 2



###### **Tích hợp dữ liệu**

* Data Integration hay quá trình tích hợp dữ liệu bao gồm việc kết hợp các dữ liệu không đồng nhất trong những nguồn khác nhau vào trong cùng một lược đồ duy nhất.
* Quá trình tích hợp dữ liệu
* Xác định các trường dữ liệu giống nhau giữa 2 DATASET

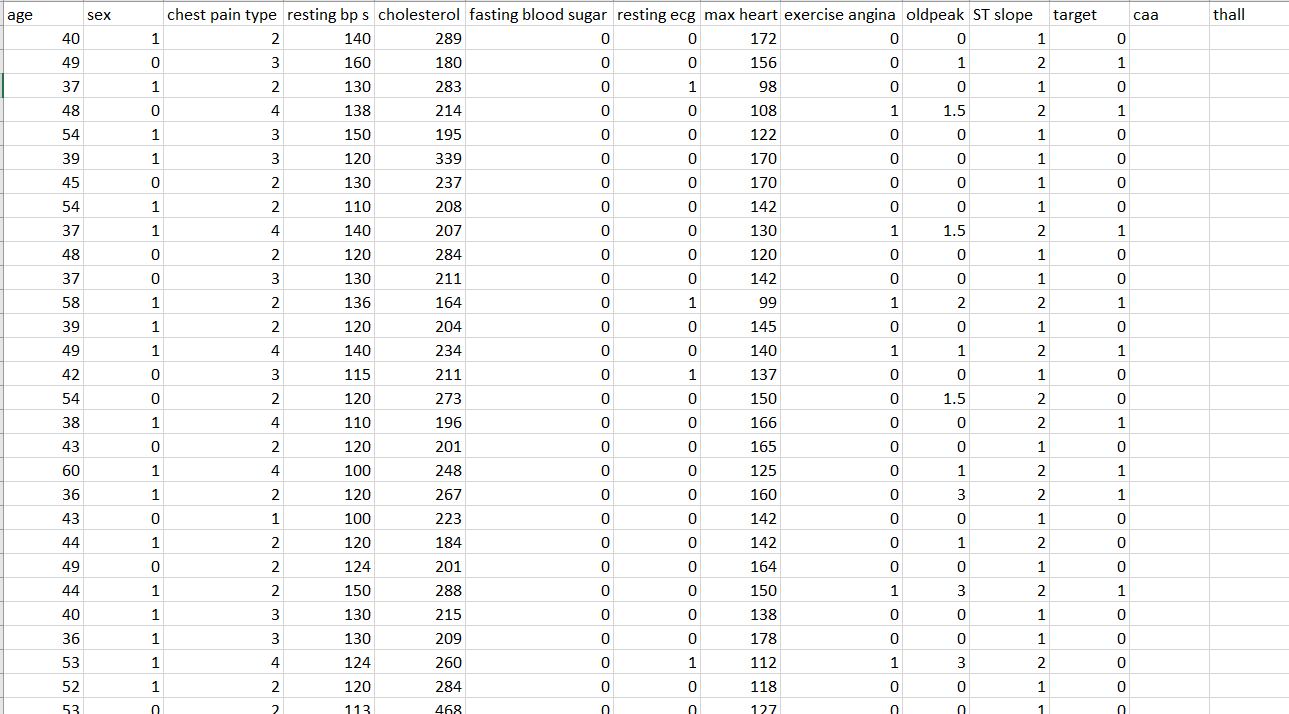
|  |  |
| --- | --- |
| **Dataset 1** | **Dataset 2** |
| Age - Tuổi | Age - Tuổi |
| Sex - Giới tính (1 Nam, 0 Nữ) | Sex - Giới tính (1 Nam, 0 Nữ) |
| Chest pain type - Kiểu đau ngực | cp - Kiểu đau ngực |
| Resting bp s - Huyết áp nghỉ ngơi (mm Hg). | trtbps - Huyết áp nghỉ ngơi (mm Hg). |
| Cholesterol: Cholesterol huyết thanh (mg/dl) | chol - Cholesterol huyết thanh (mg/dl) |
| Fasting blood sugar: Đường huyết nhanh sau 12 giờ nhanh (1 nếu > 120 mg/dl, 0 nếu không) | fbs - Đường huyết nhanh sau 12 giờ nhanh (1 nếu > 120 mg/dl, 0 nếu không) |
| Resting ecg: Kết quả điện tâm đồ nghỉ ngơi, được mã hóa theo các giá trị từ 0 đến 2 | restecg - Kết quả điện tâm đồ nghỉ ngơi, được mã hóa theo các giá trị từ 0 đến 2 |
| Max heart rate: Tần số tim tối đa đạt được | thalachh - Tần số tim tối đa đạt được |
| Exercise angina: Đau ngực do tập thể dục (1 nếu có, 0 nếu không) | exng - Đau ngực do tập thể dục (1 nếu có, 0 nếu không) |
| Oldpeak: Giảm đỉnh ST gây ra bởi tập thể dục so với nghỉ ngơi | oldpeak - Giảm đỉnh ST gây ra bởi tập thể dục so với nghỉ ngơi |
| ST slope: Giai đoạn ST của đỉnh tập thể dục, được mã hóa theo các giá trị từ 1 đến 3 | slp - Giai đoạn ST của đỉnh tập thể dục, được mã hóa theo các giá trị từ 1 đến 3 |
| Target: Mục tiêu (1 nếu có bệnh tim, 0 nếu không) | caa - number of major vessels (0-3) |
|  | thall - Thal rate |
|  | Output - Mục tiêu (1 nếu có bệnh tim, 0 nếu không) |

* Tích hợp 2 Dataset sử dụng hàm concat()



* Kết quả Dataset mới sau khi tích hợp

Dataset mới bao gồm 1221 bản ghi, 11 thuộc tính và 1 nhãn



###### **Giảm chiều dữ liệu**

+ SBS (Sequential Backward Selection) là một phương pháp tìm tập con tốt nhất của các đặc trưng trong quá trình huấn luyện mô hình học máy.

* Phương pháp SBS bắt đầu với toàn bộ tập đặc trưng và loại bỏ lần lượt các đặc trưng cho đến khi đạt được độ chính xác tốt nhất của mô hình.
* **Đầu vào:** Tập tất cả các đặc trưng Y = {y1; y2;…}.
* **Đầu ra:** Xk = {xj | j = 1,2,….k; xj Y}, k = (0, 1, 2, …, d) SBS trả về một tập hợp con các đặc trưng; số đặc trưng k (k < d) và phải được chỉ định trước.
* **Khởi tạo:** X0 = Y; k = d. Khởi tạo thuật toán cùng với việc đưa tất cả các đặc trưng từ Y vào X.
* **Bước 1 (bao gồm):** x- = argmaxJ (Xk - x), x Xk

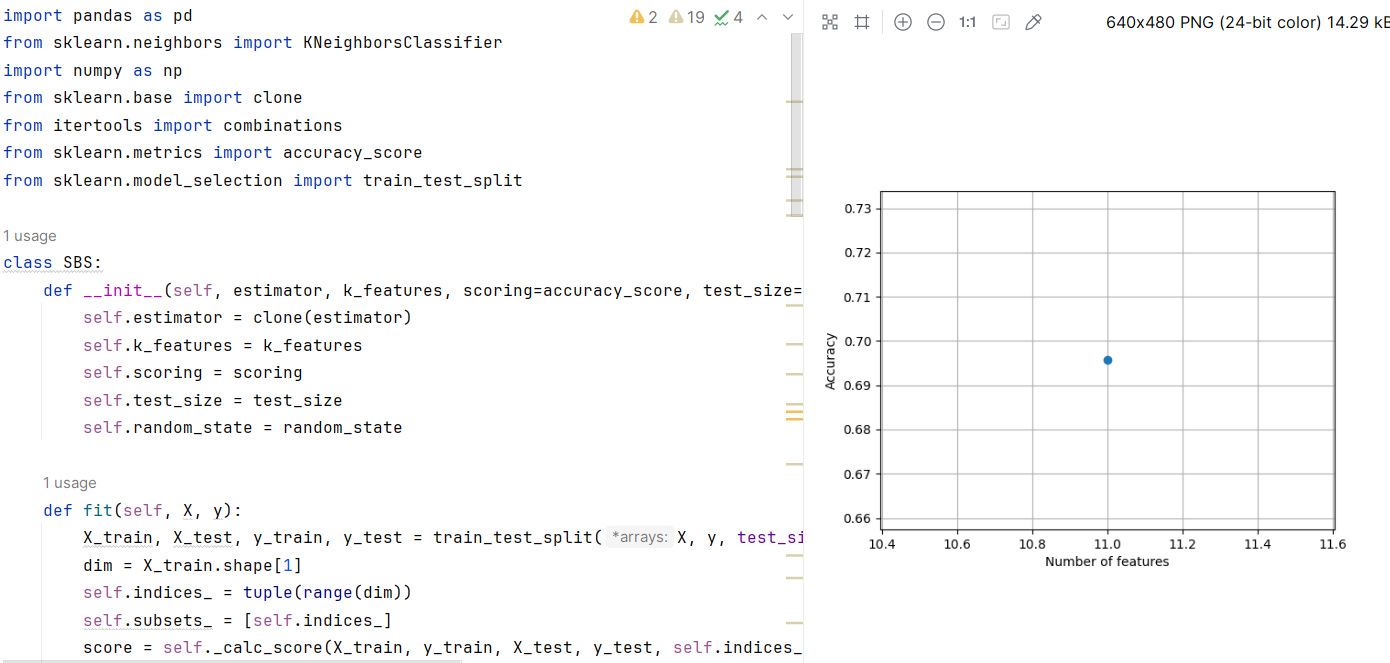
Xk-1 = Xk – x-

k = k - 1

Trong bước này, bỏ X- khỏi Xk, X- là đặc trưng đem lại hiệu suất phân loại cao nhất nếu nó được bỏ ra khỏi Xk. Lặp lại cho đến khi tiêu chí dừng được xác lập.

* **Dừng:** k = p thêm các đặc trưng vào tập con Xk cho đến khi số đặc trưng k bằng p được chỉ định trước hoặc đạt một hiệu suất nào đó.

Áp dụng phương pháp SBS vào giảm chiều dữ liệu vào DATASET mới, ta thấy không giảm đi còn 11 cột dữ liệu.

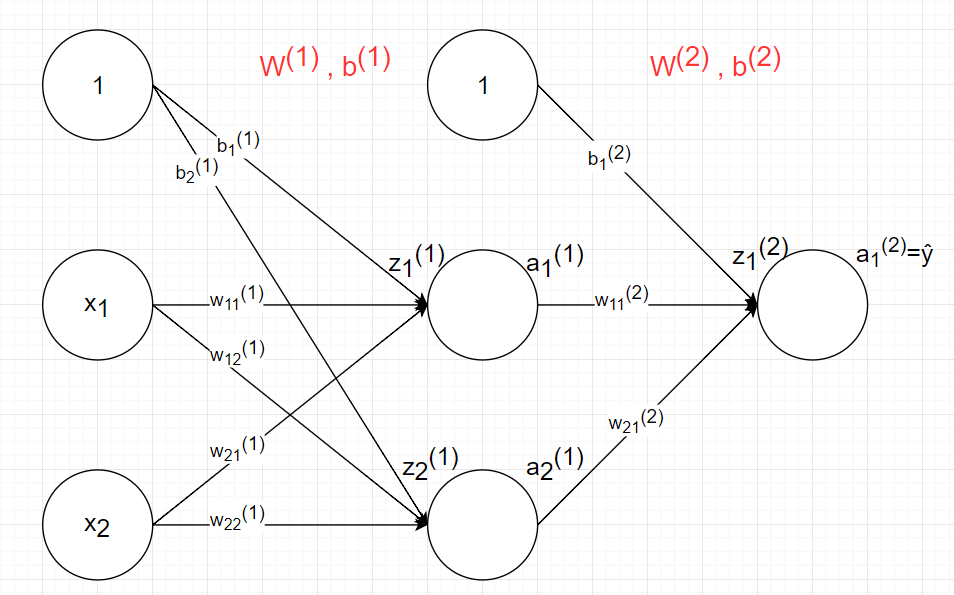


### **3.3. Xây dựng mô hình mạng nơ ron**

#### **3.3.1. Sử dụng mô hình mạng nơron MLP (Multi-layer Perceptron)**

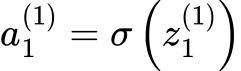
##### **3.3.1.1. Cơ sở lý thuyết**

* Một mạng Neural sẽ có 3 kiểu tầng:
* Tầng vào (Input layer): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
* Tầng ra (Output layer): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
* Tầng ẩn (Hidden layer): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.
* Mô hình mạng nơron được sử dụng rộng rãi nhất là mô hình mạng nhiều tầng truyền thẳng (MLP: Multi Layer Perceptron). Một mạng MLP tổng quát là mạng có n (n≥2) tầng (thông thường tầng đầu vào không được tính đến): trong đó gồm một tầng đầu ra (tầng thứ n) và (n-1) tầng ẩn.

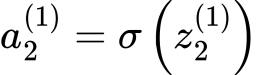


* Các hệ số và bias tương ứng được ký hiệu như trong hình Feedforward:

wps



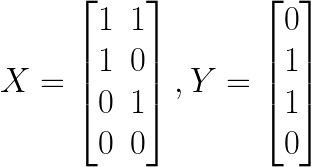
wps



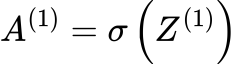
wps

wps

* Viết dưới dạng ma trận:



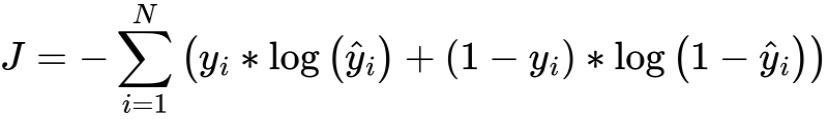
wps



wps

wps

* Loss function (Hàm mất mát):
* Với mỗi điểm (*x*[*i*]*,yi*), gọi hàm loss function
* Hàm loss function trên toàn bộ dữ liệu:



* Gradient descent
* Để áp dụng gradient descent ta cần tính được đạo hàm của các hệ số W và bias b với hàm loss function.
* **Ký hiệu chuẩn về đạo hàm**
* Khi hàm f(x) là hàm 1 biến x, ví dụ: *f*(*x*)=2∗*x*+1. Đạo hàm của f đối với biến x kí hiệu là:

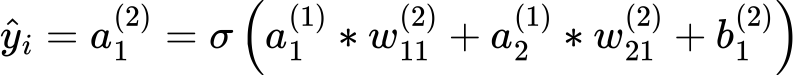
wps

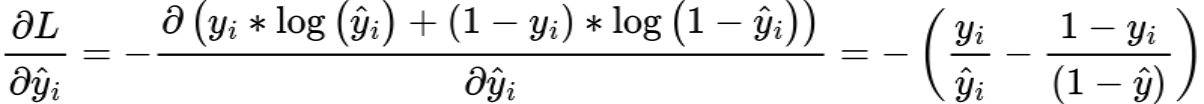
* Khi hàm f(x, y) là hàm nhiều biến, ví dụ *f*(*x,y*)= *x*2+*y*2. Đạo hàm f với biến x kí hiệu là:



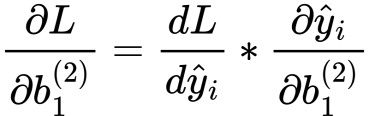
* Với mỗi điểm (*x*([*i*]*,yi*), hàm loss function là giá trị mà model dự đoán, còn *yi* là giá trị thật của dữ liệu.

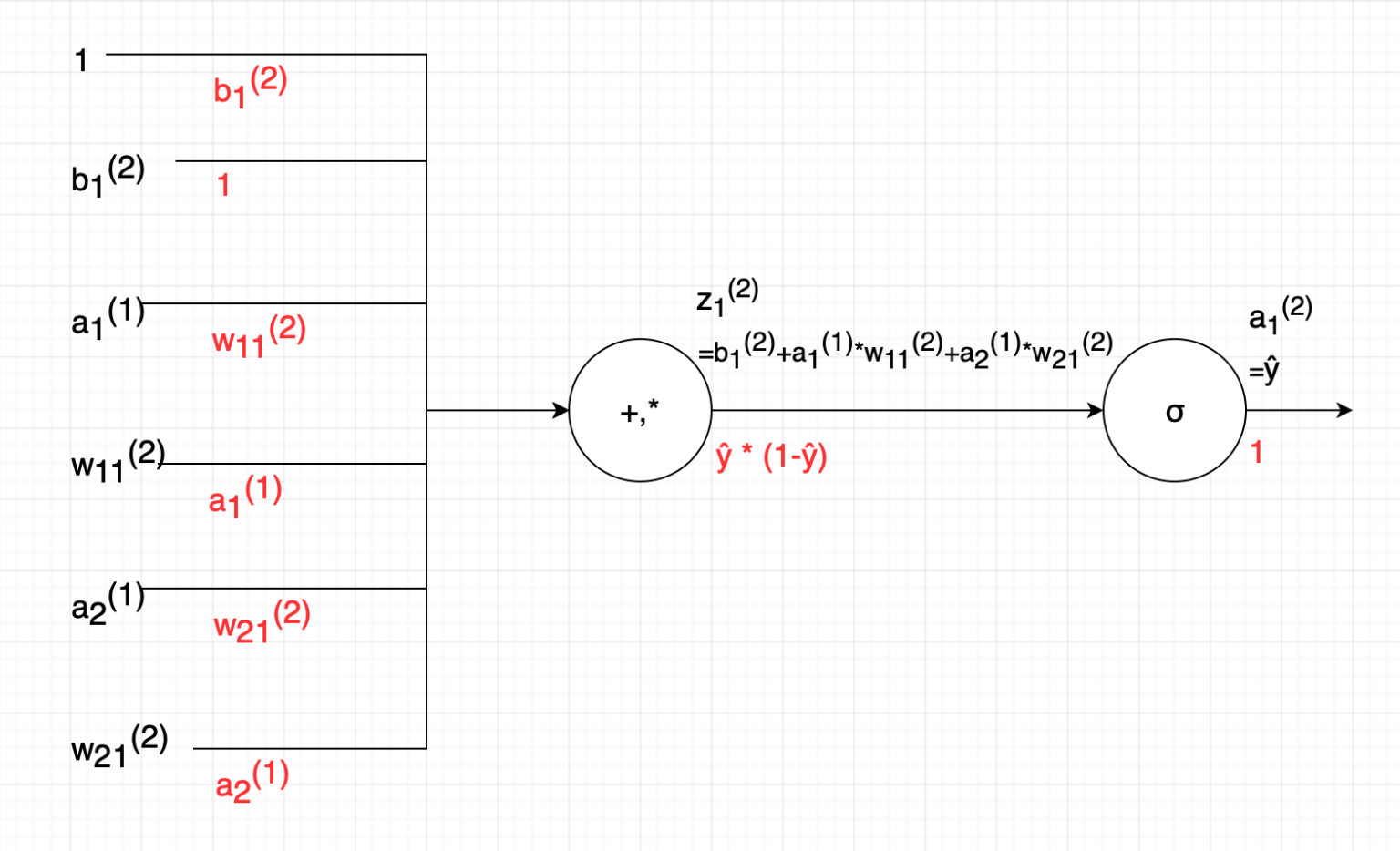
wps



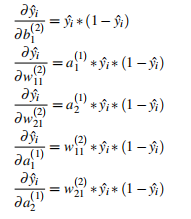


* Tính đạo hàm L với *W*(2)*,b*(2)
* Áp dụng chain rule ta có:

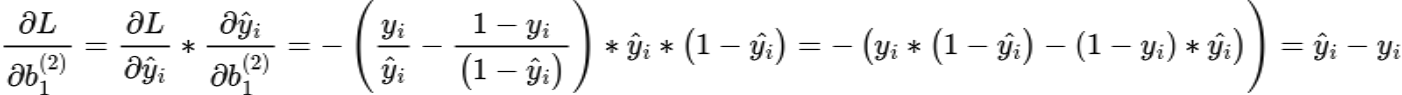




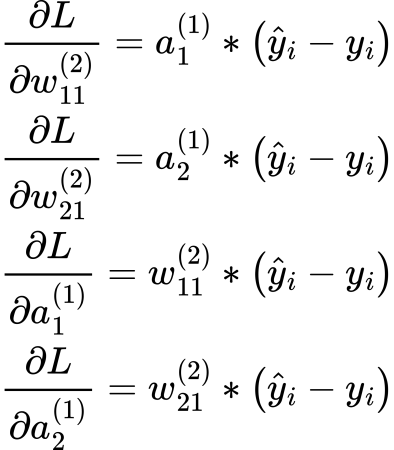
* Từ đồ thị ta thấy:



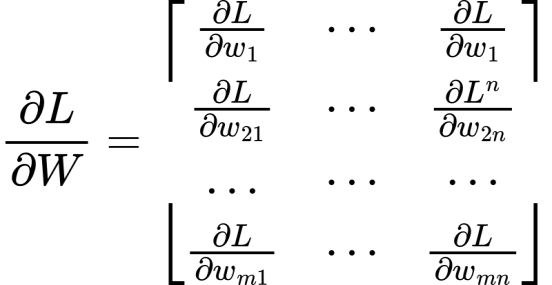
* Do đó:



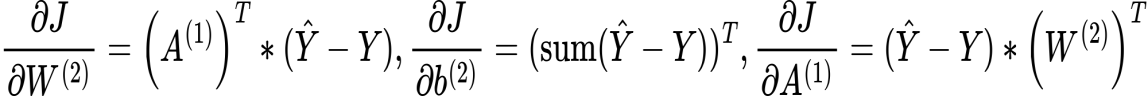
* Tương tự:



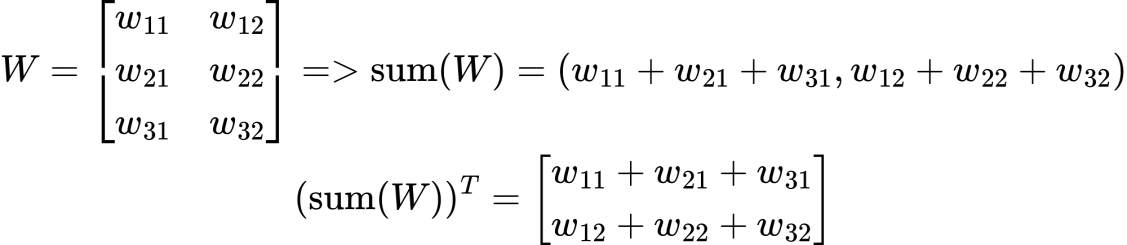
* **Biểu diễn dưới dạng ma trận**
* Lưu ý: Đạo hàm của L đối với ma trận W kích thước m\*n cũng là một ma trận cùng kích thước m\*n.



* Do đó:

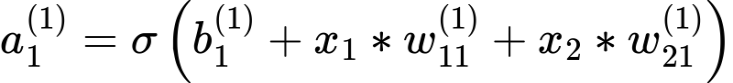


* Phép tính sum tính tổng các cột của ma trận:

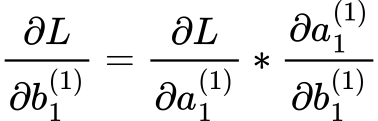


Vậy là đã tính xong đạo hàm của L với hệ số *W*(2) *,b* (2). Giờ sẽ đi tính đạo hàm của L với hệ số *W*(1) *,b* (1) vì khi tính đạo hàm của hệ số và bias trong layer trước đấy sẽ cần dùng đến.

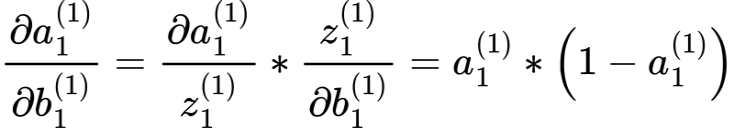
* Tính đạo hàm L với *W*(1) *, b*(1)



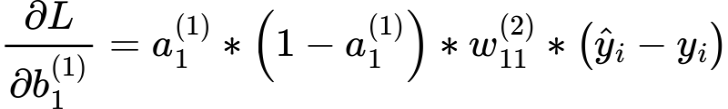
* Áp dung chain rule ta có:



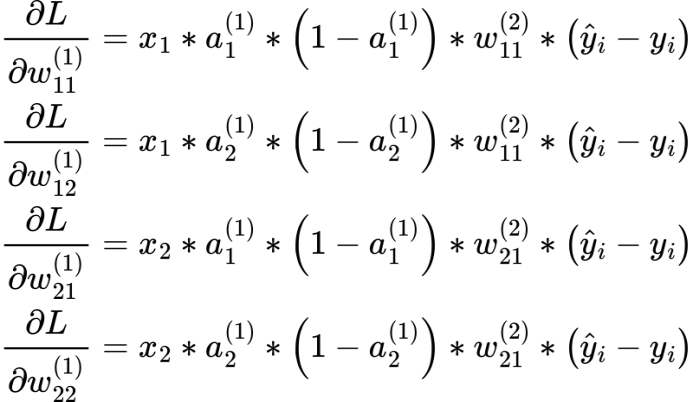
* Ta có:



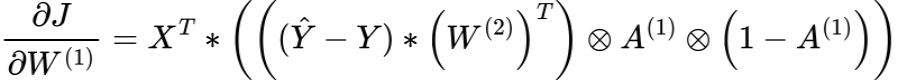
* Do đó:



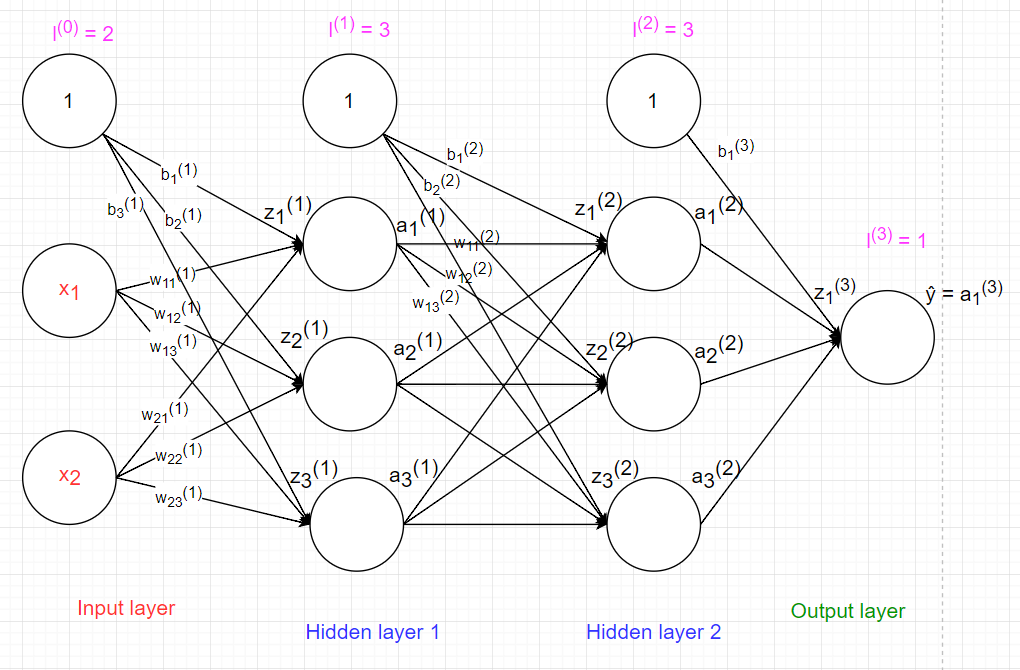
* Tương tự:

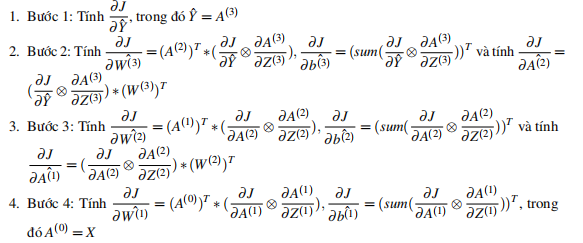


* Viết dưới dạng ma trận:

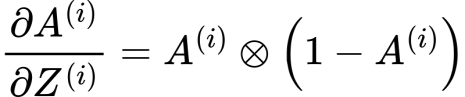


* Mô hình tổng quát:



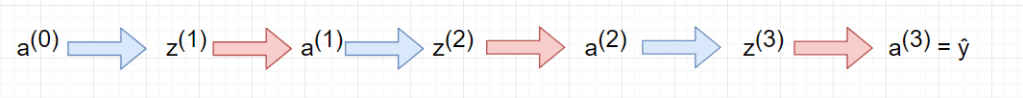


Nếu network có nhiều layer hơn thì cứ tiếp tục cho đến khi tính được đạo hàm của loss function J với tất cả các hệ số W và bias b.

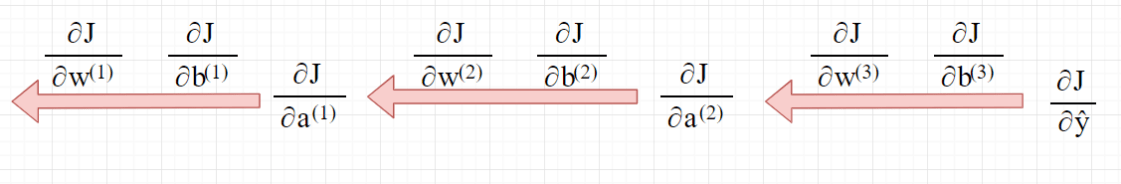


Nếu hàm activation là sigmoid thì

Quá trình Feedforward:



Quá trình Backpropagation:



##### **3.3.1.2. Vì sao chọn MLP cho bài toán?**

* Khả năng học đã chiều: MLP có khả năng học và biểu diễn các mối quan hệ phức tạp giữa các biến đầu vào và đầu ra thông qua việc sử dụng nhiều lớp ẩn, giúp có khả năng mô hình hóa dữ liệu không tuyến tính và xử lý các mối quan hệ phức tạp trong dữ liệu bệnh Tiểu đường.
* Linh hoạt trong cấu trúc mạng: MLP điều chỉnh số lượng lớp ẩn, số lượng neural trong mỗi lớp và hàm kích hoạt, giúp tối ưu hóa hiệu suất của mô hình.
* Xử lý dữ liệu không đồng nhất (không tuyến tính): Dữ liệu trong bệnh Tim thường đa dạng và không đồng nhất, bao gồm các biến số liên tục và các biến mô tả. MLP có khả năng xử lý dữ liệu đa dạng này và thích ứng với các mẫu dữ liệu phức tạp.
* Hiệu suất đáng tin cậy: MLP đã được chứng minh là có hiệu suất tốt trong nhiều bài toán phân loại và dự đoán trong lĩnh vực y học, bao gồm cả dự đoán bệnh Tim.
* Dễ sử dụng và triển khai.

##### 

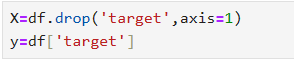
##### **3.3.1.3. Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán**

Sử dụng mô hình mạng nơron MLP (Multi-layer Perceptron)

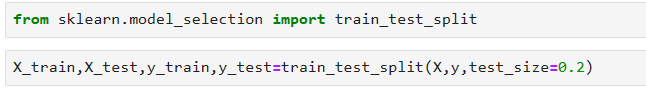
* Đọc dữ liệu từ file CSV



* Phân chia dữ liệu thành features và target



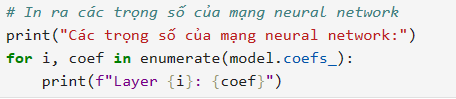
* Chia dữ liệu thành tập training và tập test (theo tỷ lệ 8:2)



* Khởi tạo và huấn luyện mô hình MLP



* In ra các trọng số của mạng neural network



* Lưu mô hình vào file



#### **3.3.2. Sử dụng mô hình mạng nơ ron CNN (Convolutional Neural Network)**

##### **3.3.2.1. Cơ sở lý thuyết**

Convolutional Neural Network (CNNs – Mạng nơron tích chập) là một trong những mô hình Deep Learning tiên tiến, giúp chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

Về cơ bản thiết kế của một mạng nơ ron tích chập 2 chiều có dạng như sau:

INPUT -> [[CONV -> RELU]*N -> POOL]*M -> [FC -> RELU]\*K -> FC

Trong đó:

INPUT: Lớp đầu vào

CONV: Lớp tích chập

RELU: Lớp biến đổi thông qua hàm kích hoạt relu để kích hoạt tính phi tuyến

POOL: Lớp tổng hợp, thông thường là Max pooling hoặc có thể là Average pooling dùng để giảm chiều của ma trận đầu vào.

FC: Lớp kết nối hoàn toàn. Thông thường lớp này nằm ở sau cùng và kết nối với các đơn vị đại diện cho nhóm phân loại.

* Các kí hiệu []*N, []*M hoặc []\*K ám chỉ cấu trúc bên trong [] có thể lặp lại nhiều lần liên tiếp nhau. M, K là số lần lặp lại. Kí hiệu -> đại diện cho các lớp liền kề nhau mà lớp đứng trước -> sẽ làm đầu vào cho lớp đứng sau ->.

- Như vậy ta có thể thấy một mạng nơ ron tích chập về cơ bản có 3 quá trình khác nhau:

* Quá trình chiết xuất đặc trưng: Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một lớp mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng hàm kích hoạt relu.
* Quá trình tổng hợp: Các lớp ở về sau quá trình chiết xuất đặc trưng sẽ có kích thước lớn do số đơn vị ở các lớp sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng hệ số và khối lượng tính toán trong mạng nơ ron. Do đó để giảm tải tính toán chúng ta sẽ cần giảm chiều của ma trận hoặc giảm số đơn vị của lớp. Vì mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng 1 bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước ma trận thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là tổng hợp.
* Quá trình kết nối hoàn toàn: Sau khi đã giảm số lượng tham số đến một mức độ hợp lý, ma trận cần được làm dẹt (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các lớp. Quá trình này sẽ diễn ra cuối mạng tích chập và sử dụng hàm kích hoạt là relu. Kết nối cuối cùng sẽ dẫn tới các đơn vị là đại diện cho mỗi lớp với hàm kích hoạt là softmax nhằm mục đích tính xác suất.

##### **3.3.2.2. Vì sao chọn mô hình mạng nơ ron CNN cho bài toán?**

- Có thể xử lý dữ liệu hình ảnh: Nếu dữ liệu bao gồm các hình ảnh y khoa như hình ảnh chụp MRI hoặc hình ảnh siêu âm, CNN được thiết kế đặc biệt để xử lý và rút trích đặc trưng từ dữ liệu hình ảnh.

- Kết hợp dữ liệu hình ảnh và dữ liệu số liệu: Trong một số trường hợp, dữ liệu y tế có thể bao gồm cả hình ảnh và dữ liệu số liệu như dữ liệu xét nghiệm. CNN có thể được sử dụng để xử lý và trích xuất đặc trưng từ dữ liệu hình ảnh, sau đó kết hợp với dữ liệu số liệu để dự đoán bệnh tim.

- Hiệu suất cao trên tập dữ liệu lớn: Khi được huấn luyện trên tập dữ liệu lớn, CNN thường có khả năng học và tổng hợp thông tin từ dữ liệu hiệu quả, dẫn đến hiệu suất dự đoán cao.

- CNN có thể được thiết kế nhiều lớp và kiến trúc khác nhau để phù hợp với nhiều loại dữ liệu và yêu cầu của bài toán.

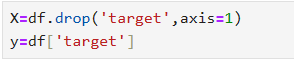
##### **3.3.2.3. Áp dụng cơ sở lý thuyết vào xây dựng mô hình cho bài toán**

Sử dụng mô hình mạng nơ ron CNN

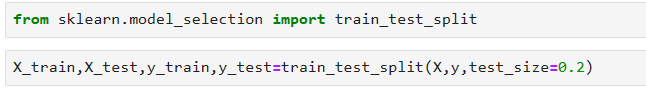
* Đọc dữ liệu từ file CSV



* Phân chia dữ liệu thành features và target



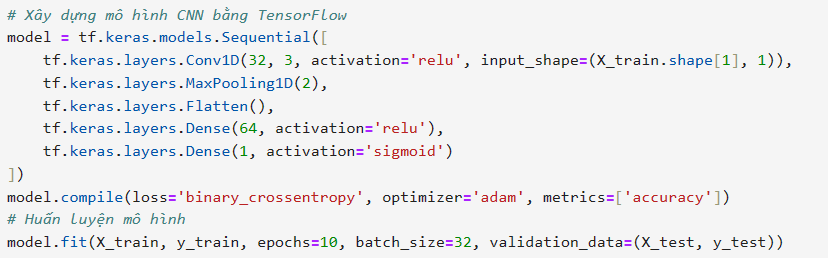
* Chia dữ liệu thành tập training và tập test (theo tỷ lệ 8:2)
* Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra



* Định hình lại dữ liệu CNN



* Khởi tạo và huấn luyện mô hình CNN



* Lưu mô hình vào file



### **3.4. Đánh giá mô hình học máy**

Đánh giá mô hình thông qua ma trận nhầm lẫn (Confusing Matrix).

* Ma trận nhầm lẫn (confusion matrix) là công cụ trực quan và hiệu quả để đánh giá hiệu suất của mô hình phân loại. Nó thể hiện rõ ràng mức độ chính xác của mô hình trong việc dự đoán các lớp (classes) khác nhau.
* Phương pháp đánh giá: Tính toán các thước đo

Accuracy: Tỉ lệ giữa số điểm dữ liệu được dự đoán đúng, tổng số điểm dữ liệu.

Precision: tỉ lệ số điểm true positive trong số những điểm được phân loại là positive.

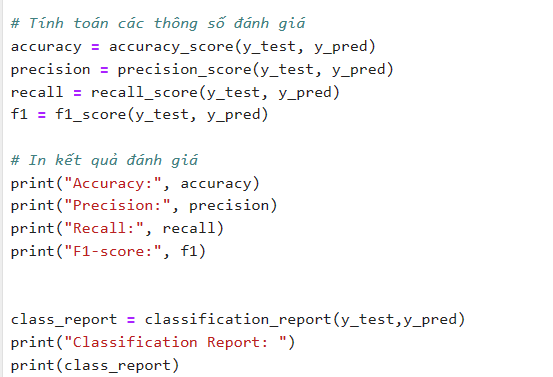
Recall: tỉ lệ số điểm true positive trong số những điểm thực sự là positive.

F1 Score: Điểm cân bằng giữa precision và recall.

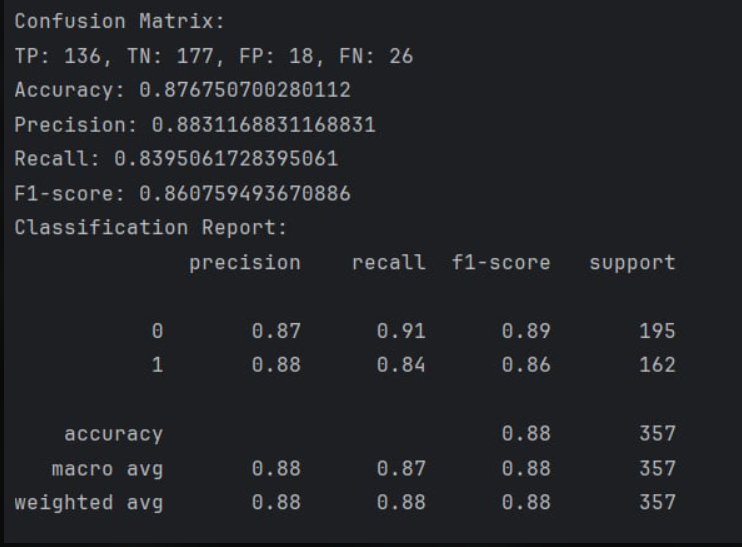
* Phân tích các thước đo trong ma trận: Một mô hình tốt khi cả Precision và Recall đều cao có nghĩa là ít phân loại nhầm và tỷ lệ bỏ sót thấp.
* Công thức tính:
* Accuracy = (TP+TN)/(TP+TN+FP+FN)
* Precision = TP/(TP+ FP)
* Recall = TP/(TP+ FN)
* Accuracy = 2 \* (Precision\*Recall )/(Precision+Recall)

Trong đó:

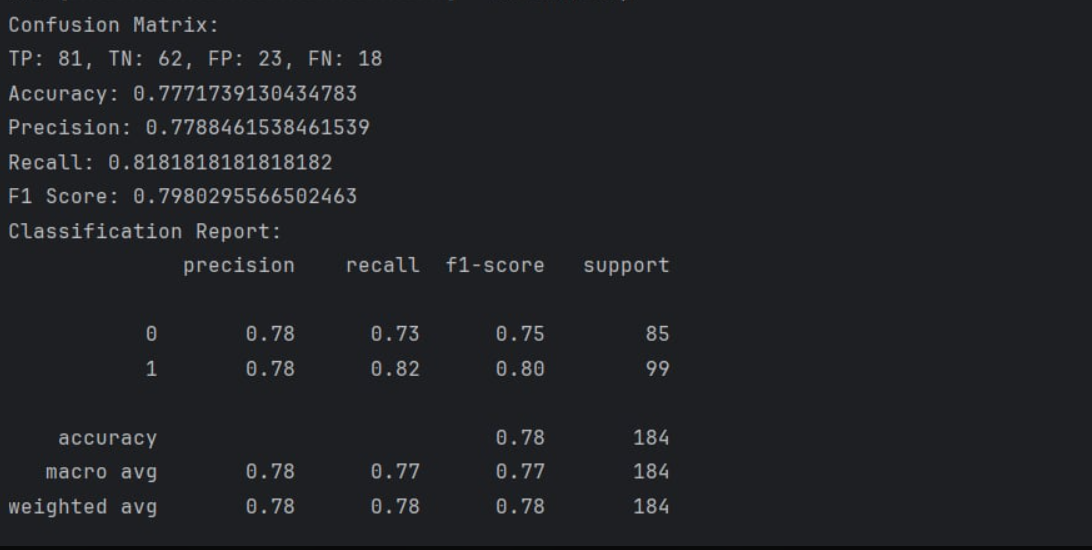
* TP (True Positive): Số lượng các trường hợp mà mô hình dự đoán đúng là positive (dương tính) và thực tế cũng là positive.
* TN (True Negative): Số lượng các trường hợp mà mô hình dự đoán đúng là negative và thực tế cũng là negative.
* FP (False Positive): Số lượng các trường hợp mà mô hình dự đoán là positive nhưng thực tế lại là negative (âm tính).
* FN (False Negative): Số lượng các trường hợp mà mô hình dự đoán là negative nhưng thực tế lại là positive.
* Đánh giá mô hình:



* Đánh giá mô hình mạng nơ ron học sâu MLP:
* Hiệu suất: 0.876750700280112
* Ma trận nhầm lẫn Confusion Matrix:



* Đánh giá mô hình mạng nơ-ron học sâu CNN:
* Hiệu suất Accuracy : 0.7065217391304348
* Ma trận nhầm lẫn Confusion Matrix:



* So sánh 2 mô hình

|  |  |  |
| --- | --- | --- |
|  | MLP | CNN |
| Accuracy | 87.6% | 77.7% |
| Precision | 88% | 77,8% |
| Recall | 83,9% | 81,8% |
| F1 – Score | 86% | 79,8% |

**Nhận xét:** Mô hình MLP có độ chính xác, Precision, Recall, F1-Score đều cao hơn so với CNN. Điều này cho thấy MLP có khả năng phân loại chính xác hơn trên tập dữ liệu kiểm tra.

* + - * Mô hình MLP có hiệu suất tốt hơn cho bài toán đặt ra

### **3.5. Tích hợp mô hình máy học vào ứng dụng**

##### **3.5.1. Mô tả**

* Ứng dụng có 11 input để người dùng nhập dữ liệu.
* Các input được làm để người dùng thuận tiện trong việc nhập dữ liệu liên quan.
* Đánh giá được người bệnh có bị bệnh.

##### **3.5.2. Triển khai**

import tkinter as tk

from tkinter import ttk

import joblib

import numpy as np

*# Tải mô hình được đào tạo trước*

neural\_network\_model = joblib.load('D:/New folder/Python/Data/DataBTL/neural\_network\_model.pkl')

*# Chức năng lấy giá trị thanh trượt*

def get\_slider\_value(slider, var):

value = float(slider.get())

var.set(f"{value}")

*# Chức năng dự đoán bệnh tim dựa vào thông tin đầu vào của người dùng*

def predict\_diabetes\_disease():

age = int(age\_entry.get())

sex = int(sex\_entry.get())

chest\_pain\_type = int(chest\_pain\_entry.get())

resting\_bp\_s = int(resting\_bp\_s\_entry.get())

cholesterol = int(cholesterol\_entry.get())

fasting\_blood\_sugar = int(fasting\_blood\_sugar\_entry.get())

resting\_ecg = int(resting\_ecg\_entry.get())

max\_heart\_rate = int(max\_heart\_rate\_entry.get())

exercise\_angina = int(exercise\_angina\_entry.get())

oldpeak = float(oldpeak\_entry.get())

ST\_slope = int(ST\_slope\_entry.get())

input\_data = np.array([[age, sex, chest\_pain\_type, resting\_bp\_s, cholesterol, fasting\_blood\_sugar, resting\_ecg, max\_heart\_rate, exercise\_angina, oldpeak, ST\_slope]])

prediction = neural\_network\_model.predict(input\_data)

result\_label.config(text=f'Kết Quả Dự Đoán: {"Bị bệnh tim" if prediction == 1 else "Không bị bệnh tim"}')

*# Chức năng thiết lập lại biểu mẫu*

def reset\_form():

result\_label.config(text='Kết Quả Dự Đoán: ')

age\_entry.delete(0, 'end')

sex\_entry.delete(0, 'end')

chest\_pain\_entry.delete(0, 'end')

resting\_bp\_s\_entry.delete(0, 'end')

cholesterol\_entry.delete(0, 'end')

fasting\_blood\_sugar\_entry.delete(0, 'end')

resting\_ecg\_entry.delete(0, 'end')

max\_heart\_rate\_entry.delete(0, 'end')

exercise\_angina\_entry.delete(0, 'end')

oldpeak\_entry.delete(0, 'end')

ST\_slope\_entry.delete(0, 'end')

root = tk.Tk()

root.title("Dự đoán bệnh tim")

*# Widget để nhập dữ liệu*

age\_label = ttk.Label(root, text="Tuổi:")

age\_label.grid(row=0, column=0, padx=10, pady=5, sticky="w")

age\_entry = ttk.Entry(root)

age\_entry.grid(row=0, column=1, padx=10, pady=5)

sex\_label = ttk.Label(root, text="Giới tính(Nữ-0, Nam-1):")

sex\_label.grid(row=1, column=0, padx=10, pady=5, sticky="w")

sex\_entry = ttk.Entry(root)

sex\_entry.grid(row=1, column=1, padx=10, pady=5)

chest\_pain\_label = ttk.Label(root, text="Loại đau ngực(1-4):")

chest\_pain\_label.grid(row=2, column=0, padx=10, pady=5, sticky="w")

chest\_pain\_entry = ttk.Entry(root)

chest\_pain\_entry.grid(row=2, column=1, padx=10, pady=5)

resting\_bp\_s\_label = ttk.Label(root, text="Huyết áp nghỉ(mm/Hg):")

resting\_bp\_s\_label.grid(row=3, column=0, padx=10, pady=5, sticky="w")

resting\_bp\_s\_entry = ttk.Entry(root)

resting\_bp\_s\_entry.grid(row=3, column=1, padx=10, pady=5)

cholesterol\_label = ttk.Label(root, text="Cholesterol(mg/dl):")

cholesterol\_label.grid(row=4, column=0, padx=10, pady=5, sticky="w")

cholesterol\_entry = ttk.Entry(root)

cholesterol\_entry.grid(row=4, column=1, padx=10, pady=5)

fasting\_blood\_sugar\_label = ttk.Label(root, text="Đường huyết nhanh(1 nếu > 120 mg/dl, 0 nếu không):")

fasting\_blood\_sugar\_label.grid(row=5, column=0, padx=10, pady=5, sticky="w")

fasting\_blood\_sugar\_entry = ttk.Entry(root)

fasting\_blood\_sugar\_entry.grid(row=5, column=1, padx=10, pady=5)

resting\_ecg\_label = ttk.Label(root, text="Điện tâm đồ nghỉ(0-2):")

resting\_ecg\_label.grid(row=6, column=0, padx=10, pady=5, sticky="w")

resting\_ecg\_entry = ttk.Entry(root)

resting\_ecg\_entry.grid(row=6, column=1, padx=10, pady=5)

max\_heart\_rate\_label = ttk.Label(root, text="Tần số tim tối đa:")

max\_heart\_rate\_label.grid(row=7, column=0, padx=10, pady=5, sticky="w")

max\_heart\_rate\_entry = ttk.Entry(root)

max\_heart\_rate\_entry.grid(row=7, column=1, padx=10, pady=5)

exercise\_angina\_label = ttk.Label(root, text="Đau ngực do tập thể dục (1 nếu có, 0 nếu không):")

exercise\_angina\_label.grid(row=8, column=0, padx=10, pady=5, sticky="w")

exercise\_angina\_entry = ttk.Entry(root)

exercise\_angina\_entry.grid(row=8, column=1, padx=10, pady=5)

oldpeak\_label = ttk.Label(root, text="Giảm đỉnh ST gây ra bởi tập thể dục so với nghỉ ngơi:")

oldpeak\_label.grid(row=9, column=0, padx=10, pady=5, sticky="w")

oldpeak\_entry = ttk.Entry(root)

oldpeak\_entry.grid(row=9, column=1, padx=10, pady=5)

ST\_slope\_label = ttk.Label(root, text="Giai đoạn ST của đỉnh tập thể dục:")

ST\_slope\_label.grid(row=10, column=0, padx=10, pady=5, sticky="w")

ST\_slope\_entry = ttk.Entry(root)

ST\_slope\_entry.grid(row=10, column=1, padx=10, pady=5)

*# Reset button*

reset\_button = ttk.Button(root, text="Reset", command=reset\_form)

reset\_button.grid(row=11, column=0, columnspan=2, padx=10, pady=5, sticky="we")

*# Prediction button*

predict\_button = ttk.Button(root, text="Submit", command=predict\_diabetes\_disease)

predict\_button.grid(row=12, column=0, columnspan=2, padx=10, pady=5, sticky="we")

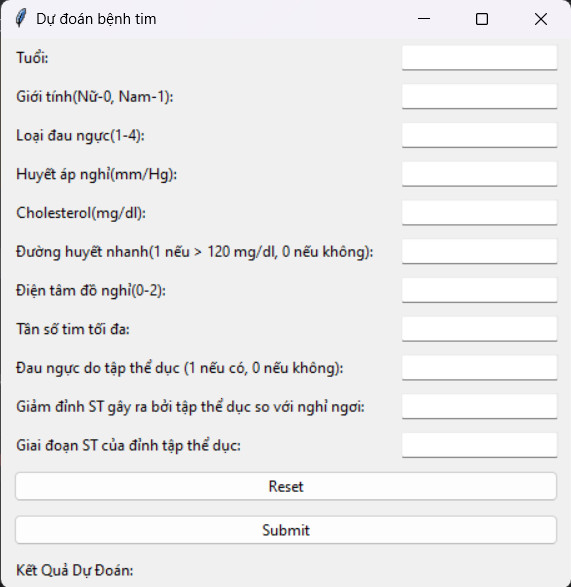
*# Result label*

result\_label = ttk.Label(root, text="Kết Quả Dự Đoán: ")

result\_label.grid(row=13 ,column=0, columnspan=2, padx=10, pady=5, sticky="we")

root.mainloop()

##### **3.5.3. Kết quả**

****

### **3.6. Hiệu suất**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | sex | chest pain type | resting bp s | cholesterol | fasting blood sugar | resting ecg | max heart rate | exercise angina | oldpeak | ST slope | target | target |
| 40 | 1 | 2 | 140 | 289 | 0 | 0 | 172 | 0 | 0 | 1 | 0 | 0 |
| 49 | 0 | 3 | 160 | 180 | 0 | 0 | 156 | 0 | 1 | 2 | 1 | 0 |
| 37 | 1 | 2 | 130 | 283 | 0 | 1 | 98 | 0 | 0 | 1 | 0 | 0 |
| 48 | 0 | 4 | 138 | 214 | 0 | 0 | 108 | 1 | 1.5 | 2 | 1 | 1 |
| 54 | 1 | 3 | 150 | 195 | 0 | 0 | 122 | 0 | 0 | 1 | 0 | 0 |
| 39 | 1 | 3 | 120 | 339 | 0 | 0 | 170 | 0 | 0 | 1 | 0 | 0 |
| 45 | 0 | 2 | 130 | 237 | 0 | 0 | 170 | 0 | 0 | 1 | 0 | 0 |
| 54 | 1 | 2 | 110 | 208 | 0 | 0 | 142 | 0 | 0 | 1 | 0 | 0 |
| 37 | 1 | 4 | 140 | 207 | 0 | 0 | 130 | 1 | 1.5 | 2 | 1 | 1 |
| 48 | 0 | 2 | 120 | 284 | 0 | 0 | 120 | 0 | 0 | 1 | 0 | 0 |

🡺 Sau khi test 10 trường hợp thì kết quả dự đoán đúng 9 sai 1

Hiệu suất đạt được đến 90%

### **3.7. Kết luận**

Trong quá trình làm bài tập lớn, nhóm đã thực hiện các công việc sau:

* Nghiên cứu về các phương pháp học máy có giám sát.
* Áp dụng các giai đoạn của chu trình học máy vào bài toán thực tế
* Thực hiện đánh giá, so sánh 2 mô hình học để lựa chọn mô hình học máy tốt nhất
* Tích hợp được mô hình học máy vào ứng dụng để giải quyết các vấn đề trong lĩnh vực nghiên cứu y tế.

Trong tương lai, nhóm sẽ tiến hành cải tiến mô hình, nghiên cứu các thuật toán mới để tiến hành tăng hiệu suất mô hình, dưới đây là đề xuất phương án cải tiến trong tương lai của nhóm

* Tăng kích thước tập dữ liệu sẽ giúp hiệu suất thực tế tốt hơn, cải thiện FPR để giảm xuống dưới 9%
* Thu thập thêm dữ liệu từ nhiều nguồn dữ liệu (tăng thuộc tính) để cải thiện hiệu suất của mô hình

# **Tài Liệu Tham Khảo**

[1] *pandas.DataFrame.drop\_duplicates — pandas 2.2.2 documentation*. (n.d.).

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html>

[2] GfG. (2024, March 12). Python Pandas dataframe.drop duplicates(). GeeksforGeeks. <https://www.geeksforgeeks.org/python-pandas-dataframe-drop_duplicates/>

[3] Sklearn.impute.SimpleImputer. (n.d.). Scikit-learn.

<https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html>

[4] L6-Hoc\_dua\_tren\_Xac\_Suat.pdf. (n.d.). Google Docs.

<https://drive.google.com/file/d/1_Gsz68GAulRJUzaGIgq93B70PVHYSuPC/view>

[6] *Deep Learning Framework Power Scores 2018*. URL:

[https://towardsdatascience. com / deep - learning - framework - power - scores - 2018 - 23607ddf297a.](about:blank)

[7] *Convolutional Neural Networks (CNNs / ConvNets)*. URL:

[https://www.qubole.com/ blog/deep-learning-the-latest-trend-in-ai-and-ml](https://www.qubole.com/%20blog/deep-learning-the-latest-trend-in-ai-and-ml%20%0D%0Acom/transfer-learning-for-deep-learning/)

[com/transfer-learning-for-deep-learning/](https://www.qubole.com/%20blog/deep-learning-the-latest-trend-in-ai-and-ml%20%0D%0Acom/transfer-learning-for-deep-learning/)

[8] (n.d.). Elcom Corporation.

<https://www.elcom.com.vn/mang-no-ron-nhan-tao-la-gi-tai-sao-chung-ta-can-mang-no-ron-nhan-tao-1669018888>

[9] *A Gentle Introduction to Transfer Learning for Deep Learning*. URL: <https://machinelearningmastery>