**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG MÔ HÌNH MẠNG NEURON DỰ ĐOÁN**

**BỆNH TIM**

**Giáo viên hướng dẫn: TS. Trần Đăng Công**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1771020296 | Phạm Minh Hoàng | CNTT:17-08 |
| 2 | 1771020217 | Nguyễn Quang Duy | CNTT:17-08 |
| 3 | 1771020328 | Nguyễn Ngọc Hưng | CNTT:17-08 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG MÔ HÌNH MẠNG NEURON DỰ ĐOÁN**

**BỆNH TIM**

**Giáo viên hướng dẫn: TS. Trần Đăng Công**

**Sinh viên thực hiện:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1771020296** | **Phạm Minh Hoàng** | **01/01/2005** |  |  |
| **2** | **1771020217** | **Nguyễn Quang Duy** | **08/06/2005** |  |  |
| **3** | **1771020328** | **Nguyễn Ngọc Hưng** | **08/10/2005** |  |  |

**CÁN BỘ CHẤM THI**

**LỜI NÓI ĐẦU**

Trong những năm gần đây, bệnh tim mạch đã trở thành một trong những nguyên nhân chính gây tử vong trên toàn thế giới. Các yếu tố nguy cơ của bệnh tim như huyết áp cao, cholesterol cao, tiểu đường, thói quen ăn uống không lành mạnh và thiếu vận động đang ngày càng gia tăng, đặc biệt trong bối cảnh lối sống hiện đại. Do đó, việc phát hiện sớm bệnh tim là vô cùng quan trọng để có thể áp dụng các biện pháp điều trị kịp thời và hiệu quả.

Truyền thống, việc chẩn đoán bệnh tim phụ thuộc vào kinh nghiệm và khả năng quan sát của bác sĩ, kết hợp với các xét nghiệm y tế. Tuy nhiên, với sự phát triển mạnh mẽ của công nghệ, đặc biệt là trong lĩnh vực trí tuệ nhân tạo (AI), các mô hình học máy, đặc biệt là mạng nơ-ron, đang trở thành công cụ hỗ trợ đắc lực trong việc dự đoán và chẩn đoán bệnh tim một cách nhanh chóng và chính xác hơn.

Đề tài này tập trung vào việc xây dựng một mô hình mạng nơ-ron nhằm dự đoán khả năng mắc bệnh tim dựa trên các yếu tố nguy cơ có sẵn trong dữ liệu. Mạng nơ-ron, với khả năng học từ dữ liệu lớn và nhận diện các mối quan hệ phức tạp giữa các yếu tố đầu vào, hứa hẹn sẽ cung cấp những kết quả chính xác, góp phần hỗ trợ quá trình chẩn đoán và điều trị bệnh tim.

Mục tiêu của nghiên cứu là phát triển một mô hình mạng nơ-ron có thể dự đoán nguy cơ mắc bệnh tim cho từng cá nhân dựa trên các yếu tố đầu vào như tuổi, giới tính, huyết áp, mức độ cholesterol, thói quen hút thuốc, và các yếu tố sức khỏe khác. Ngoài ra, nghiên cứu cũng sẽ đánh giá hiệu quả của mô hình thông qua các chỉ số thống kê và so sánh với các phương pháp dự đoán truyền thống.

Với những tiềm năng lớn từ việc áp dụng công nghệ AI trong lĩnh vực y tế, chúng tôi hy vọng rằng mô hình mạng nơ-ron này sẽ mở ra những cơ hội mới trong việc phòng ngừa và điều trị bệnh tim, từ đó góp phần giảm thiểu tỷ lệ tử vong và nâng cao chất lượng cuộc sống cho cộng đồng.

**MỤC LỤC**

**[CHƯƠNG 1. GIỚI THIỆU VÀ PHÂN TÍCH](#_Toc193493039)** [8](#_Toc193493039)

**[1.1.](#_Toc193493040)****[Giới thiệu chung và mục tiêu của đề tài](#_Toc193493040)** [8](#_Toc193493040)

**[1.2.](#_Toc193493041)****[Tổng quan về phương pháp học máy](#_Toc193493041)** [9](#_Toc193493041)

*[1.2.1.Giới thiệu về học máy](#_Toc193493042)* [9](#_Toc193493042)

*[1.3.1.](#_Toc193493043)**[Cấu trúc của mạng nơ-ron trong bài toán dự đoán bệnh tim](#_Toc193493043)* [9](#_Toc193493043)

*[1.3.2.](#_Toc193493044)**[Quá trình huấn luyện mạng nơ-ron](#_Toc193493044)* [11](#_Toc193493044)

*[1.3.3.Thuật toán tối ưu hóa sử dụng trong mạng nơ-ron](#_Toc193493045)* [12](#_Toc193493045)

**[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP](#_Toc193493046)** [14](#_Toc193493046)

**[2.1. Nguồn và Đặc điểm Dữ liệu:](#_Toc193493047)** [14](#_Toc193493047)

*[2.1.1.Nguồn dữ liệu](#_Toc193493048)* [14](#_Toc193493048)

*[2.1.2.Tiền xử lý Dữ liệu:](#_Toc193493049)* [14](#_Toc193493049)

**[2.2. Hàm kích hoạt (Activation Function)](#_Toc193493050)** [15](#_Toc193493050)

**[2.3. Hàm mất mát (Loss Function)](#_Toc193493051)** [16](#_Toc193493051)

**[2.4. Thuật toán tối ưu (Optimizer)](#_Toc193493052)** [16](#_Toc193493052)

**[2.5.Đánh giá cách chia dữ liệu (train/test), chỉ số đánh giá (accuracy, precision, recall).](#_Toc193493053)** [17](#_Toc193493053)

*[2.5.1. Cách chia dữ liệu (Train/Test)](#_Toc193493054)* [17](#_Toc193493054)

*[2.5.2 Chỉ số đánh giá (Accuracy, Precision, Recall)](#_Toc193493055)* [18](#_Toc193493055)

*[2.5.3.Quy trình đánh giá thực tế](#_Toc193493056)* [20](#_Toc193493056)

*[2.5.4.Phân tích và cân nhắc](#_Toc193493057)* [20](#_Toc193493057)

**[CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU](#_Toc193493058)** [21](#_Toc193493058)

**[3.1. Đọc dữ liệu](#_Toc193493059)** [21](#_Toc193493059)

**[3.2. Lấy dữ liệu và xử lý](#_Toc193493060)** [21](#_Toc193493060)

**[Kết luận](#_Toc193493061)** [31](#_Toc193493061)

**CHƯƠNG 1. GIỚI THIỆU VÀ PHÂN TÍCH**

## **Giới thiệu chung v****à mục tiêu của đề tài**

Bệnh tim mạch là một trong những nguyên nhân hàng đầu gây tử vong trên toàn thế giới, ảnh hưởng nghiêm trọng đến chất lượng cuộc sống của người bệnh và tạo ra gánh nặng cho hệ thống y tế. Các yếu tố nguy cơ như tuổi tác, huyết áp, cholesterol, tiểu đường, thói quen ăn uống, hút thuốc lá, và mức độ hoạt động thể chất có thể ảnh hưởng lớn đến nguy cơ mắc bệnh tim.

Với sự phát triển mạnh mẽ của trí tuệ nhân tạo và học máy, việc sử dụng các mô hình học máy để dự đoán bệnh tim từ dữ liệu y tế đã trở thành một xu hướng quan trọng trong việc hỗ trợ bác sĩ và các chuyên gia y tế trong việc phát hiện sớm và đưa ra các biện pháp điều trị kịp thời. Mô hình mạng nơ-ron (Artificial Neural Network - ANN) là một công cụ mạnh mẽ trong học máy, có khả năng học các mối quan hệ phi tuyến tính giữa các yếu tố nguy cơ và khả năng mắc bệnh tim.

Mục tiêu của đề tài:

**Thu thập và xử lý dữ liệu:**

- Thu thập một bộ dữ liệu có chứa các yếu tố nguy cơ liên quan đến bệnh tim, như tuổi, huyết áp, cholesterol, tiểu đường, và thói quen sống.

- Tiến hành xử lý dữ liệu như loại bỏ các giá trị thiếu, mã hóa các đặc trưng phân loại, chuẩn hóa dữ liệu để đảm bảo rằng các mô hình học máy có thể hoạt động hiệu quả.

**Xây dựng mô hình mạng nơ-ron**:

- Xây dựng và huấn luyện một mô hình mạng nơ-ron (như MLP - Multi-layer Perceptron) với các lớp ẩn để có thể học các mối quan hệ phức tạp giữa các yếu tố nguy cơ và khả năng mắc bệnh tim.

- Tinh chỉnh các siêu tham số của mô hình để tối ưu hóa hiệu suất dự đoán.

**Đánh giá và tối ưu hóa mô hình**:

- Đánh giá mô hình dựa trên các chỉ số đánh giá như độ chính xác, độ nhạy, độ đặc hiệu, và AUC-ROC để xác định khả năng phân loại bệnh tim của mô hình.

- Áp dụng các kỹ thuật tối ưu hóa như regularization, sử dụng bộ kiểm tra chéo (cross-validation) để tránh overfitting và cải thiện độ tổng quát của mô hình.

**So sánh với các mô hình học máy khác**:

- So sánh hiệu suất của mô hình mạng nơ-ron với các mô hình học máy khác như hồi quy logistic, cây quyết định, và rừng ngẫu nhiên để xác định mô hình nào phù hợp nhất cho bài toán dự đoán bệnh tim.

**Ứng dụng thực tế**:

- Cung cấp một công cụ hỗ trợ bác sĩ và các chuyên gia y tế trong việc dự đoán nguy cơ mắc bệnh tim cho bệnh nhân, từ đó đưa ra các biện pháp can thiệp kịp thời để ngăn ngừa bệnh tim.

- Tích hợp mô hình vào các hệ thống y tế thông minh, giúp cải thiện hiệu quả trong việc chẩn đoán và phòng ngừa bệnh tim.

## **Tổng quan về phương pháp học máy**

### *1.2.1.Giới thiệu về học máy*

**Học máy có thể được chia thành ba loại chính:**

**- Học có giám sát (Supervised Learning)**: Là phương pháp học mà trong đó dữ liệu đầu vào (features) và đầu ra (target) đã được xác định trước. Mô hình học từ dữ liệu này để dự đoán kết quả cho những dữ liệu chưa biết. Đây là phương pháp phổ biến trong các bài toán phân loại như **dự đoán bệnh tim**, nơi đầu ra có thể là **1** (mắc bệnh tim) hoặc **0** (không mắc bệnh tim).

**- Học không giám sát (Unsupervised Learning)**: Dữ liệu không có nhãn đầu ra, mô hình sẽ tìm kiếm các mẫu và mối quan hệ trong dữ liệu. Tuy nhiên, trong bài toán này, học có giám sát là phương pháp chính.

**- Học tăng cường (Reinforcement Learning)**: Mô hình học từ các quyết định và phản hồi trong môi trường, nhưng ít được áp dụng trong bài toán dự đoán bệnh tim vì đây là bài toán phân loại có giám sát.

Trong bài toán **dự đoán bệnh tim**, ta sẽ sử dụng **học có giám sát**, cụ thể là **mạng nơ-ron nhân tạo (ANN)** để học và phân tích mối quan hệ giữa các yếu tố nguy cơ và khả năng mắc bệnh tim.

**Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN)**

Mạng nơ-ron nhân tạo là một mô hình học máy mô phỏng cấu trúc và cách thức hoạt động của não bộ con người. Mạng nơ-ron bao gồm các **nơ-ron** (đơn vị tính toán) được kết nối với nhau thành các lớp, mỗi lớp thực hiện các phép toán giúp mô hình học và dự đoán kết quả.

***1.2.2.Cấu trúc của mạng nơ-ron***

**Cấu trúc của mạng nơ-rơn bao gồm:**

**- Lớp đầu vào (Input Layer)**: Nhận dữ liệu đầu vào từ các đặc trưng như tuổi, huyết áp, cholesterol, v.v.

**- Các lớp ẩn (Hidden Layers)**: Thực hiện quá trình học, các lớp này chứa các nơ-ron giúp mô hình tìm ra các mối quan hệ phức tạp giữa các đặc trưng.

**- Lớp đầu ra (Output Layer)**: Cung cấp kết quả dự đoán cuối cùng, trong bài toán dự đoán bệnh tim, lớp này sẽ cho ra kết quả 0 hoặc 1, tương ứng với việc người bệnh có mắc bệnh tim hay không.

**Ưu điểm của mạng nơ-ron trong dự đoán bệnh tim:**

**- Khả năng học các mối quan hệ phi tuyến tính**: Mạng nơ-ron có thể xử lý các mối quan hệ phi tuyến tính giữa các yếu tố nguy cơ và bệnh tim, điều này rất quan trọng vì trong thực tế, mối quan hệ giữa các yếu tố này có thể rất phức tạp.

**- Khả năng xử lý dữ liệu phức tạp**: Mạng nơ-ron có thể làm việc với một lượng lớn dữ liệu, giúp cải thiện độ chính xác trong việc dự đoán bệnh tim.

**- Tính linh hoạt và khả năng mở rộng**: Mạng nơ-ron có thể được mở rộng để xử lý các bài toán phức tạp hơn khi cần thiết, đồng thời có thể tối ưu hóa qua các kỹ thuật điều chỉnh siêu tham số.

**Các phương pháp học máy khác trong dự đoán bệnh tim**

- Mặc dù mạng nơ-ron là phương pháp mạnh mẽ trong việc dự đoán bệnh tim, nhưng cũng có nhiều phương pháp học máy khác có thể được sử dụng để giải quyết bài toán này:

**- Hồi quy logistic (Logistic Regression)**: Một phương pháp học máy đơn giản nhưng hiệu quả trong phân loại nhị phân. Tuy nhiên, hồi quy logistic chỉ phù hợp với các mối quan hệ tuyến tính giữa các yếu tố và mục tiêu.

**- Cây quyết định (Decision Trees)** và **Rừng ngẫu nhiên (Random Forest)**: Cây quyết định phân loại dữ liệu bằng cách chia nhỏ theo các đặc trưng. Rừng ngẫu nhiên kết hợp nhiều cây quyết định để tăng độ chính xác.

**- SVM (Support Vector Machine)**: SVM tìm kiếm một siêu phẳng trong không gian đặc trưng để phân loại dữ liệu, và có thể hiệu quả trong các bài toán phân loại với không gian đặc trưng phức tạp.

**- K-Nearest Neighbors (KNN)**: Phương pháp dựa trên việc tìm các đối tượng gần nhất trong không gian đặc trưng và dự đoán dựa trên những đối tượng này.

***1.2.3.Tại sao chọn học máy để dự đoán bệnh tim?***

Trong bài toán dự đoán bệnh tim, học máy, đặc biệt là **mạng nơ-ron**, được sử dụng vì các lý do sau:

**- Xử lý các mối quan hệ phức tạp và phi tuyến tính**: Các yếu tố nguy cơ đối với bệnh tim có thể tương tác với nhau theo các cách phức tạp. Mạng nơ-ron có khả năng học được những mối quan hệ phi tuyến tính giữa các đặc trưng và kết quả dự đoán.

**- Dự đoán chính xác hơn**: So với các phương pháp truyền thống, mô hình học máy có khả năng phân loại và dự đoán chính xác hơn khi có một lượng lớn dữ liệu và các yếu tố ảnh hưởng đa dạng.

**- Tối ưu hóa khả năng tổng quát**: Mạng nơ-ron có khả năng học từ dữ liệu huấn luyện và tối ưu hóa các tham số để làm việc tốt với dữ liệu chưa biết, từ đó đưa ra dự đoán chính xác cho bệnh tim.

***1.2.4.Quy trình xây dựng mô hình học máy cho dự đoán bệnh tim***

Quy trình xây dựng mô hình học máy để dự đoán bệnh tim bao gồm các bước chính sau:

**- Thu thập và chuẩn bị dữ liệu**: Thu thập dữ liệu y tế liên quan đến bệnh tim, bao gồm các yếu tố như tuổi, huyết áp, cholesterol, tình trạng tiểu đường, v.v.

**- Tiền xử lý dữ liệu**: Làm sạch dữ liệu, xử lý các giá trị thiếu, mã hóa các đặc trưng phân loại, và chuẩn hóa dữ liệu để mô hình học máy có thể hoạt động hiệu quả hơn.

**- Chia dữ liệu**: Chia dữ liệu thành các bộ huấn luyện và kiểm tra (train/test split) để kiểm tra hiệu suất mô hình trên dữ liệu chưa thấy.

**- Xây dựng và huấn luyện mô hình mạng nơ-ron**: Sử dụng các kỹ thuật học sâu (deep learning) để xây dựng mạng nơ-ron với các lớp ẩn và lớp đầu ra phù hợp với bài toán phân loại.

**- Đánh giá mô hình**: Đánh giá hiệu suất mô hình thông qua các chỉ số như độ chính xác (accuracy), độ nhạy (sensitivity), độ đặc hiệu (specificity), và diện tích dưới đường cong ROC (AUC-ROC).

**- Tối ưu hóa mô hình**: Sử dụng các kỹ thuật như **cross-validation**, điều chỉnh siêu tham số, và regularization để cải thiện mô hình.

* 1. **Cấu trúc của mạng nơ-ron, quá trình huấn luyện và các thuật toán tối ưu hóa sử dụng trong mạng nơ-ron**

### *****Cấu trúc của mạng nơ-ron trong bài toán dự đoán bệnh tim*****

Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một mô hình học máy mô phỏng cách thức hoạt động của bộ não con người để giải quyết các vấn đề phức tạp như dự đoán bệnh tim. Mạng nơ-ron bao gồm các thành phần cơ bản sau:

**Lớp đầu vào (Input Layer):**

- Lớp này nhận dữ liệu đầu vào từ các yếu tố nguy cơ ảnh hưởng đến bệnh tim. Mỗi yếu tố nguy cơ như tuổi, huyết áp, mức cholesterol, giới tính, tình trạng hút thuốc, v.v., sẽ là một đặc trưng trong dữ liệu và sẽ trở thành một nơ-ron trong lớp đầu vào.

- Số lượng nơ-ron trong lớp này bằng với số lượng đặc trưng (features) trong dữ liệu.

**Lớp ẩn (Hidden Layers):**

- Mạng nơ-ron có thể có một hoặc nhiều lớp ẩn. Lớp ẩn là nơi mà mạng nơ-ron học các đặc trưng phức tạp từ dữ liệu. Các nơ-ron trong lớp ẩn không trực tiếp tương tác với đầu vào hoặc đầu ra mà chỉ tương tác qua các nơ-ron khác trong lớp ẩn.

- Số lượng lớp ẩn và số lượng nơ-ron trong mỗi lớp ẩn phụ thuộc vào mức độ phức tạp của bài toán. Trong bài toán dự đoán bệnh tim, thường sử dụng từ một đến ba lớp ẩn với một số lượng nơ-ron đủ lớn để mô hình có khả năng học các đặc trưng phức tạp.

**Lớp đầu ra (Output Layer):**

- Lớp này trả về kết quả dự đoán của mô hình. Trong bài toán phân loại bệnh tim, lớp đầu ra có thể có một nơ-ron duy nhất nếu mô hình chỉ cần đưa ra dự đoán có hoặc không mắc bệnh (0 hoặc 1).

- Nếu là một mô hình phân loại nhị phân (như trong dự đoán bệnh tim), hàm kích hoạt **Sigmoid** được sử dụng để đưa ra xác suất mắc bệnh.

**Kết nối giữa các lớp:**

- Các lớp trong mạng nơ-ron được kết nối với nhau qua các trọng số (weights). Trọng số này được điều chỉnh trong quá trình huấn luyện để tối ưu hóa độ chính xác của mô hình.

- Quá trình truyền tín hiệu từ lớp này sang lớp khác diễn ra theo nguyên lý của **forward propagation**, trong đó dữ liệu đầu vào được tính toán qua các lớp ẩn và cuối cùng đưa ra kết quả dự đoán.

### *****Quá trình huấn luyện mạng nơ-ron*****

**Khởi tạo trọng số:**

- Trọng số ban đầu của các kết nối giữa các nơ-ron được khởi tạo ngẫu nhiên hoặc theo một phương pháp nhất định (ví dụ, phương pháp Xavier hoặc He Initialization). Điều này giúp tránh tình trạng bị "kẹt" vào các điểm tối thiểu địa phương trong quá trình tối ưu.

**Truyền tín hiệu (Forward Propagation):**

- Dữ liệu đầu vào được đưa qua lớp đầu vào, truyền qua các lớp ẩn, tính toán đầu ra ở lớp cuối cùng.

- Mỗi nơ-ron trong lớp tính toán giá trị đầu ra của mình bằng cách nhân đầu vào của nó với trọng số tương ứng và cộng với một hệ số bias. Sau đó, giá trị này được đưa vào một hàm kích hoạt (activation function) để quyết định đầu ra của nơ-ron.

**Tính toán hàm mất mát (Loss Function):**

- Sau khi có được kết quả dự đoán từ mạng nơ-ron, mô hình so sánh với kết quả thực tế (ground truth) để tính toán **hàm mất mát (loss function)**.

- Trong bài toán phân loại bệnh tim, hàm mất mát thường sử dụng là **Binary Cross-Entropy**. Mục tiêu là giảm thiểu giá trị của hàm mất mát này.

- Hàm mất mát này đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế.

**Cập nhật trọng số (Backpropagation):**

- Sau khi tính toán hàm mất mát, mô hình sử dụng **Backpropagation** để tính toán gradient của hàm mất mát đối với từng trọng số trong mạng.

**- Backpropagation** giúp xác định hướng và mức độ thay đổi cần thiết cho các trọng số để giảm thiểu hàm mất mát.

- Quá trình này diễn ra qua các lớp từ lớp đầu ra quay ngược trở lại đến lớp đầu vào, cập nhật trọng số để cải thiện kết quả dự đoán.

**Cập nhật trọng số với thuật toán tối ưu hóa:**

- Sau khi tính toán gradient, thuật toán tối ưu hóa như **Gradient Descent** hoặc **Adam** được sử dụng để cập nhật trọng số nhằm giảm thiểu hàm mất mát.

**- Gradient Descent** là một thuật toán tối ưu hóa phổ biến, nhưng **Adam (Adaptive Moment Estimation)** thường được ưa chuộng trong các mạng nơ-ron vì nó kết hợp cả phương pháp gradient descent truyền thống và tính toán trung bình động để điều chỉnh tốc độ học cho mỗi trọng số.

### *1.3.3.Thuật toán tối ưu hóa sử dụng trong mạng nơ-ron*

**Gradient Descent:**

**- Gradient Descent** là thuật toán tối ưu hóa cơ bản được sử dụng trong mạng nơ-ron để cập nhật trọng số. Thuật toán này điều chỉnh trọng số theo hướng ngược lại với gradient của hàm mất mát.

- Tốc độ cập nhật trọng số được điều khiển bởi một tham số gọi là **learning rate** (tốc độ học). Nếu learning rate quá lớn, mô hình có thể không hội tụ; nếu quá nhỏ, quá trình huấn luyện sẽ rất chậm.

**Stochastic Gradient Descent (SGD):**

- Trong **SGD**, mỗi lần cập nhật trọng số chỉ sử dụng một mẫu dữ liệu ngẫu nhiên thay vì toàn bộ tập dữ liệu. Điều này giúp giảm thiểu chi phí tính toán nhưng có thể dẫn đến sự dao động trong quá trình huấn luyện.

**Adam (Adaptive Moment Estimation):**

**- Adam** là một thuật toán tối ưu hóa nâng cao, kết hợp giữa Gradient Descent và các kỹ thuật mô phỏng moment (trung bình động). Adam điều chỉnh tốc độ học cho mỗi trọng số, giúp tăng tốc quá trình huấn luyện và đạt được kết quả tốt hơn trong nhiều tình huống.

- Adam có hai tham số quan trọng: **learning rate** và **beta1/beta2**, giúp kiểm soát tốc độ thay đổi.

**RMSprop:**

**- RMSprop** điều chỉnh tốc độ học cho mỗi trọng số dựa trên bình phương gradient trong một cửa sổ thời gian cố định. Thuật toán này giúp cải thiện tốc độ hội tụ trong một số bài toán, đặc biệt khi có dữ liệu không ổn định hoặc có nhiều sự thay đổi.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP**

**2.1. Nguồn và Đặc điểm Dữ liệu:**

### *2.1.1.Nguồn dữ liệu*

Nguồn dữ liệu có thể lấy từ các cơ sở dữ liệu y tế nổi tiếng như **UCI Machine Learning Repository**. Một ví dụ cụ thể là bộ dữ liệu **Heart Disease** từ UCI, được sử dụng phổ biến trong các nghiên cứu và mô hình dự đoán bệnh tim mạch. Bộ dữ liệu này chứa các thông tin về bệnh nhân như:

**- Tuổi tác**: Độ tuổi của bệnh nhân.

**- Giới tính**: Nam hoặc nữ.

**- Huyết áp**: Đo huyết áp trong quá trình khám bệnh.

**- Mức cholesterol**: Mức cholesterol trong máu (thường tính theo đơn vị mg/dl).

**- Đau ngực (Chest pain type)**: Loại đau ngực của bệnh nhân.

**-Điện tâm đồ (Electrocardiographic results)**: Kết quả từ xét nghiệm điện tâm đồ.

**- Nhịp tim (Maximum heart rate)**: Tốc độ nhịp tim tối đa của bệnh nhân.

**- Xã hội (Exercise induced angina)**: Thông tin về bệnh nhân có đau ngực khi vận động hay không.

**- Dòng máu (ST depression induced by exercise)**: Thông tin về sự thay đổi ST trong khi tập thể dục.

**- Tình trạng bệnh tim (Heart Disease)**: Biến mục tiêu, thể hiện bệnh nhân có mắc bệnh tim hay không (biến phân loại).

Các yếu tố nguy cơ chính được sử dụng trong mô hình thường là **tuổi tác, huyết áp, mức cholesterol, nhịp tim tối đa, tình trạng đau ngực** và các yếu tố khác liên quan đến tim mạch.

### *2.1.2.Tiền xử lý Dữ liệu:*

**Tiền xử lý dữ liệu**

Tiền xử lý là bước quan trọng để đảm bảo dữ liệu phù hợp với mạng neuron. Quy trình bao gồm làm sạch và chuẩn hóa như sau:

1. Làm sạch dữ liệu

**Xử lý giá trị thiếu (Missing Values):**

Trong UCI Heart Disease Dataset, một số đặc trưng như "ca" hoặc "thal" có giá trị bị thiếu (ký hiệu NaN hoặc "?").

Cách xử lý:

- Thay thế: Dùng giá trị trung bình (mean) hoặc trung vị (median) cho đặc trưng số (ví dụ: thay giá trị thiếu của "chol" bằng trung bình 246 mg/dl).

- Loại bỏ: Nếu số lượng mẫu thiếu ít (<5%), có thể bỏ các hàng đó.

- Dự đoán: Sử dụng mô hình đơn giản (như KNN) để điền giá trị thiếu dựa trên mẫu tương tự.

**Xử lý giá trị ngoại lai (Outliers):**

- Ví dụ: Huyết áp "trestbps" = 0 hoặc cholesterol "chol" = 500 mg/dl là không thực tế.

- Cách xử lý: Thay bằng giá trị trung bình hoặc giới hạn trong khoảng hợp lý (ví dụ: trestbps từ 90-200 mmHg).

- Kiểm tra tính nhất quán: Đảm bảo không có giá trị âm ở các cột như "age" hoặc "chol".

2. Chuẩn hóa dữ liệu

Mục đích: Đưa các đặc trưng về cùng thang đo để mạng neuron học hiệu quả hơn, vì các giá trị như "chol" (hàng trăm) và "oldpeak" (0-6) có phạm vi khác nhau.

Phương pháp:

**Chuẩn hóa Min-Max (Min-Max Scaling):**

Công thức



Đưa tất cả giá trị về khoảng [0, 1]. Ví dụ: "trestbps" từ 94-200 mmHg được chuẩn hóa thành 0-1.

**Chuẩn hóa Z-score (Standardization):**

Công thức

​ (với μ\muμ là trung bình, σ\sigmaσ là độ lệch chuẩn).

Đưa giá trị về phân phối chuẩn với trung bình 0, độ lệch chuẩn 1. Thường dùng khi dữ liệu có phân phối gần chuẩn.

Mã hóa biến phân loại (Encoding Categorical Variables):

- Các đặc trưng như "cp" (1-4), "restecg" (0-2) là dạng phân loại.

- One-Hot Encoding: Chuyển thành vector nhị phân (ví dụ: "cp" = 1 thành [1, 0, 0, 0], "cp" = 2 thành [0, 1, 0, 0]).

- Kết quả: Tăng số đặc trưng đầu vào (từ 13 lên khoảng 20-25 cột).

Cân bằng dữ liệu (nếu cần):

- Nếu nhãn 0 và 1 không cân bằng (ví dụ: 165 không bệnh, 138 có bệnh), có thể dùng kỹ thuật như oversampling (SMOTE) hoặc undersampling để tránh bias.

**2.2. Hàm kích hoạt (Activation Function)**

Hàm kích hoạt xác định cách neuron biến đổi tổng trọng số đầu vào thành đầu ra, ảnh hưởng đến khả năng học của mạng:

**Lớp ẩn:**

Hàm ReLU (Rectified Linear Unit): f(x)=max⁡(0,x) f(x) = \max(0, x) f(x)=max(0,x).

- Cách hoạt động: Nếu tổng trọng số x<0 x < 0 x<0, đầu ra = 0; nếu x≥0 x \geq 0 x≥0, đầu ra = x x x.

- Ví dụ: Tổng trọng số = -0.3 → ReLU(-0.3) = 0; Tổng = 1.5 → ReLU(1.5) = 1.5.

- Ưu điểm:

Tránh vanishing gradient (gradient bằng 0 ở vùng âm của sigmoid/tanh).

Tăng tốc độ học nhờ tính toán đơn giản.

Phù hợp với dữ liệu phi tuyến như mối quan hệ giữa huyết áp và nhịp tim.

Áp dụng: Dùng cho cả 16 neuron ở lớp ẩn 1 và 8 neuron ở lớp ẩn 2.

**Lớp đầu ra:**

Hàm Sigmoid:



Cách hoạt động: Chuyển đổi tổng trọng số thành giá trị từ 0 đến 1.

Ví dụ: Tổng = 2 → Sigmoid(2) ≈ 0.88; Tổng = -1 → Sigmoid(-1) ≈ 0.27.

Ưu điểm:

- Diễn giải trực tiếp thành xác suất (0.88 = 88% nguy cơ mắc bệnh).

- Phù hợp với phân loại nhị phân.

- Nhược điểm: Có thể gặp vanishing gradient nếu tổng trọng số quá lớn/nhỏ, nhưng ít ảnh hưởng vì chỉ dùng ở lớp cuối.

Lý do lựa chọn:

- ReLU ở lớp ẩn để tối ưu tốc độ và khả năng học phi tuyến.

- Sigmoid ở lớp đầu ra để phù hợp với bài toán nhị phân và yêu cầu đầu ra xác suất.

**2.3. Hàm mất mát (Loss Function)**

Hàm mất mát đo lường mức độ sai lệch giữa dự đoán và nhãn thực tế, định hướng quá trình tối ưu:

**Hàm mất mát: Binary Cross-Entropy (BCE).**

Công thức:



- N N N: Số mẫu (ví dụ: 303).

- yi y\_i yi​: Nhãn thực tế (0 hoặc 1).

- y^i \hat{y}\_i y^​i​: Dự đoán (xác suất từ 0-1).

Ưu điểm:

- Phù hợp với phân loại nhị phân, tăng penalty khi dự đoán sai xa.

- Tương thích với Sigmoid ở lớp đầu ra (đạo hàm đơn giản).

Lý do chọn: Chuẩn mực cho bài toán nhị phân, đảm bảo mô hình học tốt sự khác biệt giữa hai lớp.

**2.4. Thuật toán tối ưu (Optimizer)**

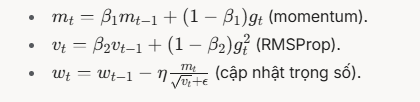
Thuật toán tối ưu cập nhật trọng số và bias dựa trên gradient của hàm mất mát:

Thuật toán: Adam (Adaptive Moment Estimation).

Cách hoạt động:

- Kết hợp momentum (trung bình động gradient) và RMSProp (trung bình động bình phương gradient).

- Công thức cập nhật trọng số:



- Trong đó: gt g\_t gt​ là gradient, η \eta η là learning rate, ϵ \epsilon ϵ là hằng số nhỏ (10⁻⁸).

Tham số:

- Learning rate: 0.001 (nhỏ để học từ từ, tránh vượt điểm tối ưu).

- Beta1: 0.9 (trọng số cho momentum).

- Beta2: 0.999 (trọng số cho RMSProp).

Ưu điểm:

- Tự động điều chỉnh tốc độ học, phù hợp với dữ liệu noisy hoặc gradient dao động.

- Hội tụ nhanh hơn SGD thông thường.

Ví dụ:

- Sau 1 epoch, trọng số từ neuron 1 (lớp ẩn 1) đến neuron 1 (lớp ẩn 2) có thể giảm từ 0.5 xuống 0.498 dựa trên gradient.

Thay thế (nếu cần):

SGD với Momentum:

Công thức:



- Learning rate: 0.01, Momentum: 0.9.

Nhược điểm: Chậm hơn Adam, cần tinh chỉnh nhiều hơn.

Lý do chọn Adam:

- Hiệu quả cao với bài toán thực tế, ít cần thử nghiệm tham số.

- Phù hợp với dữ liệu y khoa có biến động (như huyết áp, cholesterol).

**2.5. Đánh giá cách chia dữ liệu (train/test), chỉ số đánh giá (accuracy, precision, recall).**

### *2.5.1. Cách chia dữ liệu (Train/Test)*

Chia dữ liệu là bước quan trọng để huấn luyện và kiểm tra hiệu suất của mô hình mạng neuron. Với bộ dữ liệu nhỏ như UCI Heart Disease Dataset (303 mẫu), việc chia cần được thực hiện cẩn thận để đảm bảo mô hình học tốt và đánh giá khách quan.

**Tỷ lệ chia:**

80% Train - 20% Test:

- Train: 242 mẫu (80% của 303).

- Test: 61 mẫu (20% của 303).

- Lý do: Tỷ lệ phổ biến trong học máy, đảm bảo đủ dữ liệu để huấn luyện (242) và kiểm tra (61) với kích thước dữ liệu nhỏ.

Thay thế:

- 70% Train - 30% Test (212 train, 91 test) nếu muốn kiểm tra kỹ hơn trên tập test lớn hơn.

- K-fold Cross-Validation (k=5 hoặc 10) nếu cần đánh giá ổn định hơn (chia thành 5 hoặc 10 phần, mỗi phần lần lượt làm test).

**Phương pháp chia:**

Ngẫu nhiên có phân tầng (Stratified Random Split):

- Đảm bảo tỷ lệ nhãn (0: không bệnh, 1: có bệnh) trong tập train và test giống với tỷ lệ tổng thể (ví dụ: 54% không bệnh, 46% có bệnh trong UCI dataset).

- Công cụ: Hàm train\_test\_split trong thư viện scikit-learn với tham số stratify=y.

- Lý do: Tránh trường hợp tập train toàn mẫu không bệnh hoặc tập test toàn mẫu có bệnh, gây bias trong đánh giá.

Tập validation (nếu cần):

- Lấy 20% từ tập train (khoảng 48 mẫu) làm tập validation để tinh chỉnh siêu tham số (learning rate, số epoch).

- Cấu hình cuối: 60% train (182), 20% validation (48), 20% test (61).

**Quy trình:**

- Xáo trộn dữ liệu ngẫu nhiên (random shuffle).

- Chia thành train/test theo tỷ lệ đã chọn.

- Kiểm tra phân phối nhãn để đảm bảo không bị mất cân bằng nghiêm trọng.

### *2.5.2 Chỉ số đánh giá (Accuracy, Precision, Recall)*

Các chỉ số đánh giá đo lường hiệu suất mô hình trong việc phân loại nguy cơ mắc bệnh tim. Với bài toán nhị phân (0 = không bệnh, 1 = có bệnh), các chỉ số cơ bản được tính từ ma trận nhầm lẫn (confusion matrix):

**Ma trận nhầm lẫn (Confusion Matrix):**

- True Positive (TP): Dự đoán có bệnh và thực tế có bệnh.

- True Negative (TN): Dự đoán không bệnh và thực tế không bệnh.

- False Positive (FP): Dự đoán có bệnh nhưng thực tế không bệnh (lỗi loại I).

- False Negative (FN): Dự đoán không bệnh nhưng thực tế có bệnh (lỗi loại II).

- Ví dụ: Với 61 mẫu test, giả sử:

TP = 25, TN = 30, FP = 3, FN = 3.

Accuracy (Độ chính xác):

Công thức:



- Ý nghĩa: Tỷ lệ dự đoán đúng trên tổng số mẫu.

- Ưu điểm: Dễ hiểu, phản ánh hiệu suất tổng thể.

-Nhược điểm: Không đáng tin nếu dữ liệu mất cân bằng (ví dụ: 90% không bệnh, dự đoán toàn "0" vẫn cho accuracy cao).

Precision (Độ chính xác của dự đoán dương):

Công thức:



- Ý nghĩa: Tỷ lệ các trường hợp dự đoán "có bệnh" là đúng.

- Tầm quan trọng: Cao precision nghĩa là ít báo động sai (false positives), tránh gây lo lắng không cần thiết cho bệnh nhân.

- Ứng dụng: Quan trọng khi chi phí của FP cao (ví dụ: điều trị không cần thiết).

Recall (Độ nhạy, Tỷ lệ phát hiện dương):

Công thức



- Ý nghĩa: Tỷ lệ các trường hợp thực tế "có bệnh" được phát hiện đúng.

- Tầm quan trọng: Cao recall nghĩa là ít bỏ sót bệnh nhân thực sự có bệnh (false negatives), rất quan trọng trong y học để đảm bảo không bỏ qua ca bệnh nguy hiểm.

- Ứng dụng: Ưu tiên recall khi bỏ sót bệnh nhân (FN) nguy hiểm hơn báo động sai (FP).

Chỉ số bổ sung (nếu cần):

- F1-Score: Trung bình hài hòa của Precision và Recall.

Công thức



- Ý nghĩa: Cân bằng giữa Precision và Recall, hữu ích khi dữ liệu mất cân bằng.

- ROC-AUC: Diện tích dưới đường cong ROC, đo lường khả năng phân biệt giữa 2 lớp (gần 1 là tốt).

### *2.5.3.Quy trình đánh giá thực tế*

Bước 1: Huấn luyện mô hình trên tập train (242 mẫu) với kiến trúc (23-16-8-1), Adam, Binary Cross-Entropy.

Bước 2: Dự đoán trên tập test (61 mẫu).

Bước 3: Tính ma trận nhầm lẫn và các chỉ số:

- Giả sử: TP = 25, TN = 30, FP = 3, FN = 3.

- Accuracy = 90.2%, Precision = 89.3%, Recall = 89.3%, F1 = 89.3%.

Bước 4: So sánh với mục tiêu:

- Accuracy > 85%: Đạt yêu cầu cơ bản.

- Recall > 90%: Có thể cần cải thiện để giảm FN (bỏ sót bệnh nhân).

### *2.5.4.Phân tích và cân nhắc*

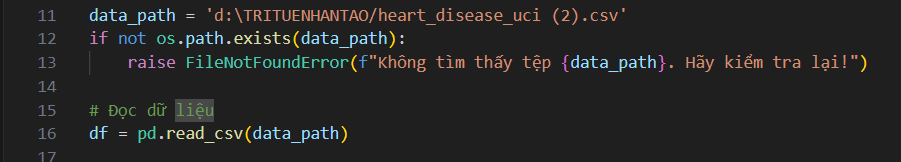
- Dữ liệu nhỏ: Với 303 mẫu, kết quả có thể biến động tùy lần chia. K-fold Cross-Validation (k=5) sẽ cho đánh giá ổn định hơn (trung bình các chỉ số qua 5 lần chia).

- Mất cân bằng: Nếu nhãn không đều (ví dụ: 165 không bệnh, 138 có bệnh), Recall quan trọng hơn Accuracy để tránh bỏ sót ca bệnh.

- Cải thiện: Điều chỉnh ngưỡng xác suất (thay vì 0.5, thử 0.4 để tăng Recall nếu cần).

**CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU**

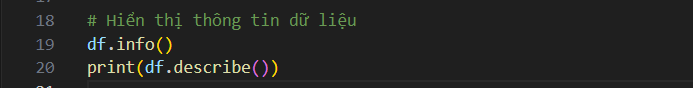
**3.1. Đọc dữ liệu**



**3.2. Lấy dữ liệu và xử lý**

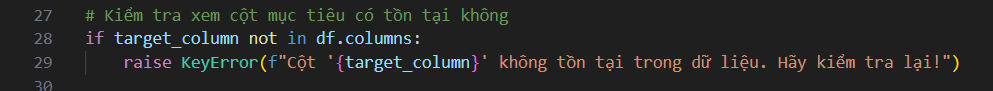
- Hiển thị dữ liệu:

Hiển thị ra các dữ liệu trong bảng dữ liệu mà mình cung cấp.



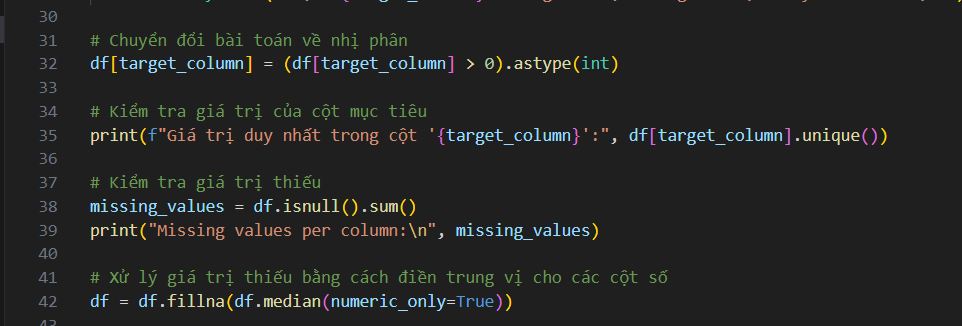
- Kiểm tra xem có tồn tại không

Hệ thống sẽ kiểm tra xem trong bảng dự liệu đó thì cột mục tiêu đó có tồn tại hay không và nêu hiển thi ra không tồn tại dữ cột dự liệu đó thì đề nghị kiểm tra lại dữ liệu ban đầu.



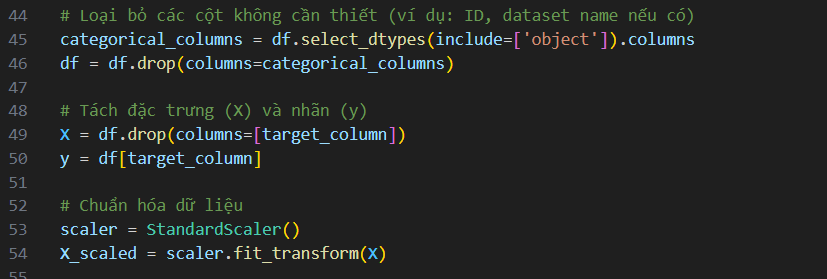
- Chuyển đổi, kiểm tra và xử lý

Chuyển đổi các bài toán dữ liệu về dạng bài toán nhị phân và đưa ra kiểm tra các giá trị của cột mục tiêu đó có là cột giá trị duy nhất trong đó không. Sau đó kiểm tra lại các giá trị còn thiếu rồi xywr lý điềm thêm các dự liệu trung vị trong cột.



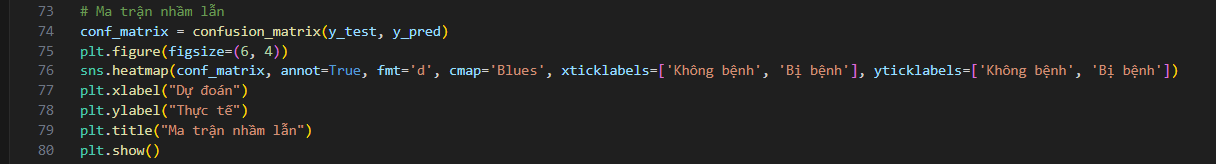
- Loại bỏ, tách các đặc trưng và chuẩn hóa dữ liệu

Loại bọ các cột mà không tồn tại và không cần thiết trong bảng dữ liệu đó xem có cần thiết hay không. Sau đó tiếp tục tách các đăng trung và chuẩn hóa các dữ liệu.

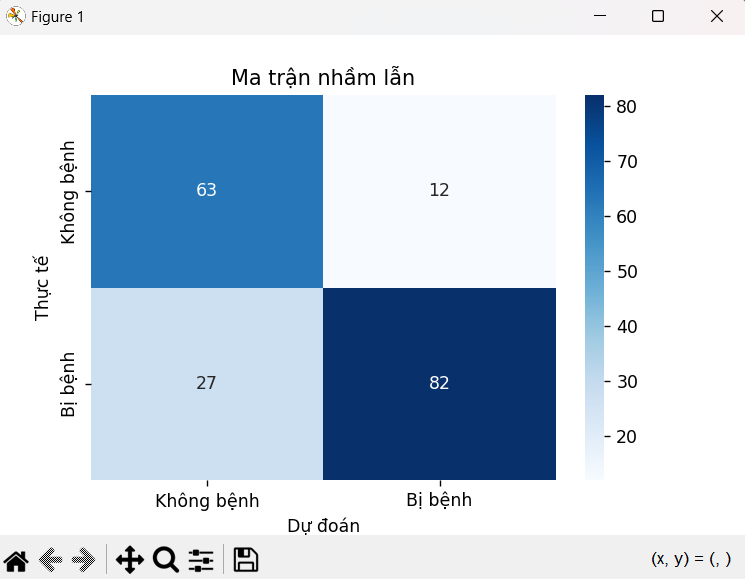


- Ma trận nhầm lần

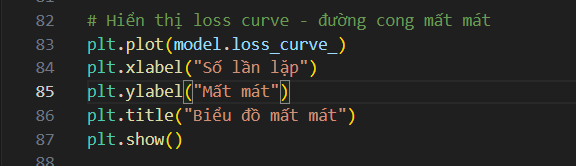
Đưa ra bảng ma trận sát suất nhầm lần người bị bênh hay không bị bệnh, hiển thị ra dự đoán thực tế.



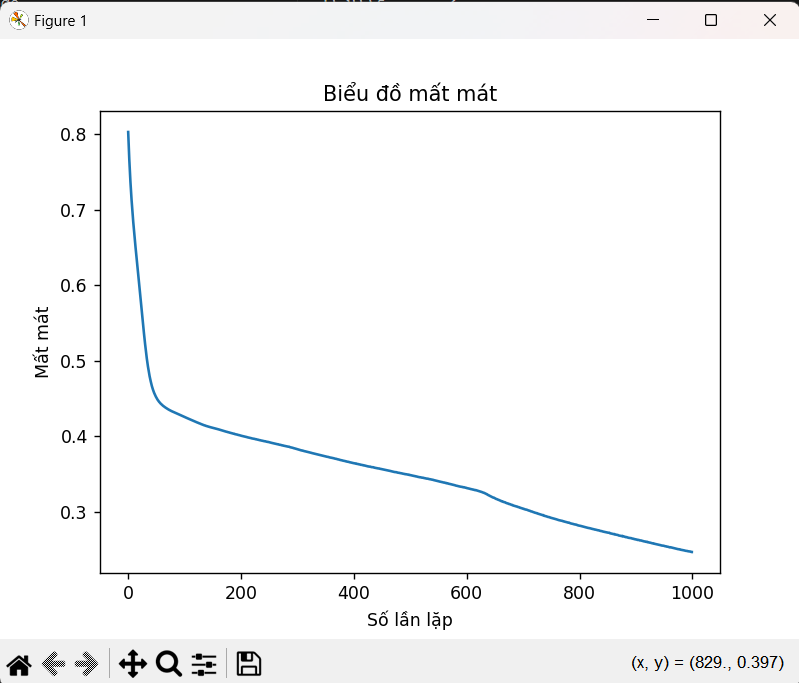
Kết quả:



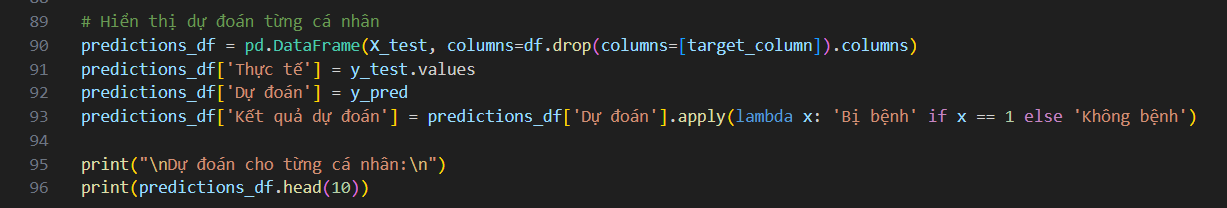
- Hiển thị loss curve - đường cong mất mát



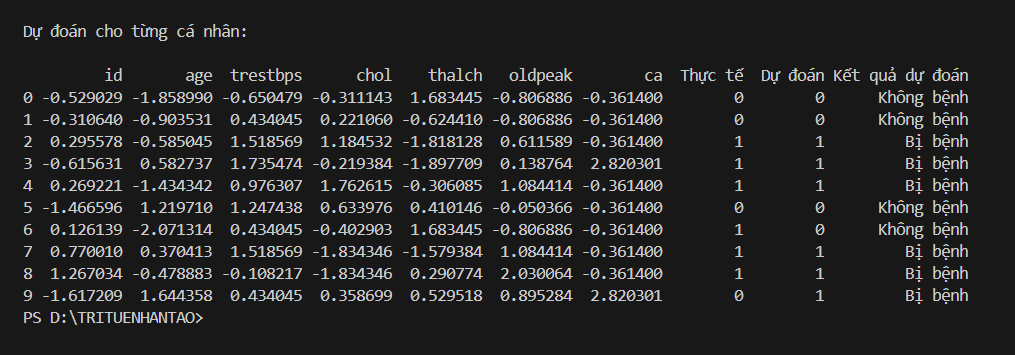
Kết quả



- Hiển thị dự đoán cá nhân



Kết quả



**KẾT LUẬN**

**Ưu điểm:** Mô hình dự đoán bệnh bằng mạng nơ-ron nhân tạo (ANN) là khả năng học hỏi từ dữ liệu và phát hiện các mẫu phức tạp mà hệ thống truyền thống kê mô hình có thể bỏ qua. Công việc chuẩn hóa dữ liệu giúp mô hình hoạt động ổn định hơn, trong khi thuật toán tối ưu hóa như Adam giúp tăng tốc độ hội tụ. Ngoài ra, ANN có thể mở rộng các cấu trúc phức tạp hơn để cải thiện độ chính xác khi có nhiều dữ liệu hơn.

**Nhược điểm:** Phương pháp này yêu cầu tính toán tài chính cao, đặc biệt khi mở rộng số lượng tầng ẩn và số lượng nơ-ron. Hơn nữa, ANN có thể bị trang bị quá mức nếu không có đủ lượng dữ liệu lớn hoặc không áp dụng các tiêu chuẩn kỹ thuật. Quá trình huấn luyện cũng cần phải thử nghiệm nhiều lần để tìm được cấu hình tối ưu về số lớp, số nơron và siêu tham số như hệ số học (tốc độ học tập).

**Hướng phát triển:** Mô hình bao gồm cải thiện độ chính xác bằng cách sử dụng các kỹ thuật nâng cao như bỏ học để giảm trang bị quá mức, thử nghiệm độ sâu của mạng kiến trúc hơn hoặc tích hợp với các mô hình khác như Random Forest hoặc XGBoost để tạo mô hình lai. Ngoài ra, việc thu thập và xử lý đa dạng dữ liệu có thể giúp mô hình hóa tổng hợp tốt hơn trên dữ liệu thực tế.

**PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSV** | **Họ và tên** | **Nhiệm vụ** |
| 1 | 1771020296 | Phạm Minh Hoàng | Làm word c1 + phần đầu code |
| 2 | 1771020217 | Nguyễn Quang Duy | Làm word c2 + phần giữa code |
| 3 | 1771020328 | Nguyễn Ngọc Hưng | Làm word c3 +phần cuối |

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Trí tuệ nhân tạo*  (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Trí tuệ nhân tạo (AI)*, Đại học SPKT Hưng Yên.