







**MỤC LỤC**

[**DANH MỤC CÁC BẢNG BIỂU 4**](#_heading=h.u33bg6gqtf52)

[**DANH MỤC CÁC HÌNH ẢNH 5**](#_heading=h.ruxvvamdv742)

[**MỞ ĐẦU 6**](#_heading=h.uz92a449im92)

[**PHẦN I. TỔNG QUAN 7**](#_heading=h.bv0yumu2igab)

[1.1. Giới thiệu bài toán 8](#_heading=h.ssoyak449ux6)

[1.2. Một số thông tin liên quan 8](#_heading=h.ptltxs1nigju)

[1.2.1.Thách thức trong dự đoán động kinh 8](#_heading=h.yzjj9cexy9x3)

[1.2.2. Các phương pháp học máy phổ biến 9](#_heading=h.x3s97qrphe4z)

[1.2.3. Các nghiên cứu và ứng dụng thực tế 9](#_heading=h.h2fk1cpmhzl1)

[1.2.3.1. Các nghiên cứu trước đây 9](#_heading=h.age1tsode3je)

[1.2.3.2. Ứng dụng thực tế 9](#_heading=h.j9vhsoqnb3g6)

[1.2.4. Mục tiêu nghiên cứu 10](#_heading=h.3r5zkb8ko8n)

[1.2.4.1. Mục tiêu tổng quát 10](#_heading=h.trpdlho972h9)

[1.2.4.2. Mục tiêu cụ thể 10](#_heading=h.o7hfakdmgmgv)

[1.2.5. Đối tượng nghiên cứu và phạm vi nghiên cứu 10](#_heading=h.s3x02vzi3p49)

[1.3. Kết luận 11](#_heading=h.yxnylywrhmnm)

[**PHẦN II. PHƯƠNG PHÁP THỰC HIỆN 11**](#_heading=h.pus3z6rnohdg)

[2.1. Đặt vấn đề 12](#_heading=h.25fiidx24aym)

[2.2. Một số mô hình giải quyết bài toán 12](#_heading=h.cj1mpcl7g22x)

[2.2.1.Thuật toán Iterative Dichotomiser 3 12](#_heading=h.m34utvfonzbb)

[2.2.1.1. Nguyên lý hoạt động 12](#_heading=h.lziyontl0dm8)

[2.2.1.2. Đầu vào và đầu ra 13](#_heading=h.6scvrna2kq8g)

[2.2.1.3. Đặc điểm 13](#_heading=h.b23wxwz7179p)

[2.2.1.4. Ưu điểm 13](#_heading=h.qe8lnc96vh3h)

[2.2.1.5. Ứng dụng thực tế 13](#_heading=h.504dm0ua2e8e)

[2.2.2. Thuật toán Support Vector Machine 13](#_heading=h.9opp0bhw8ys3)

[2.2.2.1.Nguyên lý hoạt động 14](#_heading=h.4zpowegkr0uq)

[2.2.2.2. Đầu vào và đầu ra 14](#_heading=h.w1c0z8wrna9n)

[2.2.2.3. Ưu điểm 14](#_heading=h.2qxead6qvgd)

[2.2.2.4. Ứng dụng thực tế 14](#_heading=h.jr4xug6sokmb)

[2.2.3. Thuật toán Artificial Neural Network 15](#_heading=h.iwvcpyj3wq0c)

[2.2.3.1.Nguyên lý hoạt động 15](#_heading=h.4nit4iikc810)

[2.2.3.2. Đầu vào và đầu ra 15](#_heading=h.fdweovdr8qeu)

[2.2.3.3. Đặc điểm 15](#_heading=h.khhdbmrnq8xb)

[2.2.3.4. Ưu điểm 15](#_heading=h.smf6jnu2qi9v)

[2.2.3.5. Ứng dụng thực tế 16](#_heading=h.1zvqkby63hmi)

[2.3. Kết luận 16](#_heading=h.rftp3pjcfvgm)

[**PHẦN III. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 16**](#_heading=h.iix46b831uww)

[3.1. Công nghệ sử dụng 17](#_heading=h.cqf63gkmx3h6)

[3.1.1. Ngôn ngữ 17](#_heading=h.xjyuc2w4a7s6)

[3.1.2. Các thư viện sử dụng 17](#_heading=h.258hlwe3i731)

[3.1.3. Phát biểu bài toán 17](#_heading=h.gqhc603ppi66)

[3.2. Triển khai thực nghiệm 18](#_heading=h.n2pw69ha8g3q)

[3.2.1. Quy trình tổng quan bài toán 18](#_heading=h.sw23yxhzy69n)

[3.2.1.1.Thu thập dữ liệu 18](#_heading=h.exi76atl3gqz)

[3.2.1.2. Làm sạch dữ liệu 18](#_heading=h.suj6mmb6uhe)

[3.2.1.2.1. Xóa các thông tin bị thiếu 18](#_heading=h.wy4s6t852a4d)

[3.2.1.2.2. Xóa dữ liệu trùng lặp 19](#_heading=h.f8hks91wvsbw)

[3.2.1.3. Phân tích dữ liệu 19](#_heading=h.nafmi7q6xjob)

[3.2.1.4. Tiền xử lý dữ liệu 20](#_heading=h.1vg2basa5yg7)

[3.2.1.4.1. Chuẩn hóa dữ liệu 20](#_heading=h.dlbvu1xkk4to)

[3.2.1.4.2. Cân bằng dữ liệu 20](#_heading=h.c2tnzoqrlanb)

[3.2.1.5. Lựa chọn mô hình 20](#_heading=h.2mphi0lfjw3x)

[3.2.1.6. Huấn luyện mô hình 21](#_heading=h.o0xw4hhkuubj)

[3.3. Đánh giá mô hình 22](#_heading=h.8se5pdioyvb3)

[3.3.1. Iterative Dichotomiser 3 22](#_heading=h.7jao2cutuof)

[3.3.2. SVM 23](#_heading=h.8y9ghlptkb7v)

[3.3.3. ANN 24](#_heading=h.s7h8cspe98ec)

[3.4. Nhận xét chung 24](#_heading=h.ekh444u3mer1)

[3.5. Triển khai mô hình 25](#_heading=h.6vrkpeo66yh1)

[3.5.1. Lưu mô hình 25](#_heading=h.9n5vwbjkb0xx)

[3.5.2. Triển khai mô hình 25](#_heading=h.a76oas3epp8p)

[**PHẦN IV. KẾT LUẬN 26**](#_heading=h.rzd8mo4r9ojl)

[**TÀI LIỆU THAM KHẢO 27**](#_heading=h.u2a7s5vr5tc)

# 

# 

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3.4. Bảng dự đoán của 3 mô hình 24](#_heading=h.5jjrv3a1vvde)

# 

# DANH MỤC CÁC HÌNH ẢNH

[Hình 3.2.1. Quy trình tổng quan bài toán 18](#_heading=h.gp8judrsjvza)

[Hình 3.2.1.1. Thông tin dữ liệu thu thập 18](#_heading=h.f1uau7ypb7bs)

[Hình 3.2.1.2.1. Hình ảnh thông tin dữ liệu 19](#_heading=h.9z4azdxyatxb)

[Hình 3.2.1.4.2. Giá trị sau khi cân bằng 20](#_heading=h.xqc0yz5ywoqw)

[Hình 3.3.1.1. Ma trận matrix thuật toán Iterative Dichotomiser 3 22](#_heading=h.z29v6cw35rve)

[Hình 3.3.1.2. Kết quả báo cáo phân loại thuật toán Iterative Dichotomiser 3 22](#_heading=h.vpc4fhh47ib)

[Hình 3.3.2.1. Ma trận matrix thuật toán SVM 23](#_heading=h.nxjghx63iq9d)

[Hình 3.3.2.2. Kết quả báo cáo phân loại thuật toán SVM 23](#_heading=h.qkou2w88aqqe)

[Hình 3.3.3. Ma trận matrix thuật toán ANN 24](#_heading=h.34vmarmdkde0)

[Hình 3.5.2. Giao diện người dùng 26](#_heading=h.fs58x74wk2fr)

**DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ**

| **STT** | **Ký hiệu chữ viết tắt** | **Chữ viết đầy đủ** |
| --- | --- | --- |
| 1 | NLP | Natural Language Processing |
| 2 | LSTM | Long Short-Term Memory |
| 3 | WHO | World Health Organization |
| 4 | EEG | Electroencephalogram |
| 5 | ANN | Artificial Neural Networks |
| 6 | CNN | Convolutional Neural Networks |
| 7 | ID3 | Iterative Dichotomiser 3 |
| 8 | SVM | Support Vector Machine |

# MỞ ĐẦU

Động kinh là một rối loạn thần kinh mãn tính ảnh hưởng đến hàng triệu người trên thế giới, gây ra các cơn co giật không kiểm soát do hoạt động điện bất thường trong não. Chẩn đoán và theo dõi bệnh chủ yếu dựa vào dữ liệu điện não đồ (EEG), tuy nhiên phương pháp truyền thống phụ thuộc nhiều vào chuyên gia y tế, tốn thời gian và khó phát hiện sớm cơn động kinh. Việc dự đoán chính xác thời điểm xảy ra cơn động kinh có ý nghĩa quan trọng trong điều trị và nâng cao chất lượng cuộc sống cho bệnh nhân.

Sự phát triển của trí tuệ nhân tạo và học máy mở ra hướng đi mới trong việc phân tích tín hiệu EEG nhằm tự động hóa quá trình phát hiện và dự đoán cơn động kinh. Các thuật toán như Support Vector Machine (SVM), Artificial Neural Networks (ANN), và Iterative Dichotomiser 3 (ID3) được nghiên cứu nhằm cải thiện độ chính xác trong nhận diện và dự đoán cơn co giật. Việc áp dụng các phương pháp học máy không chỉ giúp tăng tốc độ chẩn đoán mà còn hỗ trợ bác sĩ đưa ra quyết định chính xác hơn.

Nghiên cứu này tập trung vào phát triển mô hình học máy để phân tích dữ liệu EEG, đánh giá hiệu suất dự đoán và đề xuất phương pháp tối ưu. Các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision) và điểm F1 (F1-score) sẽ được sử dụng để đánh giá mô hình. Từ đó, nghiên cứu hướng đến việc xây dựng một hệ thống có thể ứng dụng trong thực tế, giúp hỗ trợ bệnh nhân động kinh và bác sĩ trong quá trình chẩn đoán.

Báo cáo gồm bốn phần chính:

- Phần I trình bày tổng quan về bài toán và các phương pháp liên quan

- Phần II mô tả chi tiết phương pháp thực hiện

- Phần III phân tích thực nghiệm và đánh giá kết quả

- Phần IV tổng kết nghiên cứu và đề xuất hướng phát triển trong tương lai.

# PHẦN I. TỔNG QUAN

## 1.1. Giới thiệu bài toán

Động kinh là một bệnh lý thần kinh phổ biến, ảnh hưởng đến khoảng 50 triệu người trên toàn cầu. Bệnh đặc trưng bởi các cơn co giật tái diễn do rối loạn hoạt động điện của não, gây ảnh hưởng nghiêm trọng đến sức khỏe và chất lượng cuộc sống. Hiện nay, chẩn đoán động kinh chủ yếu dựa trên phân tích điện não đồ (EEG), nhưng phương pháp này phụ thuộc nhiều vào trình độ chuyên môn của bác sĩ và có thể gặp khó khăn trong việc nhận diện tín hiệu phức tạp.

Học máy (machine learning) đang mở ra hướng tiếp cận mới, giúp tăng độ chính xác và tự động hóa chẩn đoán động kinh. Các mô hình như SVM, ANN và CNN có khả năng xử lý lượng lớn dữ liệu EEG, phát hiện các mẫu tín hiệu mà con người khó nhận ra. Đặc biệt, CNN đã đạt độ chính xác trên 90% trong nhiều nghiên cứu, giúp phát hiện và dự đoán cơn co giật hiệu quả. Một số hệ thống còn có thể dự đoán trước vài phút, giúp bệnh nhân chuẩn bị và giảm nguy cơ chấn thương.

Việc ứng dụng học máy vào dự đoán động kinh không chỉ khắc phục hạn chế của phương pháp truyền thống mà còn mang lại lợi ích vượt trội về độ chính xác, chi phí và tính cá nhân hóa. Từ thực tế đó, nghiên cứu về “Ứng dụng các mô hình học máy trong dự đoán cơn động kinh” là cần thiết và có ý nghĩa thực tiễn cao.

## 1.2. Một số thông tin liên quan

### 1.2.1.Thách thức trong dự đoán động kinh

Việc dự đoán chính xác thời điểm xảy ra cơn động kinh không hề đơn giản do nhiều yếu tố tác động đến tín hiệu EEG. Một số thách thức chính bao gồm:

* Sự biến động của tín hiệu EEG theo thời gian: Hoạt động điện não của mỗi người không giống nhau và có thể thay đổi tùy thuộc vào trạng thái cơ thể (tỉnh táo, ngủ, mệt mỏi). Điều này khiến mô hình học máy khó tìm ra một quy luật chung áp dụng cho tất cả bệnh nhân.
* Sự nhiễu loạn từ môi trường: Tín hiệu EEG rất nhạy cảm với nhiễu từ các chuyển động của bệnh nhân, thiết bị đo, hoặc các tác nhân bên ngoài, làm ảnh hưởng đến chất lượng dữ liệu.
* Thiếu dữ liệu gán nhãn chất lượng cao: Việc thu thập dữ liệu EEG đòi hỏi chuyên gia y tế phân loại chính xác đâu là tín hiệu bình thường, đâu là tín hiệu báo hiệu cơn động kinh. Tuy nhiên, quy trình này rất tốn thời gian và công sức, dẫn đến số lượng dữ liệu được gán nhãn chính xác còn hạn chế.

Những thách thức này đặt ra yêu cầu cần có các phương pháp xử lý dữ liệu tiên tiến và lựa chọn mô hình học máy phù hợp để đạt hiệu suất dự đoán cao nhất.

### 1.2.2. Các phương pháp học máy phổ biến

Để giải quyết bài toán dự đoán cơn động kinh, nhiều phương pháp học máy đã được áp dụng. Một số thuật toán tiêu biểu bao gồm:

* Support Vector Machine (SVM): Một thuật toán mạnh trong phân loại, có khả năng xác định ranh giới giữa các trạng thái não bộ bằng cách tối ưu hóa siêu phẳng phân tách dữ liệu EEG.
* Iterative Dichotomiser 3 (ID3): Mô hình cây quyết định đơn giản nhưng hiệu quả trong việc xác định các đặc trưng quan trọng từ tín hiệu EEG.
* Artificial Neural Networks (ANN): Hệ thống mạng nơ-ron có khả năng học các mẫu tín hiệu phức tạp và tạo ra các dự đoán chính xác hơn.
* Convolutional Neural Networks (CNN): Mạng nơ-ron tích chập giúp trích xuất đặc trưng tự động từ dữ liệu EEG, giảm bớt sự phụ thuộc vào các phương pháp trích xuất đặc trưng thủ công.

### 1.2.3. Các nghiên cứu và ứng dụng thực tế

#### 1.2.3.1. Các nghiên cứu trước đây

Nhiều nghiên cứu đã ứng dụng học máy vào dự đoán cơn động kinh dựa trên dữ liệu EEG, thay thế phương pháp truyền thống. Một số nghiên cứu tiêu biểu:

* Shoeb & Guttag (2010): Dùng SVM, đạt 96% độ chính xác trong phân loại trạng thái bình thường và tiền co giật.
* Acharya et al. (2018): Áp dụng CNN, đạt trên 90% độ chính xác khi trích xuất đặc trưng từ EEG.
* Daoud & Bayoumi (2019): Kết hợp CNN và LSTM để phân tích chuỗi thời gian EEG, cải thiện hiệu suất dự đoán.

Các nghiên cứu cho thấy mô hình học sâu (Deep Learning), đặc biệt là CNN, mang lại hiệu quả cao trong nhận diện tín hiệu động kinh.

#### 1.2.3.2. Ứng dụng thực tế

Nhờ các tiến bộ trong học máy, nhiều hệ thống hỗ trợ giám sát cơn động kinh đã được phát triển:

* Thiết bị đeo thông minh: Empatica Embrace giám sát EEG liên tục, cảnh báo sớm khi có dấu hiệu co giật.
* Ứng dụng điện thoại: EpiWatch theo dõi hoạt động thần kinh, hỗ trợ quản lý bệnh động kinh.
* Hệ thống giám sát y tế từ xa: AI kết hợp dữ liệu đám mây giúp bác sĩ theo dõi bệnh nhân và phát hiện bất thường kịp thời.

Những ứng dụng này giúp nâng cao độ chính xác trong dự đoán và cải thiện chất lượng cuộc sống của bệnh nhân động kinh.

### 1.2.4. Mục tiêu nghiên cứu

#### 1.2.4.1. Mục tiêu tổng quát

Xây dựng một ứng dụng học máy có khả năng dự đoán cơn động kinh dựa trên dữ liệu điện não đồ (EEG) và các đặc điểm lâm sàng của bệnh nhân.

#### 1.2.4.2. Mục tiêu cụ thể

* Tìm hiểu bài toán dự đoán động kinh: Nghiên cứu về động kinh và các đặc điểm giúp dự đoán cơn động kinh.
* Nghiên cứu các kỹ thuật học máy: Tìm hiểu và áp dụng các mô hình như SVM, Iterative Dichotomiser 3 (ID3) để phân loại và dự đoán chính xác cơn động kinh.
* Xây dựng và huấn luyện mô hình học máy: Thu thập dữ liệu EEG, thực hiện tiền xử lý dữ liệu, loại bỏ nhiễu và tối ưu hóa mô hình để nâng cao hiệu suất dự đoán.
* Đánh giá mô hình: Sử dụng các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision) và điểm F1 (F1-score) để đo lường hiệu suất mô hình. Ngoài ra, áp dụng kỹ thuật cross-validation để kiểm tra khả năng tổng quát hóa của mô hình trên dữ liệu mới.

### 1.2.5. Đối tượng nghiên cứu và phạm vi nghiên cứu

* Đối tượng nghiên cứu

Nghiên cứu các mô hình học máy được áp dụng trong dự đoán cơn động kinh từ dữ liệu thu thập được.

* Phạm vi nghiên cứu
* Loại dữ liệu: Dữ liệu được thu thập từ tín hiệu EEG và các đặc điểm lâm sàng của bệnh nhân động kinh.
* Phương pháp học máy: Sử dụng các thuật toán như SVM, Iterative Dichotomiser 3 (ID3) để phân loại và dự đoán cơn động kinh.
* Đánh giá mô hình: Kiểm tra hiệu suất mô hình dựa trên các chỉ số như accuracy, precision, recall và F1-score. Kỹ thuật cross-validation sẽ được sử dụng để đảm bảo mô hình có thể tổng quát hóa tốt trên dữ liệu chưa thấy, giúp đánh giá khả năng dự đoán chính xác các lớp trong các tình huống khác nhau. là phát triển và triển khai một mô hình học máy để dự đoán nguy cơ động kinh dựa trên các yếu tố đầu vào từ dữ liệu EEG và đặc điểm lâm sàng của bệnh nhân. Các nội dung chính của nghiên cứu bao gồm:
* Tìm hiểu bài toán dự đoán động kinh: Bệnh động kinh có thể được dự đoán dựa trên các đặc điểm cụ thể của bệnh nhân.
* Tìm hiểu và nghiên cứu các kỹ thuật học máy: Việc lựa chọn các mô hình học máy phù hợp như SVM, Iterative Dichotomiser 3 (ID3) giúp mô hình dự đoán đạt độ chính xác cao trong việc phân loại các cơn động kinh.
* Áp dụng các kỹ thuật học máy để dự đoán động kinh: Thu thập và tiền xử lý dữ liệu, loại bỏ nhiễu, xây dựng và huấn luyện mô hình học máy, sau đó tối ưu hóa mô hình để đạt hiệu quả dự đoán tốt nhất.
* Đánh giá mô hình và tính hiệu quả của thuật toán: Sử dụng các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision) và điểm F1 (F1-score) để đánh giá hiệu suất của mô hình. Ngoài ra, áp dụng kỹ thuật đánh giá như cross-validation để kiểm tra khả năng tổng quát hóa của mô hình khi áp dụng trên dữ liệu mới.

## 1.3. Kết luận

Từ những vấn đề đã phân tích, có thể thấy rằng việc áp dụng học máy vào dự đoán động kinh không chỉ giúp nâng cao độ chính xác trong chẩn đoán mà còn mở ra khả năng cảnh báo sớm, hỗ trợ bệnh nhân và bác sĩ đưa ra biện pháp can thiệp kịp thời. Tuy nhiên, để đạt hiệu quả cao, cần có phương pháp tiền xử lý dữ liệu EEG phù hợp, lựa chọn mô hình tối ưu và tinh chỉnh các tham số mô hình.

Nghiên cứu này sẽ tập trung vào việc phát triển mô hình dự đoán động kinh có độ chính xác cao, đồng thời đánh giá hiệu suất của các thuật toán khác nhau để tìm ra phương pháp tối ưu nhất. Kết quả của nghiên cứu không chỉ đóng góp vào lĩnh vực y học mà còn có tiềm năng ứng dụng rộng rãi trong thực tế, giúp cải thiện chất lượng cuộc sống của bệnh nhân động kinh.

# PHẦN II. PHƯƠNG PHÁP THỰC HIỆN

## 2.1. Đặt vấn đề

Động kinh là một rối loạn thần kinh phổ biến, ảnh hưởng đến hàng triệu người trên toàn thế giới. Việc dự đoán chính xác cơn động kinh có ý nghĩa quan trọng trong việc giúp bệnh nhân chủ động phòng tránh nguy hiểm và cải thiện chất lượng cuộc sống.

Tuy nhiên, các phương pháp chẩn đoán truyền thống dựa trên phân tích tín hiệu điện não đồ (EEG) gặp nhiều thách thức do tính phức tạp, dễ nhiễu của dữ liệu và sự khác biệt giữa các bệnh nhân. Điều này đòi hỏi các phương pháp phân tích tự động, chính xác và đáng tin cậy hơn.

Học máy đang trở thành một công cụ quan trọng trong việc phát hiện và phân loại các mẫu dữ liệu EEG để dự đoán cơn động kinh. Các mô hình học máy có thể trích xuất đặc trưng quan trọng từ tín hiệu EEG, giúp bác sĩ đưa ra quyết định nhanh chóng và hiệu quả. Nghiên cứu này tập trung vào việc áp dụng và so sánh các mô hình học máy khác nhau để đánh giá hiệu suất dự đoán cơn động kinh, từ đó tìm ra phương pháp tối ưu nhất.

## 2.2. Một số mô hình giải quyết bài toán

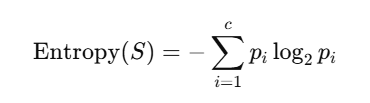
Trong nghiên cứu này, ba mô hình học máy chính được sử dụng để giải quyết bài toán dự đoán cơn động kinh:

### 2.2.1.Thuật toán Iterative Dichotomiser 3

ID3 là một thuật toán xây dựng cây quyết định. Thuật toán này sử dụng khái niệm entropy và gain thông tin để xác định thuộc tính tốt nhất tại mỗi nút trong cây quyết định. Quá trình hoạt động của ID3 dựa trên việc phân loại dữ liệu một cách đệ quy, tiếp tục chia nhỏ các dữ liệu cho đến khi đạt được phân loại hoàn chỉnh hoặc không thể chia tiếp được nữa.

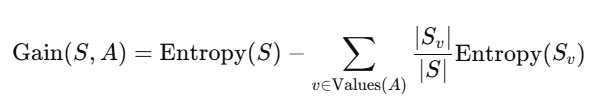
#### 2.2.1.1. Nguyên lý hoạt động

- Tính Entropy



* pi​: Tỷ lệ phần tử thuộc lớp i trong tập dữ liệu S.

- Tính Gain thông tin

****

* A: Thuộc tính đang xét.
* Sv​: Tập con dữ liệu khi A=v

- Chọn thuộc tính tốt nhất: Chọn thuộc tính có Gain thông tin lớn nhất để làm nút phân chia tại mỗi bước.

- Lặp lại: Áp dụng đệ quy trên các tập con, tiếp tục phân chia cho đến khi tất cả các điểm dữ liệu trong tập con thuộc cùng một lớp hoặc không còn thuộc tính nào để chia.

#### 2.2.1.2. Đầu vào và đầu ra

- Đầu vào: Dữ liệu huấn luyện gồm các đặc trưng (features) và nhãn (labels).

- Đầu ra: Một cây quyết định có thể sử dụng để phân loại hoặc dự đoán dữ liệu mới.

#### 2.2.1.3. Đặc điểm

- Phân loại dựa trên thuộc tính: Sử dụng các thuộc tính để chia dữ liệu thành các nhóm.

- Cây quyết định: Kết quả của thuật toán là một cây quyết định với các nút lá biểu thị kết quả phân loại.

- Dựa vào Entropy và Gain thông tin: ID3 chọn thuộc tính chia nhỏ dữ liệu dựa trên giá trị Gain thông tin cao nhất.

#### 2.2.1.4. Ưu điểm

- Dễ hiểu và dễ triển khai: Cây quyết định có cấu trúc rõ ràng, dễ dàng giải thích và biểu diễn trực quan.

- Tốc độ nhanh: Xây dựng cây với tốc độ nhanh đối với tập dữ liệu nhỏ hoặc trung bình.

- Hiệu quả với dữ liệu phân loại: Phù hợp khi các thuộc tính có dạng phân loại.

#### 2.2.1.5. Ứng dụng thực tế

- Hệ thống chẩn đoán y khoa: Xây dựng cây quyết định để hỗ trợ bác sĩ chẩn đoán bệnh dựa trên triệu chứng.

- Quản lý rủi ro tài chính: Phân loại các khoản vay có rủi ro cao và thấp dựa trên thông tin tài chính.

- Thương mại điện tử: Phân loại khách hàng theo hành vi mua sắm để tối ưu hóa quảng cáo.

- Phân tích giáo dục: Phân loại học sinh theo thành tích dựa trên các yếu tố như thời gian học, điểm số.

### 2.2.2. Thuật toán Support Vector Machine

Support Vector Machine là một thuật toán học máy dựa trên lý thuyết không gian hình học. Trong các bài toán phân loại, mục tiêu của Support Vector Machine là tìm ra một siêu phẳng (hyperplane) trong không gian nhiều chiều sao cho có thể phân chia các lớp dữ liệu khác nhau mà không làm sai lệch dữ liệu của lớp nào. Siêu phẳng này không chỉ phân tách các điểm dữ liệu mà còn tối đa hóa khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất của mỗi lớp. Những điểm này được gọi là vector hỗ trợ, vì chúng là các điểm quan trọng nhất trong việc xác định siêu phẳng tối ưu.

#### 2.2.2.1.Nguyên lý hoạt động

- SVM cố gắng tìm một siêu phẳng trong không gian đặc trưng để phân chia dữ liệu thành các lớp khác nhau.

- Siêu phẳng tối ưu là siêu phẳng tối đa hóa khoảng cách giữa các điểm dữ liệu gần nhất thuộc các lớp khác nhau, được gọi là vector hỗ trợ.

- SVM có thể sử dụng hạt nhân để chuyển đổi dữ liệu từ không gian ban đầu sang không gian đặc trưng cao hơn, giúp giải quyết các bài toán không tuyến tính.

#### 2.2.2.2. Đầu vào và đầu ra

- Đầu vào:

* Tập dữ liệu huấn luyện với các đặc trưng và nhãn.
* Lựa chọn hạt nhân (kernel) phù hợp (linear, polynomial, RBF).

- Đầu ra:

* Với phân loại: Dự đoán nhãn cho dữ liệu mới.
* Với hồi quy: Dự đoán giá trị liên tục.

#### 2.2.2.3. Ưu điểm

SVM, là một kỹ thuật phân lớp phổ biến, có nhiều ưu điểm nổi bật, đặc biệt là trong việc tính toán hiệu quả trên các tập dữ liệu lớn.

- Xử lý trong không gian chiều cao: SVM là một công cụ mạnh mẽ khi làm việc với các không gian có số chiều cao, điều này đặc biệt hữu ích trong các bài toán phân loại văn bản và phân tích cảm xúc, nơi mà không gian đặc trưng có thể rất lớn.

- Tiết kiệm bộ nhớ: SVM chỉ lưu trữ một tập con các điểm dữ liệu trong quá trình huấn luyện và khi đưa ra quyết định cho các điểm mới, điều này giúp giảm thiểu việc sử dụng bộ nhớ. Chỉ những điểm dữ liệu quan trọng nhất, gọi là vector hỗ trợ, được lưu trữ và sử dụng trong quá trình dự đoán.

- Tính linh hoạt cao: SVM cho phép áp dụng các hàm hạt nhân (kernel), giúp chuyển đổi dữ liệu vào không gian chiều cao hơn để phân tách các lớp dữ liệu phi tuyến tính. Khả năng linh động trong việc sử dụng các kernel khác nhau giúp SVM có thể chuyển đổi giữa phương pháp phân lớp tuyến tính và phi tuyến tính, cải thiện hiệu suất phân loại.

#### 2.2.2.4. Ứng dụng thực tế

- Nhận dạng chữ viết tay: SVM hỗ trợ chuyển đổi chữ viết tay từ tài liệu giấy thành văn bản kỹ thuật số, giúp tự động hóa quá trình số hóa tài liệu và xử lý các mẫu đơn, tài liệu hành chính.

- Hệ thống gợi ý: Được sử dụng trong các hệ thống khuyến nghị, chẳng hạn như trên các nền tảng thương mại điện tử, để gợi ý sản phẩm hoặc dịch vụ cho người dùng dựa trên thói quen mua sắm và sở thích của họ.

- Dự báo khoa học: Sử dụng trong dự báo thời tiết, giúp phân loại các hiện tượng khí tượng như bão, mưa lớn hoặc sóng nhiệt.

- Mạng xã hội: SVM giúp phân loại người dùng trong các nền tảng mạng xã hội như Facebook hoặc Twitter dựa trên các yếu tố như hành vi, sở thích hoặc độ tuổi, từ đó cá nhân hóa trải nghiệm người dùng

### 2.2.3. Thuật toán Artificial Neural Network

Mạng Nơ-ron Nhân tạo là một thuật toán học máy được thiết kế dựa trên cách thức hoạt động của hệ thần kinh con người. ANN giúp xử lý và phân tích các dữ liệu phức tạp thông qua các mô hình học sâu. Mạng này bao gồm nhiều lớp nơ-ron, với mỗi lớp có vai trò xử lý thông tin theo cách tương tự cách bộ não con người xử lý tín hiệu.

#### 2.2.3.1.Nguyên lý hoạt động

- Dữ liệu đầu vào được đưa vào các nơ-ron của lớp đầu tiên.

- Các nơ-ron trong mỗi lớp sẽ tính toán giá trị đầu ra bằng cách sử dụng trọng số và hàm kích hoạt.

- Kết quả đầu ra từ lớp cuối cùng được so sánh với giá trị mục tiêu để tính toán lỗi.

- Thuật toán sẽ sử dụng các phương pháp tối ưu để điều chỉnh trọng số nhằm giảm thiểu lỗi.

#### 2.2.3.2. Đầu vào và đầu ra

- Đầu vào: Tập dữ liệu huấn luyện với các đặc trưng và nhãn.

- Đầu ra: Dự đoán nhãn cho dữ liệu mới.

#### 2.2.3.3. Đặc điểm

- Cấu trúc mạng: Bao gồm các lớp nơ-ron, bao gồm lớp đầu vào, một hoặc nhiều lớp ẩn, và lớp đầu ra.

- Học giám sát và không giám sát: Có thể huấn luyện với cả dữ liệu có nhãn hoặc không có nhãn.

- Hàm kích hoạt: Các hàm kích hoạt như ReLU, Sigmoid, hoặc Tanh được sử dụng để điều chỉnh tín hiệu trong các nơ-ron.

- Khả năng tự học: ANN có khả năng tự điều chỉnh và học từ dữ liệu mà không cần phải lập trình chi tiết.

#### 2.2.3.4. Ưu điểm

- Khả năng học phi tuyến tính: Mạng nơ-ron có thể mô hình hóa các mối quan hệ phi tuyến tính rất tốt, điều mà các phương pháp học máy truyền thống khó thực hiện.

- Khả năng tự học: Không cần phải biết rõ quy luật chi tiết của vấn đề, mạng có thể tự học từ dữ liệu.

- Xử lý dữ liệu phức tạp: ANN có thể xử lý các vấn đề phức tạp như nhận diện hình ảnh, phân loại văn bản, dự đoán thời gian, ...

- Khả năng mở rộng: Có thể mở rộng kích thước mạng, thêm các lớp và nơ-ron để tăng độ chính xác và hiệu suất.

#### 2.2.3.5. Ứng dụng thực tế

- Nhận dạng hình ảnh: Các ứng dụng như nhận diện khuôn mặt, nhận diện chữ viết tay, nhận diện đối tượng trong video.

- Xử lý ngôn ngữ tự nhiên: Dùng trong các hệ thống dịch máy, chatbot, phân tích cảm xúc, nhận dạng giọng nói.

- Dự báo và phân tích dữ liệu: Dự đoán giá cổ phiếu, phân tích tín dụng, dự báo nhu cầu tiêu dùng.

- Xe tự lái: Các mạng nơ-ron giúp xe tự lái nhận diện môi trường xung quanh và đưa ra quyết định điều khiển.

- Y tế: Chẩn đoán bệnh qua hình ảnh y tế, phân tích gen, dự đoán hiệu quả điều trị.

## 2.3. Kết luận

Việc áp dụng các mô hình học máy vào dự đoán cơn động kinh mang lại tiềm năng lớn trong hỗ trợ chẩn đoán và điều trị bệnh. Trong nghiên cứu này, ba mô hình chính đã được xem xét: SVM, ANN và ID3.

* SVM phù hợp với dữ liệu có nhiễu cao và giúp phân loại trạng thái não bộ hiệu quả.
* ANN có khả năng học các đặc trưng phức tạp nhưng đòi hỏi dữ liệu lớn và tài nguyên tính toán cao.
* ID3 đơn giản, dễ triển khai nhưng nhạy cảm với nhiễu và có thể bị overfitting.

Kết quả thực nghiệm cho thấy việc kết hợp tiền xử lý dữ liệu (lọc nhiễu, chuẩn hóa, trích xuất đặc trưng) giúp cải thiện đáng kể hiệu suất dự đoán. Lựa chọn mô hình tối ưu phụ thuộc vào tính chất dữ liệu EEG và yêu cầu thực tế của ứng dụng. Trong phần tiếp theo, chúng tôi sẽ trình bày chi tiết về thực nghiệm và so sánh kết quả của các mô hình để xác định phương pháp hiệu quả nhất.

# PHẦN III. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

## 3.1. Công nghệ sử dụng

### 3.1.1. Ngôn ngữ

- Python:

Một ngôn ngữ lập trình phổ biến, dễ học, mạnh mẽ, tương thích với nhiều hệ điều hành, ứng dụng rộng rãi trong phát triển web, phần mềm, khoa học dữ liệu và học máy.

- HTML CSS:

HTML giúp cấu trúc nội dung trang web, trong khi CSS định dạng giao diện, tạo sự tách biệt giữa nội dung và thiết kế, giúp trang web linh hoạt và dễ duy trì.

### 3.1.2. Các thư viện sử dụng

- NumPy: Thư viện quan trọng giúp xử lý dữ liệu số, cung cấp cấu trúc mảng nhiều chiều (ndarray), hỗ trợ các phép toán đại số tuyến tính, thống kê và tối ưu hóa hiệu suất tính toán với C và Fortran.

- Pandas: Hỗ trợ thao tác với dữ liệu dạng bảng thông qua DataFrame và Series, giúp xử lý dữ liệu thiếu, chuyển đổi định dạng, phân tích và trực quan hóa dữ liệu EEG một cách hiệu quả.

- Scikit-learn: Cung cấp nhiều thuật toán học máy như SVM, hồi quy tuyến tính, cây quyết định, giúp tiền xử lý dữ liệu, huấn luyện mô hình và đánh giá hiệu suất dự đoán cơn động kinh.

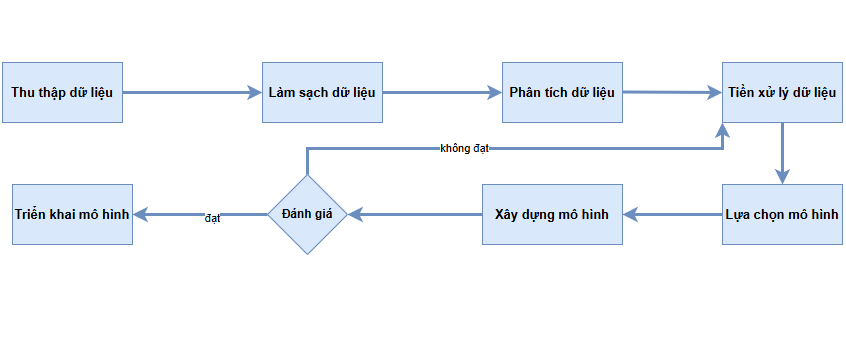
- Matplotlib: Thư viện trực quan hóa dữ liệu mạnh mẽ, hỗ trợ vẽ các loại biểu đồ như đường, cột, phân tán, giúp phân tích xu hướng và biểu diễn kết quả mô hình học máy trên dữ liệu EEG.

### 3.1.3. Phát biểu bài toán

Bài toán dự đoán cơn động kinh dựa trên dữ liệu điện não đồ (EEG) bằng các phương pháp học máy. Thách thức chính là xử lý lượng lớn dữ liệu phức tạp, nhiễu cao, yêu cầu các kỹ thuật phân tích tiên tiến. Mục tiêu là xây dựng mô hình nhận diện mẫu tín hiệu đặc trưng của cơn động kinh, dự đoán chính xác để hỗ trợ bệnh nhân giảm nguy cơ chấn thương. Các thuật toán như ANN, SVM, ID3 sẽ được áp dụng nhằm tối ưu hóa khả năng phát hiện và dự đoán, góp phần nâng cao chất lượng cuộc sống cho bệnh nhân.

## 3.2. Triển khai thực nghiệm

### 3.2.1. Quy trình tổng quan bài toán



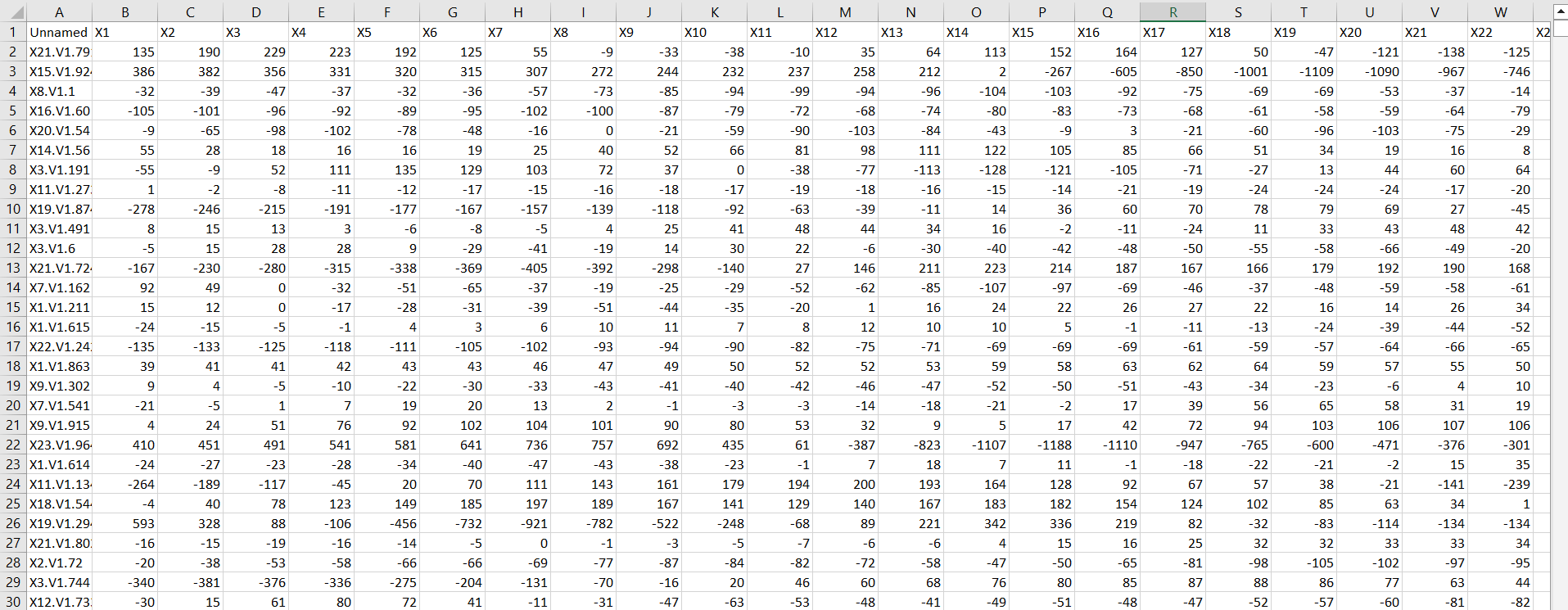
###### Hình 3.2.1. Quy trình tổng quan bài toán

#### 3.2.1.1.Thu thập dữ liệu

Tập dữ liệu này được tổ chức lại từ một nguồn phổ biến để phát hiện cơn động kinh, gồm 500 cá nhân với tín hiệu EEG dài 23,6 giây, chia thành 11.500 mẫu (mỗi mẫu 178 điểm dữ liệu). Nó chứa 178 đặc trưng EEG và 5 lớp nhãn, với mục tiêu phân loại cơn động kinh (lớp 1) và không động kinh (lớp 2-5). Dữ liệu hỗ trợ phát triển thuật toán học máy giúp chẩn đoán và điều trị hiệu quả hơn.

Nguồn dữ liệu:

<https://www.kaggle.com/code/harunshimanto/machine-learning-algorithms-for-epileptic-seizures/notebook>

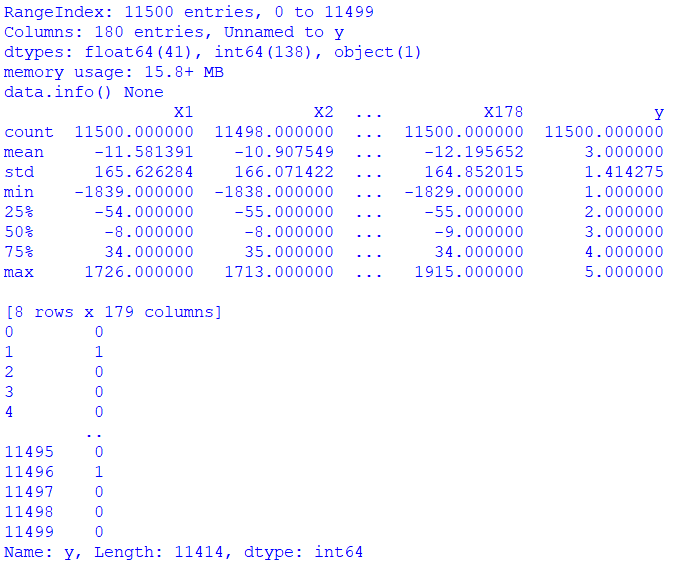


###### Hình 3.2.1.1. Thông tin dữ liệu thu thập

#### 3.2.1.2. Làm sạch dữ liệu

##### 3.2.1.2.1. Xóa các thông tin bị thiếu

Trong quá trình thu thập dữ liệu thiếu có thể xuất hiện do nhiều nguyên nhân, chẳng hạn như lỗi trong quá trình thu thập dữ liệu hoặc sự cố trong thiết bị đo tín hiêu não bộ. Việc xóa các dòng bị thiếu là một bước quan trọng nhằm đảm bảo tính chính xác và độ tin cậy của các phân tích sau này.



###### Hình 3.2.1.2.1. Hình ảnh thông tin dữ liệu

| data.dropna(inplace=True) |
| --- |

##### 3.2.1.2.2. Xóa dữ liệu trùng lặp

Thông tin trùng lặp không mang lại giá trị bổ sung cho mô hình mà có thể làm giảm hiệu quả và độ chính xác của nó. Do đó, việc loại bỏ các dữ liệu dư thừa không cần thiết là một bước quan trọng, giúp tối ưu hóa mô hình và cải thiện khả năng học tập, đồng thời giảm thiểu sự phức tạp không cần thiết.

| data.drop\_duplicates(inplace=True) |
| --- |

#### 3.2.1.3. Phân tích dữ liệu

Dữ liệu ban đầu được thu thập với 4097 điểm dữ liệu từ tín hiệu EEG. Sau đó, dữ liệu này được chia thành các khối nhỏ hơn, mỗi khối chứa 178 điểm dữ liệu (tương đương với 1 giây). Mỗi mẫu dữ liệu trong tập chứa 178 đặc trưng EEG, phản ánh hoạt động của não bộ trong khoảng thời gian 1 giây, và một nhãn kết quả thể hiện trạng thái của tín hiệu EEG trong khoảng thời gian đó.

Chính vì thế nên ta sẽ sử dụng tất cả các đặc trưng để huấn luyện mô hình.

#### 3.2.1.4. Tiền xử lý dữ liệu

##### 3.2.1.4.1. Chuẩn hóa dữ liệu

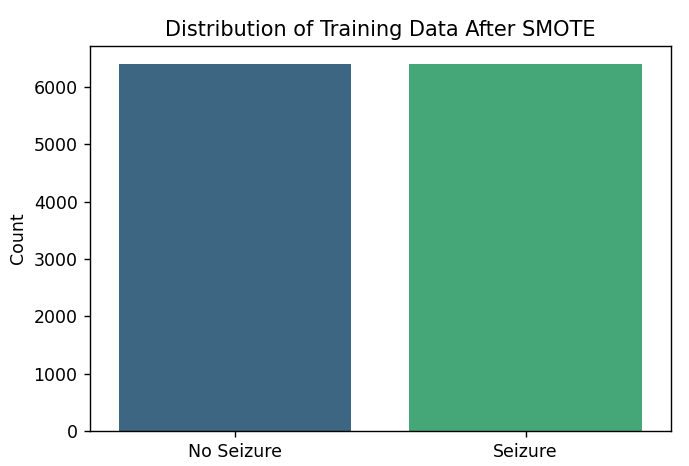
Chuẩn hóa dữ liệu là bước quan trọng trong tiền xử lý dữ liệu, mang lại nhiều lợi ích cho việc phát hiện cơn động kinh từ tín hiệu EEG. Việc này giúp cải thiện tính chính xác của mô hình bằng cách đưa các biến về cùng một thang đo, từ đó giảm thiểu sự thiên lệch và tăng tốc độ hội tụ của các thuật toán tối ưu hóa. Chuẩn hóa cũng làm giảm độ phức tạp của dữ liệu, tăng cường khả năng so sánh giữa các biến và giảm thiểu ảnh hưởng của các biến ngoại lai. Nhờ đó, nó tạo ra một nền tảng vững chắc cho phân tích và diễn giải kết quả, đặc biệt quan trọng trong các ứng dụng nhạy cảm như chẩn đoán cơn động kinh.

| scaler = StandardScaler()  X = scaler.fit\_transform(X) |
| --- |

##### 3.2.1.4.2. Cân bằng dữ liệu

Nhận thấy rằng dữ liệu có sự phân phối không đồng đều giữa các nhãn kết quả. Để khắc phục vấn đề này, em sẽ sử dụng phương pháp SMOTEđể cân bằng dữ liệu. SMOTE giúp tạo ra các mẫu tổng hợp cho lớp thiểu số, từ đó giảm thiểu sự thiên lệch trong dữ liệu.

| smote = SMOTE(random\_state=42)  X\_train\_smote, y\_train\_smote = smote.fit\_resample(X\_train, y\_train) |
| --- |



###### Hình 3.2.1.4.2. Giá trị sau khi cân bằng

#### 3.2.1.5. Lựa chọn mô hình

Sử dụng các mô hình phân loại:

- SVM

- ID3

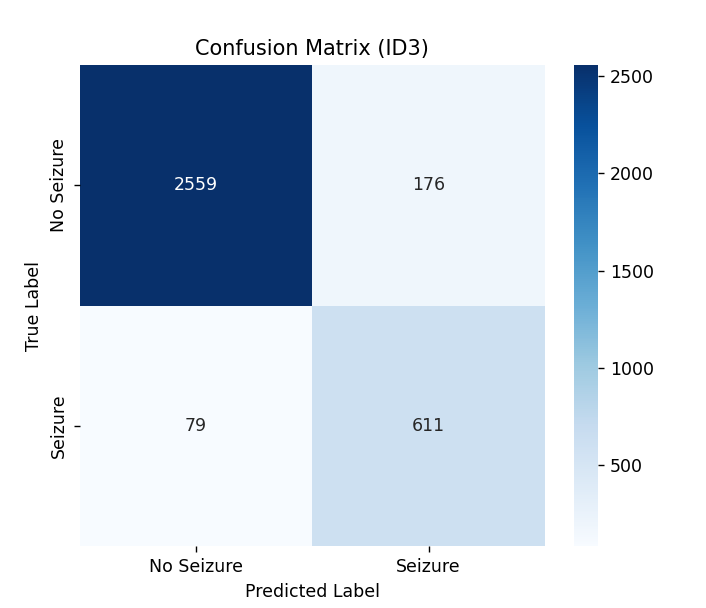
- ANN

#### 3.2.1.6. Huấn luyện mô hình

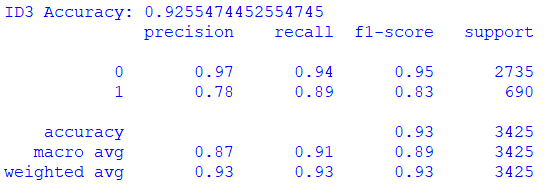
| # 5. Huấn luyện mô hình Iterative Dichotomiser 3  rf = RandomForestClassifier(n\_estimators=100, random\_state=42)  rf.fit(X\_train\_smote, y\_train\_smote)  y\_pred\_rf = rf.predict(X\_test)  y\_pred\_prob\_rf = rf.predict\_proba(X\_test)[:, 1] # Xác suất dự đoán để vẽ ROC  print("Iterative Dichotomiser 3 Accuracy:", accuracy\_score(y\_test, y\_pred\_rf))  print(classification\_report(y\_test, y\_pred\_rf))  # 5. Huấn luyện mô hình SVM  svm = SVC(probability=True, random\_state=42)  svm.fit(X\_train\_smote, y\_train\_smote)  y\_pred\_svm = svm.predict(X\_test)  y\_pred\_prob\_svm = svm.predict\_proba(X\_test)[:, 1] # Xác suất dự đoán để vẽ ROC  print("SVM Accuracy:", accuracy\_score(y\_test, y\_pred\_svm))  print(classification\_report(y\_test, y\_pred\_svm))  # Xây dựng mô hình ANN  model = Sequential([      Dense(128, activation='relu', input\_shape=(X\_train.shape[1],)),      Dropout(0.3),      Dense(64, activation='relu'),      Dropout(0.3),      Dense(1, activation='sigmoid')  # Output layer for binary classification  ])  model.compile(optimizer=Adam(learning\_rate=0.001), loss='binary\_crossentropy', metrics=['accuracy']) |
| --- |

## 3.3. Đánh giá mô hình

### 3.3.1. Iterative Dichotomiser 3



###### Hình 3.3.1.1. Ma trận matrix thuật toán Iterative Dichotomiser 3

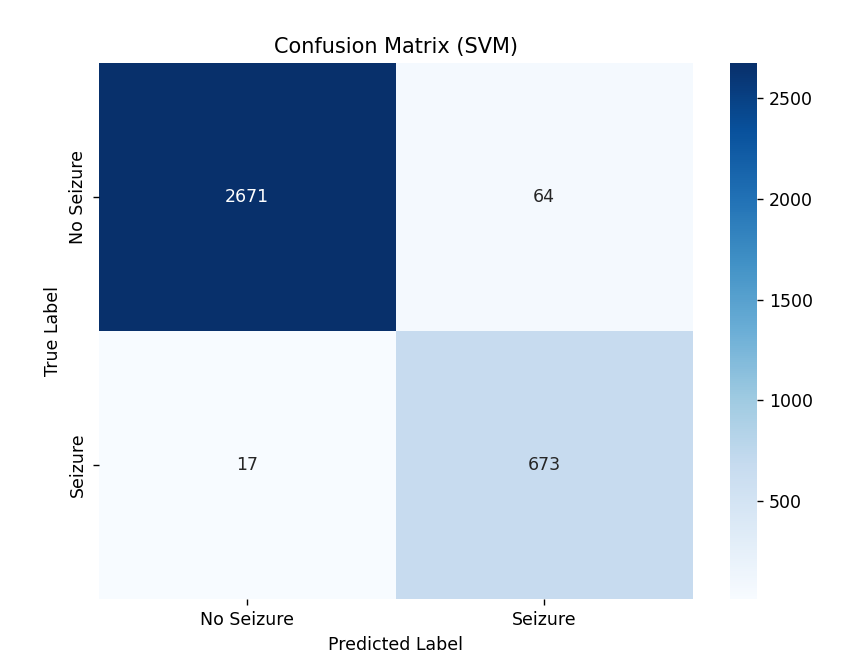


###### Hình 3.3.1.2. Kết quả báo cáo phân loại thuật toán Iterative Dichotomiser 3

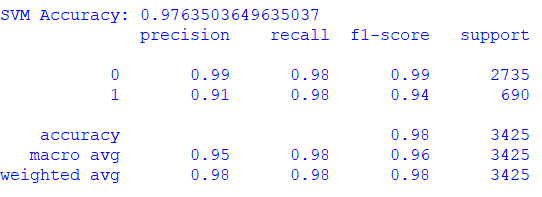
Nhận xét:

* Tỷ lệ dự đoán của mô hình: 92%
* Số lượng dự đoán "Không co giật" đúng là 2599.
* Số lượng dự đoán "Co giật" đúng là 611.
* Số lượng dự đoán "Không co giật" nhưng thực tế là "Co giật" là 176.
* Số lượng dự đoán "Co giật" nhưng thực tế là "Không co giật" là 79.

### 3.3.2. SVM



###### Hình 3.3.2.1. Ma trận matrix thuật toán SVM

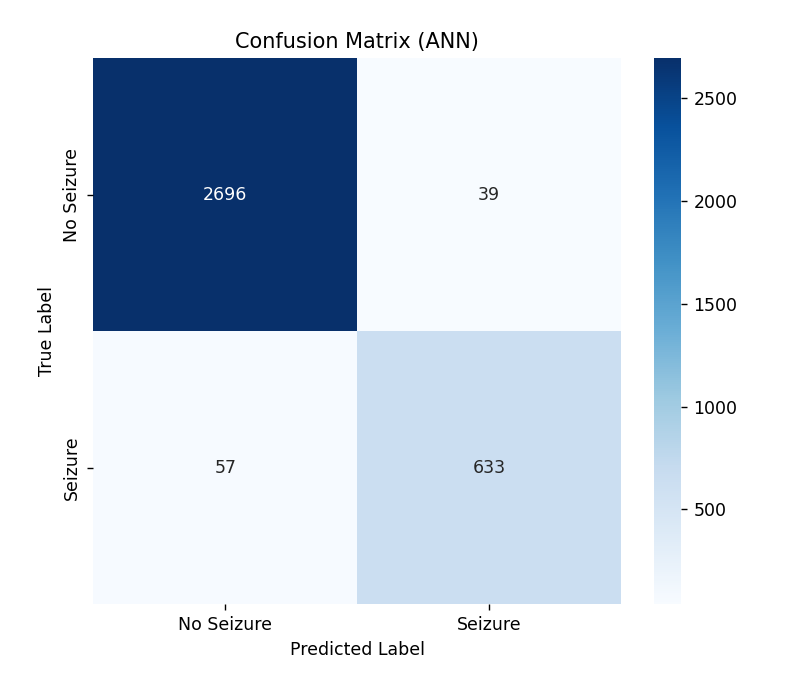


###### Hình 3.3.2.2. Kết quả báo cáo phân loại thuật toán SVM

Nhận xét:

* Tỷ lệ dự đoán của mô hình: 98%
* Số lượng dự đoán "Không co giật" đúng là 2671.
* Số lượng dự đoán "Co giật" đúng là 673.
* Số lượng dự đoán "Không co giật" nhưng thực tế là "Co giật" là 64.
* Số lượng dự đoán "Co giật" nhưng thực tế là "Không co giật" là 17.

### 3.3.3. ANN



###### Hình 3.3.3. Ma trận matrix thuật toán ANN

Nhận xét

* Tỷ lệ dự đoán mô hình: 97%
* Số lượng dự đoán "Không co giật" đúng là 2696.
* Số lượng dự đoán "Co giật" đúng là 633.
* Số lượng dự đoán "Không co giật" nhưng thực tế là "Co giật" là 39.
* Số lượng dự đoán "Co giật" nhưng thực tế là "Không co giật" là 57.

## 3.4. Nhận xét chung

###### Bảng 3.4. Bảng dự đoán của 3 mô hình

| **STT** | **Thuật toán** | **Tỷ lệ dự đoán** |
| --- | --- | --- |
| **1** | **Iterative Dichotomiser 3** | 92% |
| **2** | **SVM** | 98% |
| **3** | **ANN** | 97% |

Ba mô hình Iterative Dichotomiser 3, SVM, và ANN đều thể hiện hiệu suất tốt trong bài toán nhận diện cơn động kinh từ tín hiệu EEG, với độ chính xác lần lượt là 92%, 98%, và 97%. Mỗi mô hình có ưu điểm riêng: Iterative Dichotomiser 3 xử lý hiệu quả dữ liệu phức tạp và không cần nhiều điều chỉnh, ANN có khả năng học các mối quan hệ phi tuyến tính từ dữ liệu lớn, và SVM vượt trội trong phân loại dữ liệu nhị phân nhờ tìm siêu phẳng phân tách tối ưu. Trong số đó, SVM được xem là lựa chọn tối ưu nhất với độ chính xác cao nhất (98%) và khả năng phân loại hiệu quả trên tập dữ liệu đã chuẩn hóa.

## 3.5. Triển khai mô hình

### 3.5.1. Lưu mô hình

Thực hiện lưu các mô hình để tiến hành đưa vào sử dụng

| with open('rf\_model.pkl', 'wb') as file:  pickle.dump(rf, file) |
| --- |

Việc lưu mô hình học máy sau khi huấn luyện mang lại nhiều lợi ích quan trọng, bao gồm:

- Tiết kiệm thời gian: Quá trình huấn luyện mô hình có thể mất nhiều thời gian, đặc biệt với các mô hình phức tạp và dữ liệu lớn. Lưu mô hình đã huấn luyện giúp không phải huấn luyện lại từ đầu mỗi khi cần sử dụng.

- Tái sử dụng mô hình: Mô hình đã được lưu có thể được tái sử dụng trong các ứng dụng thực tế hoặc các dự án tương tự mà không cần xử lý lại dữ liệu hay huấn luyện từ đầu.

- Triển khai mô hình: Một mô hình đã lưu có thể được tích hợp vào các ứng dụng thực tế như API, ứng dụng web, hoặc hệ thống dự báo mà không yêu cầu truy cập mã nguồn hoặc dữ liệu huấn luyện ban đầu.

- Chia sẻ mô hình: Có thể chia sẻ mô hình đã lưu với mọi người để sử dụng hoặc kiểm tra.

- Khả năng phục hồi: Nếu có sự cố xảy ra, chẳng hạn mất dữ liệu hoặc thay đổi môi trường, mô hình đã lưu sẽ đảm bảo rằng bạn vẫn có thể tiếp tục sử dụng nó mà không mất tiến trình trước đó.

### 3.5.2. Triển khai mô hình

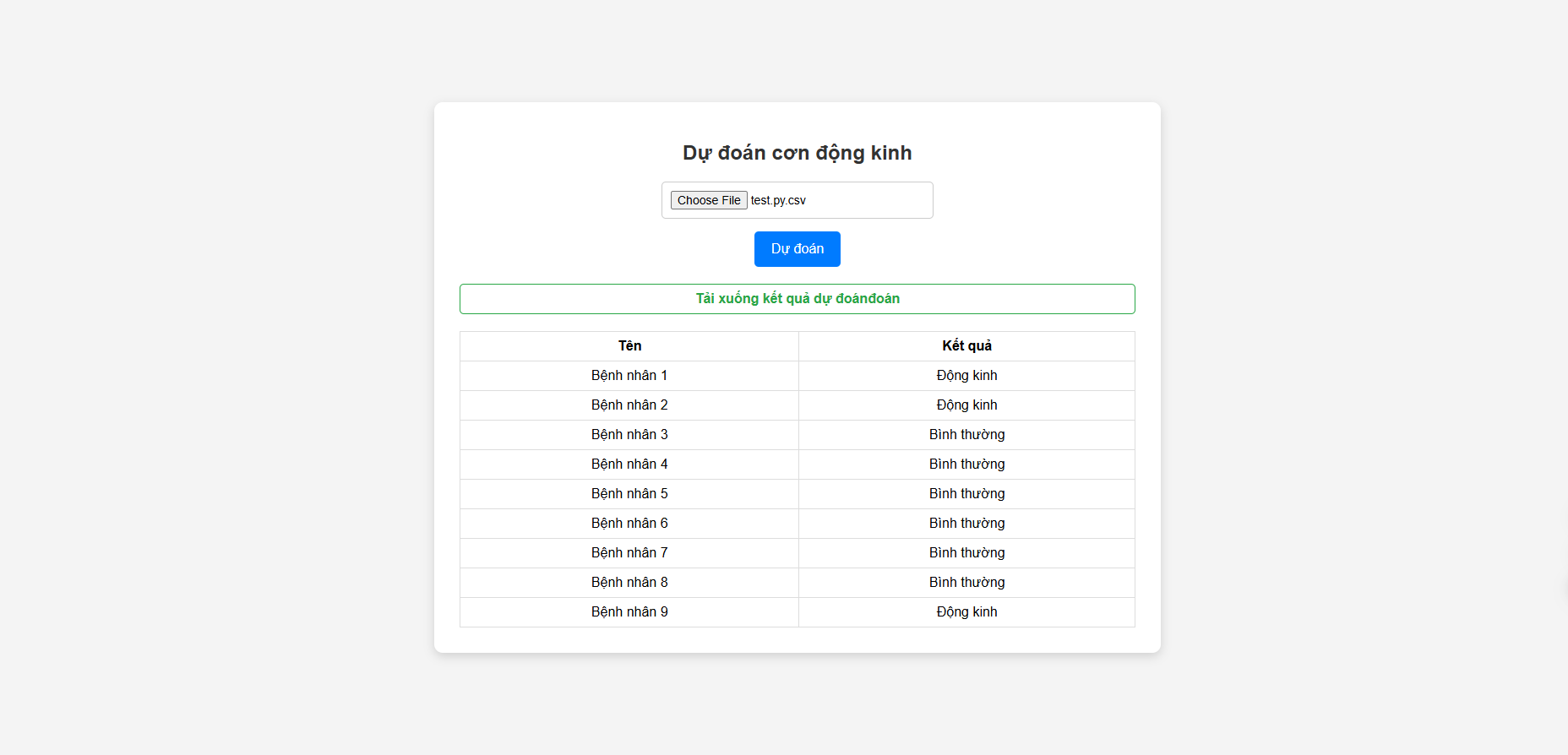
Thực hiện xây dựng giao diện người dùng bằng html và flask và triển khai đến người sử dùng. Cụ thể sẽ xây dựng giao diện cho người dùng nhập file csv thông tin điện não đồ để đưa vào dự đoán(Hình 3.10)

Hệ thống sẽ cho phép người dùng tải lên file CSV chứa thông tin về các bệnh nhân. Sau khi tải lên, người dùng chỉ cần nhấn nút dự đoán và hệ thống sẽ tiến hành xử lý dữ liệu và trả về một file kết quả dự đoán. File này sẽ được cung cấp cho người dùng để tải về, giúp họ dễ dàng xem xét và sử dụng các kết quả từ hệ thống.

Code giao diện sẽ có càng hàm chính sau:

- index(): Thực hiện hiển thị trang giao diện người dùng cho ứng dụng dự đoán cơn động kinh

- predict():Thực hiện tiếp nhận dữ liệu do người dùng nhập từ giao diện, sau đó thực hiện các bước tiền xử lý để đảm bảo dữ liệu đầu vào đáp ứng đúng định dạng và yêu cầu của mô hình dự đoán. Sau khi chuẩn bị dữ liệu, chức năng sẽ đưa dữ liệu vào mô hình đã huấn luyện và lưu trữ, thực hiện quá trình dự đoán, rồi xử lý kết quả để trả về thông tin dự báo một cách rõ ràng và dễ hiểu cho người dùng.



###### Hình 3.5.2. Giao diện người dùng

# PHẦN IV. KẾT LUẬN

Trong đồ án này, chúng em đã phát triển một hệ thống dự đoán cơn động kinh dựa trên dữ liệu EEG, sử dụng các mô hình học máy khác nhau như Iterative Dichotomiser 3, SVM và ANN để phân loại các tín hiệu EEG thành các lớp tương ứng. Các mô hình này đã được huấn luyện và đánh giá hiệu quả với độ chính xác cao, với SVM đạt hiệu suất tốt nhất. Hệ thống không chỉ mang lại kết quả dự đoán chính xác mà còn cung cấp giao diện người dùng đơn giản, dễ sử dụng, cho phép người dùng tải lên dữ liệu và nhận kết quả dự đoán trong thời gian ngắn. Việc ứng dụng các kỹ thuật học máy vào phân tích dữ liệu y tế sẽ góp phần nâng cao khả năng phát hiện và chẩn đoán bệnh một cách tự động, hỗ trợ các bác sĩ trong công tác điều trị và quản lý bệnh nhân

Hoàn thành các mục tiêu ban đầu đề ra:

- Mục tiêu 1: Tìm hiểu bài toán dự đoán cơn động kinh

- Mục tiêu 2: Tìm hiểu, nghiên cứu một số mô hình học máy

- Mục tiêu 3: Huấn luyện mô hình với bộ dữ liệu

- Mục tiêu 4: Đánh giá mô hình

- Mục tiêu 5: Kết luận

Trong tương lai, hệ thống có thể được cải tiến theo nhiều hướng để nâng cao độ chính xác và tính ứng dụng thực tế.

- Cải thiện mô hình học máy: Tinh chỉnh và thử nghiệm các thuật toán tiên tiến để tối ưu hóa độ chính xác.

- Mở rộng tập dữ liệu: Thu thập thêm dữ liệu từ nhiều nguồn khác nhau để tăng tính tổng quát.

- Tích hợp vào hệ thống giám sát y tế: Kết nối hệ thống với các thiết bị giám sát để cung cấp cảnh báo sớm và hỗ trợ bác sĩ trong việc chăm sóc bệnh nhân.

# TÀI LIỆU THAM KHẢO

| [1] | https://www.who.int/news-room/fact-sheets/detail/epilepsy |
| --- | --- |
| [2] | https://www.elcom.com.vn/12-ung-dung-may-hoc-machine-learning-hang-dau-trong-thuc-tien-1693998666 |
| [3] | https://viettelidc.com.vn/tin-tuc/machine-learning-la-gi |
| [4] | https://www.ibm.com/topics/random-forest#Random+forest+applications |
| [5] | https://viblo.asia/p/support-vector-machine-trong-hoc-may-mot-cai-nhin-don-gian-hon-XQZkxoQmewA |
| [6] | https://www.w3schools.com/python/default.asp |
| [7] | https://www.kaggle.com/code/harunshimanto/machine-learning-algorithms-for-epileptic-seizures |
| [8] | https://vjol.info.vn/index.php/TCDHDL/article/view/46662 |

# 