

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**      **BỘ NÔNG NGHIỆP VÀ PTNT**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**PHẠM VĂN TIẾN**

**ỨNG DỤNG THUẬT TOÁN AI VÀO DỰ ĐOÁN SỐM NGUY  
CƠ ĐỘT QUÝ Ở NGƯỜI GIÀ**

**ĐỒ ÁN TỐT NGHIỆP**

HÀ NỘI, NĂM 2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**      **BỘ NÔNG NGHIỆP VÀ PTNT**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**PHẠM VĂN TIẾN**

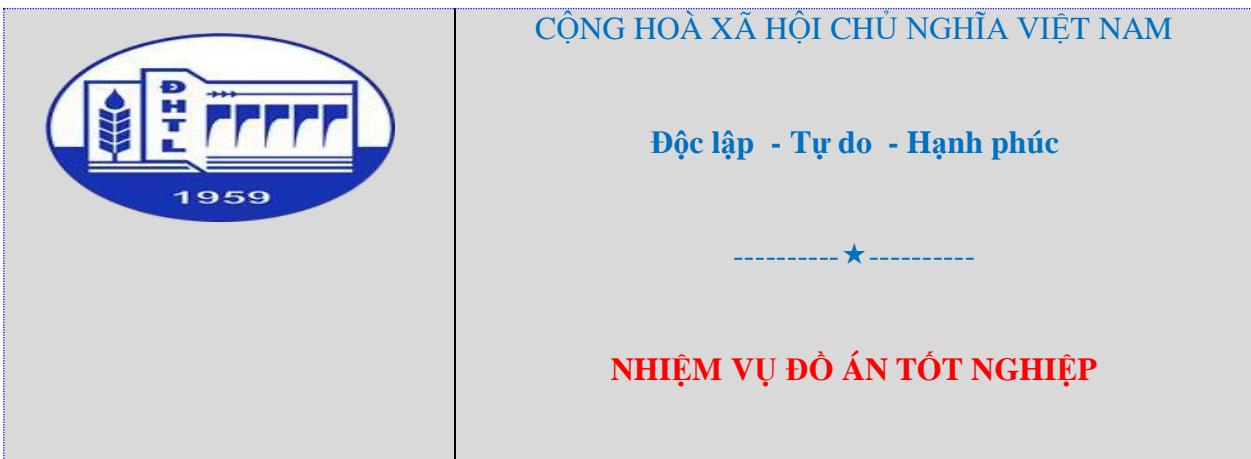
**ỨNG DỤNG THUẬT TOÁN AI VÀO DỰ ĐOÁN SỐM NGUY  
CƠ ĐỘT QUÝ Ở NGƯỜI GIÀ**

Ngành : Công nghệ thông tin

Mã số: 7480201

**NGƯỜI HƯỚNG DẪN: TS. NGUYỄN VĂN THẮNG**

**HÀ NỘI, NĂM 2022**



Họ tên sinh viên: **PHẠM VĂN TIẾN**

Hệ đào tạo: **Đại học chính quy**

Lớp: **60TH2**

Ngành: **Công nghệ thông tin**

Khoa: **Công nghệ thông tin**

### **1. TÊN ĐỀ TÀI:**

## **ỨNG DỤNG THUẬT TOÁN AI VÀO DỰ ĐOÁN SỐM NGUY CƠ ĐỘT QUÝ Ở NGƯỜI GIÀ**

### **2. CÁC TÀI LIỆU CƠ BẢN:**

[1] <https://machinelearningcoban.com/2017/04/09/sm/>

[2] <https://miae.vn/2021/01/18/k-fold-cross-validation-tuyet-chieu-train-khi-it-du-lieu/>

[3] Tham khảo slide “Học máy” của cô TS.Nguyễn Thị Kim Ngân.

### **3. NỘI DUNG CÁC PHẦN THUYẾT MINH:**

- **Chương 1: Tổng quan về học máy và trí tuệ nhân tạo AI.**

- **Chương 2: Thuật toán áp dụng và mô tả bài toán.**
- **Chương 3: Xây dựng hệ thống và đánh giá mô hình.**

#### **4. GIÁO VIÊN HƯỚNG DẪN TÙNG PHẦN**

Giáo viên hướng dẫn toàn bộ quá trình thực hiện đồ án: TS. Nguyễn Văn Thắng

#### **5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày .....tháng ..... năm 2022

**Trưởng Bộ môn**

(Ký và ghi rõ Họ tên)

**Giáo viên hướng dẫn chính**

(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . . . tháng. . . . . năm 2022

**Chủ tịch Hội đồng**

(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày ... tháng ... năm 2022

**Sinh viên làm Đồ án tốt nghiệp**

(Ký và ghi rõ Họ tên)

## **LỜI CAM ĐOAN**

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

**Tác giả ĐATN**

**PHẠM VĂN TIẾN**

## LỜI CẢM ƠN

Sau hơn 4 năm học tập và nghiên cứu tại Khoa Công nghệ thông tin - Trường Đại học Thủy Lợi, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè và đã được trải nghiệm trong môi trường đào tạo tốt và nhận được sự chỉ dạy nhiệt tình của các thầy, các cô trong khoa.

Trước hết, em xin được bày tỏ lòng biết ơn và gửi lời cảm ơn chân thành đến TS. Nguyễn Văn Thắng đã tận tình chỉ bảo, nhắc nhở và hướng dẫn em trong suốt quá trình làm đồ án tốt nghiệp.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin nói riêng và trường Đại học Thủy Lợi nói chung đã trang bị cho em những kiến thức quý báu làm hành trang trong những năm học vừa qua.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến Cha mẹ và những người thân trong gia đình đã chăm sóc, nuôi dạy, hỗ trợ, động viên và tạo mọi điều kiện thuận lợi nhất cho em trong suốt thời gian qua và đặc biệt trong thời gian em làm đồ án tốt nghiệp.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên mặc dù trong quá trình nghiên cứu được sự hướng dẫn nhiệt tình của thầy Nguyễn Văn Thắng cùng với sự nỗ lực của cá nhân nhưng chắc chắn không tránh khỏi thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn với các dự án thực tế sau này.

Em xin chân thành cảm ơn!

## MỤC LỤC

CHƯƠNG I. TỔNG QUAN VỀ HỌC MÁY VÀ TRÍ TUỆ NHÂN TẠO AI.....	1
1.1    Tổng quan về học máy và trí tuệ nhân tạo AI.....	1
1.1.1 Giới thiệu về học máy .....	1
1.1.2 Phân loại các phương pháp học máy.....	2
1.1.3 Những cột mốc quan trọng. ....	6
1.1.4 Ứng dụng học máy .. ..	8
1.2    Giới thiệu về bài toán phân lớp.....	10
1.2.1    Giới thiệu bài toán.....	10
1.2.2    Phân loại bài toán phân lớp .....	11
1.2.3    Thuật toán tiêu biểu của bài toán phân lớp .....	14
CHƯƠNG II. THUẬT TOÁN ÁP DỤNG VÀ MÔ TẢ BÀI TOÁN.....	20
2.1 Mô tả bài toán:.....	20
2.1.1 Định nghĩa bệnh đột quy: .....	20
2.1.2 Triệu chứng bệnh đột quy:.....	21
2.1.3 Các biến chứng có thể gặp sau khi đột quy: .....	22
2.2 Mô tả dữ liệu bài toán.....	23
2.2.1 Thu thập dữ liệu: .....	23
2.2.2 Mô tả dữ liệu bài toán.....	31
2.3 Thuật toán áp dụng .....	34
2.3.1 Phương pháp K-fold cross validation để cải thiện dữ liệu .....	35
2.3.2 Thuật toán Support Vector Machines (SVM).....	38
CHƯƠNG 3. XÂY DỰNG HỆ THỐNG VÀ ĐÁNH GIÁ MÔ HÌNH.....	45
3.1 Xây dựng hệ thống .....	45
3.1.1 Ngôn ngữ lập trình được sử dụng trong xây dựng hệ thống .....	45

3.1.2 Phần mềm được sử dụng trong xây dựng hệ thống .....	47
3.1.3 Các bước thực hiện xây dựng hệ thống: .....	51
3.2 Đánh giá và demo giao diện chương trình: .....	61
3.2.1. Form nhập dữ liệu kiểm tra: .....	62
3.2.2. Khả năng dự đoán: .....	62
3.2.3. Giao diện người dùng: .....	64
3.2.4. Đánh giá kết quả:.....	64
3.2.5 . Nhận xét:.....	65
KẾT LUẬN .....	68
TÀI LIỆU THAM KHẢO .....	69

## BẢNG THUẬT NGỮ

<b>Thuật ngữ</b>	<b>Giải thích</b>
Artificial Intelligence (AI)	Trí tuệ nhân tạo
Machine Learning (ML)	Học máy
Support vector machine(SVM)	Thuật toán giám sát
Supervised learning	Học có giám sát
Unsupervised learning	Học không giám sát
Semi-Supervised Learning	Học bán giám sát
Reinforcement learning	Học củng cố
Principal Component Analysis (PCA)	Thuật toán giảm chiều dữ liệu
Perceptron Learning Algorithm(PLA)	Thuật toán perceptron.
K-fold cross validation	Kỹ thuật lấy mẫu
k-Nearest Neighbors	Thuật toán lảng giềng gần nhất
Dimension reduction	Giảm số chiều của dữ liệu
Pattern recognition	Nhận dạng mẫu
Naive Bayes	Thuật toán phân loại nhị phân
Random Forest.	Thuật toán phân lớp phô biến
Labeled data	Dữ liệu được gán nhãn
Logistic Regression	Thuật toán hồi quy logistic
Gradient Boosting	Thuật toán phân lớp phô biến
Clustering	Phân nhóm
Multinoulli	Phối xác suất rời rạc
Chỉ Số Khối	Cân nặng chia chiều cao bình phương.

Dường Máu	Chỉ số Glucose xác định bệnh tiểu đường.
Cholesterol	Chỉ số cholesterol xác định lipid trong máu.
Triglycerid.	Chỉ số mỡ máu.
RANKIN	Thang điểm nhận thức .
ICA	Phân tích thành phần động lập
K	Số phần được chia trong kỹ thuật lấy mẫu K-fold
Model	Mô hình
All data	Tất cả dữ liệu
Train	Huấn luyện
Predict	Dự đoán
Test set	Dữ liệu thử nghiệm trong K-fold cross validation
Training data	Dữ liệu huấn luyện
Data mining	Khai phá dữ liệu
Test data	Dữ liệu kiểm thử

## **DANH MỤC CÁC HÌNH ẢNH**

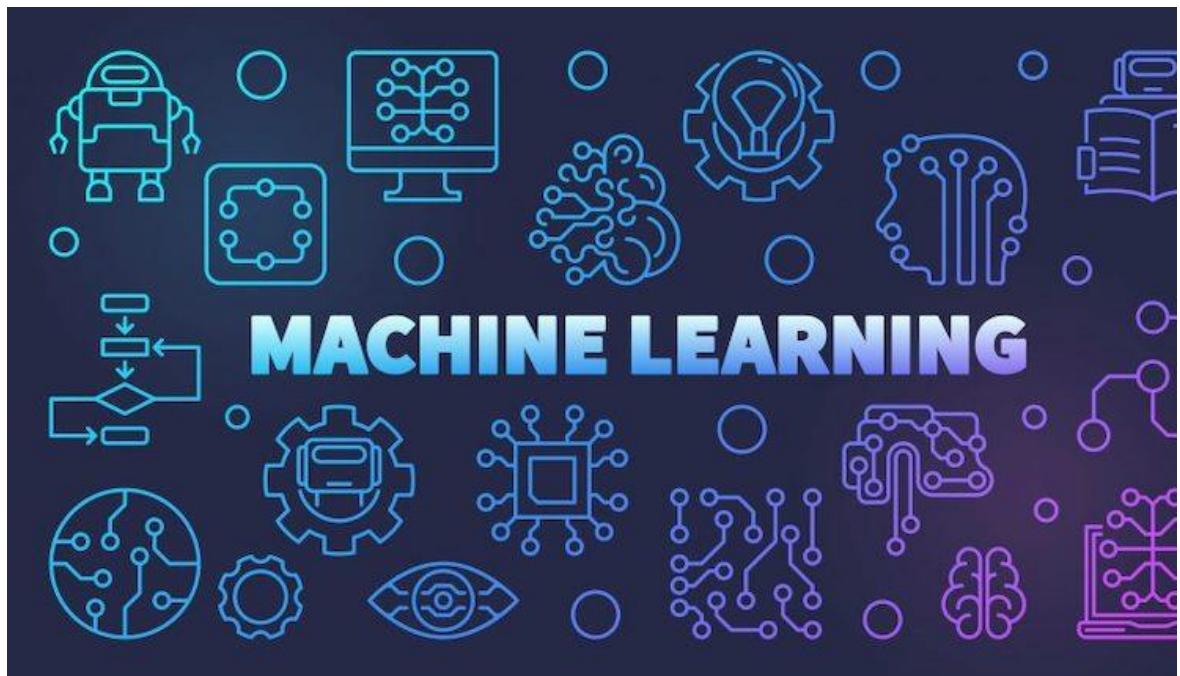
Hình 1. 1 Hình minh họa học máy .....	1
Hình 1. 2 Hình minh họa học máy có giám sát.....	3
Hình 1. 3 Hình minh họa học máy không giám sát.....	4
Hình 1. 4 Hình minh họa học máy bán giám sát.....	5
Hình 1. 5 Hình minh họa học máy bán giám sát.....	5
Hình 1. 6 Hình minh họa ứng dụng học máy.....	10
Hình 1. 7 Hình minh họa bài toán phân lớp.....	11
Hình 1. 8 Hình minh họa bài toán phân lớp nhị phân .....	12
Hình 1. 9 Hình minh họa bài toán phân lớp nhiều lớp .....	13
Hình 1. 10 Hình minh họa Hồi quy logistic.....	15
Hình 1. 11 Hình minh họa thuật toán svm.....	16
Hình 1. 12 Hình minh họa thuật toán Naive Bayes.....	17
Hình 1. 13 Hình minh họa phân tích thành phần độc lập .....	18
Hình 1. 14 Hình minh họa thuật toán k-Nearest Neighbors .....	19
Hình 2. 1 Hình minh họa bệnh đột quy .....	20
Hình 2. 2 Hình minh họa triệu chứng bệnh đột quy.....	21
Hình 2. 3 Hình minh họa dữ liệu bệnh nhân đột quy gốc.....	24
Hình 2. 4 Hình minh họa phiếu xét nghiệm định kỳ .....	26
Hình 2. 5 Hình minh họa dữ liệu bệnh nhân đột quy sau khi chọn lọc các thuộc tính .	27
Hình 2. 6 Hình mẫu báo cáo khám sức khỏe định kỳ tại bệnh viện Đa Khoa Sóc Sơn Hà Nội .....	28
Hình 2. 7 Hình minh họa dữ liệu bệnh nhân khỏe mạnh sau khi chọn lọc và tập hợp .	30
Hình 2. 8 Hình minh họa sử dụng Kutools để xáo trộn dữ liệu .....	30
Hình 2. 9 Hình minh họa kiểu dữ liệu của từng thuộc tính .....	32
Hình 2. 10 Hình minh họa kiểu dữ liệu int của thuộc tính đầu ra.....	32
Hình 2. 11 Hình minh họa đầu vào (input) của bài toán .....	33
Hình 2. 12 Hình minh họa đầu ra (output) của bài toán .....	34
Hình 2. 13 Hình minh họa svm và k-fold cross validation .....	35
Hình 2. 14 Hình minh họa Phương pháp K-fold cross validation .....	36

Hình 2. 15 Hình minh họa siêu phẳng trong svm. ....	39
Hình 2. 16 Hình minh họa hai lớp dữ liệu đỏ và xanh. Có vô số các đường thẳng có thể phân tách chính xác hai lớp dữ liệu này.....	41
Hình 2. 17 Hình minh họa xây dựng bài toán tối ưu cho SVM .....	42
Hình 3. 1 Hình ảnh minh họa ngôn ngữ lập trình python.....	45
Hình 3. 2 Hình ảnh minh họa html css .....	46
Hình 3. 3 Hình ảnh minh họa phần mềm lập trình Python 3.10 .....	47
Hình 3. 4 Hình ảnh minh họa phần mềm lập trình Visual Studio Code.....	48
Hình 3. 5 Hình ảnh minh họa phần mềm phát triển sản phẩm HEROKU .....	49
Hình 3. 6 Hình ảnh minh họa phần mềm lưu trữ mã nguồn GitHub .....	50
Hình 3. 7 Hình ảnh minh họa phần mềm điều phối lượng truy cập giữa máy chủ CloudFlare .....	51
Hình 3. 8 Hình ảnh các thư viện được sử dụng trong bài toán .....	52
Hình 3. 9 Hình ảnh sơ đồ hoạt động của tiền xử lý dữ liệu đầu vào .....	53
Hình 3. 10 Hình ảnh sơ đồ hoạt động của thuật toán .....	56
Hình 3. 11 Hình ảnh sơ đồ hoạt động của form dữ liệu .....	60
Hình 3. 12 Hình ảnh sơ đồ hoạt động của thuật toán kết nối giao diện .....	61
Hình 3. 13 Hình minh họa form nhập dữ liệu đầu vào bài toán .....	62
Hình 3. 14 Hình minh họa hoạt kết quả dự đoán .....	63
Hình 3. 15 Hình minh họa dữ liệu đầu và kết quả dự đoán tại form.....	63
Hình 3. 16 Hình minh họa giao diện người dùng.....	64
Hình 3. 17 Hình minh họa độ chính xác của thuật toán .....	65
Hình 3. 18 Hình minh họa độ chính xác của thuật toán svm.....	65
Hình 3. 19 Hình minh họa độ chính xác của thuật toán K-fold cross validation kết hợp SVM .....	66
Hình 3. 20 Hình minh họa độ chính xác của thuật toán SVM kết hợp xáo trộn dữ liệu .....	66
Hình 3. 21 Hình minh họa độ chính xác kết hợp của ba phương pháp .....	67

# CHƯƠNG I. TỔNG QUAN VỀ HỌC MÁY VÀ TRÍ TUỆ NHÂN TẠO AI

## 1.1 Tổng quan về học máy và trí tuệ nhân tạo AI

### 1.1.1 Giới thiệu về học máy



Hình 1. 1 Hình minh họa học máy<sup>[1]</sup>

Những năm gần đây, lĩnh vực Trí tuệ nhân tạo hay Học máy ảnh hưởng mạnh mẽ tới nhiều lĩnh vực, gắn liền với nó là thuật ngữ công nghệ machine learning.

Trong suy nghĩ phổ biến, AI hay học máy thường được hiểu rằng máy móc có khả năng tiếp thu kiến thức và suy nghĩ, hành động giống như người. Điều đó có thực sự chính xác? Học máy là gì, vì sao các nhà khoa học lại đầu tư phát triển công nghệ này?

Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo. Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian. ML vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kỹ thuật phù hợp để phân tích dữ liệu.<sup>[2][3]</sup>

Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

### **1.1.2 Phân loại các phương pháp học máy**

Dựa theo phương pháp học tập, các phương pháp học máy được chia làm 4 loại ML chính bao gồm học có giám sát (supervised learning) và học không giám sát (unsupervised learning) và học bán giám sát (Semi-Supervised Learning), học củng cố (Reinforcement learning).<sup>[4]</sup>

**Học có giám sát (Supervised learning) :** Trong học có giám sát, máy tính học cách mô hình hóa các mối quan hệ dựa trên dữ liệu được gán nhãn (labeled data). Sau khi tìm hiểu cách tốt nhất để mô hình hóa các mối quan hệ cho dữ liệu được gán nhãn, các thuật toán được huấn luyện và được sử dụng cho các bộ dữ liệu mới.

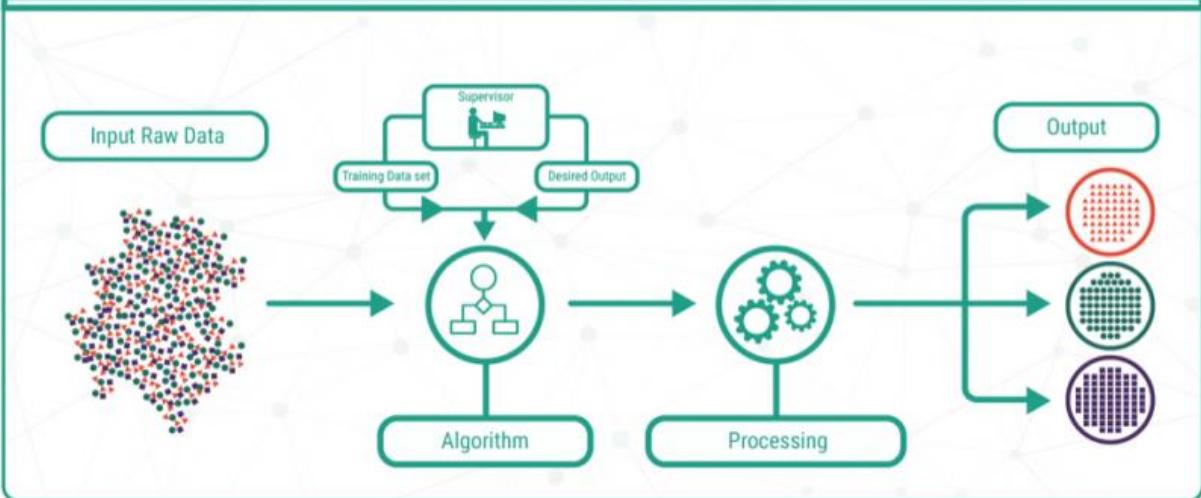
Học có giám sát là khi chúng ta có một tập hợp biến đầu vào:  $X = \{x_1, x_2, \dots, x_n\}$  và một tập hợp nhãn tương ứng  $Y = \{y_1, y_2, \dots, y_n\}$ , trong đó:  $x_i, y_i$  là các vector.

Các cặp dữ liệu biết trước  $(x_i, y_i) \in X \times Y$  được gọi là tập training data (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập  $X$  sang một phần tử (xấp xỉ) tương ứng của tập  $Y$ .

$$y_i \approx f(x_i), \quad \forall i=1,2,\dots,n$$

Mục đích là xấp xỉ hàm số  $f$  thật tốt để khi có một dữ liệu  $x$  mới, chúng ta có thể tính được nhãn tương ứng của nó  $y = f(x)$ <sup>[4]</sup>

# SUPERVISED LEARNING



Hình 1. 2 Hình minh họa học máy có giám sát<sup>[5]</sup>

Học không giám sát (unsupervised learning): Trong học không giám sát, máy tính không được cung cấp dữ liệu được gán nhãn mà thay vào đó chỉ được cung cấp dữ liệu mà thuật toán tìm cách mô tả dữ liệu và cấu trúc của chúng.

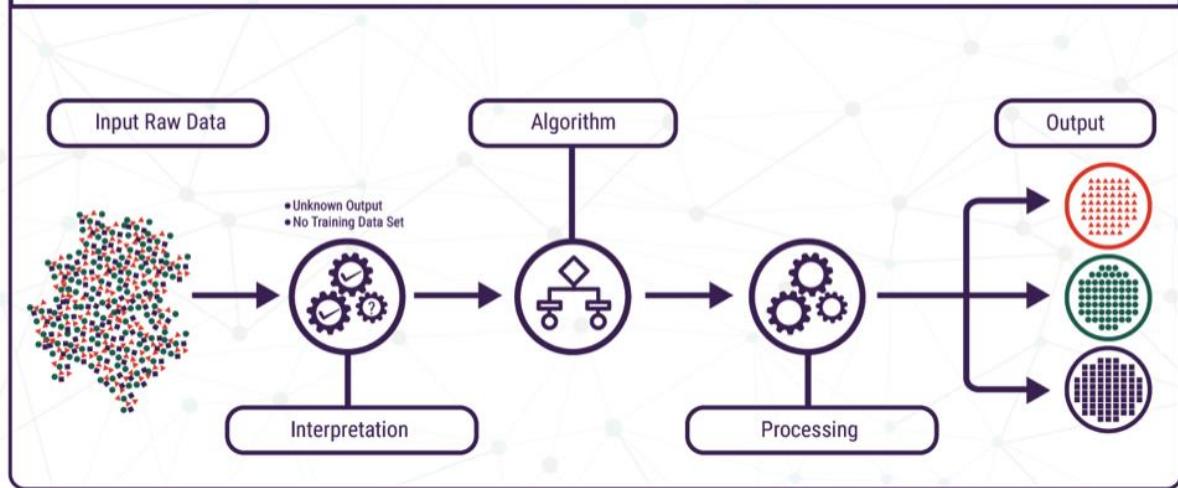
Trong thuật toán này, chúng ta không biết được nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào.

[4]

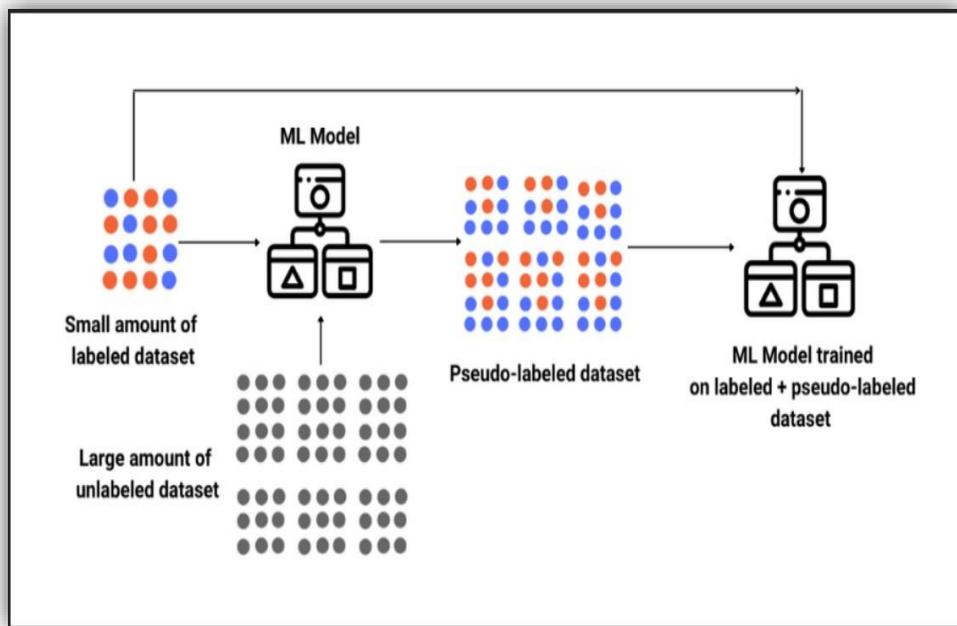
# UNSUPERVISED LEARNING



Hình 1. 3 Hình minh họa học máy không giám sát<sup>[5]</sup>

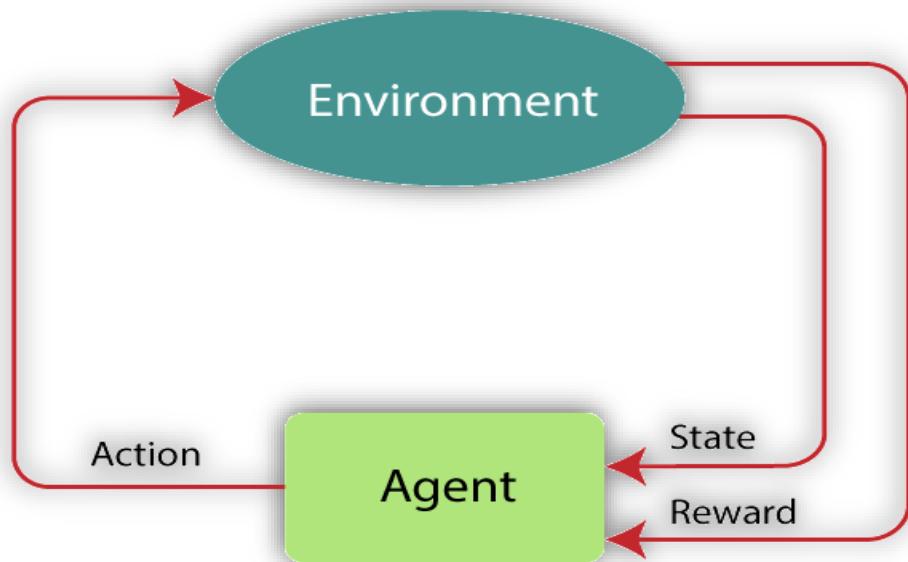
Học bán giám sát (Semi-Supervised Learning): Học nửa giám sát là một lớp của kỹ thuật học máy, sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện. Diễn hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn.

Các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.<sup>[4]</sup>



Hình 1. 4 Hình minh họa học máy bán giám sát<sup>[6]</sup>

Học củng cố (Reinforcement learning): là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất. <sup>[4]</sup>



Hình 1. 5 Hình minh họa học máy bán giám sát<sup>[7]</sup>

### **1.1.3 Những cột mốc quan trọng.**

**Năm 1950** – Nhà bác học Alan Turing đã tạo ra “Turing Test (phép thử Turing)” để xác định xem liệu một máy tính có trí thông minh thực sự hay không. Để vượt qua bài kiểm tra đó, một máy tính phải có khả năng đánh lừa một con người tin rằng nó cũng là con người.

**Năm 1952** – Arthur Samuel đã viết ra chương trình học máy (computer learning) đầu tiên. Chương trình này là trò chơi cờ đam, và hãng máy tính IBM đã cải tiến trò chơi này để nó có thể tự học và tổ chức những nước đi trong chiến lược để giành chiến thắng.

**Năm 1957** – Frank Rosenblatt đã thiết kế mạng nơron (neural network) đầu tiên cho máy tính, trong đó mô phỏng quá trình suy nghĩ của bộ não con người.

**Năm 1967** – Thuật toán “nearest neighbor” đã được viết, cho phép các máy tính bắt đầu sử dụng những mẫu nhận dạng (pattern recognition) rất cơ bản. Nó được sử dụng để vẽ ra lộ trình cho một người bán hàng có thể bắt đầu đi từ một thành phố ngẫu nhiên nhưng đảm bảo anh ta sẽ đi qua tất cả các thành phố khác theo một quãng đường ngắn nhất.

**Năm 1979** – Sinh viên tại trường đại học Stanford đã phát minh ra giỏ hàng “Stanford Cart” có thể điều hướng để tránh các chướng ngại vật trong một căn phòng.

**Năm 1981** – Gerald DeJong giới thiệu về khái niệm Explanation Based Learning (EBL), trong đó một máy tính phân tích dữ liệu huấn luyện và tạo ra một quy tắc chung để nó có thể làm theo bằng cách loại bỏ đi những dữ liệu không quan trọng.

**Năm 1985** – Terry Sejnowski đã phát minh ra NetTalk, nó có thể học cách phát âm các từ giống như cách một đứa trẻ tập nói.

**Năm 1990** – Machine Learning đã dịch chuyển từ cách tiếp cận hướng kiến thức (knowledge-driven) sang cách tiếp cận hướng dữ liệu (data-driven). Các nhà khoa học bắt đầu tạo ra các chương trình cho máy tính để phân tích một lượng lớn dữ liệu và rút ra các kết luận – hay là “học” từ các kết quả đó.

**Năm 1997** – Deep Blue của hãng IBM đã đánh bại nhà vô địch cờ vua thế giới.

**Năm 2006** – Geoffrey Hinton đã đưa ra một thuật ngữ “deep learning” để giải thích các thuật toán mới cho phép máy tính “nhìn thấy” và phân biệt các đối tượng và văn bản trong các hình ảnh và video.

**Năm 2010** – Microsoft Kinect có thể theo dõi 20 hành vi của con người ở một tốc độ 30 lần mỗi giây, cho phép con người tương tác với máy tính thông qua các hành động và cử chỉ.

**Năm 2011** – Máy tính Watson của hãng IBM đã đánh bại các đối thủ là con người tại Jeopardy.

**Năm 2011** – Google Brain đã được phát triển, và mạng deep nơron (deep neural network) của nó có thể học để phát hiện và phân loại nhiều đối tượng theo cách mà một con mèo thực hiện.

**Năm 2012** – X Lab của Google phát triển một thuật toán machine learning có khả năng tự động duyệt qua các video trên YouTube để xác định xem video nào có chứa những con mèo.

**Năm 2014** – Facebook phát triển DeepFace, một phần mềm thuật toán có thể nhận dạng hoặc xác minh các cá nhân dựa vào hình ảnh ở mức độ giống như con người có thể.

**Năm 2015** – Amazon ra mắt nền tảng machine learning riêng của mình.

**Năm 2015** – Microsoft tạo ra Distributed Machine Learning Toolkit, trong đó cho phép phân phối hiệu quả các vấn đề machine learning trên nhiều máy tính.

**Năm 2015** – Hơn 3.000 nhà nghiên cứu AI và Robotics, được sự ủng hộ bởi những nhà khoa học nổi tiếng như Stephen Hawking, Elon Musk và Steve Wozniak (và nhiều người khác), đã ký vào một bức thư ngỏ để cảnh báo về sự nguy hiểm của vũ khí tự động trong việc lựa chọn và tham gia vào các mục tiêu mà không có sự can thiệp của con người.

**Năm 2016** – Thuật toán trí tuệ nhân tạo của Google đã đánh bại nhà vô địch trò chơi Cờ Vây, được cho là trò chơi phức tạp nhất thế giới (khó hơn trò chơi cờ vua rất nhiều). Thuật toán AlphaGo được phát triển bởi Google DeepMind đã giành chiến thắng 4/5 trước nhà vô địch Cờ Vây.<sup>[8]</sup>

#### **1.1.4 *Ứng dụng học máy :***

Có rất nhiều ứng dụng thực tế khác nhau của học máy. Hai lĩnh vực ứng dụng lớn nhất của học máy là khai phá dữ liệu (data mining) và nhận dạng mẫu (pattern recognition).

Khai phá dữ liệu là: ứng dụng kỹ thuật học máy vào các cơ sở dữ liệu hoặc các tập dữ liệu lớn để phát hiện quy luật hay tri thức trong dữ liệu đó hoặc để dự đoán các thông tin quan tâm trong tương lai. Ví dụ, từ tập hợp hóa đơn bán hàng có thể phát hiện ra quy luật “những người mua bánh mì thường mua bơ”.

Nhận dạng mẫu là: ứng dụng các kỹ thuật học máy để phát hiện các mẫu có tính quy luật trong dữ liệu, thường là dữ liệu hình ảnh, âm thanh. Bài toán nhận dạng mẫu cụ thể thường là xác định nhãn cho đầu vào cụ thể, ví dụ cho ảnh chụp mặt người, cần xác định đó là ai.<sup>[9]</sup>

#### ***Ứng dụng cụ thể***

Sau đây là một số ví dụ ứng dụng cụ thể của học máy:

- Nhận dạng ký tự: phân loại hình chụp ký tự thành các loại, mỗi loại ứng với một ký tự tương ứng.
- Phát hiện và nhận dạng mặt người: phát hiện vùng có chứa mặt người trong ảnh, xác định đó là mặt người nào trong số những người đã có ảnh trước đó, tức là phân chia ảnh thành những loại tương ứng với những người khác nhau.
- Lọc thư rác, phân loại văn bản: dựa trên nội dung thư điện tử, chia thư thành loại “thư rác” hay “thư bình thường”; hoặc phân chia tin tức thành các thể loại khác nhau như “xã hội”, “kinh tế”, “thể thao”.v.v.

- Dịch tự động: dựa trên dữ liệu huấn luyện dưới dạng các văn bản song ngữ, hệ thống dịch tự động học cách dịch từ ngôn ngữ này sang ngôn ngữ khác. Hệ thống dịch tự động tiêu biểu dạng này là Google Translate.
- Chẩn đoán y tế: học cách dự đoán người bệnh có mắc hay không mắc một số bệnh nào đó dựa trên triệu chứng quan sát được.
- Phân loại khách hàng và dự đoán sở thích: sắp xếp khách hàng vào một số loại, từ đây dự đoán sở thích tiêu dùng của khách hàng.
- Dự đoán chỉ số thị trường: căn cứ giá trị một số tham số hiện thời và trong lịch sử, đưa ra dự đoán, chẳng hạn dự đoán giá chứng khoán, giá vàng.v.v.
- Các hệ khuyến nghị, hay hệ tư vấn lựa chọn: cung cấp một danh sách ngắn các loại hàng hóa, phim, video, tin tức v.v. mà người dùng nhiều khả năng quan tâm. Ví dụ ứng dụng loại này là phần khuyến nghị trên Youtube hay trên trang mua bán trực tuyến Amazon.
- Ứng dụng lái xe tự động: dựa trên các mẫu học chứa thông tin về các tình huống trên đường, hệ thống học máy cho phép tự ra quyết định điều khiển xe, và do vậy không cần người lái. Hiện Google đã có kế hoạch thương mại hóa xe ôtô tự động lái như vậy.<sup>[9]</sup>



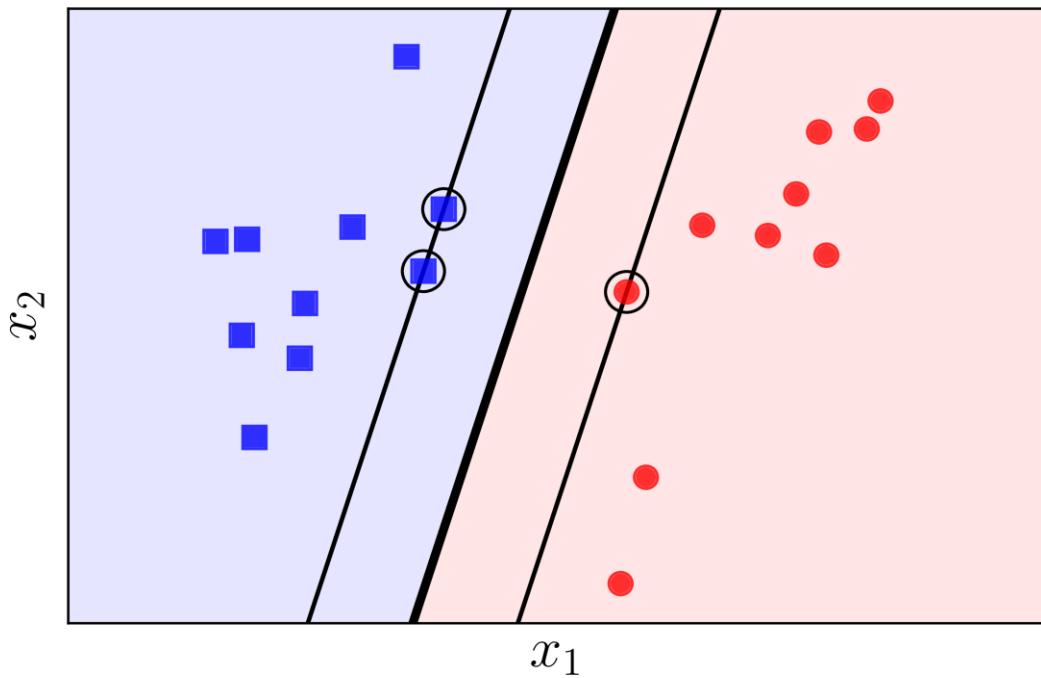
Hình 1. 6 Hình minh họa ứng dụng học máy<sup>[10][11][12][13]</sup>

## 1.2 Giới thiệu về bài toán phân lớp

### 1.2.1 Giới thiệu bài toán

Phân lớp là một nhiệm vụ yêu cầu sử dụng các thuật toán học máy để học cách gán nhãn lớp cho các mẫu. Một ví dụ dễ hiểu đó là phân lớp email là “spam” hoặc “không phải spam”. Có nhiều loại nhiệm vụ phân lớp khác nhau mà chúng ta có thể gặp phải trong học máy và các phương pháp tiếp cận chuyên biệt để lập mô hình có thể được sử dụng cho từng loại.

Từ góc độ mô hình hóa, phân lớp yêu cầu một tập dữ liệu đào tạo với nhiều mẫu về đầu vào và đầu ra để học hỏi. Một mô hình sẽ sử dụng tập dữ liệu huấn luyện và sẽ tính toán cách ánh xạ các mẫu tốt nhất về dữ liệu đầu vào, vào các nhãn lớp cụ thể. Như vậy, tập dữ liệu huấn luyện phải đủ lớn để đại diện cho vấn đề và có nhiều mẫu về mỗi nhãn lớp. Các nhãn lớp thường là các giá trị chuỗi, ví dụ: “spam”, “not spam” và phải được ánh xạ tới các giá trị số trước khi được cung cấp cho một thuật toán để lập mô hình. Việc ánh xạ này thường được gọi là mã hóa nhãn (label encoding), trong đó một số nguyên duy nhất được gán cho mỗi nhãn lớp, ví dụ: “spam” = 0, “not spam” = 1.<sup>[14]</sup>



Hình 1. 7 Hình minh họa bài toán phân lớp<sup>[20]</sup>

Các thuật toán mô hình dự đoán phân lớp được đánh giá dựa trên kết quả của chúng. Độ chính xác (accuracy) của phân lớp là một số liệu phổ biến được sử dụng để đánh giá hiệu suất của một mô hình dựa trên các nhãn lớp được dự đoán. Độ chính xác của phân lớp không phải là hoàn hảo nhưng là một điểm khởi đầu tốt cho nhiều nhiệm vụ phân lớp. Thay vì nhãn lớp, một số nhiệm vụ có thể yêu cầu dự đoán xác suất thành viên của lớp cho mỗi mẫu.<sup>[14]</sup>

### 1.2.2 Phân loại bài toán phân lớp

Có nhiều loại thuật toán phân lớp khác nhau để mô hình hóa các bài toán mô hình dự báo phân lớp chính gồm:

- Phân lớp nhị phân.
- Phân lớp nhiều lớp.<sup>[14]</sup>

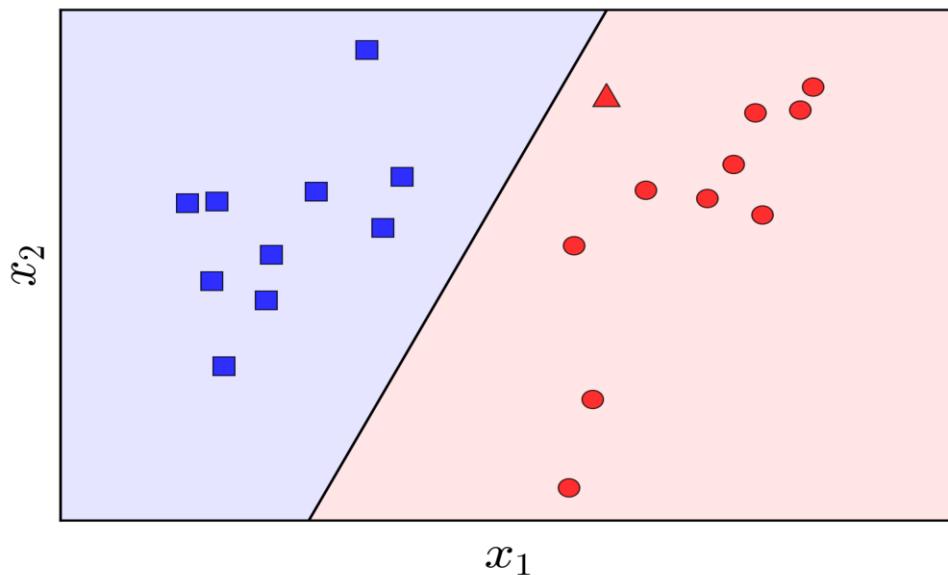
Tìm hiểu chi tiết:

- a. Bài toán phân lớp nhị phân.

Phân lớp nhị phân đề cập đến các nhiệm vụ phân loại có hai nhãn lớp.

Những ví dụ bao gồm:

- Phát hiện thư rác email (spam hay không).
- Dự đoán bệnh nhân có nguy cơ đột quỵ hay không...



Hình 1.8 Hình minh họa bài toán phân lớp nhị phân<sup>[15]</sup>

Thông thường, các nhiệm vụ phân loại nhị phân liên quan đến một lớp là trạng thái bình thường và một lớp khác là trạng thái bất thường.

Ví dụ là “ung thư không được phát hiện” là trạng thái bình thường của một nhiệm vụ liên quan đến xét nghiệm y tế và ‘ung thư được phát hiện’ là trạng thái bất thường.

Lớp cho trạng thái bình thường được gán nhãn lớp 0 và lớp có trạng thái bất thường được gán nhãn lớp 1. Người ta thường lập mô hình nhiệm vụ phân lớp nhị phân với một mô hình dự đoán phân phối xác suất Bernoulli cho mỗi ví dụ.

Phân phối Bernoulli là một phân phối xác suất rời rạc bao gồm trường hợp một sự kiện sẽ có kết quả nhị phân là 0 hoặc 1. Đối với phân lớp, điều này có nghĩa là mô hình dự đoán xác suất của một mẫu thuộc loại 1, hoặc trạng thái bất thường.

Các thuật toán phổ biến có thể được sử dụng để phân lớp nhị phân bao gồm:

- Hồi quy logistic.

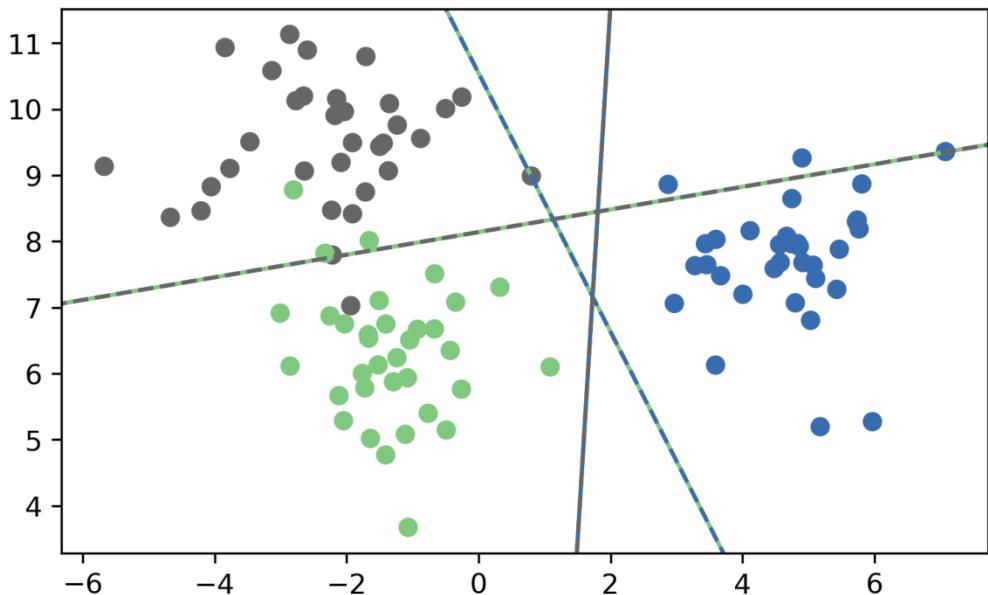
- k-Nearest Neighbors.
- Cây quyết định.
- Support Vector Machine.
- Naive Bayes. [14]

## b. Bài toán phân lớp nhiều lớp.

Phân lớp nhiều lớp đề cập đến các nhiệm vụ phân loại có nhiều hơn hai nhãn lớp.

Những ví dụ bao gồm:

- Phân loại khuôn mặt.
- Phân loại loài thực vật.
- Nhận dạng ký tự quang học...



Hình 1. 9 Hình minh họa bài toán phân lớp nhiều lớp<sup>[16]</sup>

Số lượng nhãn lớp có thể rất lớn đối với một số bài toán. Ví dụ: một mô hình có thể dự đoán một bức ảnh thuộc về một trong số hàng nghìn hoặc hàng chục nghìn khuôn mặt trong hệ thống nhận dạng khuôn mặt.

Các vấn đề liên quan đến dự đoán một chuỗi từ, chẳng hạn như mô hình dịch văn bản, cũng có thể được coi là một kiểu phân loại nhiều lớp đặc biệt. Mỗi từ trong chuỗi các từ được dự đoán liên quan đến một phân loại nhiều lớp trong đó kích thước của từ vựng xác định số lượng các lớp có thể được dự đoán và có thể có kích thước hàng

chục hoặc hàng trăm nghìn từ. Người ta thường lập mô hình nhiệm vụ phân lớp nhiều lớp với một mô hình dự đoán phân phối xác suất Multinoulli cho mỗi mẫu.

Phân phối Multinoulli là một phân phối xác suất rời rạc bao gồm trường hợp một sự kiện sẽ có kết quả phân loại, ví dụ: K trong  $\{1, 2, 3, \dots, K\}$ . Đối với phân loại, điều này có nghĩa là mô hình dự đoán xác suất của một ví dụ thuộc về mỗi nhãn lớp.

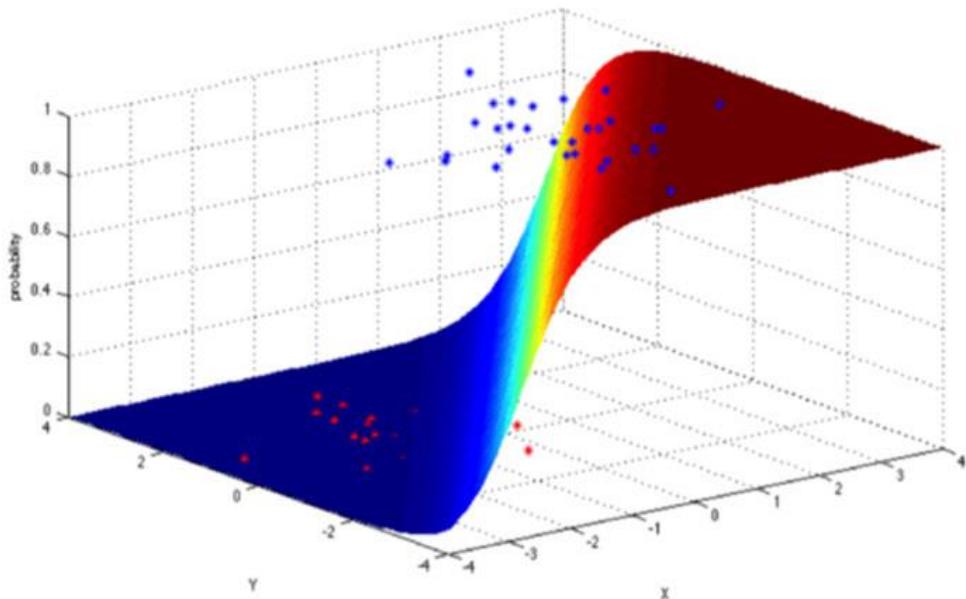
Các thuật toán phổ biến có thể được sử dụng để phân loại nhiều lớp bao gồm:

- k-Nearest Neighbors.
- Cây quyết định.
- Naive Bayes.
- Random Forest.
- Gradient Boosting.<sup>[14]</sup>

### **1.2.3 Thuật toán tiêu biểu của bài toán phân lớp**

a. Giới thiệu thuật toán hồi quy logistic:

Hồi quy logistic là một cách thống kê mạnh mẽ để mô hình hóa một kết quả nhị thức với một hoặc nhiều biến giải thích. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.<sup>[17]</sup>



Hình 1. 10 Hình minh họa Hồi quy logistic<sup>[18]</sup>

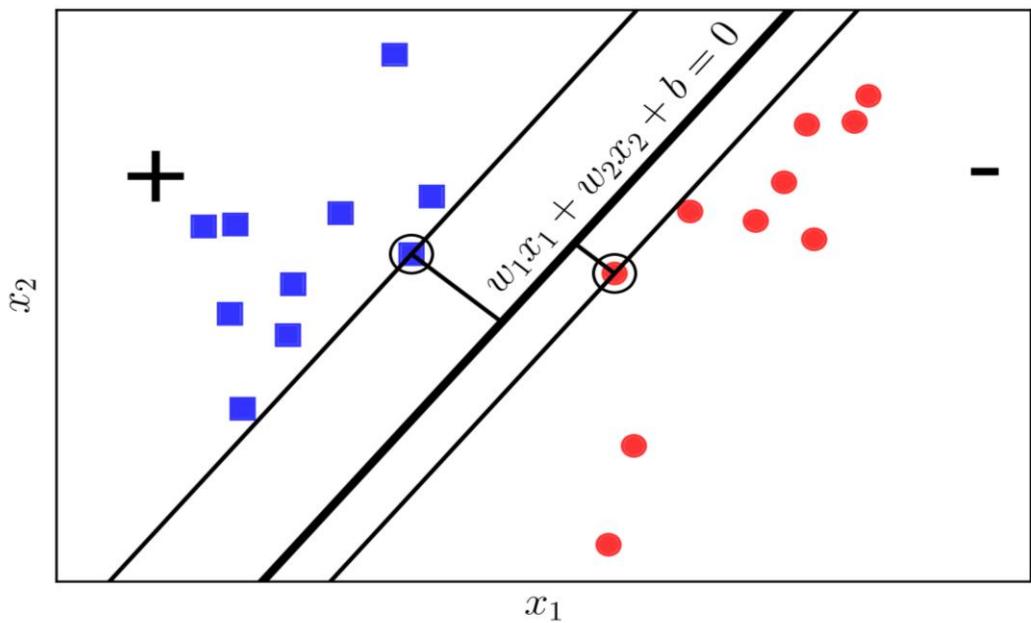
Thuật toán này được sử dụng trong một số trường hợp:

- Dự đoán doanh thu của một sản phẩm nhất định
- Dự đoán y khoa.
- Dự đoán động đất ...<sup>[17]</sup>

#### b. Giới thiệu thuật toán Support Vector Machines (SVM).

SVM là phương pháp phân loại nhị phân. Cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm.

Ví dụ, cho một tập các điểm thuộc 2 loại như hình bên dưới, SVM sẽ tìm ra một đường thẳng nhằm phân cách các điểm đó thành 2 nhóm sao cho khoảng cách giữa đường thẳng và các điểm xa nhất có thể. <sup>[19]</sup>



Hình 1. 11 Hình minh họa thuật toán svm<sup>[20]</sup>

Thuật toán này được sử dụng trong một số trường hợp:

- Mô hình chẩn đoán bệnh.
- Phân loại hình ảnh.
- Mô hình phân loại tin tức
- Mô hình phát hiện gian lận...<sup>[19]</sup>

### c. Giới thiệu thuật toán phân loại Naive Bayes

Naive Bayes là một thuật toán phân loại cho các vấn đề phân loại nhị phân (hai lớp) và đa lớp. Kỹ thuật này dễ hiểu nhất khi được mô tả bằng các giá trị đầu vào nhị phân hoặc phân loại.

Thuật toán Naive Bayes tính xác suất cho các yếu tố, sau đó chọn kết quả với xác suất cao nhất.

Tuy nhiên, ta cần lưu ý giả định của thuật toán Naive Bayes là các yếu tố đầu vào được cho là độc lập với nhau.<sup>[21]</sup>



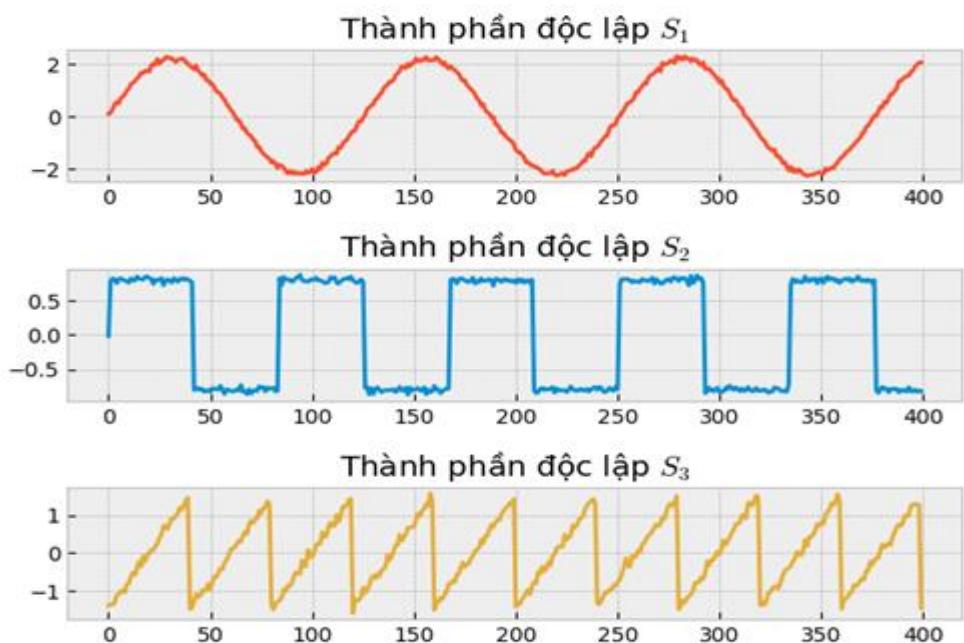
Hình 1. 12 Hình minh họa thuật toán Naive Bayes<sup>[22]</sup>

Thuật toán này được sử dụng trong một số trường hợp:

- Ứng dụng cho Y học .
- Đánh dấu một email là spam hay không.
- Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
- Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
- Sử dụng cho các phần mềm nhận diện khuôn mặt...<sup>[21]</sup>

#### d. Giới thiệu thuật toán phân tích thành phần độc lập

ICA là một kỹ thuật thống kê nhằm tìm ra các yếu tố ẩn nằm dưới các bộ biến ngẫu nhiên, các phép đo hoặc tín hiệu. ICA định nghĩa một mô hình phát sinh cho dữ liệu đa biến quan sát được, thường được đưa ra như một cơ sở dữ liệu lớn các mẫu. Trong mô hình, các biến số dữ liệu giả định là hỗn hợp tuyến tính của một số biến tiềm ẩn chưa biết, và hệ thống hỗn hợp cũng không rõ. Các biến tiềm ẩn được giả định không gian và độc lập với nhau, và chúng được gọi là các thành phần độc lập của dữ liệu được quan sát.<sup>[23]</sup>



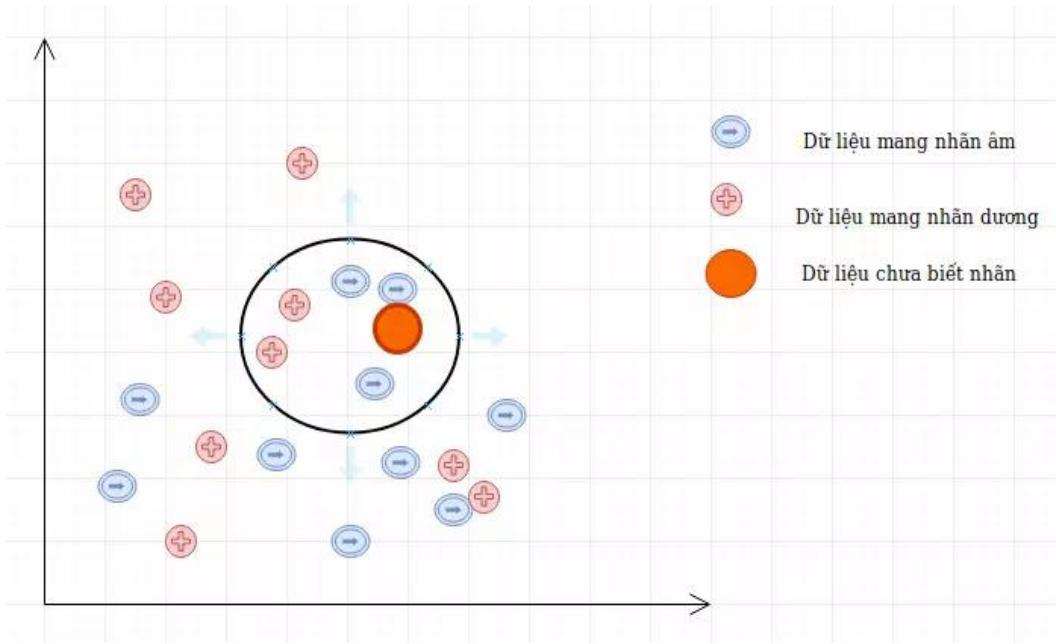
Hình 1. 13 Hình minh họa phân tích thành phần độc lập<sup>[23]</sup>

Thuật toán này được sử dụng trong một số trường hợp:

- Úng dụng đo lường tâm lý.
- Úng dụng hình ảnh kỹ thuật số .
- Úng dụng cơ sở dữ liệu tài liệu.
- Úng dụng đo lường chỉ số kinh tế...<sup>[23]</sup>

#### e. Giới thiệu thuật toán k-Nearest Neighbors

Thuật toán k-Nearest Neighbors -lắng giềng gần nhất, còn được gọi là KNN hoặc k-NN, là một bộ phân loại học có giám sát, phi tham số, sử dụng khoảng cách gần nhất để thực hiện phân loại hoặc dự đoán về việc nhóm một điểm dữ liệu riêng lẻ. Trong khi nó có thể được sử dụng cho các bài toán hồi quy (Regression) hoặc phân loại (Classification), nó thường được sử dụng như một thuật toán phân loại, giải quyết giả định rằng các điểm tương tự có thể được tìm thấy gần nhau.<sup>[24]</sup>



Hình 1. 14 Hình minh họa thuật toán  $k$ -Nearest Neighbors<sup>[25]</sup>

Thuật toán này được sử dụng trong một số trường hợp:

- Xếp hạng tín dụng.
- Phê duyệt khoản vay.
- Dự đoán giá cổ phiếu.
- Thị giác máy tính.
- Tiết xử lý dữ liệu...<sup>[24]</sup>

## CHƯƠNG II. THUẬT TOÁN ÁP DỤNG VÀ MÔ TẢ BÀI TOÁN

### 2.1 Mô tả bài toán:

Bối cảnh của bài toán: Theo thống kê, trên thế giới hằng năm có khoảng 15 triệu người đột quy, trong đó Việt Nam có 200.000 ca. Trung bình mỗi 45 giây có 1 người bị đột quy và mỗi 3 phút có một người tử vong. Tỉ lệ tử vong do đột quy ở Việt Nam cao hơn thế giới. Bệnh này có thể xảy ra ở mọi lứa tuổi, tuy nhiên, chủ yếu là người từ 55 tuổi trở lên. Có thể nói đột quy có thể xảy ra ở mọi lứa tuổi, tuy nhiên, tập trung chủ yếu ở người cao tuổi. Đột quy có thể gây ra nhiều biến chứng nghiêm trọng, thậm chí tử vong. Nếu được phát hiện, cảnh báo sớm sẽ giúp cho người bệnh có biện pháp phòng ngừa kịp thời, giảm nguy cơ xấu xảy ra.<sup>[26]</sup>

#### 2.1.1 Định nghĩa bệnh đột quy:

Đột quy là một căn bệnh cấp tính. Đột quy xảy ra khi xuất hiện hiện tượng vỡ mạch máu não hoặc tắc mạch khiến dòng máu lên não bị ngưng trệ, không tuần hoàn.

Nếu không được điều trị kịp thời, các tế bào trong não sẽ nhanh chóng bị ngừng hoạt động. Điều này có thể khiến cho người bệnh đổi mặt với di chứng tàn tật, thậm chí là tử vong.<sup>[27]</sup>



Hình 2. 1 Hình minh họa bệnh đột quy<sup>[29]</sup>

### **2.1.2 Triệu chứng bệnh đột quy:**

Đột quy thường được phát hiện với 7 dấu hiệu đặc trưng dưới đây:

1. Người bệnh có hiện tượng tê hoặc yếu cơ, đặc biệt thường xảy ra ở một bên cơ thể.
2. Người bệnh có dấu hiệu thay đổi thị lực ở một hoặc cả hai mắt.
3. Xuất hiện cảm giác khó nuốt.
4. Người bệnh bị nhức đầu nghiêm trọng không rõ nguyên nhân.
5. Cảm thấy chóng mặt, đi lại khó khăn, khó cử động.
6. Xuất hiện hiện tượng nói ngọng, khó nói, lưỡi bị tê cứng.
7. Bị rối loạn trí nhớ.

Các triệu chứng báo hiệu đột quy thường không kéo dài vì thế dấu hiệu sớm đột quy thường hay bị nhầm lẫn với một số biểu hiện thông thường, khi phát hiện bất kể một biểu hiện bất thường nào của người bệnh thì không nên chủ quan, mà hãy thực hiện việc cấp cứu kịp thời.<sup>[27]</sup>



Hình 2. 2 Hình minh họa triệu chứng bệnh đột quy<sup>[30]</sup>

### **2.1.3 Các biến chứng có thể gặp sau khi đột quy:**

Các biến chứng liên quan đến tim, viêm phổi, nghẽn tĩnh mạch, sốt, đau, khó nuốt, co cứng chi, trầm cảm,... đều là những biến chứng phổ biến của bệnh. Các biến chứng của đột quy tai biến mạch máu não khiến bệnh nhân bị ảnh hưởng nặng nề về sức khỏe, tâm lý, có thể dẫn đến khuyết tật tạm thời hoặc khuyết tật vĩnh viễn.<sup>[28]</sup>

Biến chứng ở vị trí nào còn phụ thuộc vào vị trí não bị ảnh hưởng và khoảng thời gian não không được cung cấp oxy bao gồm:

- Phù nề não sau đột quy.
- Viêm phổi: bệnh nhân đột quy tai biến mạch máu não thường gặp khó khăn trong việc nuốt, từ đó dẫn đến hiện tượng thức ăn, đồ uống dễ đi vào phổi gây viêm phổi.
- Đau tim: 1/2 trường hợp đột quy có liên quan đến tình trạng xơ vữa động mạch, gia tăng nguy cơ đau tim sau đột quy do sự tồn tại của mảng xơ vữa.
- Trầm cảm: Bệnh có thể trở nên tồi tệ hơn với những bệnh nhân đã bị trầm cảm trước khi đột quy.
- Loét do tỳ đè (thời gian nằm liệt giường kéo dài): Người bị đột quy thường mất khả năng vận động, phải nằm hoặc ngồi yên tại chỗ trong thời gian dài gây viêm loét.
- Động kinh: Sau đột quy, não có thể có những hoạt động bất thường, gây ra co giật.
- Rối loạn thị giác: Người bị đột quy có thể bị giảm hoặc mất thị lực ở 1 hoặc cả hai mắt.
- Co cứng chi: Mất khả năng vận động, một tay bị yếu hoặc liệt.
- Nghẽn mạch máu: Mất khả năng vận động hoặc hạn chế vận động khiến cục máu đông hình thành trong tĩnh mạch chân.
- Nhiễm trùng đường tiết niệu: Xảy ra ở bệnh nhân đột quy có đặt ống thông foley.
- Giảm nhận thức (mất trí nhớ).

- Mất chức năng nói, khó nói, nói không đầy đủ, nói từ vô nghĩa, không hiểu người khác nói gì...<sup>[28]</sup>

Đột quy có quá trình phục hồi chậm và lâu dài, cần kiên trì, không nóng vội, nghe theo các phương pháp điều trị phản khoa học. Đột quy thường phục hồi tốt trong 3 tháng đầu, chậm hơn 3 tháng tiếp theo, đột quy đã ngoài 6 tháng thì khả năng phục hồi rất chậm.

Việc phát hiện sớm bệnh đột quy là vô cùng quan trọng. Phát hiện đột quy thông qua các dữ liệu của mỗi người sẽ được Bác sĩ phát hiện và báo cho người bệnh biết để có biện pháp phòng, chống. Tuy nhiên, khi không có triệu chứng, thường thì người bệnh sẽ không biết được, chủ quan không đi khám xét. Hơn nữa, khi tổ chức khám xét, Bác sĩ sẽ mất nhiều thời gian để đọc, kết luận người nào có có dấu hiệu bị đột quy hay không. Nghiên cứu được trình bày trong đồ án này là dùng thuật toán trí tuệ nhân tạo (AI) để giúp Bác sĩ chẩn đoán nhanh hơn, thậm chí sau khi có kết quả khám bệnh, mọi người có thể nhập các dữ liệu vào phần mềm, phần mềm sẽ tự động đưa ra kết quả thay vì phải thông qua Bác sĩ, hoặc phải chờ đợi tại Bệnh viện trong bối cảnh quá tải tại các bệnh viện. Dưới đây là cách thực hiện thuật toán đó.<sup>[28]</sup>

## 2.2 Mô tả dữ liệu bài toán

Bài toán dự đoán sớm nguy cơ đột quy của người già dựa trên dữ liệu bệnh án của 338 bệnh nhân bị đột quy và người khỏe mạnh. Trong đó, có 161 dữ liệu bệnh nhân bị đột quy được thu thập ở bệnh viện E Trung ương và 177 dữ liệu bệnh nhân khỏe mạnh được thu thập từ bệnh viện Đa Khoa Sóc Sơn Hà Nội.

### 2.2.1 Thu thập dữ liệu:

A. Dữ liệu bệnh nhân bị đột quy:

Dữ liệu gốc 161 dữ liệu bệnh nhân bị đột quy được thu thập ở bệnh viện E Trung ương:

stt	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1																											
2	1	54	1	##### TX Cao Be	15:25:00	#####	14:30:00	8:3	2:1	55	65	120	124	100	60	120	36.4	26	15	2	1	1	128	1			
3	2	81	1	##### Thuong Ti	14:20:00	#####	13:00:00	1:1	2:1	80	70	150	125	160	100	60	36.0	18.9	10	8	7	6	128	1			
4	3	54	1	##### Dai Tu, Tr	09:15:00	#####	08:00:00	2:3	1:1	75	65	140	12	190	100	116	36.5	19.2	17	13	7	6	128	2			
5	4	58	1	##### Hiep Hoa,	19:20:00	#####	17:15:00	1:1	3:1	125	50	175	125	120	80	82	37.0	32	11	8	6	5	1248	1			
6	5	75	1	##### Quinh Phu	08:05:00	#####	06:15:00	1:3	1:1	110	40	150	12	110	70	100	36.4	17.4	12	#NULL!	#NULL!	#NULL!	1289	1			
7	6	51	2	##### Hai Ba Tru	15:25:00	#####	14:35:00	1:1	2:1	50	50	100	12	140	80	84	36.6	36	11	3	1	1	1289	2			
8	7	51	1	##### Ung Hoa, I	08:20:00	#####	07:30:00	7:1	1:1	50	85	135	123	130	90	110	36.8	18.4	11	10	8	6	1289	1			
9	8	60	1	##### Hoang Ma	11:00:00	#####	10:15:00	2:1	1:1	45	75	120	12	210	120	85	36.9	22.7	10	6	4	3	1289	1			
10	9	60	2	##### Tay Ho, H	14:40:00	#####	12:35:00	6:1	2:1	125	50	175	1,235	130	80	78	36.8	19.9	17	3	1	0	1289	2			
11	10	58	2	##### Hai Ba Tru	07:05:00	#####	05:30:00	2:1	1:1	95	95	180	12	120	80	105	36.5	22.1	10	6	1	0	1289	1			
12	11	53	1	##### Dong Da,	14:10:00	#####	13:15:00	1:4	2:1	55	60	115	124	110	70	97	37.1	20.6	10	8	7	7	1248	1			
13	12	70	2	##### Thach Th	14:50:00	#####	14:00:00	2:1	2:1	50	70	120	12	12	7	99	36.3	16.8	10	5	5	4	124	2			
14	13	58	1	##### Thach Th	14:00:00	#####	12:45:00	2:1	2:1	75	75	150	125	600	100	87	37.0	23.1	15	13	10	9	1258	1			
15	14	50	2	##### Dong Da,	12:45:00	#####	12:15:00	2:1	2:1	30	40	70	125	230	130	78	36.8	19.7	15	10	8	7	1258	1			
16	15	71	2	##### Ba Dinh, H	15:10:00	#####	13:30:00	3:1	2:1	100	50	150	125	120	70	80	36.7	20.2	19	19	19	#NULL!	1259	2			
17	16	49	1	##### Phu Xuyer	14:35:00	#####	13:30:00	7:1	2:1	65	65	130	128	110	70	66	37.0	21.1	14	7	4	2	1248	1			
18	17	76	1	##### Hai Ba Tru	20:20:00	#####	19:30:00	3:1	3:1	50	65	135	128	120	80	96	37.0	19.8	17	8	3	3	1249	2			
19	18	56	2	##### Phu Xuyer	23:00:00	#####	21:00:00	7:1	3:1	120	45	165	12	110	70	164	37.0	18.9	17	13	11	8	1249	2			
20	19	56	2	##### Hoai Duc,	14:30:00	#####	13:00:00	1:1	2:1	90	50	140	12	130	80	78	36.5	16.2	10	7	5	5	1248	2			
21	20	55	2	##### Dong Da,	21:15:00	#####	20:15:00	2:1	3:1	60	30	90	123	170	80	80	37.0	22.6	7	3	1	0	12	1			
22	21	57	2	##### Van Giang	07:30:00	#####	05:30:00	2:1	1:1	120	55	175	124	190	100	90	37.0	20.3	15	8	7	5	1248	1			
23	22	61	1	##### Khoa Cha	13:30:00	#####	11:30:00	1:1	2:1	120	45	165	12	160	90	80	37.0	18.7	6	2	1	0	12	2			
24	23	61	1	##### Ba Dinh, H	13:15:00	#####	11:30:00	2:1	2:1	105	55	160	1,234	140	90	95	36.8	29.3	10	2	2	1	1248	1			
25	24	73	2	##### Yen My, H	07:05:00	#####	05:10:00	2:1	1:1	110	50	160	124	140	80	96	37.0	27.8	10	3	2	1	1248	2			
26	25	63	2	##### Thanh Tru	08:30:00	#####	07:00:00	2:1	1:1	90	50	140	124	130	80	80	37.0	28.5	9	7	11	11	1248	1			
27	26	62	1	##### Hoang Ma	16:50:00	#####	15:00:00	2:1	2:1	110	55	165	12	140	90	80	36.0	24.1	5	2	1	1	1248	1			
28	27	69	1	##### Hai Ba Tru	09:45:00	#####	09:15:00	1:4	1:1	30	45	75	125	140	90	87	36.0	24.9	7	3	1	1	1248	1			
29	28	70	1	##### My Hao, H	12:30:00	#####	10:30:00	2:1	2:1	120	55	175	125	110	70	73	36.8	22.4	8	2	1	1	128	1			
30	29	62	1	##### Thanh Xu	15:15:00	#####	14:45:00	7:1	2:1	30	90	120	125	130	80	85	36.8	23.2	14	8	4	#NULL!	1248	1			
31	30	58	1	##### Hoang Ma	18:10:00	#####	16:40:00	3:1	3:1	90	55	145	125	130	80	92	37.0	21.7	15	12	4	1	1248	2			
32	31	67	1	##### Dong Da,	19:25:00	#####	19:00:00	2:1	3:1	25	120	145	12	140	70	85	36.8	20.8	13	9	6	5	1249	1			
33	32	66	2	##### Hai Ba Tru	10:00:00	#####	08:36:00	6:1	1:1	74	70	154	125	130	80	75	37.0	20.3	10	4	2	1	1248	2			
34	33	61	1	##### Ba Dinh, H	16:35:00	#####	15:26:00	2:1	2:1	69	75	144	125	170	90	75	36.0	22.4	9	5	4	4	1248	1			
35	34	53	2	##### Anh Dien	21:00:00	#####	19:00:00	2:1	3:1	120	50	170	125	130	90	80	37.0	24.9	12	9	2	1	1248	1			
36	35	57	1	##### Yen Thinh	15:55:00	#####	13:55:00	2:1	2:1	120	65	185	1,258	130	90	97	36.8	29.9	14	3	2	0	1248	1			
37	36	77	1	##### Hoang Ma	08:45:00	#####	07:15:00	3:1	1:1	90	35	125	127	180	90	85	36.2	21.7	10	14	20	20	128	2			
38	37	62	2	##### Phuong Li	22:05:00	#####	21:00:00	1:1	3:1	65	55	120	125	150	100	101	36.5	19.5	15	15	14	14	1258	2			

Hình 2.3 Hình minh họa dữ liệu bệnh nhân đột quy gốc

Chi tiết:

- Dữ liệu bệnh nhân đột quy gốc ban đầu:

Dữ liệu gốc 161 bệnh nhân bị đột quy bao gồm các 69 cột thuộc tính dữ liệu (gồm cả thuộc tính nhiễu) là:

Tuổi, giới tính, mã bệnh nhân, địa chỉ, thời điểm nhập viện, ngày vào viện, thời gian khởi phát, tiền sử bệnh, dd khởi phát, khung thời gian khởi phát, kiểu khởi phát, khởi phát vào viện, nv dùng thuốc, thời gian khởi phát dùng thuốc, triệu chứng, huyết áp thực tế vào viện, huyết áp trong viện, nhịp tim vào viện, nhiệt độ vào viện, chỉ số khói, NIHSS vào viện, NIHSS 1 giờ, NIHSS 24 giờ, NIHSS ra viện, triệu chứng thực tế, liệt nửa người, hồng cầu, bạch cầu, hematocrit, tiểu cầu, INR, PTs, PToo, APTTs, FIB, đường máu, HbA1C, cholesterol, HDLcho, LDLcho, triglycerid, SGOT, SGPT, natri, kali, DTD, CTso, mạch não lần 1, mạch não lần 2, SA tim, HKnhi, SA mạch, phân loại đột quy, lieutPA, MORI, XH não, TG chảy máu, thé Xã Hội, XH hệ thống, đại máu vi thể, LQxuathuyet, tai tắc mạch, NMlanrong, NIHSS, GLASGOW, RANKIN, thời gian nằm viện, kết quả xuất viện.

- Xử lý thuộc tính dữ liệu nhiễu (các thuộc tính không tác động đến kết quả đầu ra):

➤ Loại bỏ 17 thuộc tính dữ liệu nhiều gồm:

Mã bệnh nhân, địa chỉ, thời điểm nhập viện, ngày vào viện, thời gian khởi phát, dd khởi phát, khung thời gian khởi phát, kiểu khởi phát, khởi phát vào viện, nv dùng thuốc, thời gian khởi phát dùng thuốc, phân loại đột quy, thẻ Xã Hội, XH hệ thống, Lqxuathuyet, thời gian năm viện, kết quả xuất viện.

➤ Thuộc tính dữ liệu còn lại 52 thuộc tính gồm:

Tuổi, giới tính, triệu chứng, huyết áp thực tế vào viện, huyết áp trong viện, nhịp tim vào viện, nhiệt độ vào viện, chỉ số khói, NIHSS vào viện, NIHSS 1 giờ, NIHSS 24 giờ, NIHSS ra viện, triệu chứng thực tế, liệt nửa người, hồng cầu, bạch cầu, hematocrit, tiểu cầu, INR, PTs, PToo, APTTs, FIB, đường máu, HbA1C, cholesterol, HDLcho, LDLcho, triglycerid, SGOT, SGPT, natri, kali, DTD, CTso, mạch não lần 1, mạch não lần 2, SA tim, HKnhi, SA mạch, lieuTPA, MORI, XH não, TG chảy máu, đại máu vi thể, tai tắc mạch, NMIanrong, NIHSS, GLASGOW, RANKIN.

• Chọn lọc thuộc tính áp dụng cho bài toán:

Thuộc tính dữ liệu được chọn để áp dụng cho bài toán dựa trên cách tiếp cận dữ liệu của người bệnh và đặc biệt là những thuộc tính chính ảnh hưởng trực tiếp tới kết quả.

Đồng thời dựa trên “Trường Môn Tim mạch Hoa Kỳ/Hội đồng Tim mạch Hoa Kỳ (American College of Cardiology/American Heart Association - ACC/AHA)” để chọn các thuộc tính chính để phát triển dự án.<sup>[31]</sup>

Tên xét nghiệm	Trị số bình thường	Kết quả	Tên xét nghiệm	Trị số bình thường	Kết quả
Urê	2,5 - 7,5 mmol/L		Sắt	Nam: 11- 27 µmol/L Nữ: 7- 26 µmol/L	
Glucose	3,9 - 6,4 mmol/L		Magiê	0,8- 1,00 mmol/L	
Creatinin	Nam: 62- 120 µmol/L Nữ: 53- 100 µmol/L		AST (GOT)	≤ 37 U/L- 37°C	
Acid Uric	Nam: 180- 420µmol/L Nữ: 150-360µmol/L		ALT (GPT)	≤ 40 U/L- 37°C	
Bilirubin T.P	≤ 17µmol/L		Amylase		
Bilirubin T.T	≤ 4,3µmol/L		CK	Nam: 24- 190 U/L- 37° Nữ: 24- 167 U/L- 37°	
Bilirubin G.T	≤ 12,7 µmol/L		CK- MB	≤ 24 U/L- 37°	
Protein T.P	65- 82 g/L		LDH	230- 460 U/L- 37°	
Albumin	35- 50 g/L		GGT	Nam: 11- 50 U/L- 37° Nữ: 7- 32 U/L- 37°	
Globulin	24- 38 g/L		Cholinesterase	5300- 12900 U/L- 37°	
Tỉ lệ A/G	1,3- 1,8		Phosphattase kiềm		
Fiblinogen	2- 4 g/L		<b>Các xét nghiệm khí máu</b>		

Hình 2. 4 Hình minh họa phiếu xét nghiệm định kỳ<sup>[52]</sup>

Sau khi chọn lọc:

- Gồm 10 thuộc tính chính được chọn áp dụng cho bài toán gồm:  
Tuổi, giới tính, chỉ số khối, nhịp tim, huyết áp, đường máu, Cholesterol, Triglycerid, Tiền Sứ Bệnh, RANKIN.

	A	B	C	D	E	F	G	H	I	J	K	L
1	tuo	gioitinh	chisokhoi	nhiptim	huyetap	duongmau	cholesterol	triglycerid	tiensubent	RANKIN	ketqua	
2	54	1	26	120	100	6.30	4.39	2.57	8	1	1	
3	81	1	18.9	60	160	7.91	6.51	1.46	1	4	1	
4	54	1	19.2	116	190	6.54	5.13	1.25	2	2	1	
5	58	1	32	82	120	5.87	4.79	2.15	1	3	1	
6	75	1	17.4	100	110	6.46	6.66	2.34	1	6	1	
7	51	2	36	84	140	7.63	4.49	1.49	1	1	1	
8	51	1	18.4	110	130	6.74	4.45	3.64	7	4	1	
9	60	1	22.7	85	210	5.37	6.08	3.99	2	4	1	
10	60	2	19.9	78	130	5.80	4.58	1.46	6	0	1	
11	58	2	22.1	105	120	9.34	4.13	1.42	2	0	1	
12	53	1	20.6	97	110	20.24	3.95	0.87	1	5	1	
13	70	2	16.8	99	120	8.98	5.10	4.30	2	4	1	
14	58	1	23.1	87	600	8.46	5.91	6.55	2	5	1	
15	50	2	19.7	78	230	7.25	5.68	0.66	2	4	1	
16	71	2	20.2	80	120	14.15	4.90	1.77	3	6	1	
17	49	1	21.1	66	110	6.43	5.24	1.10	7	2	1	
18	76	1	19.8	96	120	10.33	5.08	0.91	3	3	1	
19	56	2	18.9	164	110	9.37	4.25	0.73	7	4	1	
20	56	2	16.2	78	130	5.96	3.60	0.85	1	3	1	
21	55	2	22.6	80	170	5.70	5.70	2.42	2	0	1	
22	57	2	20.3	90	190	7.70	7.29	1.77	2	3	1	
23	61	1	18.7	80	160	6.31	4.19	0.73	1	0	1	
24	61	1	29.3	95	140	5.60	6.18	6.82	2	1	1	
25	73	2	27.8	96	140	10.47	6.10	1.98	2	1	1	

*Hình 2. 5 Hình minh họa dữ liệu bệnh nhân đột quỵ sau khi chọn lọc các thuộc tính*

b. Dữ liệu người khỏe mạnh:

Dữ liệu người khỏe mạnh gồm 177 dữ liệu, những người khỏe mạnh được thu thập từ bệnh viện Đa Khoa Sóc Sơn Hà Nội.

Dữ liệu ban đầu được tập hợp từ các mẫu báo cáo khám sức khỏe định kỳ tại cơ sở bệnh viện Đa Khoa Sóc Sơn Hà Nội.

<p><b>KHÁM SỨC KHỎE ĐỊNH KỲ</b></p> <p><b>I. HÀNH CHÍNH</b></p> <p>Họ và tên : Trần Thanh Tâm Giới Tính : Nữ Sinh năm : 18/07/1953 Địa chỉ : Kim Sơn - Hồng Kỳ – SÓC Sơn – Hà Nội SĐT : 0376878468 Nhập viện : Ngày 08 tháng 03 năm 2021</p> <p><b>II. Hồi bệnh</b></p> <p>1. Lý do vv: Khám Sức Khỏe Định Kỳ . 2. Tiền sử: * Bản thân: Bệnh nhân có tiền sử bệnh đột quỵ triều chứng Tê, yếu hoặc liệt tay chân 1 bên cơ thể, đôi khi người bệnh chỉ cảm giác nặng, Rối loạn giọng nói, nói đờ lưỡi, nói khó Chóng mặt, choáng váng, xây xẩm hoặc có thể ngất xỉu, Thay đổi dáng đi, mất đồng bộ và khả năng phối hợp vận động. Bệnh nhân có tiền sử bệnh huyết áp cao triệu chứng tăng huyết áp nhức đầu, hoa mắt, chóng mặt, ù tai, mất ngủ nhẹ, đau nhói vùng tim, suy giảm thị lực, thở gấp, mặt đỏ bừng, da tái xanh, nôn ói, hồi hộp, đánh trống ngực, hốt hoảng. * Gia đình : Khỏe mạnh</p> <p><b>III. KHÁM BỆNH :</b></p> <p>1. Toàn thân: Bệnh nhân tình trạng sức khỏe bình thường</p> <p>Gan lách không to 2.4 Thận – Tiểu niệu : Hồ thận 2 bên không đầy , Chạm thận (-) bẹp bẹn thận (-) 2.5 Thần kinh: Hội chứng não , màng não (-). Không liệt thần kinh khu trú 2.6 Cơ – xương – khớp : Không teo cơ cứng khớp Khớp vận động trong giới hạn bình thường</p> <p>IV. Cận lâm sàng:</p> <p>1.CTM: HC: 4,3 T/L( 3.8 – 5.0 T/L) ; Hb: 130 g/L( 120 - 150 g/L) ; Hct: 0,418 L/L( 0.336-0.450 L/L); MCV: 81 fL (75 - 96 fL) MCH : 32 pg (24- 33pg) ; MCHC: 326 – 382 g/L BC : 9.8 G/L (4.0 đến 10.0G/L) 2. Sinh hóa máu : Glucose( đái): 4.22 mmol/L Lipid máu : Cholesterol : 5.2 mmol/L ; Triglycerid: 0.67 mmol/L GOT/GPT: 42/53 ; Aciduric: 356 ; Ure: Creatinin 100</p> <p>3. Điện giải đồ : Natri : bình thường; Kali: Bình thường</p> <p>4. Tổng phân tích nước tiểu : Protein niệu: bình thường</p>	<p><b>Thể trạng bình thường</b></p> <p>BMI: 21.7</p> <p>Da niêm mạc hồng Không tim môi không tim đầu chi Không khó thở, không run tay Không phù, Không sốt Tuyến giáp không to , Hạch ngoại vi không sờ thấy Điểm Rankin : 2.</p> <p>Mach : 79 lần / phút , nhiệt độ 36 độ 7 , Huyết áp: 120/95 mmHg</p> <p>2. Bộ phận</p> <p>2.1 Tuần hoàn : Móm tim đập ở khaong liên sườn V đường giữa đòn trái Tim nhịp đều , Tso 79ck / p Không có tiếng tim bệnh lý Dhieu Hartzer âm tính</p> <p>2.2. Hô hấp : Lồng ngực hai bên cân đối di động theo nhịp thở Phổi rì rào phế nang rõ Không có ranh bệnh lý</p> <p>2.3 Tiêu hóa : Bụng mềm không chướng ,</p> <p>5. Điện tim đồ : nhịp xoang, trục trung gian ts 79ck/ p 6. Siêu âm tim: bình thường 7. Soi đáy mắt: bình thường</p> <p>V. Tóm tắt bệnh án:</p> <p>Bệnh nhân nữ 69 tuổi vào viện với lý do : Khám Sức Khỏe Định Kỳ. Qua hỏi và khám thấy: Bệnh nhân có tiền sử bệnh đột quỵ triều chứng Tê, yếu hoặc liệt tay chân. Bệnh nhân có tiền sử bệnh huyết áp cao triệu chứng tăng huyết áp nhức đầu, hoa mắt, chóng mặt, ù tai.</p> <p>Gia đình : Khỏe mạnh BMI : 21.7 Glucose( đái): 4.22 mmol/L Lipid máu : Cholesterol : 5.2 mmol/L ; Triglycerid: 0.67 mmol/L GOT/GPT: 42/53 ; Aciduric: 356 ; Ure: Creatinin 100</p> <p>Chẩn đoán : Sức Khỏe bình thường.</p> <p>VII. Điều trị</p> <p>1 Mục tiêu Ăn nhạt , giảm muối (50% natri bình thường ) Giảm mỡ, hạn chế mỡ động vật và các thức ăn chứa nhiều cholesterol như da , nội tạng động vật , lòng đỏ trứng gà ...</p>
---	--

*Hình 2. 6 Hình mẫu báo cáo khám sức khỏe định kỳ tại bệnh viện Đa Khoa Sóc Sơn Hà Nội*

Chi tiết:

- Các thuộc tính dữ liệu bệnh nhân khỏe mạnh ban đầu:

Dữ liệu gốc gồm 177 dữ liệu bệnh nhân khỏe mạnh bao gồm tập hợp các dữ liệu (gồm cả dữ liệu nhiễu) là:

Họ và tên, giới tính, sinh năm (tuổi), địa chỉ, số điện thoại, ngày nhập viện, lý do vào viện, tiền sử bệnh, tiếp xúc, chỉ số BMI, nhịp tim, nhiệt độ, huyết áp, Rankin,

tuần hoàn, hô hấp, tiêu hóa, thận tiết liệu, thần kinh, cơ xương khớp, HC, HB, HCT, MCV, MCH, MCHC, BC, glucose (đường máu), cholesterol, triglycerid, GOT/GPT, aciduric, ure, creatinin, natri, kali, protein niệu, điện tim đồ ,siêu âm tim, soi đáy mắt, tiên lượng.

- Xử lý thuộc tính dữ liệu nhiễu ( các thuộc tính không tác động đến kết quả đầu ra):
  - Loại bỏ 14 thuộc tính dữ liệu nhiễu gồm:

Họ và tên, địa chỉ, số điện thoại , ngày nhập viện , tiếp xúc , lý do vào viện, hô hấp , tiêu hóa, thận tiết liệu, cơ xương khớp, điện tim đồ ,siêu âm tim, soi đáy mắt, tiên lượng.

- Thuộc tính dữ liệu còn lại 27 thuộc tính gồm:

Giới tính, sinh năm (tuổi), tiền sử bệnh , chỉ số BMI, nhịp tim , nhiệt độ , huyết áp, Rankin, tuần hoàn, thần kinh, HC, HB, HCT, MCV, MCH, MCHC, BC, glucose (đường máu), cholesterol, triglycerid, GOT/GPT , aciduric, ure, creatinin, natri , kali, protein niệu.

- Chọn lọc thuộc tính áp dụng cho bài toán:

Thuộc tính dữ liệu được chọn để áp dụng cho bài toán dựa trên cách tiếp cận dữ liệu của người bệnh và đặc biệt là những thuộc tính chính ảnh hưởng trực tiếp tới kết quả.

Đồng thời dựa trên “Trường Môn Tim mạch Hoa Kỳ/Hội đồng Tim mạch Hoa Kỳ (*American College of Cardiology/American Heart Association - ACC/AHA*)” để chọn các thuộc tính chính để phát triển dự án .<sup>[31]</sup>

Sau khi chọn lọc:

- Gồm 10 thuộc tính chính được chọn áp dụng cho bài toán gồm:
  - Tuổi, giới tính, chỉ số khối, nhịp tim, huyết áp , đường máu, Cholesterol, Triglycerid, Tiền Sử Bệnh, RANKIN.

	A	B	C	D	E	F	G	H	I	J	K
1	tuoi	gioitinh	chisokhoi	nhiptim	huyetap	duongmau	cholesterc	triglycerid	tiensubent	RANKIN	ketqua
2	52	1	19.6	75	140	4.81	2.32	0.74	0	0	0
3	77	2	23.3	86	140	4.62	3.68	1.04	0	0	0
4	88	1	24.1	63	130	4.81	2.4	1.53	0	0	0
5	59	2	18.4	84	120	3.35	3.02	1.64	1	1	0
6	85	1	26.3	76	110	5.96	4.5	1.71	1	2	0
7	91	1	22.8	85	120	3.8	3.48	1.38	0	0	0
8	75	1	22.4	80	120	3.41	3.96	0.82	0	0	0
9	92	1	18.2	75	100	5.12	4.87	1.15	0	1	0
10	90	2	23.2	85	110	4.86	1.73	1.22	0	0	0
11	66	2	17.8	80	130	3.3	4	1.51	0	1	0
12	57	2	24.1	83	110	3.73	3	1.35	0	1	0
13	89	1	22.8	69	120	5.54	4.62	0.81	0	0	0
14	52	2	21.8	79	130	5.1	2.73	1.56	0	0	0
15	64	1	22	73	130	5.26	4.21	1.14	0	0	0
16	62	2	25.7	73	130	5.6	3.47	1.91	0	1	0
17	92	1	19.1	68	130	6.14	3.63	1.67	1	1	0
18	79	1	22.2	78	120	4.47	4.38	1.61	1	1	0
19	99	2	24.3	80	120	4.8	4.19	1.77	0	0	0
20	99	1	23.1	71	140	5.3	3.48	1.08	0	0	0
21	90	1	24.2	82	120	4.26	3.69	1.95	0	1	0
22	45	1	22.9	82	90	3.08	4.7	0.78	0	0	0
23	80	2	20.3	65	90	4	3.6	1.25	0	0	0
24	81	1	18.4	72	110	3.74	4.16	1.41	0	0	0
25	56	1	19.5	87	110	5.4	3.5	1.67	0	0	0

Hình 2. 7 Hình minh họa dữ liệu bệnh nhân khỏe mạnh sau khi chọn lọc và tập hợp

### C. Gộp dữ liệu bệnh nhân đột quy và người khỏe mạnh

Sau khi xử lý và chọn lọc dữ liệu kết quả thu được 338 dữ liệu gồm 10 thuộc tính của các bệnh nhân đột quy và bệnh nhân khỏe mạnh.

Sau đó gộp và xáo trộn dữ liệu bằng công cụ Kutools trong Excel.<sup>[32]</sup>

A	B	C	D	E	F	G	H	I	J	K	
1	tuoi	gioitinh	huyetap	nhiptimw	cholesterc	triglycerid	duongmau	tiensubent	chisokhoi	RANKIN	ketqua
2	54	1	100	120	4.39	2.57	6.30	8	26	1	1
3	81	1	160	60	6.51	1.46	7.91	1	18.9	4	1
4	54	1	190	116	5.13	1.25	6.54	2	19.2	2	1
5	58	1	120	82	4.79	2.15	5.87	1	32	3	1
6	75	1	110	100	6.66	2.34	6.46	1	17.4	6	1
7	51	2	140	84	4.49	1.49	7.63	1	36	1	1
8	51	1	130	110	4.45	3.64	6.74	7	18.4	4	1
9	60	1	210	85	6.08	3.99	5.37	2	22.7	4	1
10	60	2	130	78	4.58	1.46	5.80	6	19.9	0	1
11	58	2	120	105	4.13	1.42	9.34	2	22.1	0	1
12	53	1	110	97	3.95	0.87	20.24	1	20.6	5	1
13	70	2	120	99	5.10	4.30	8.98	2	16.8	4	1
14	58	1	600	87	5.91	6.55	8.46	2	23.1	5	1
15	50	2	230	78	5.68	0.66	7.25	2	19.7	4	1
16	71	2	120	80	4.90	1.77	14.15	3	20.2	6	1
17	49	1	110	66	5.24	1.10	6.43	7	21.1	2	0
18	76	1	120	96	5.08	0.91	10.33	3	19.8	3	0
19	56	2	110	164	4.25	0.73	9.37	7	18.9	4	0
20	56	2	130	78	3.60	0.85	5.96	1	16.2	3	0
21	55	2	170	80	5.70	2.42	5.70	2	22.6	0	0
22	57	2	190	90	7.29	1.77	7.70	2	20.3	3	0
23	61	1	160	80	4.19	0.73	6.31	1	18.7	0	0
24	61	1	140	95	6.18	6.82	5.60	2	29.3	1	0
25	73	2	140	96	6.10	1.98	10.47	2	27.8	1	0

Hình 2. 8 Hình minh họa sử dụng Kutools để xáo trộn dữ liệu

### **2.2.2 Mô tả dữ liệu bài toán**

- Ma trận dữ liệu (x):

Sau khi tiền xử lý ma trận dữ liệu gồm có 10 thuộc tính và mỗi thuộc tính có 338 dữ liệu:

Ta có các thuộc tính thuộc kiểu số nguyên (int) gồm:

- Tuổi.
- Giới tính.
- Nhịp tim.
- Huyết áp.
- Tiền sử bệnh.
- Rankin.

Ta có các thuộc tính thuộc kiểu số thực (float) gồm:

- Chỉ số khối.
- Đường Máu.
- Cholesterol.
- Triglycerid.

	A	B	C	D	E	F	G	H	I	J	K
1	tuoi	gioitinh	chisokhoi	nhiptim	huyetap	duongmau	cholesterol	triglycerid	tiensuben	RANKIN	ketqua
2	70	1	21.1	105	140	5.91	5.1	0.62	6	1	1
3	57	1	20.3	90	190	7.7	7.29	1.77	2	3	1
4	65	1	27.9	100	120	10.94	5.82	1.58	5	4	1
5	75	2	22.7	66	110	5.8	4.73	1.79	1	2	0
6	69	1	24.9	87	140	12.78	4.02	2.07	1	1	1
7	93	2	23.7	82	130	4.7	3.63	0.76	0	0	0
8	81	1	18.1	83	110	4.37	4.94	1.63	0	0	0
9	77	1	21.7	85	180	10.14	5.31	1.72	3	5	1
10	73	1	19.7	78	130	4.04	4.15	1.27	0	0	0
11	57	1	21.8	90	150	9.16	3.27	1.86	2	5	1
12	74	1	22.5	78	160	5.68	6.78	0.93	2	1	1
13	69	1	28.3	76	170	8.66	3.11	0.8	1	2	1
14	81	1	21.3	80	120	4.75	5.8	1.42	0	0	0
15	75	1	17.6	88	120	6.36	4.36	2.08	1	1	1
16	77	1	19.1	85	160	6.39	5.34	1.11	2	0	1
17	84	1	22.6	82	120	5.71	4.86	0.63	1	1	0
18	63	1	28.5	80	130	8.66	4.72	1.76	2	4	1
19	81	1	17.4	85	130	3.94	4.1	1.63	0	1	0
20	92	1	19.1	68	130	6.14	3.63	1.67	1	1	0
21	81	1	18.4	72	110	3.74	4.16	1.41	0	0	0
22	68	1	27.9	86	180	5.52	5.76	1.28	1	1	1
23	88	1	24.1	63	130	4.81	2.4	1.53	0	0	0

Hình 2. 9 Hình minh họa kiểu dữ liệu của từng thuộc tính

- Vecto đầu ra (y):

Sau khi tiền xử lý vecto đầu ra gồm có 1 thuộc tính và 338 dữ liệu.

Kiểu dữ liệu đầu ra (kết quả) thuộc kiểu dữ liệu số nguyên (int).

ketqua	1	1	1	0	1	0	0	1	0	1	1	0	1	0
--------	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Hình 2. 10 Hình minh họa kiểu dữ liệu int của thuộc tính đầu ra

- Input và Output của bài toán:

Input của bài toán gồm:

- $x_1$  = Tuổi.
- $x_2$  = Giới tính.
- $x_3$  = Chỉ số khối.
- $x_4$  = Nhịp tim.
- $x_5$  = Huyết áp.
- $x_6$  = Đường máu.
- $x_7$  = Cholesterol.
- $x_8$  = Triglycerid.

➤  $x_9 =$  Tiền sử bệnh.

➤  $x_{10} =$  Rankin.

**FORM DỰ ĐOÁN NGUY CƠ ĐỘT QUY**

---

Tuổi	<input type="text" value="số tuổi của bạn"/>	
Giới Tính	<input type="text" value="Nam nhập : 1   Nữ nhập : 2"/>	
Chi Só Khối	<input type="text" value="CSK = Cân nặng : [ Chiều cao x 2 ]"/>	
Nhip Tim	<input type="text" value="Chỉ Số Nhịp Tim : Phút"/>	
Huyết Áp	<input type="text" value="Chỉ Số Huyết Áp"/>	
Đường Máu	<input type="text" value="Chỉ Số Đường Máu"/>	
Cholesterol	<input type="text" value="Chỉ Số cholesterol"/>	
Triglycerid	<input type="text" value="Chỉ Số triglycerid"/>	
Tiền Sử Bệnh	<input type="text" value="Số Lần Bị Đột Quy."/>	
Rankin	<input type="text" value="Chỉ Số Rankin"/>	

 Chuẩn Đoán

*Hình 2. 11 Hình minh họa đầu vào (input) của bài toán*

Output của bài toán gồm:

➤  $Y =$  Kết quả người bệnh có nguy cơ bị đột quy hay không.

**FORM DỰ ĐOÁN NGUY CƠ ĐỘT QUỴ**

Tuổi	<input type="text" value="số tuổi của bạn"/>	❶
Giới Tính	<input type="text" value="Nam nhập : 1   Nữ nhập : 2"/>	❷
Chi Số Khối	<input type="text" value="CSK = Cân nặng : [ Chiều cao x 2 ]"/>	❸
Nhịp Tim	<input type="text" value="Chỉ Số Nhịp Tim : Phút"/>	❹
Huyết Áp	<input type="text" value="Chỉ Số Huyết Áp"/>	❺
Đường Máu	<input type="text" value="Chỉ Số Đường Máu"/>	❻
Cholesterol	<input type="text" value="Chỉ Số cholesterol"/>	❽
Triglycerid	<input type="text" value="Chỉ Số triglycerid"/>	❾
Tiền Sử Bệnh	<input type="text" value="Số Lần Bị Đột Quỵ"/>	❿
RanKin	<input type="text" value="Chỉ Số Rankin"/>	❻
<input type="button" value="🔍 Chuẩn Đoán"/>		
<b>Bệnh Nhân Khôe Mạnh</b>		

Hình 2. 12 Hình minh họa đầu ra (output) của bài toán

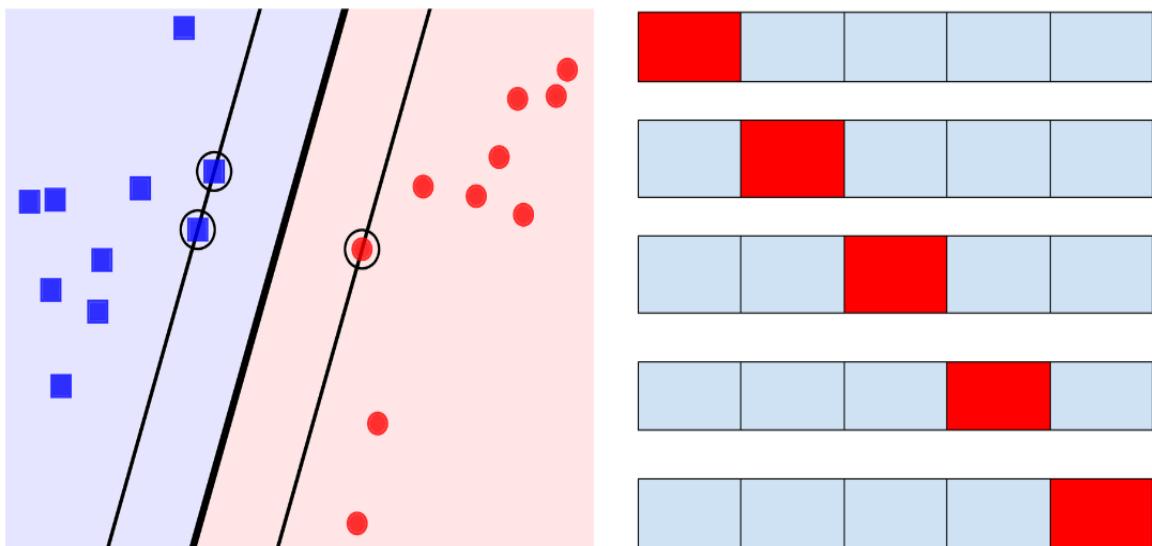
### 2.3 Thuật toán áp dụng

Sau khi tìm hiểu và nghiên cứu bài toán “dự đoán bệnh đột quỵ ở người già” dựa trên nhiều thuộc tính liên quan tới bài toán, nhưng do không thể xác định được xác suất của từng thuộc tính liên quan ảnh hưởng đến từng nhãn. Vì vậy, sử dụng thuật toán Support Vector Machines (SVM) để giải quyết bài toán là một phương án hợp lý nhất và sử dụng thuật toán K-fold Cross validation để lấy mẫu để đánh giá mô hình học máy trong trường hợp dữ liệu hiện tại của bài toán không được dồi dào.<sup>[33]</sup>

Vì sao lại sử dụng thuật toán SVM để giải quyết bài toán?

- SVM rất hiệu quả để giải quyết bài toán dữ liệu có số chiều lớn.
- SVM giải quyết vấn đề overfitting rất tốt.

- SVM là phương pháp phân lớp nhanh.
- SVM có hiệu suất tổng hợp tốt và hiệu suất tính toán cao.
- SVM tiết kiệm bộ nhớ khi sử dụng.
- SVM có tính linh hoạt - phân lớp thường là phi tuyến tính.



Hình 2. 13 Hình minh họa svm và k-fold cross validation<sup>[20][34]</sup>

### 2.3.1 Phương pháp K-fold cross validation để cải thiện dữ liệu

a. Giới thiệu:

K-fold cross validation là một kỹ thuật lấy mẫu để đánh giá mô hình học máy trong trường hợp dữ liệu không được dồi dào cho lắm (số mẫu hạn chế).

K-fold cross validation: Chia training set thành K tập con không giao nhau, có kích thước gần bằng nhau. Tại mỗi lần kiểm thử:

- 1 tập con được lấy ra làm validation set.
- K-1 tập còn lại được dùng để xây dựng mô hình.
- Mô hình cuối được xác định dựa trên trung bình của các train error và validation error là nhỏ nhất.

Giả sử  $y$  là đầu ra thực sự, và  $\hat{y}$  là đầu ra dự đoán bởi mô hình.

Train error: Mức độ sai khác giữa đầu ra thực và đầu ra dự đoán của mô hình, thường là giá trị của hàm mất mát áp dụng lên training set.<sup>[35]</sup>

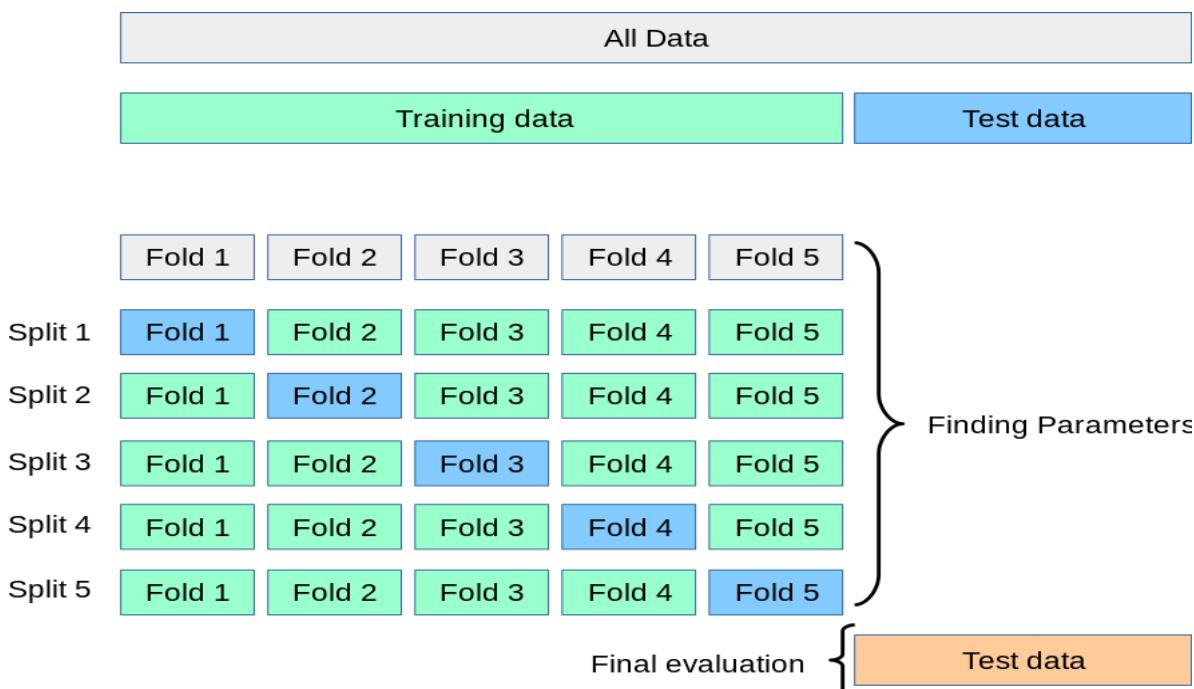
$$\text{train error} = \frac{1}{N_{\text{train}}} \sum_{\text{training set}} \|\mathbf{y} - \hat{\mathbf{y}}\|_p^2$$

Test error: Mức độ sai khác giữa đầu ra thực và đầu ra dự đoán của mô hình, thường là giá trị của hàm mất mát áp dụng lên test set.

$$\text{test error} = \frac{1}{N_{\text{test}}} \sum_{\text{test set}} \|\mathbf{y} - \hat{\mathbf{y}}\|_p^2$$

Mô hình cuối được xác định dựa trên trung bình của các train error và validation error (CV(w) là nhỏ nhất).<sup>[35]</sup>

$$CV(w) = \frac{1}{K} \sum_{i=1}^K E_i(w)$$



Hình 2. 14 Hình minh họa Phương pháp K-fold cross validation<sup>[36]</sup>

b. Ý Tưởng và cách thực hiện của phương pháp:

Ý tưởng:

- Phần dữ liệu Test data sẽ được để riêng và dành cho bước đánh giá cuối cùng nhằm kiểm tra “phản ứng” của model khi gặp các dữ liệu unseen hoàn toàn.

- Phần dữ liệu Training thì sẽ được chia ngẫu nhiên thành K phần (K là một số nguyên, hay chọn là 5 hoặc 10). Sau đó train model K lần, mỗi lần train sẽ chọn 1 phần làm dữ liệu validation và K-1 phần còn lại làm dữ liệu training. Kết quả đánh giá model cuối cùng sẽ là trung bình cộng kết quả đánh giá của K lần train. Đó chính là lý do vì sao ta đánh giá khách quan và chính xác hơn.

- Sau khi đánh giá xong model và thấy kết quả (ví dụ accuracy trung bình) chấp nhận được thì ta có thể thực hiện một trong 2 cách sau để tạo ra model cuối cùng (để mang đi dùng dự đoán predict): [36]

➤ Cách một: Trong quá trình train các fold, ta lưu lại model tốt nhất và mang model đó đi dùng luôn. Cách này sẽ có ưu điểm là không cần train lại nhưng lại có nhược điểm là model sẽ không nhìn được all data và có thể không làm việc tốt với các dữ liệu trong thực tế.

➤ Cách hai: Train model 1 lần nữa với toàn bộ dữ liệu (không chia train, val nữa) và sau đó save lại và mang đi predict với test set để xem kết quả. [36]

❖ Cách thực hiện phương pháp:

1. Xáo trộn dataset một cách ngẫu nhiên
2. Chia dataset thành k nhóm
3. Với mỗi nhóm:

- Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình.
- Các nhóm còn lại được sử dụng để huấn luyện mô hình.
- Huấn luyện mô hình.
- Đánh giá và sau đó hủy mô hình.

4. Tổng hợp hiệu quả của mô hình từ các số liệu đánh giá:

Kết quả tổng hợp thường là trung bình của các lần đánh giá. Ngoài ra, việc bổ sung thông tin về phương sai và độ lệch chuẩn vào kết quả tổng hợp cũng được sử dụng trong thực tế. [36]

c. Ba chiến thuật để lựa chọn k:

- Đại diện: Giá trị của k được chọn để mỗi tập train/test đủ lớn, có thể đại diện về mặt thống kê cho dataset chứa nó.

- $K = 10$ : Giá trị của  $k$  được gán cố định bằng 10, một giá trị thường được sử dụng và được chứng minh là cho sai số nhỏ, phương sai thấp (through qua thực nghiệm).
- $K = n$ : Giá trị của  $k$  được gán cố định bằng  $n$ , với  $n$  là kích thước của dataset, như vậy mỗi mẫu sẽ được sử dụng để đánh giá mô hình một lần.<sup>[37]</sup>

d. Đánh giá phương pháp:

- Số lượng mô hình cần huấn luyện tỉ lệ thuận với  $k$ .
- Nếu trong bài toán, có nhiều tham số cần xác định, khoảng giá trị của mỗi tham số rộng thì việc huấn luyện nhiều mô hình là khó khả thi.
- Mỗi điểm được kiểm tra chính xác một lần.
- Giải pháp khi có ít dữ liệu để xây dựng mô hình. <sup>[37]</sup>

### **2.3.2 Thuật toán Support Vector Machines (SVM).**

A. Giới thiệu:

Support vector machine (SVM) là một trong những thuật toán phân lớp phổ biến và hiệu quả.

Phương pháp SVM được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính được các tác giả Vapnik và Chervonenkis phát triển mạnh mẽ năm 1995.

Phương pháp này thực hiện phân lớp dựa trên nguyên lý Cực tiểu hóa rủi ro có Cấu trúc SRM (Structural Risk Minimization), được xem là một trong các phương pháp phân lớp giám sát không tham số tinh vi nhất cho đến nay. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp.<sup>[20]</sup>

B. Định nghĩa:

Là phương pháp dựa trên nền tảng của lý thuyết thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả tìm được là chính xác.

Là thuật toán học giám sát được sử dụng cho phân lớp dữ liệu.

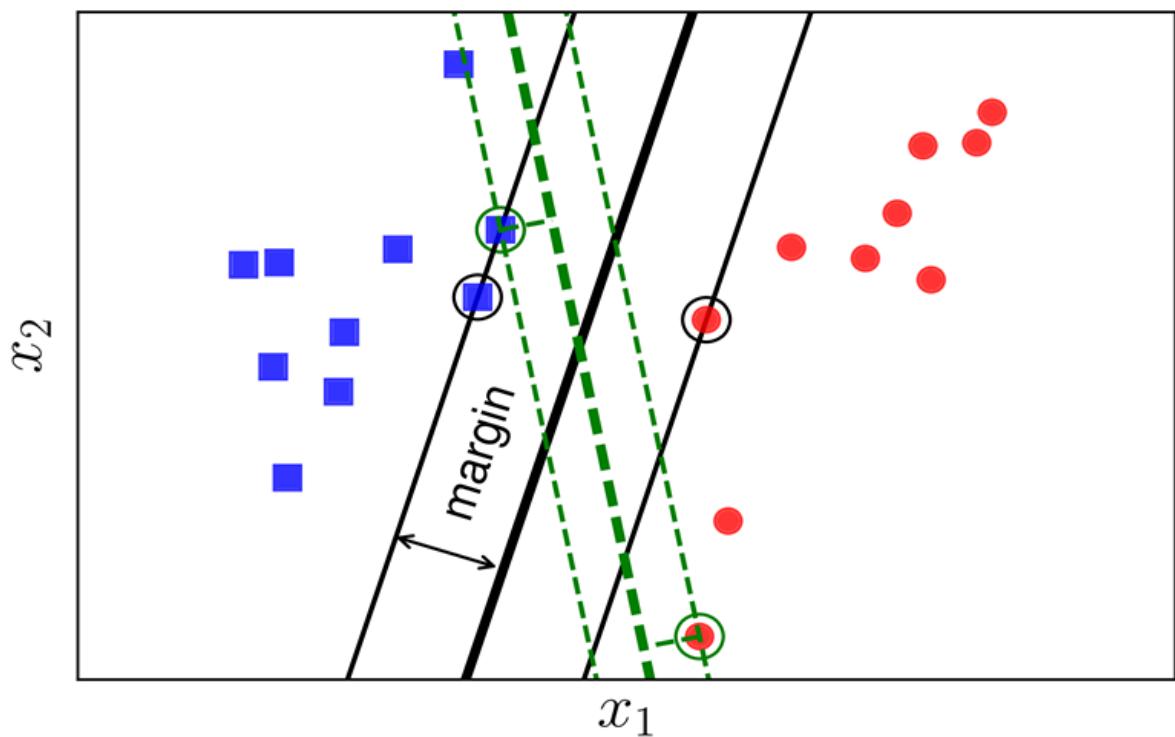
Là 1 phương pháp thử nghiệm, đưa ra 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu.

SVM là một phương pháp có tính tổng quát cao nên có thể được áp dụng cho nhiều loại bài toán nhận dạng và phân loại.<sup>[20][35]</sup>

### C. Ý tưởng của SVM

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi dữ liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp “+” và lớp “-”.

Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này (gọi là margin). Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt và khoảng cách của hai lớp đến mặt phẳng này phải bằng nhau và lớn nhất có thể.<sup>[20][35]</sup>



Hình 2. 15 Hình minh họa siêu phẳng trong svm.<sup>[20]</sup>

### D. Input và Output của SVM

Input:

Giả sử ma trận chứa các điểm dữ liệu:

$$X = [x_1, x_2, x_3, \dots, x_N] \in \mathbb{R}^{d \times N}$$

Mỗi cột  $x_i \in \mathbb{R}^{d \times 1}$  là một điểm dữ liệu trong không gian d chiều.

Các nhãn của một điểm dữ liệu được lưu trong một vector hàng

$$Y = [y_1, y_2, y_3, \dots, y_N] \in \mathbb{R}^{1 \times N}$$

Output:

Tìm ra được 1 siêu phẳng thỏa mãn Khoảng cách từ điểm gần nhất của mỗi lớp tới đường phân chia là như nhau và khoảng cách này phải là lớn nhất.

- Từ đó, phân chia được các class của tập dữ liệu đầu vào

E. Cách giải quyết bài toán:

- Khoảng cách từ một điểm tới một siêu mặt phẳng:

Trong không gian hai chiều, khoảng cách từ một điểm có tọa độ  $(x_0, y_0)$  tới đường thẳng có phương trình  $w_1x + w_2y + b = 0$  được xác định bởi:<sup>[35]</sup>

$$\frac{|w_1x_0 + w_2y_0 + b|}{\sqrt{w_1^2 + w_2^2}}$$

Trong không gian ba chiều, khoảng cách từ một điểm có tọa độ  $(x_0, y_0, z_0)$  tới một mặt phẳng có phương trình  $w_1x + w_2y + w_3z + b = 0$  được xác định bởi:

$$\frac{|w_1x_0 + w_2y_0 + w_3z_0 + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}}$$

Trong không gian d chiều khoảng cách từ một điểm (vector) có tọa độ  $(x_{10}, x_{20}, \dots, x_{d0})$  tới siêu mặt phẳng có phương trình  $w_1x_1 + w_2x_2 + \dots + w_dx_d + b = 0$  được xác định bởi:

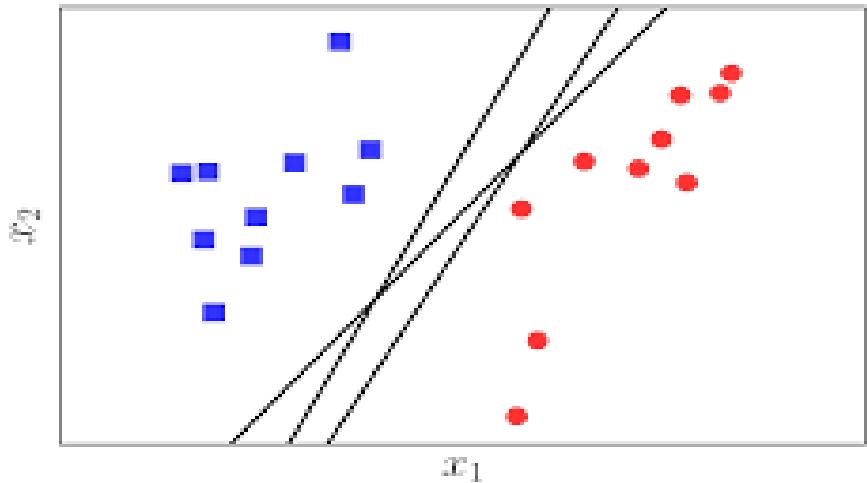
$$\frac{|w_1x_{10} + w_2x_{20} + \dots + w_dx_{d0} + b|}{\sqrt{w_1^2 + w_2^2 + \dots + w_d^2}} = \frac{|\mathbf{w}^T \mathbf{x}_0 + b|}{\|\mathbf{w}\|_2}$$

với  $\mathbf{x}_0 = [x_{10}, x_{20}, \dots, x_{d0}]^T$ ,  $\mathbf{w} = [w_1, w_2, \dots, w_d]^T$

Nếu bỏ trị tuyệt đối ở tử số, ta biết được điểm đó nằm về phía nào của mặt phẳng đang xét:

- Những điểm mang dấu dương nằm về cùng một phía .
- Những điểm mang dấu âm nằm về phía còn lại.

- Những điểm nằm trên mặt phẳng làm cho tử số có giá trị bằng 0, tức khoảng cách bằng 0.<sup>[35]</sup>



*Hình 2. 16 Hình minh họa hai lớp dữ liệu đỏ và xanh. Có vô số các đường thẳng có thể phân tách chính xác hai lớp dữ liệu này.*<sup>[20]</sup>

➤ Xây dựng bài toán tối ưu cho SVM

Giả sử các cặp dữ liệu trong tập huấn luyện là  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  với:

- Vector  $x_i \in \mathbb{R}^d$  thể hiện đầu vào của một điểm dữ liệu.
- $y_i$  là nhãn của điểm dữ liệu đó.
- $d$  là số chiều của dữ liệu.
- $N$  là số điểm dữ liệu.

Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi  $y_i = 1$  (class 1) hoặc  $y_i = -1$  (class 2).

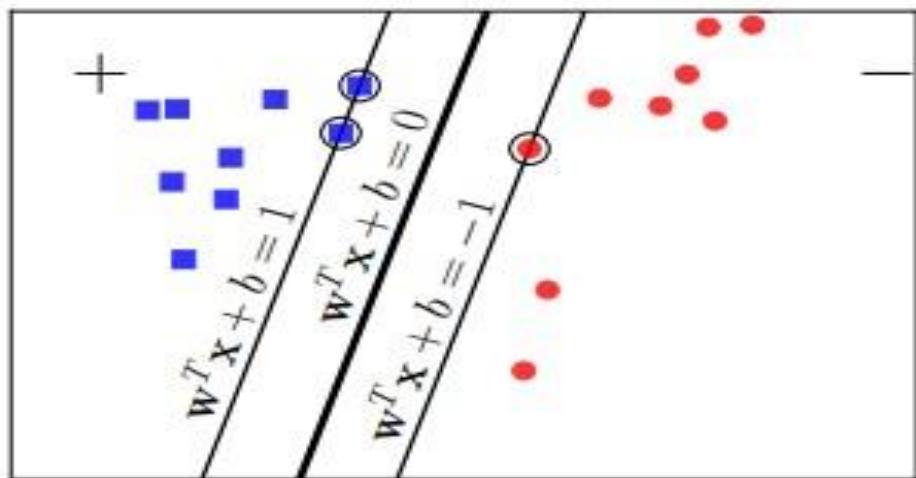
Xét trong không gian hai chiều:

Giả sử :

- Các điểm vuông xanh thuộc class 1.
- Các điểm tròn đỏ thuộc class -1.
- Mặt phẳng phân chia giữa 2 class là :  $w^T x + b = w_1 x_1 + w_2 x_2 + b = 0$ .

- Class 1 nằm về phía dương . Class -1 nằm về phía âm của mặt phân chia.<sup>[35]</sup>

Ta cần tìm  $\mathbf{w}$  và  $\mathbf{b}$  để tìm được siêu phẳng .



Hình 2. 17 Hình minh họa xây dựng bài toán tối ưu cho SVM<sup>[20]</sup>

- Với cặp dữ liệu  $(\mathbf{x}_n, y_n)$  bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2}$$

- Vì  $y_n$  luôn cùng dấu với phía của  $\mathbf{x}_n$ . Từ đó suy ra  $y_n$  cùng dấu với  $(\mathbf{w}^T \mathbf{x}_n + b)$ , vì vậy tử số luôn là một số không âm.
- Với mặt phân chia như trên , margin được tính là khoảng cách gần nhất từ một điểm tới mặt đó.

$$margin = \min_n \frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2}$$

- Bài toán tối ưu của SVM chính là việc tìm  $w$  và  $b$  sao cho margin này đạt giá trị lớn nhất:

$$(\mathbf{w}, b) = \arg \max_{\mathbf{w}, b} \left\{ \min_n \frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2} \right\} = \arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_n y_n(\mathbf{w}^T \mathbf{x}_n + b) \right\}$$

- Việc giải trực tiếp bài toán này sẽ rất phức tạp , nhưng ta có cách để đưa nó về bài toán đơn giản hơn.
- Nhận xét quan trọng là nếu ta thay vector hệ số  $\mathbf{w}$  bởi  $k\mathbf{w}$  và  $b$  bởi  $kb$  trong đó  $k$  là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi.<sup>[35]</sup>
- Dựa trên nhận xét quan trọng trên, ta có thể giả sử:

$$y_n(\mathbf{w}^T \mathbf{x}_n + b) = 1$$

- Như vậy, với mọi  $n$  ta luôn có:

$$y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1$$

- Vậy bài toán tối ưu có thể đưa về bài toán tối ưu có ràng buộc sau đây:

$$(\mathbf{w}, b) = \arg \max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|_2}$$

thoả mãn:  $y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1, \forall n = 1, 2, \dots, N$

- Xác định lớp cho một điểm dữ liệu mới:

Sau khi đã tìm được mặt phân cách  $\mathbf{w}^T \mathbf{x} + b = 0$ .

Nhãn của bất kỳ một điểm nào sẽ được xác định đơn giản bằng:

$$\text{class}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$$

Khi thay điểm dữ liệu mới vào biểu thức này nếu:

- $\text{sgn}(\mathbf{w}^T \mathbf{x} + b) > 0$  : điểm dữ liệu mới được gán nhãn là: 1.
- $\text{sgn}(\mathbf{w}^T \mathbf{x} + b) < 0$  : điểm dữ liệu mới được gán nhãn là: -1.

## G. Tóm tắt bài toán SVM:

- Với bài toán phân lớp nhị phân mà hai lớp dữ liệu là linearly separable, có vô số các mặt phân cách phẳng giúp phân biệt hai lớp đó.
  - Với mỗi mặt phân cách ta có một phân lớp .
  - Khoảng cách gần nhất từ một điểm dữ liệu tới mặt phân cách ấy được gọi là margin của bộ phân lớp đó.

- Support vector machine là bài toán đi tìm mặt phân cách sao cho margin có được là lớn nhất, đồng nghĩa với việc các điểm dữ liệu có một khoảng cách an toàn tới mặt phân cách.
- Bài toán tối ưu trong SVM là một bài toán lồi với hàm mục tiêu là strictly convex, nghiệm của bài toán này là duy nhất . Hơn nữa, bài toán tối ưu đó là một quadratic programming (QP).<sup>[35]</sup>

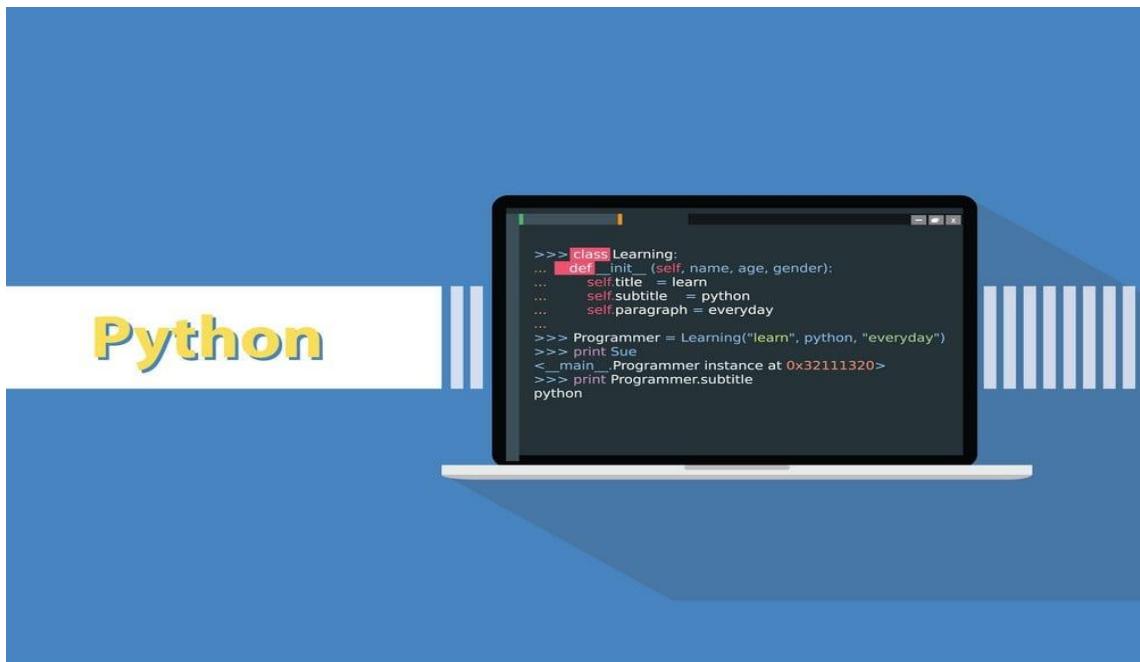
## CHƯƠNG 3. XÂY DỰNG HỆ THỐNG VÀ ĐÁNH GIÁ MÔ HÌNH

### 3.1 Xây dựng hệ thống

Xây dựng hệ thống gồm những vấn đề quan trọng sau: Ngôn ngữ lập trình áp dụng, phần mềm áp dụng và các bước thực hiện xây dựng hệ thống.

#### 3.1.1 Ngôn ngữ lập trình được sử dụng trong xây dựng hệ thống

Ngôn ngữ lập trình PYTHON: Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng. Ngôn ngữ lập trình Python được tạo bởi Guido van Rossum và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.<sup>[38]</sup>



Hình 3. 1 Hình ảnh minh họa ngôn ngữ lập trình python<sup>[39]</sup>

Tính năng chính của ngôn ngữ python:

- Ngôn ngữ lập trình đơn giản, dễ học.
- Miễn phí, mã nguồn mở.
- Ngôn ngữ thông dịch cấp cao.
- Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến.
- Đặc biệt được dùng rộng rãi trong phát triển trí tuệ nhân tạo. [38]

Ngôn ngữ đánh dấu HTML CSS:

HTML: là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web, nghĩa là các mảng thông tin được trình bày trên World Wide Web.

CSS: Định nghĩa về cách hiển thị của một tài liệu HTML. CSS đặc biệt hữu ích trong việc thiết kế Web. Nó giúp cho người thiết kế dễ dàng áp đặt các phong cách đã được thiết kế lên bất kỳ trang nào của website một cách nhanh chóng, đồng bộ. [40]



Hình 3. 2 Hình ảnh minh họa html css<sup>[41]</sup>

Những ưu điểm của HTML CSS:

- Nguồn tài nguyên hỗ trợ lớn.
- Hoạt động mượt mà trên phần lớn các trình duyệt phổ biến hiện nay.
- Cách sử dụng dễ dàng.

- Mã nguồn mở, miễn phí.
- Dễ dàng tích hợp với nhiều loại ngôn ngữ như PHP, Node.js...

### **3.1.2 Phần mềm được sử dụng trong xây dựng hệ thống**

Phần mềm lập trình Python 3.10:

Phần mềm lập trình Python 3.10 hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động, do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý, phần mềm lập trình Python 3.10 hầu như tương thích trên mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.<sup>[42]</sup>



Hình 3. 3 Hình ảnh minh họa phần mềm lập trình Python 3.10<sup>[43]</sup>

Những ưu điểm của phần mềm lập trình Python 3.10

- Hỗ trợ đa nền tảng: Linux, Mac, Windows,...
- Ít dung lượng
- Tính năng mạnh mẽ
- Intellisense chuyên nghiệp
- Giao diện thân thiện
- Số lượng người sử dụng lớn tạo nên cộng đồng hỗ trợ rộng rãi

## Phần mềm lập trình Visual Studio Code :

Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux.<sup>[44]</sup>



Hình 3. 4 Hình ảnh minh họa phần mềm lập trình Visual Studio Code<sup>[45]</sup>

### Những ưu điểm của phần mềm lập trình Visual Studio Code

- Hỗ trợ đa nền tảng: Linux, Mac, Windows,...
- Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, JavaScript, HTML, CSS,...
- Ít dung lượng.
- Tính năng mạnh mẽ.
- Intellisense chuyên nghiệp.
- Giao diện thân thiện.
- Số lượng người sử dụng lớn tạo nên cộng đồng hỗ trợ rộng rãi.

### Phần mềm phát triển sản phẩm HEROKU:

Heroku là nền tảng đám mây cho phép các lập trình viên xây dựng, triển khai, quản lý và mở rộng ứng dụng (PaaS – Platform as a service).

Nó rất linh hoạt và dễ sử dụng, cung cấp cho một con đường đơn giản nhất để đưa sản phẩm tiếp cận người dùng. Nó giúp các nhà phát triển tập trung vào phát triển sản phẩm mà không cần quan tâm đến việc vận hành máy chủ hay phần cứng...<sup>[46]</sup>



Hình 3. 5 Hình ảnh minh họa phần mềm phát triển sản phẩm HEROKU<sup>[47]</sup>

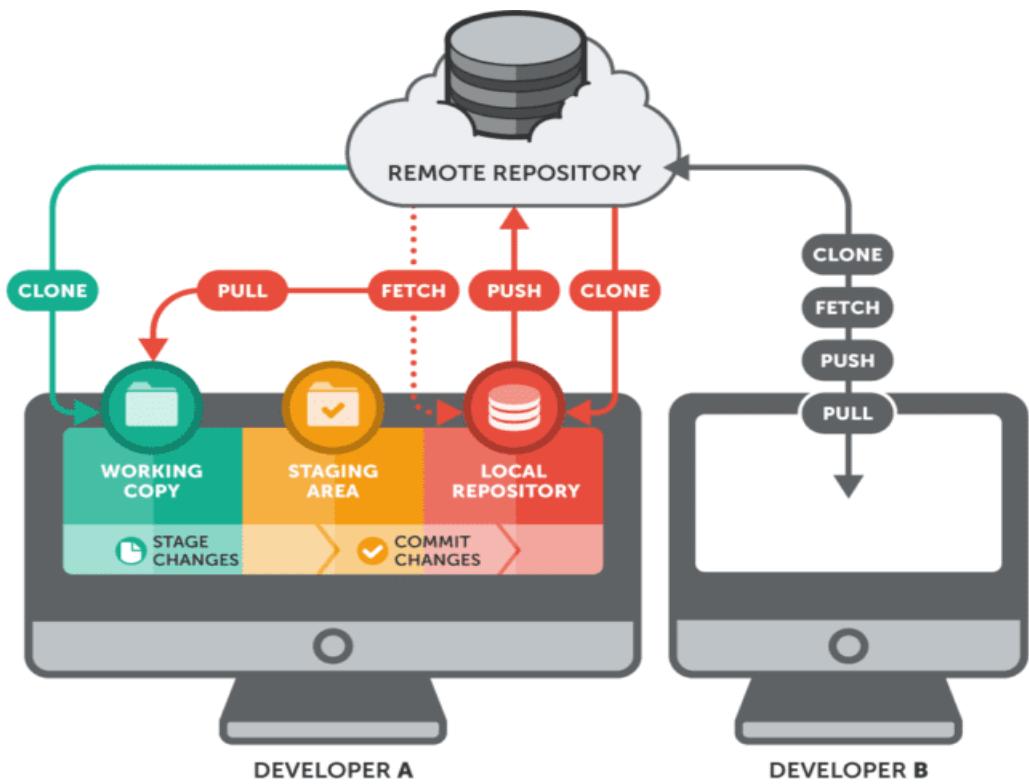
Những ưu điểm của phần mềm phát triển sản phẩm HEROKU:

- Cung cấp trải nghiệm ưu việt, hệ sinh thái dịch vụ đa dạng.
- Hỗ trợ kết nối với salesforce cho phép người dùng đồng bộ 2 chiều salesforce.
- Heroku có thể hoạt động trọn tru với tất cả những ngôn ngữ lập trình phổ biến nhất như Nodejs, Ruby, PHP, Python hay Java.

Phần mềm lưu trữ mã nguồn GitHub:

GitHub là một dịch vụ lưu trữ trên web dành cho các dự án có sử dụng hệ thống kiểm soát Git revision.

GitHub cung cấp chức năng mạng xã hội như là nguồn cấp dữ liệu, người theo dõi và biểu đồ mạng để các Developer học hỏi kinh nghiệm làm việc thông qua lịch sử làm việc.<sup>[48]</sup>



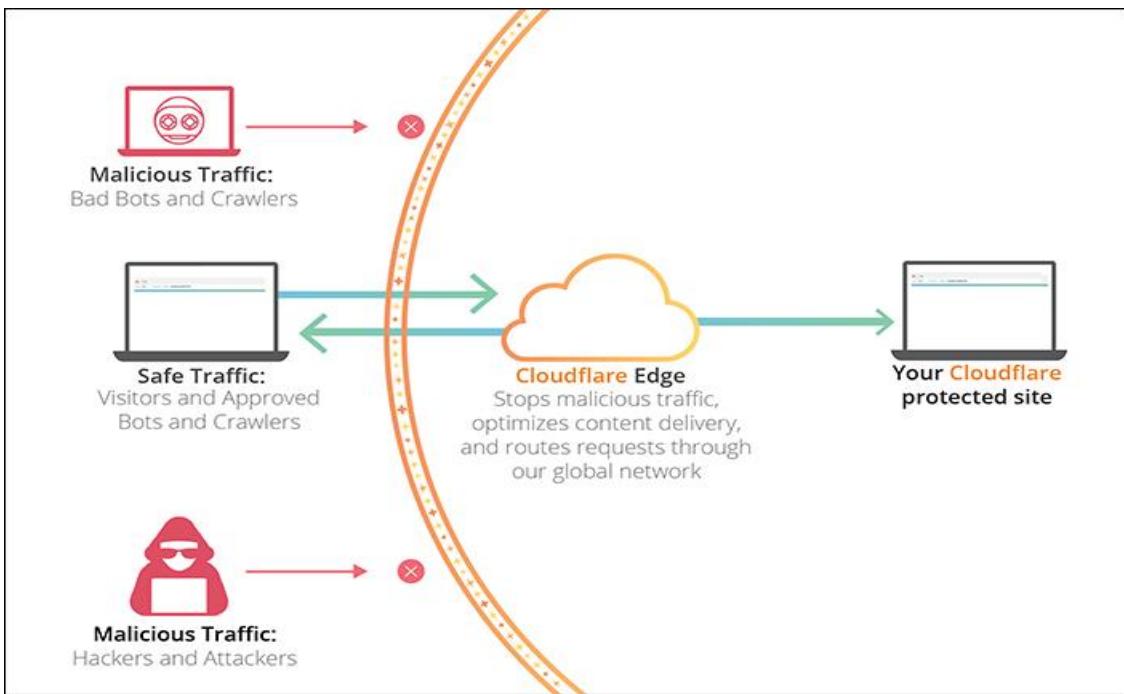
Hình 3. 6 Hình ảnh minh họa phần mềm lưu trữ mã nguồn GitHub<sup>[49]</sup>

Những ưu điểm của phần mềm lưu trữ mã nguồn GitHub:

- Dễ dàng quản lý Source Code.
- Dễ dàng theo dõi những thay đổi .
- Khẳng định chuyên môn.
- Cải thiện khả năng Code.
- Kho tài nguyên tuyệt vời.

Phần mềm điều phối lượng truy cập giữa máy chủ CloudFlare:

CloudFlare chính là dịch vụ DNS trung gian, nơi mà điều phối lượng truy cập giữa máy chủ với máy của khách hàng qua lớp bảo vệ CloudFlare. Tức là thay vì phải truy cập trực tiếp vào website thông qua phân giải tên miền DNS thì có thể sử dụng máy chủ phân giải tên của CloudFlare. Ngoài các chức năng trên, CloudFlare còn cung cấp nhiều dịch vụ như CNS, SPDY, tường lửa chống Ddos, Chứng chỉ số SSL,...<sup>[50]</sup>



Hình 3. 7 Hình ảnh minh họa phần mềm điều phối lượng truy cập giữa máy chủ CloudFlare<sup>[51]</sup>

Những ưu điểm của phần mềm điều phối lượng truy cập giữa máy chủ CloudFlare:

- Hạn chế truy cập trực tiếp vào máy chủ, giúp tiết kiệm được băng thông cho máy chủ.
- Tăng khả năng bảo mật của website, hạn chế sự tấn công của DDoS.
- Sử dụng cloudflare như SSL miễn phí nhằm thêm giao thức HTTPS cho website.
- Hạn chế truy cập từ những quốc gia đã được chỉ định.
- Cấm truy cập với những IP nhất định.
- Công nghệ tường lửa trong ứng dụng website.

### 3.1.3 Các bước thực hiện xây dựng hệ thống:

- Để thuận tiện cho làm thuật toán K-fold cross validation và SVM, ta cần thêm các thư viện như pandas để đọc file .csv(tập ví dụ), từ sklearn ta import SVM, numpy để cho phép làm việc hiệu quả với ma trận và mảng. Đặc biệt có thêm thư viện tkinter để tạo form cho bài.

```

from tkinter import *
from tkinter.ttk import *
from tkinter import messagebox
import tkinter
import numpy as np
import pandas as pd
from sklearn.svm import SVC
from sklearn import svm
from sklearn.model_selection import KFold
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn import decomposition
from sklearn import datasets

```

*Hình 3. 8 Hình ảnh các thư viện được sử dụng trong bài toán*

- Đầu tiên ta sẽ tiến xử lý dữ liệu đầu vào:

```

original_data = pd.read_csv('TongHop.csv')

original_data_null = original_data.replace(['#NULL!', 'N', 'null', 'NaN'], [np.nan, np.nan, np.nan, np.nan])

later_data = original_data_null[~pd.isna(original_data_null).any(axis=1)].reset_index(drop=True)

later_data.to_csv('thu.csv', index=False)

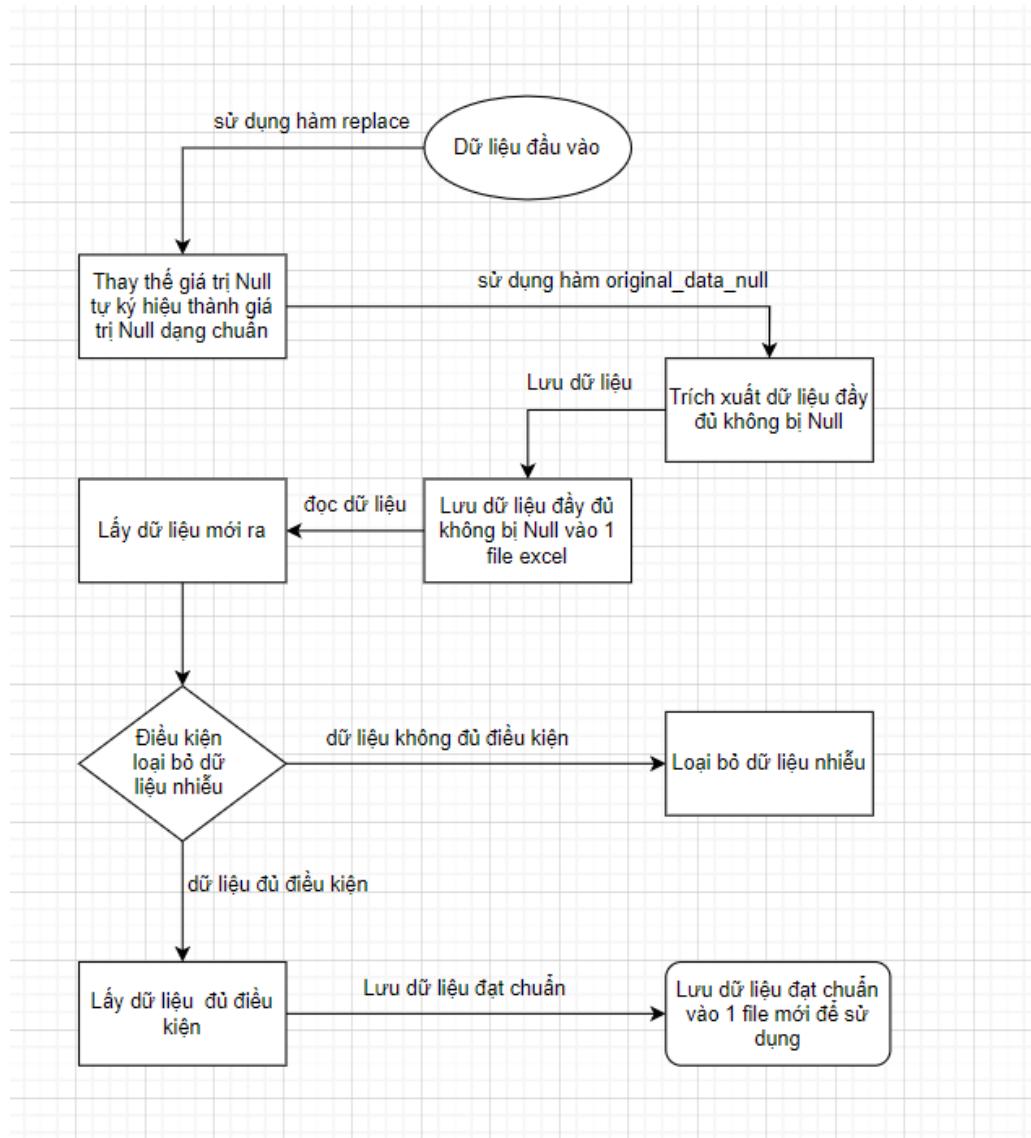
df = pd.read_csv('thu.csv')

data_condition = df[(df['tuoi'] >= 0) & (df['chisokhoi'] >= 0) & (df['nhiptim'] >= 0) & (df['huyetap'] >= 0) & (df['duongmau'] >= 0) & (df['cholesterol'] >= 0) & (df['triglycerid'] >= 0) & (df['tiensubenh'] >= 0) & (df['RANKIN'] >= 0) & ((df['gioitinh'] == 1) | (df['gioitinh'] == 2))]

data_condition.to_csv('thu_nghiem.csv', index=False)

```

Sơ đồ hoạt động:



Hình 3. 9 Hình ảnh sơ đồ hoạt động của tiền xử lý dữ liệu đầu vào

- Tìm mô hình tốt nhất của thuật toán:

$\min 1 = 0$

$a=0$

for abc in range(10000):

```
data = pd.read_csv('thu_nghiem.csv')
```

```
data_Train, data_Test = train_test_split(data, test_size=0.3 , shuffle = True)
```

$k = 10$

```
kf = KFold(n_splits=k, random_state=None)
```

$a = a+1$

$i=1$

$\min = 0$

```

for train_index, test_index in kf.split(data_Train):
    X_train, X_test = data_Train.iloc[train_index,:-1], data_Train.iloc[test_index, :-1]
    y_train, y_test = data_Train.iloc[train_index, -1], data_Train.iloc[test_index, -1]

    SVM = svm.SVC()
    ## Huấn luyện (đào tạo) mô hình SVM với tập dữ liệu huấn luyện
    SVM.fit(X_train,y_train)

    ## dùng mô hình SVM đã huấn luyện để dự đoán dữ liệu cho tập kiểm thử k_form
    Y_pred_test=SVM.predict(X_test)

    ## sử dụng hàm accuracy_score để tính toán điểm chính xác cho một tập hợp các nhãn được dự đoán so với các nhãn thực.
    accuracy_kf = accuracy_score(Y_pred_test,y_test)
    good_svm_kf = SVM.fit(X_train, y_train)

    ## dùng mô hình SVM đã huấn luyện để dự đoán dữ liệu cho tập kiểm thử thực tế
    y_predict=good_svm_kf.predict(data_Test.iloc[:,:-1])
    y_reality = np.array(data_Test.iloc[:, -1])

    ## sử dụng hàm accuracy_score để tính toán điểm chính xác cho một tập hợp các nhãn được dự đoán so với các nhãn thực tế .
    accuracy_reality = accuracy_score(y_predict, y_reality)

    ## tính số dữ liệu dự đoán đúng trên tập kiểm thử thực tế
    x = (len(y_reality) * accuracy_reality)
    # so sánh độ chính xác tìm ra mô hình tốt nhất qua từng vòng lặp
    if(accuracy_reality > min):
        min = accuracy_reality
        y = (len(y_reality) * accuracy_reality)
        good_svm_reality = SVM.fit(X_train, y_train)
    i = i+1
    if(min > min1):
        min1 = min
        z = y
    very_good_svm = good_svm_reality

```

```

print("SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA_TEST THỰC TẾ LÀ :",'Đúng', round(z), "trên
tổng", len(y_reality))

print('==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ '['a,'] LÀ :',round(min1 * 100,3),'%', '\n')

import pickle

#luu mo hinh tot nhat vao model.pkl de su dung

score_svm = pickle.load(open('E:\VD\TN\CT\score_svm.pkl','rb'))

print('==> Tỷ Lệ chính xác cao nhất của lần train trước đó được lưu là :',round(score_svm * 100,3),'%')

# chỉ lưu mô hình có tỷ lệ đúng cao nhất qua các lần train

if(min1 > score_svm):

    score_svm = min1

    pickle.dump(score_svm, open('E:\VD\TN\CT\score_svm.pkl','wb'))

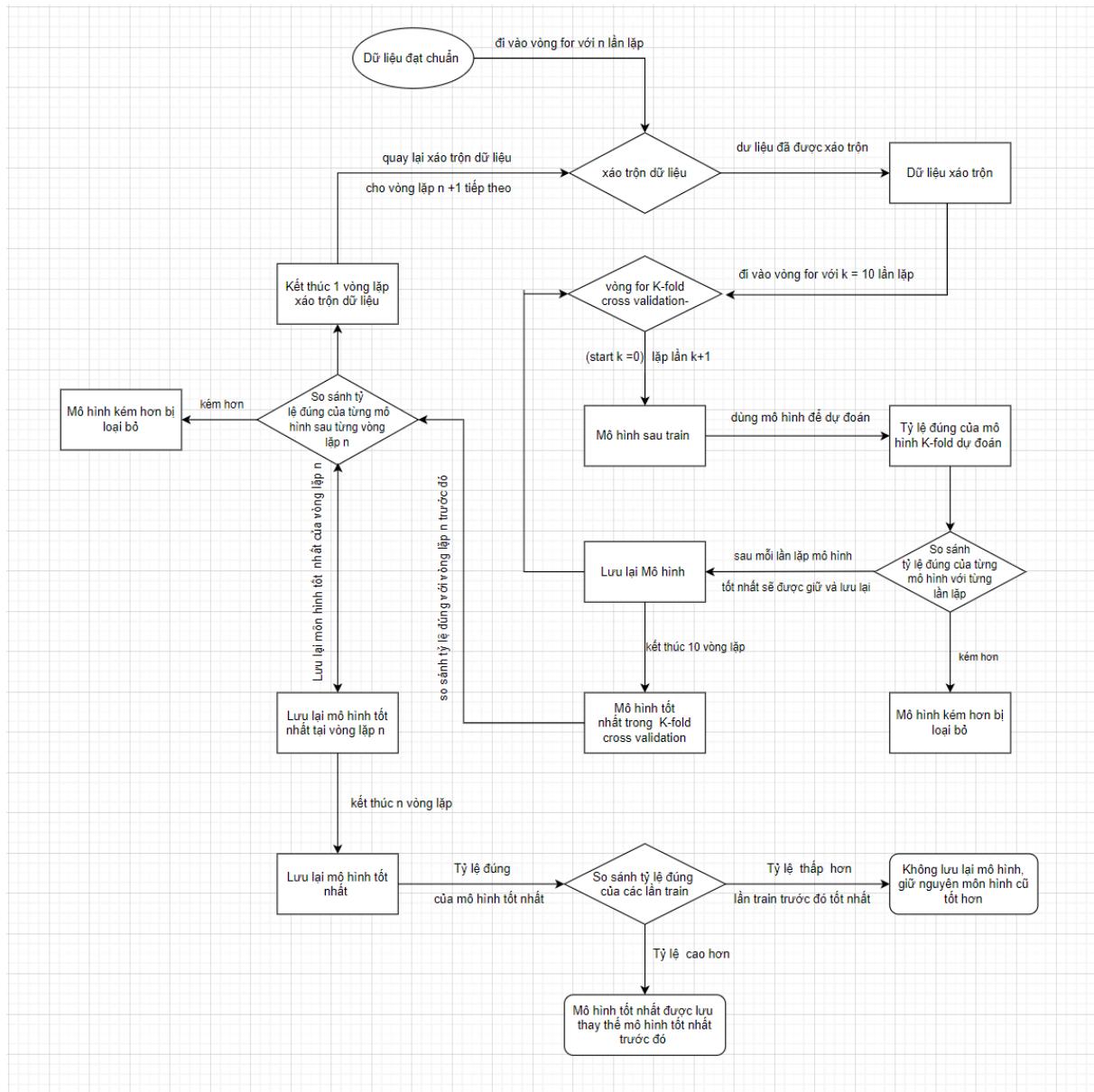
    pickle.dump(very_good_svm, open('E:\VD\TN\CT\model.pkl','wb'))


model = pickle.load(open('E:\VD\TN\CT\model.pkl','rb'))

print('==> Tỷ Lệ chính xác cao nhất từ các lần train là :', round(score_svm * 100,3),'%')

```

Sơ đồ hoạt động :



Hình 3. 10 Hình ảnh sơ đồ hoạt động của thuật toán

- Tạo giao diện:

# Bắt lỗi nhập dữ liệu

```

def showPredict():
    if(txttuo1.get() == "" or txtgioitinh.get() == "" or txtchisokhoi.get() == "" or txtnhiptim.get() == "" or
       txthuyetap.get() == "" or txtduongmau.get() == "" or txtcholesterol.get() == "" or txttriglycerid.get() ==
       "" or txttiensubenh.get() == "" or txtRANKIN.get() == ""):
        messagebox.showerror("Lỗi Thiếu Thông Tin ", "Vui lòng điền đầy đủ thông tin.");
  
```

```

x_input = np.array([float(txttuo1.get()), float(txtgioitinh.get()), float(txtchisokhoi.get()),
                   float(txtnhiptim.get()), float(txthuyetap.get()), float(txtduongmau.get()), float(txtcholesterol.get()),
                   float(txttriglycerid.get()), float(txttiensubenh.get()), float(txtRANKIN.get())]).reshape(1, -1)
  
```

```

if ((float(txttuoi.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng số tuổi ")
    x_input = np.nan;

if ( (float(txtgioitinh.get()) <= 0) or (float(txtgioitinh.get()) >= 3) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng giới tính : 1 Là Nam | 2 Là Nữ ")
    x_input = np.nan;

if ((float(txtchisokhoi.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Khối ")
    x_input = np.nan;

if ((float(txtnhiptim.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Nhịp Tim ")
    x_input = np.nan;

if ((float(txthuyetap.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Huyết Áp ")
    x_input = np.nan;

if ((float(txtduongmau.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Đường Máu ")
    x_input = np.nan;

if ((float(txtcholesterol.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Cholesterol ")
    x_input = np.nan;

if ((float(txttriglycerid.get()) <= 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Chỉ Số Triglycerid ")
    x_input = np.nan;

if ((float(txttiensubenh.get()) < 0) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng Tiền Sử Bệnh ")
    x_input = np.nan;

if ( (float(txtRANKIN.get()) < 0) or (float(txtRANKIN.get()) >= 7) ):
    messagebox.showerror("Lỗi Sai Thông Tin", " Vui Lòng điền đúng RANKIN ")

```

```

x_input = np.nan;

y_dd = model.predict(x_input)
print('Kết Quả :', y_dd[0])
print('input :', x_input)

if(y_dd == 1):
    messagebox.showinfo("Kết quả dự đoán: ", "Bạn Có Nguy Cơ Bị Đột Quỵ ")
else :
    messagebox.showinfo("Kết quả dự đoán: ", "Sức Khỏe Bình Thường ")

def dochinhxac():

    messagebox.showinfo("Khả năng dự đoán của SVM", "Độ chính xác của phương pháp: " +
str(round(score_svm * 100,3)) + '%')

#khởi tạo cửa sổ giao diện
window = Tk()
window.geometry("550x300")
window.title("DỰ ĐOÁN NGUY CƠ ĐỘT QUỴ")

#Tạo thông số
lbltuoi = tkinter.Label (window, text =("Tuổi"), font = ("Arial",10))
lbltuoi.grid(column = 1, row = 2)
txttuoi = Entry(window, width = 25)
txttuoi.grid(column = 2, row = 2)

lblgioitinh = tkinter.Label (window, text =("Giới Tính"), font = ("Arial",10))
lblgioitinh.grid(column = 1, row = 4)
txtgioitinh = Entry(window, width = 25)
txtgioitinh.grid(column = 2, row = 4)

lblchisokhoi = tkinter.Label (window, text =("Chỉ Số Khối"), font = ("Arial",10))
lblchisokhoi.grid(column = 1, row = 6)
txtchisokhoi = Entry(window, width = 25)
txtchisokhoi.grid(column = 2, row = 6)

lblnhiptim = tkinter.Label (window, text =("Nhịp Tim "), font = ("Arial",10))
lblnhiptim.grid(column = 1, row = 8)
txtnhiptim = Entry(window, width = 25)

```

```
txtnhiptim.grid(column = 2, row = 8)
```

```
lblhuyetap = tkinter.Label (window, text =("Huyết Áp"), font = ("Arial",10))
```

```
lblhuyetap.grid(column = 1, row = 10)
```

```
txthuyetap = Entry(window, width = 25)
```

```
txthuyetap.grid(column = 2, row = 10)
```

```
lblduongmau = tkinter.Label (window, text =("Đường Máu"), font = ("Arial",10))
```

```
lblduongmau.grid(column = 6, row = 2)
```

```
txtduongmau = Entry(window, width = 25)
```

```
txtduongmau.grid(column = 7, row = 2)
```

```
lblcholesterol = tkinter.Label (window, text =("Cholesterol"), font = ("Arial",10))
```

```
lblcholesterol.grid(column = 6, row = 4)
```

```
txtcholesterol = Entry(window, width = 25)
```

```
txtcholesterol.grid(column = 7, row = 4)
```

```
lbltriglycerid = tkinter.Label (window, text =("Triglycerid"), font = ("Arial",10))
```

```
lbltriglycerid.grid(column = 6, row = 6)
```

```
txttriglycerid = Entry(window, width = 25)
```

```
txttriglycerid.grid(column = 7, row = 6)
```

```
lbltiensubenh = tkinter.Label (window, text =("Tiền Sử Bệnh"), font = ("Arial",10))
```

```
lbltiensubenh.grid(column = 6, row = 8)
```

```
txttiensubenh = Entry(window, width = 25)
```

```
txttiensubenh.grid(column = 7, row = 8)
```

```
lblRANKIN = tkinter.Label (window, text =("RANKIN"), font = ("Arial",10))
```

```
lblRANKIN.grid(column = 6, row = 10)
```

```
txtRANKIN = Entry(window, width = 25)
```

```
txtRANKIN.grid(column = 7, row = 10)
```

### #Tạo nút bấm

```
btketqua = Button (window, text = "Kết Quả",command = showPredict)
```

```
btketqua.place(x = 75, y = 200)
```

```
btdochinhxac = Button (window, text = "Độ Chính Xác",command = dochinhxac)
```

```
btdochinhxac.place(x = 225, y = 200)
```

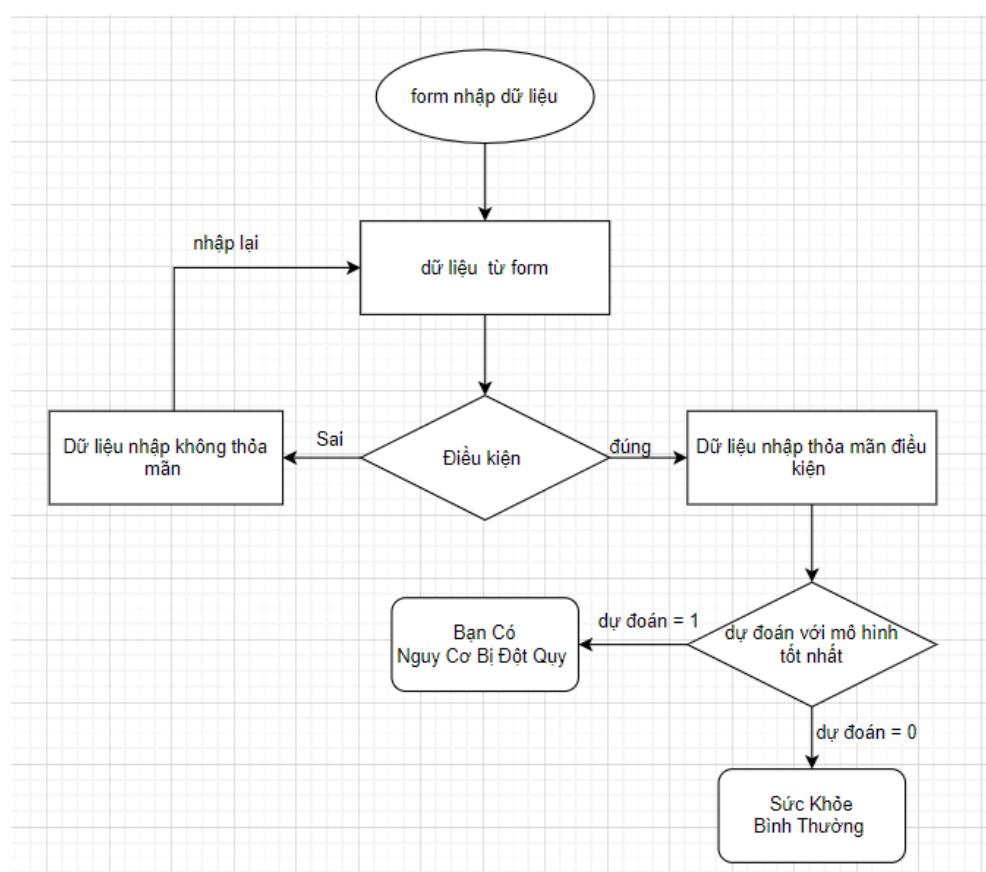
```

btthoat = Button (window, text = "Thoát", command = exit)
btthoat.place(x = 400, y = 200)

window.mainloop()

```

Sơ đồ hoạt động:



Hình 3. 11 Hình ảnh sơ đồ hoạt động của form dữ liệu

- Thuật toán kết nối giao diện dùng:

```

import numpy as np
from flask import Flask, request, render_template
import pickle

```

```

app = Flask(__name__)

model = pickle.load(open('models/model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])

```

```

def predict():

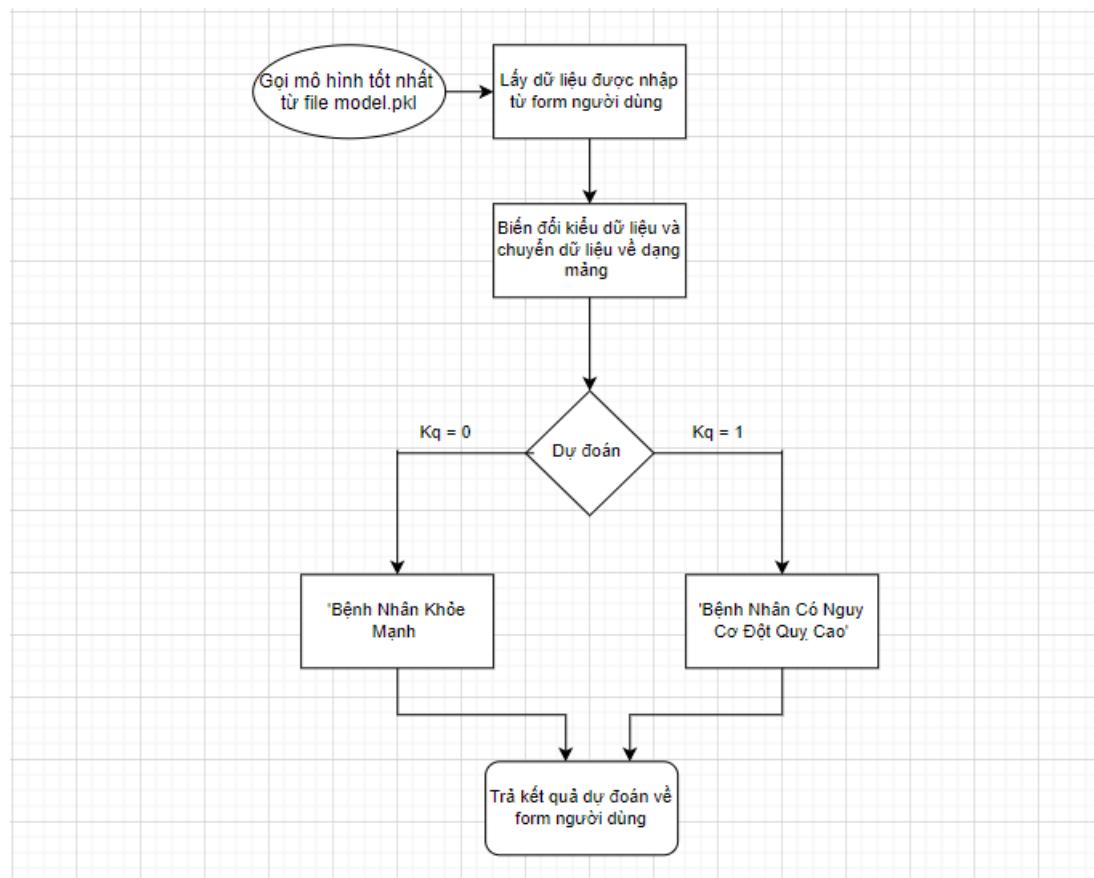
    int_features = [float(x) for x in request.form.values()]
    features = [np.array(int_features)]
    prediction = model.predict(features)

    output = prediction[0]
    if(output == 1):
        return render_template('index.html', prediction_text='Bệnh Nhân Có Nguy Cơ Đột Quy Cao')
    else:
        return render_template('index.html', prediction_text='Bệnh Nhân Khôe Mạnh')

if __name__ == "__main__":
    app.run()

```

Sơ đồ hoạt động :



Hình 3. 12 Hình ảnh sơ đồ hoạt động của thuật toán kết nối giao diện

### 3.2 Đánh giá và demo giao diện chương trình:

### 3.2.1. Form nhập dữ liệu kiểm tra:

Form nhập dữ liệu gồm 10 thuộc tính quan trọng nhất ảnh hưởng tác động trực tiếp tới kết quả bài toán.

Hình 3. 13 Hình minh họa form nhập dữ liệu đầu vào bài toán

### 3.2.2. Khả năng dự đoán:

Sau khi nhập dữ liệu đầu vào thuật toán AI sẽ dựa trên dữ liệu đầu vào và mô hình dự đoán tốt nhất đã được huấn luyện để dự đoán đưa ra kết quả đầu ra của người dùng.

Với bài toán hiện tại khả năng dự đoán sẽ đưa ra 2 kết quả “Bệnh Nhân Có Nguy Cơ Đột Quy Cao” hoặc “Bệnh Nhân Khỏe Mạnh” với độ chính xác của thuật toán nên đến 97.059% . sau khi train 10000 lần.

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9993 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9994 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9995 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9996 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9997 ] LÀ : 97.059 %

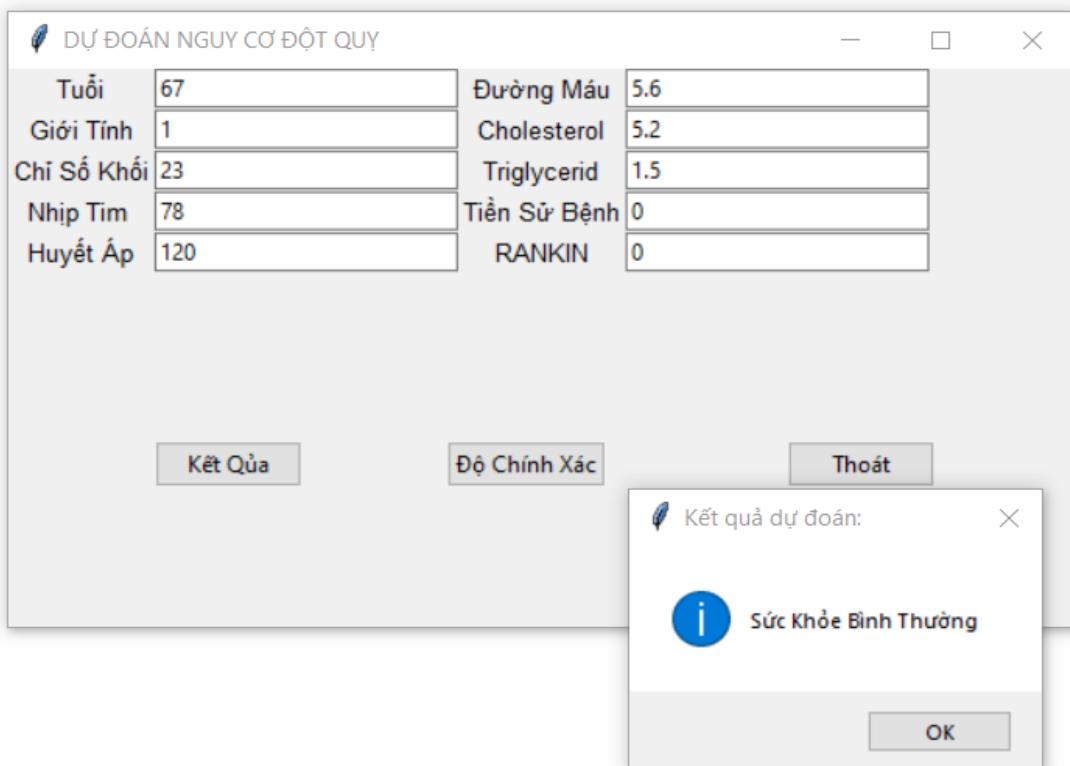
SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9998 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9999 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
==> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 10000 ] LÀ : 97.059 %

==> Tỷ Lệ chính xác cao nhất của lần train trước đó được lưu là : 95.098 %  
==> Tỷ Lệ chính xác cao nhất từ các lần train là : 97.059 %

Hình 3. 14 Hình minh họa hoạt kết quả dự đoán



Hình 3. 15 Hình minh họa dữ liệu đầu và kết quả dự đoán tại form

### 3.2.3. Giao diện người dùng:

Thuật toán AI dự đoán bệnh đột quỵ ở người già , không những chỉ có giao diện ở tại máy mà còn có giao diện trên web . Giúp mọi người có thể sử dụng có thể sử dụng cho dù ở mọi nơi miễn là thiết bị có kết nối internet.

Giao diện người dùng đơn giản dễ hiểu, dễ sử dụng, có hướng dẫn chi tiết từng thành phần, lời khuyên và hoàn toàn miễn phí.

❖ Đường Link sử dụng giao diện thuật toán :

<https://dudoanbenhdotquy.click/>

FORM ĐỰ ĐOÁN NGUY CƠ ĐỘT QUỴ		HƯỚNG DẪN SỬ DỤNG	
Tuổi	<input type="text"/>	Giới Tính :	<ul style="list-style-type: none"><li>Nhập số [1] nếu bạn là Nam .</li><li>Nhập số [2] nếu bạn là Nữ .</li></ul>
Giới Tính	<input type="text"/>	Chi Số Khối :	<ul style="list-style-type: none"><li>Chi số khối = cân nặng : [ chiều cao x 2 ].</li></ul>
Chi Số Khối	<input type="text"/>	Tiền Sử Bệnh :	<ul style="list-style-type: none"><li>Nhập chi số Lần đã từng bị đột quỵ ví dụ :</li><li>Nhập số [0] chưa từng bị đột quỵ.</li><li>Nhập số [1] với 1 Lần đã từng bị đột quỵ.</li><li>Nhập số [2] với 2 Lần đã từng bị đột quỵ.</li></ul>
Nhịp Tim	<input type="text"/>	RanKin [ Thang Điểm Nhận Thức ] :	<ul style="list-style-type: none"><li>Nhập số [0] với biểu hiện nhận thức bình thường.</li><li>Nhập số [1] triệu chứng nhẹ về thần kinh nhưng vẫn có khả năng làm những việc hàng ngày.</li><li>Nhập số [2] có chứng nhẹ biểu hiện như không còn khả năng làm những việc như trước kia.</li><li>Nhập số [3] có chứng vừa biểu hiện như cần sự giúp đỡ nhất định nhưng vẫn có thể đi lại.</li><li>Nhập số [4] có chứng tương đối nặng biểu hiện như không tự đi lại, không tự phục vụ.</li><li>Nhập số [5] có chứng nặng biểu hiện như liệt giường và sinh không tự chủ.</li></ul>
Huyết Áp	<input type="text"/>	LỜI KHUYÊN	<ul style="list-style-type: none"><li>Trường Hợp gặp các triệu chứng : Cơ thể mệt mỏi, đột nhiên cảm thấy không còn sức lực, té cung mặt hoặc một nửa mặt, nụ cười bị méo mó, cử động khó khăn.</li><li>Trường Hợp dự đoán "Bệnh nhân có nguy cơ đột quỵ cao."</li></ul> <p>➔ Khả năng cao của các triệu chứng khởi phát độ quý , cần đến ngay cơ sở Y Tế gần nhất để thăm khám.</p>
Đường Máu	<input type="text"/>		
Cholesterol	<input type="text"/>		
Triglycerid	<input type="text"/>		
Tiền Sử Bệnh	<input type="text"/>		
RanKin	<input type="text"/>		
		<input type="button" value="Chuẩn Đoán"/>	
<div style="border: 1px solid black; height: 50px;"></div>			

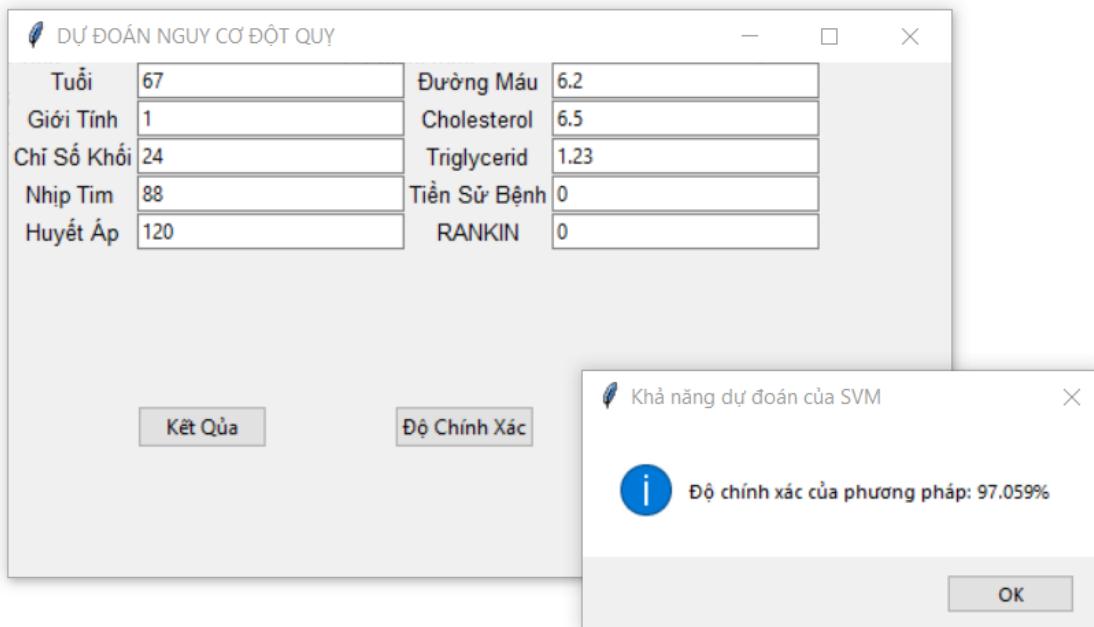
Hình 3. 16 Hình minh họa giao diện người dùng

### 3.2.4. Đánh giá kết quả:

Sau khi huấn luyện mô hình được 25 lần, mỗi lần huấn luyện, dữ liệu, mô hình được xáo trộn, xây dựng, so sánh với nhau 10000 lần. Thời gian chạy hoàn thành mỗi lần huấn luyện là 37 phút.

Kết quả: Trong 25 lần huấn luyện có 17 lần kết quả đạt 96.078% và 8 lần kết quả đạt 97.059%.

Mô hình hiện tại dùng để dự đoán có độ chính xác là : 97.059%.



Hình 3. 17 Hình minh họa độ chính xác của thuật toán

Kết quả dự đoán của thuật toán SVM khi kết hợp với thuật toán K-fold cross validation đạt và xáo trộn dữ liệu đầu vào mỗi lần lặp là: **97.059 %.**

### 3.2.5 . Nhận xét:

Dựa trên tỷ lệ dự đoán đúng của mô hình trên dữ liệu thực tế trên cùng một bộ dữ liệu ta có tỷ lệ dự đoán đúng của các bước xây dựng phương pháp áp dụng cho bài toán xây dựng với phương pháp SVM:

- Tỷ lệ dự đoán đúng là : 87.255%

---

ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN SVM

==> Số dự đoán đúng trên tập dữ liệu data\_test thực tế : 89 trên tổng 102  
==> Độ chính xác của thuật toán SVM là : 87.255 %

Hình 3. 18 Hình minh họa độ chính xác của thuật toán svm

Xây dựng với phương pháp K-fold cross validation kết hợp SVM:

- Tỷ lệ dự đoán là 88.235%.

---

MÔ HÌNH TỐT NHẤT K-fold cross validation + SVM

---

=> Mô hình tốt nhất là : 2  
=> Độ chính xác của mẫu trên K trên tập data\_train là : 100.0 %  
=> Số dự đoán đúng trên tập dữ liệu data\_test thực tế : 90 trên tổng 102  
=> Tỉ Lệ Chính xác của K-fold cross validation + SVM Là : 88.235 %

---

Hình 3. 19 Hình minh họa độ chính xác của thuật toán K-fold cross validation kết hợp SVM

Xây dựng với phương pháp SVM kết hợp xáo trộn dữ liệu mỗi lần vào (10000 lần):

- Tỷ lệ chính xác: 95.098 %.

=> Độ chính xác của lần lặp 9995 là 95.098 %  
=> Độ chính xác của lần lặp 9996 là 95.098 %  
=> Độ chính xác của lần lặp 9997 là 95.098 %  
=> Độ chính xác của lần lặp 9998 là 95.098 %  
=> Độ chính xác của lần lặp 9999 là 95.098 %  
=> Độ chính xác của lần lặp 10000 là 95.098 %

---

ĐỘ CHÍNH XÁC CỦA SVM + XÁO TRỘN DỮ LIỆU ĐẦU VÀO

---

=> Số dự đoán đúng trên tập dữ liệu data\_test thực tế : 97 trên tổng 102  
=> Tỉ Lệ Chính xác của K-fold cross validation + SVM Là : 95.098 %

Hình 3. 20 Hình minh họa độ chính xác của thuật toán SVM kết hợp xáo trộn dữ liệu

Xây dựng với sự kết hợp của 3 phương pháp K-fold cross validation kết hợp SVM + xáo trộn dữ liệu mỗi lần vào (10000 lần):

- Tỷ lệ chính xác: 97.059%

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9995 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9996 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9997 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9998 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 9999 ] LÀ : 97.059 %

SỐ DỰ ĐOÁN ĐÚNG TRÊN TẬP DỮ LIỆU DATA\_TEST THỰC TẾ LÀ : Đúng 99 trên tổng 102  
=> ĐỘ CHÍNH XÁC CỦA THUẬT TOÁN LẦN THỨ [ 10000 ] LÀ : 97.059 %

=> Tỷ Lệ chính xác cao nhất của lần train trước đó được lưu là : 95.098 %  
=> Tỷ Lệ chính xác cao nhất từ các lần train là : 97.059 %

*Hình 3. 21 Hình minh họa độ chính xác kết hợp của ba phương pháp*

❖ Nhận xét:

Kết quả nghiên cứu cho thấy:

1. Nếu chỉ sử dụng thuật toán SVM thì tỷ lệ dự đoán chính xác bệnh đột quy là 87.255%.

2. Khi sử dụng thuật toán để xuất là kết hợp K-fold cross validation với SVM thì tỷ lệ dự đoán chính xác được nâng lên 88.235%. Sau đó, tiếp tục cải tiến chỉ dùng xáo trộn dữ liệu đầu vào ra tỷ lệ đúng, sau so sánh nhiều lần lặp với nhau, ở đây là 10000 lần được tỷ lệ đúng cao nhất là 95.098% . Cuối cùng, tiếp tục cải tiến nâng cấp xây dựng với sự kết hợp của 3 phương pháp k-fold cross validation kết hợp SVM cộng xáo trộn dữ liệu mỗi lần vào (10000 lần) nâng được tỷ lệ chính xác lên 97.059%.

## KẾT LUẬN

Kết quả nghiên cứu chỉ ra rằng nếu chỉ dùng thuật toán Support Vector Machine (SVM) thì tỉ lệ dự đoán chính xác bệnh đột quy là (SVM) 87.255%. Sau khi kết hợp phương pháp k-fold cross validation với Support Vector Machine đã nâng tỷ lệ dự đoán đúng lên 95.098% . Còn khi tiếp tục kết hợp với xáo trộn dữ liệu đầu vào mỗi lần để nâng cao quá trình học hỏi của thuật toán sau cùng tỷ lệ đúng được nâng lên 97.059%.

Không những thành công về xây dựng thuật toán mà em còn thành công xây dựng được Form giao diện người dùng và thuật toán được đóng gói, lưu trữ trên máy chủ ảo Heroku (đường link sử dụng giao diện thuật toán dự đoán đột quy: <https://dudoanbenhdotquy.click/> ) để mọi người bệnh đều có thể sử dụng miễn phí. Điều này giúp người bệnh trong việc xác định dự đoán việc mình có bị đột quy hay không để có những biện pháp ứng phó tốt nhất.

Tuy nhiên, trong nghiên cứu này dữ liệu thu thập từ bệnh viện, người bệnh chưa phong phú nên có thể chưa đánh giá hết được độ chính xác của thuật toán đề xuất. Trong tương lai em hy vọng sẽ bổ sung và nghiên cứu tiếp để nâng cấp thuật toán hơn nữa và biến nó không chỉ dừng lại ở dạng một phần mềm mà có thể phát triển để trở thành thiết bị được sử dụng rộng rãi tại các bệnh viện và các hộ gia đình giúp người bệnh không cần phải đến tận bệnh viện để thăm khám, đồng thời tiết kiệm được chi phí, thời gian và giảm tải cho các bệnh viện, đặc biệt là ở tuyến trung ương.

## TÀI LIỆU THAM KHẢO

- [1] <https://wikikienthuc.com/machine-learning/>
- [2] <https://topdev.vn/blog/machine-learning-la-gi/>
- [3] <https://lytuong.net/hoc-may-machine-learning-la-gi/>
- [4] <https://machinelearningcoban.com/2016/12/27/categories/>
- [5] <https://blog.vinbigdata.org/supervised-learning-va-unsupervised-learning-khac-biet-la-gi/>
- [6] <https://teksands.ai/blog/semi-supervised-learning>
- [7] <https://www.javatpoint.com/reinforcement-learning>
- [8] <https://tuanvanle.wordpress.com/2017/04/22/nhung-cot-moc-quan-trong-cua-machine-learning/>
- [9] <https://lytuong.net/hoc-may-machine-learning-la-gi/>
- [10] <https://hai.doimoisangtao.vn/tp-ho-chi-minh-buoc-dau-ung-dung-tri-tue-nhan-tao-trong-y-te/>
- [11] <https://startup.vnexpress.net/tin-tuc/xu-huong/tuong-lai-cong-nghe-ai-4061092.html>
- [12] <https://www.enterknow.com/lich-su-cua-xe-o-to-tu-lai-ai-da-phat-minh>
- [13] <https://steamcommunity.com/sharedfiles/filedetails/?id=2657310429>
- [14] <https://tek4.vn/khoa-hoc/machine-learning-co-ban/tong-quan-ve-bai-toan-phan-lop>
- [15] <https://machinelearningcoban.com/2017/01/21/perceptron/>

- [16] <https://amueller.github.io/aml/02-supervised-learning/06-linear-models-classification.html>
- [17] <https://aws.amazon.com/vi/what-is/logistic-regression/>
- [18] <https://www.raona.com/los-10-algoritmos-esenciales-machine-learning/>
- [19] <https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB>
- [20] <https://machinelearningcoban.com/2017/04/09/sm/>
- [21] <https://rpubs.com/lengockhanhi/350533>
- [22] [https://www.youtube.com/watch?v=S2\\_SZ7WRy-Q](https://www.youtube.com/watch?v=S2_SZ7WRy-Q)
- [23] <https://thetalog.com/statistics/independent-component-analysis/>
- [24] <https://phantichkinhdoanh.net/knn-thuat-toan-lang-gieng-gan-nhat-k-nearest-neighbor-la-gi/>
- [25] <https://viblo.asia/p/knn-k-nearest-neighbors-1-djeZ14ejKWz>
- [26] <https://tuoitre.vn/moi-ngay-tphcm-co-khoang-300-benh-nhan-dot-quy-so-nguoi-tre-mac-benh-tang-nhanh-20211124114637988.htm>
- [27] <https://hongngochospital.vn/benh-dot-quy/>
- [28] <https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/suc-khoe-tong-quat/cac-bien-chung-co-the-gap-sau-dot-quy/>
- [29] <https://benhvienthucuc.vn/dot-quy-la-gi/>
- [30] <https://thanhnien.vn/4-dau-hieu-can-bao-nguy-co-dot-quy-nao-post907532.html>
- [31] <https://quantrimang.com/khoa-hoc/he-thong-ai-co-the-du-doan-nguy-co-dot-quy-chinh-xac-hon-ca-bac-si-133479>

- [32] <https://kinhdientamquoc.vn/sap-xep-ngau-nhien-trong-excel/>
- [33] <https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB>
- [34] <https://www.mltut.com/k-fold-cross-validation-in-machine-learning-how-does-k-fold-work/>
- [35] Tham khảo slide “Học máy” của cô TS.Nguyễn Thị Kim Ngân.
- [36] <https://miae.vn/2021/01/18/k-fold-cross-validation-tuyet-chieu-train-khi-it-du-lieu/>
- [37] <https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/>
- [38] <https://aws.amazon.com/vi/what-is/python/>
- [39] <https://vtiacademy.edu.vn/nhung-dieu-can-biet-ve-lap-trinh-python.html>
- [40] <https://topdev.vn/blog/html-la-gi/>
- [41] <https://www.youtube.com/watch?v=iG2jotQo9NI>
- [42] <https://mobifirst.myharavan.com/products/python-3-10-0-ngon-ngu-lap-trinh-co-ban>
- [43] <https://irender.vn/python-3-10-va-nhung-tinh-nang-moi-tuyet-voi/>
- [44] <https://vietnix.vn/visual-studio-code-la-gi/>
- [45] <https://fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gicac-tinh-nang-noi-bat-cua-visual-studio-code-146213>
- [46] <https://itnavi.com.vn/blog/heroku-la-gi>
- [47] <https://www.logicline.de/blog/2015/11/a-more-technical-look-at-heroku-about-procfiles-dynos-and-the-slug/>
- [48] [https://zinpro.vn/vn/github-la-gi-github-duoc-su-dung-nhu-the-nao\\_1154.html](https://zinpro.vn/vn/github-la-gi-github-duoc-su-dung-nhu-the-nao_1154.html)

- [49] <https://www.bacs.vn/vi/blog/cong-cu-ho-tro/huong-dan-cai-dat-github-desktop-va-su-dung-17735.html>
- [50] <https://vietnix.vn/cloudflare-la-gi/>
- [51] <https://codelearn.io/sharing/han-che-tan-cong-web-voi-cloudflare>
- [52] <https://hoatieu.vn/bieu-mau/mau-phieu-xet-nghiem-hoa-sinh-mau-123520>