ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN

KHAI PHÁ DỮ LIỆU

DỰ ĐOÁN MỨC ĐỘ BỆNH TIM

Nhóm sinh viên thực hiện:

1. **Nguyễn Thành Trung**
2. **Đỗ Đoàn Minh Thắng**
3. **Nguyễn Quốc Cường**
4. **Phan Hữu Tuấn Kiệt**
5. **Trần Văn Kiệt**

Tên học phần: KHAI PHÁ DỮ LIỆU - NHÓM 2

Giáo viên hướng dẫn: TS.NGUYỄN NGỌC THỦY(1990)

Huế, 6 - 2025

Mục lục

DANH MỤC CÁC TỪ VIẾT TẮT

AI	Artificial Intelligence – Trí tuệ nhân tạo
DL	Deep Learning – Học sâu
ML	Machine Learning – Học máy
GAN	Generative Adversarial Network – Mạng đối kháng sinh
DF	Deepfake – Công nghệ giả mạo nội dung bằng AI
CNN	Convolutional Neural Network – Mạng nơ-ron tích chập
SVM	Support Vector Machine – Máy vector hỗ trợ

LỜI CẢM ƠN

Chúng em cũng xin chân thành cảm ơn *Trường Đại học Khoa học – Đại học Huế*, đặc biệt là *Khoa Công nghệ Thông tin* đã tạo điều kiện thuận lợi về môi trường học tập và cung cấp các tài liệu hữu ích phục vụ cho việc nghiên cứu, tìm hiểu và hoàn thành đề tài.

Xin cảm ơn các thành viên trong nhóm đã nỗ lực, trách nhiệm và hợp tác tốt với nhau trong suốt quá trình làm việc để có thể hoàn thiện tiểu luận đúng thời hạn, đảm bảo chất lượng nội dung và hình thức trình bày.

Mặc dù nhóm đã cố gắng hết sức, tuy nhiên không thể tránh khỏi những thiếu sót trong quá trình thực hiện. Nhóm rất mong nhận được những góp ý quý báu từ quý thầy cô để rút kinh nghiệm và hoàn thiện hơn trong các đề tài nghiên cứu sau.

LỜI MỞ ĐẦU

Sự bùng nổ của khoa học công nghệ trong những năm gần đây đã tạo ra những thay đổi mạnh mẽ không chỉ trong các lĩnh vực kinh tế, văn hóa mà còn trong nhiều ngành khoa học khác. Một trong những lĩnh vực nhận được sự tác động mạnh mẽ từ công nghệ chính là y tế. Đặc biệt, với sự phát triển vượt bậc của trí tuệ nhân tạo (AI) và các thuật toán học máy, các phương pháp dự đoán và chẩn đoán bệnh lý đã trở nên ngày càng chính xác và hiệu quả. Trong đó, dự đoán mức độ nguy cơ mắc bệnh tim là một trong những ứng dụng quan trọng và đang được nghiên cứu rộng rãi.

Bệnh tim là một trong những nguyên nhân hàng đầu gây tử vong trên toàn thế giới. Việc phát hiện và can thiệp sớm có thể cứu sống hàng triệu người mỗi năm, nhưng để làm được điều đó, các bác sĩ và chuyên gia y tế cần phải có những công cụ hỗ trợ hiệu quả trong việc chẩn đoán. Công nghệ dự đoán bệnh tim hiện nay có thể phân tích rất nhiều yếu tố, từ những thông tin cá nhân như tuổi tác, giới tính đến các chỉ số y tế quan trọng như huyết áp, mức cholesterol, thói quen ăn uống và vận động. Các mô hình học máy tiên tiến, thông qua việc xử lý và phân tích dữ liệu khổng lồ, có thể giúp dự đoán chính xác mức độ nguy cơ mắc bệnh tim của từng cá nhân, từ đó hỗ trợ bác sĩ trong việc đưa ra các quyết định điều trị phù hợp.

Việc áp dụng trí tuệ nhân tạo vào chẩn đoán bệnh tim còn có thể giúp giảm thiểu sai sót do yếu tố con người gây ra, đồng thời tiết kiệm thời gian và chi phí cho cả bệnh nhân và các cơ sở y tế. Hơn nữa, với sự xuất hiện của các thiết bị đeo thông minh và hệ thống chăm sóc sức khỏe từ xa, người bệnh có thể được theo dõi liên tục, nhận thông báo kịp thời về tình trạng sức khỏe của mình, từ đó chủ động phòng ngừa và điều trị sớm trước khi bệnh tình trở nên nghiêm trọng.

Trong bối cảnh đó, tôi đã quyết định nghiên cứu đề tài "**Dự đoán mức độ bệnh tim**" cho tiểu luận của mình. Mục tiêu của nghiên cứu là áp dụng các thuật toán học máy để phân tích các dữ liệu y tế, xây dựng một mô hình có thể dự đoán chính xác nguy cơ mắc bệnh tim dựa trên các yếu tố khác nhau. Qua đó, mô hình không chỉ giúp cảnh báo sớm mà còn cung cấp những gợi ý về phương án điều trị hiệu quả, từ đó giảm thiểu rủi ro cho bệnh nhân.

Ngoài ra, việc nghiên cứu và ứng dụng công nghệ dự đoán bệnh tim còn giúp mở ra cơ hội phát triển các mô hình y tế thông minh, góp phần cải thiện chất lượng cuộc sống của con người, đồng thời mang lại những lợi ích to lớn cho cộng đồng và xã hội. Trong tương lai, chúng ta có thể hy vọng rằng công nghệ sẽ giúp tạo ra những giải pháp mới, giúp hàng triệu người phòng ngừa và điều trị bệnh tim một cách hiệu quả hơn, từ đó bảo vệ sức khỏe cộng đồng một cách toàn diện.

CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU

1.1 Giới thiệu

Nhóm chúng em lựa chọn đề tài **Dự đoán bệnh tim bằng thuật toán học máy** vì đây là một vấn đề mang tính thời sự và có ý nghĩa thực tiễn cao trong lĩnh vực y tế. Bệnh tim mạch hiện đang là một trong những nguyên nhân gây tử vong hàng đầu trên toàn thế giới, không chỉ ở các nước phát triển mà cả ở những quốc gia đang phát triển như Việt Nam. Theo Tổ chức Y tế Thế giới (WHO), mỗi năm có hàng triệu người tử vong do các bệnh lý liên quan đến tim mạch.

Một điểm đáng chú ý là bệnh tim thường tiến triển âm thầm, không có triệu chứng rõ ràng trong giai đoạn đầu. Điều này dẫn đến việc phát hiện muộn, gây khó khăn cho quá trình điều trị và làm tăng chi phí y tế. Mặc dù y học hiện đại đã đạt được nhiều tiến bộ trong chẩn đoán và điều trị, việc phát hiện sớm vẫn là một thách thức lớn.

Trước bối cảnh đó, nhóm nhận thấy việc ứng dụng công nghệ học máy vào dự đoán nguy cơ mắc bệnh tim là một hướng đi tiềm năng. Học máy có khả năng xử lý lượng lớn dữ liệu từ các yếu tố như tuổi tác, huyết áp, cholesterol, thói quen sinh hoạt, tiền sử bệnh,... để đưa ra dự đoán chính xác. Điều này có thể hỗ trợ bác sĩ phát hiện nguy cơ từ sớm và đưa ra các biện pháp phòng ngừa kịp thời, từ đó giảm thiểu nguy cơ xảy ra các biến chứng như nhồi máu cơ tim hay đột quỵ.

Ngoài ra, đề tài này còn kết hợp giữa y học và công nghệ - hai lĩnh vực mà các thành viên trong nhóm đều quan tâm và yêu thích. Đây là cơ hội để nhóm học hỏi, rèn luyện kỹ năng nghiên cứu khoa học, làm việc nhóm và vận dụng kiến thức đã học vào thực tiễn. Hơn nữa, nghiên cứu này không chỉ mang tính học thuật mà còn góp phần nâng cao nhận thức cộng đồng về sức khỏe tim mạch, khuyến khích lối sống lành mạnh và chủ động trong việc phòng bệnh.

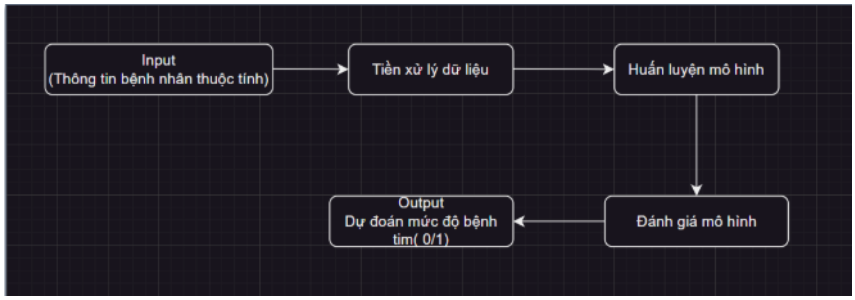
1.2 Tổng quan kiến thức về hệ thống dự đoán bệnh tim

Bệnh tim mạch là nguyên nhân tử vong hàng đầu trên toàn cầu, do đó việc xây dựng các hệ thống hỗ trợ phát hiện sớm là rất cần thiết. Một hệ thống dự đoán bệnh tim bằng học máy thường bao gồm các bước cơ bản như sau:

- **Tiền xử lý dữ liệu:** Làm sạch dữ liệu, chuẩn hóa, xử lý giá trị thiếu và mã hóa các biến phân loại.
- **Huấn luyện mô hình:** Áp dụng các thuật toán học máy như Logistic Regression, Random Forest, SVM,... trên dữ liệu đã xử lý.
- **Đánh giá mô hình:** Sử dụng các chỉ số đánh giá như Accuracy, Precision, Recall, F1-score, AUC để kiểm tra hiệu quả dự đoán.

- **Triển khai và dự đoán:** Áp dụng mô hình vào dữ liệu mới để dự đoán nguy cơ mắc bệnh tim.

Bệnh tim mạch là nguyên nhân tử vong hàng đầu, đòi hỏi công nghệ phát hiện sớm. Hệ thống dự đoán bệnh tim bằng học máy phân tích dữ liệu sức khỏe qua các bước: tiền xử lý, huấn luyện mô hình (Logistic Regression, Random Forest, SVM), đánh giá (accuracy, precision, AUC) và triển khai dự đoán. Các nghiên cứu dùng tập dữ liệu như Cleveland Heart Disease, đạt độ chính xác trên 85% với mô hình cổ điển, học sâu (ANN) hoặc ensemble (XGBoost).
Ba hướng tiếp cận gồm truyền thống, học máy cổ điển và học sâu, tùy thuộc dữ liệu và tài nguyên. Nhóm dự kiến dùng học máy cổ điển và ensemble cho bối cảnh y tế Việt Nam. Đề tài nhằm nâng cao chẩn đoán và sức khỏe cộng đồng.



Hình 1.1: Sơ đồ tổng quan hệ thống dự đoán bệnh tim

1.3 Bài toán dự đoán bệnh tim

1.3.1 Phát biểu bài toán

Bài toán đặt ra là xây dựng một mô hình học máy có khả năng phân loại mức độ nghiêm trọng của bệnh tim dựa trên các đặc trưng lâm sàng và cận lâm sàng của bệnh nhân, chẳng hạn như: tuổi, giới tính, huyết áp, cholesterol, kết quả điện tâm đồ, nhịp tim,...

Mục tiêu là phân loại bệnh nhân vào các nhóm nguy cơ (từ không mắc bệnh đến bệnh nặng), từ đó hỗ trợ bác sĩ trong việc đánh giá tình trạng bệnh và đưa ra quyết định điều trị phù hợp.

1.3.2 Dữ liệu đầu vào

Tập dữ liệu đầu vào bao gồm các đặc trưng y tế như:

- **Age** – Tuổi
- **Gender** – Giới tính
- **BloodPressure** – Huyết áp
- **Cholesterol** – Mức cholesterol
- **HeartRate** – Nhịp tim
- **QuantumPatternFeature** – Đặc trưng mô hình hóa (rút trích bằng kỹ thuật nâng cao)

1.3.3 Dữ liệu đầu ra

Mô hình sẽ dự đoán mức độ nghiêm trọng của bệnh tim, được phân chia thành 4 mức độ như sau:

- **0**: Không mắc bệnh
- **1**: Mức độ nhẹ
- **2**: Mức độ trung bình
- **3**: Mức độ nặng

# Age	# Gender	# BloodPres...	# Cholesterol	# HeartRate	# QuantumP...	# HeartDise...
68	1	105	191	107	8.362240688	1
58	0	97	249	89	9.249001528	0
44	0	93	190	82	7.94254206	1
72	1	93	183	101	6.495155116	1
37	0	145	166	103	7.653900124	1
50	1	114	271	73	8.63168402	0
68	1	156	225	73	7.559544668	1
48	0	156	236	61	9.152102641	0
52	0	116	266	114	9.146931988	0
40	1	121	255	96	9.683800175	0
40	1	139	235	64	8.732313184	0
53	1	150	176	97	8.984834605	0
65	0	140	206	104	7.545537611	1
69	1	108	180	100	6.799793664	1
53	1	110	283	113	9.190135016	0
32	1	94	247	119	9.322586014	0

Hình 1.2: Minh họa tập dữ liệu

1.4 Tiểu kết chương 1

Chương này đã trình bày tổng quan đề tài dự đoán bệnh tim, bao gồm lý do chọn đề tài, ý nghĩa và vai trò thực tiễn của việc ứng dụng học máy trong lĩnh vực y tế. Các kiến thức nền tảng về hệ thống dự đoán bệnh tim cũng đã được giới thiệu, cùng với phát biểu bài toán rõ ràng từ góc nhìn khoa học dữ liệu. Nội dung chương là nền tảng cho các chương tiếp theo về cách tiếp cận, xây dựng mô hình và đánh giá hiệu quả hệ thống.

1.5 Cấu trúc bài tiểu luận

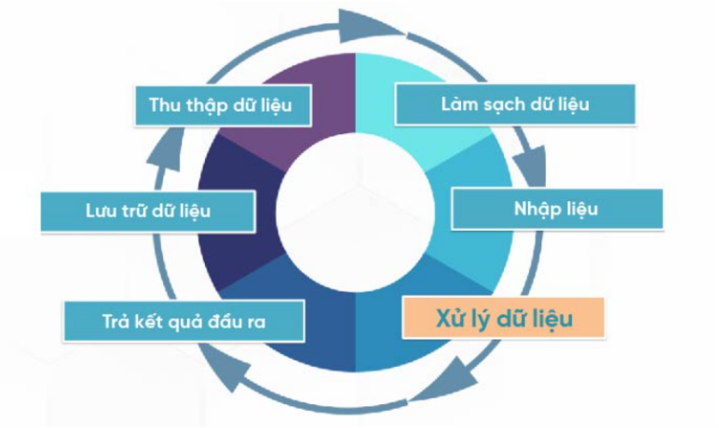
Bài tiểu luận được tổ chức thành các chương như sau:

- **Chương 1:** Tổng quan đề tài nghiên cứu
- **Chương 2:** Cách tiếp cận và phương pháp thực hiện
- **Chương 3:** Thử nghiệm và đánh giá

CHƯƠNG 2 CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP THỰC HIỆN

2.1 Tiền xử lý dữ liệu

Trong lĩnh vực học máy, dữ liệu thô ban đầu thường ở trạng thái lộn xộn, không nhất quán và chưa sẵn sàng để các thuật toán học máy có thể sử dụng trực tiếp hoặc hiệu quả.1 Quá trình này có thể được ví như việc một đầu bếp cần phải sơ chế nguyên liệu – rửa sạch, cắt thái, tẩm ướp – trước khi bắt tay vào nấu một món ăn ngon. Tương tự, các nhà khoa học dữ liệu phải ”chuẩn bị”dữ liệu thông qua một loạt các kỹ thuật được gọi chung là tiền xử lý dữ liệu. Đây là tập hợp các phương pháp nhằm biến đổi dữ liệu thô thành một định dạng sạch sẽ, giàu thông tin và phù hợp cho việc huấn luyện mô hình.3 Mặc dù thường tốn nhiều thời gian, đây lại là bước nền tảng, có ảnh hưởng quyết định đến hiệu suất của mô hình học máy.



Hình 2.1: https://www.elcom.com.vn/quy-trinh-va-cac-buoc-xu-ly-du-lieu-thong-dung-1712287428

Chất lượng của dữ liệu đầu vào có một vai trò không thể phủ nhận. Thực tế cho thấy, hiệu suất của mô hình phụ thuộc rất lớn vào chất lượng dữ liệu được sử dụng trong quá trình huấn luyện; do đó, các kỹ thuật tiền xử lý, đặc biệt là kỹ thuật đặc trưng (feature engineering), là vô cùng quan trọng.1 Dữ liệu thệu đơn thuần hiếm khi đủ để mang lại những hiểu biết sâu sắc có ý nghĩa.2 Điều này ngụ ý rằng, chất lượng của quá trình tiền xử lý có thể ảnh hưởng đến hiệu suất mô hình nhiều hơn cả việc lựa chọn thuật toán học máy. Một thuật toán phức tạp được cung cấp dữ liệu chuẩn bị kém có khả năng sẽ hoạt động kém hơn so với một thuật toán đơn giản hơn nhưng được cung cấp dữ liệu đã được chuẩn bị kỹ lưỡng. Nếu dữ liệu đầu vào nhiễu hoặc không liên quan, thuật toán, dù tiên tiến đến đâu, về cơ bản cũng chỉ học từ ”rác”(nguyên tắc ”garbage in, garbage out”). Do đó, việc đầu tư thời gian và công sức để hiểu và thực hiện tỉ mỉ các bước tiền xử lý là một hoạt động mang lại lợi ích lớn cho bất kỳ nhà khoa học dữ liệu nào. Đây không chỉ là một nhiệm vụ ”dọn dẹp”mà là một phần quan trọng của quá trình mô hình hóa, đòi hỏi kiến thức chuyên môn và sự cân nhắc cẩn thận.

Kỹ Thuật 1:Kỹ Thuật Nhãn (Label Engineering)

Trong học máy có giám sát, **kỹ thuật nhãn (Label Engineering)** là quá trình thiết kế hoặc điều chỉnh nhãn – tức "sự thật cơ bản" mà mô hình học để dự đoán, chẳng hạn mức độ bệnh tim của bệnh nhân. Nhãn này không chỉ được lấy trực tiếp từ dữ liệu thô mà có thể được xây dựng để phù hợp với bài toán cụ thể. Ví dụ, nhãn *DiseaseLevel* được thiết kế để biểu thị mức độ nghiêm trọng của bệnh tim, bao gồm các hạng mục: *Không Bệnh*, *Nguy Cơ Thấp*, *Nguy Cơ Trung Bình*, và *Nguy Cơ Cao*.

Nhóm đã tạo nhãn *DiseaseLevel* dựa trên cột *HeartDisease* (0: không bệnh, 1: có bệnh) và kết hợp các yếu tố bổ sung như tuổi, giới tính, huyết áp, và cholesterol. Quy tắc phân loại được xây dựng như sau:

- *HeartDisease = 0*: *DiseaseLevel* = "Không Bệnh".
- *HeartDisease = 1*:
  - Nếu tuổi > 60, huyết áp cao và cholesterol cao: *DiseaseLevel* = "Nguy Cơ Cao".
  - Nếu tuổi < 40, huyết áp và cholesterol bình thường: *DiseaseLevel* = "Nguy Cơ Thấp".
  - Các trường hợp còn lại: *DiseaseLevel* = "Nguy Cơ Trung Bình".

Ngoài ra, giới tính cũng được xem xét vì nguy cơ bệnh tim khác nhau giữa nam và nữ (nam giới thường có nguy cơ cao hơn).

Nhãn *DiseaseLevel* chi tiết hơn so với nhãn nhị phân *HeartDisease*, giúp bác sĩ đưa ra các can thiệp phù hợp, như khuyến nghị thay đổi lối sống hoặc điều trị tích cực. Nhãn này cũng cho phép mô hình học máy nhận diện các mẫu phức tạp hơn trong dữ liệu. Tuy nhiên, việc thiết kế nhãn đòi hỏi kiến thức y khoa chuyên sâu và sự hợp tác chặt chẽ với các bác sĩ tim mạch để đảm bảo tính chính xác, đồng thời tránh thiên vị từ các quy tắc đơn giản hóa hoặc dữ liệu lịch sử không đầy đủ.

Nhóm hy vọng nhãn *DiseaseLevel* sẽ cải thiện độ chính xác của mô hình dự đoán, từ đó đóng góp tích cực vào việc nâng cao chất lượng chăm sóc sức khỏe cộng đồng.

Kỹ Thuật 2: Kỹ Thuật Đặc Trưng (Feature Engineering)

Kỹ thuật đặc trưng là gì

Kỹ thuật đặc trưng (*Feature Engineering*) là quá trình biến đổi dữ liệu thô thành các đặc trưng giúp mô hình học máy dự đoán tốt hơn, bằng cách sử dụng kiến thức chuyên môn và phân tích dữ liệu. Nhóm nhận thấy đây là bước quan trọng vì đặc trưng tốt giúp thuật toán dễ dàng nhận ra các mẫu trong dữ liệu, từ đó nâng cao độ chính xác của mô hình. Việc tạo đặc trưng mới, như kết hợp các biến, là cách chính để làm dữ liệu trở nên rõ ràng hơn.

Tạo đặc trưng tương tác

Đặc trưng tương tác được tạo bằng cách kết hợp hai hoặc nhiều đặc trưng — thường qua phép nhân — để nắm bắt các hiệu ứng hiệp đồng. Ví dụ, huyết áp cao kết hợp cholesterol cao có thể nguy hiểm hơn

từng yếu tố riêng lẻ. Nhóm lựa chọn tạo đặc trưng tương tác dựa trên hiểu biết y tế để phản ánh mối quan hệ phức tạp trong bệnh tim.

Các đặc trưng cho dự đoán bệnh tim

Nhóm đề xuất hai đặc trưng tương tác chính:

- **BP_Cholesterol (Huyết Áp × Cholesterol)**: Đặc trưng này nắm bắt nguy cơ cao hơn khi cả huyết áp và cholesterol đều cao. Ví dụ, bệnh nhân có cả hai yếu tố ở mức trung bình có thể có nguy cơ cao hơn người chỉ có huyết áp cao. Đặc trưng này giúp mô hình nhận ra hiệu ứng kết hợp.
- **Age_BP (Tuổi × Huyết Áp)**: Huyết áp cao ở người lớn tuổi (trên 65 tuổi) thường nguy hiểm hơn ở người trẻ. Đặc trưng này giúp mô hình học mối quan hệ phụ thuộc vào tuổi tác, vì tuổi là yếu tố nguy cơ chính của bệnh tim.

Việc tạo đặc trưng dựa trên giả thuyết y tế, nhưng nhóm cẩn thận lựa chọn các tương tác hợp lý để tránh việc nhân ngẫu nhiên gây phức tạp hoặc quá khớp (*overfitting*). Điều này đòi hỏi sự phân tích dữ liệu kết hợp kiến thức chuyên ngành về bệnh tim.

Tại sao đặc trưng tương tác quan trọng?

Các đặc trưng tương tác giúp mô hình nắm bắt các mối quan hệ phi tuyến — vốn phổ biến trong bệnh tim do nhiều yếu tố tác động cùng lúc. Chúng làm mô hình chính xác hơn, cung cấp thông tin ngữ cảnh phong phú và có thể tiết lộ những hiểu biết mới, như xác định các yếu tố nguy cơ kết hợp đặc biệt. Tuy nhiên, nhóm nhận thấy cần cân bằng giữa việc tạo đặc trưng và tránh tăng số chiều, do đó sẽ kiểm tra và tinh chỉnh đặc trưng dựa trên hiệu suất của mô hình.

Tóm lại, kỹ thuật đặc trưng là một bước then chốt, kết hợp giữa khoa học và sáng tạo. Nhóm hy vọng các đặc trưng như `BP_Cholesterol` và `Age_BP` sẽ cải thiện dự đoán bệnh tim, hỗ trợ bác sĩ và nâng cao chất lượng chăm sóc sức khỏe.

Kỹ Thuật 3: Lựa Chọn Đặc Trưng (feature selection)

- Tại sao cần lựa chọn đặc trưng? Lựa chọn đặc trưng là quá trình chọn tập hợp con các đặc trưng phù hợp nhất từ dữ liệu, loại bỏ đặc trưng không liên quan hoặc dư thừa. Nhóm nhận thấy không phải đặc trưng nào cũng hữu ích cho dự đoán bệnh tim, một số có thể gây nhiễu hoặc tương quan cao, làm phức tạp mô hình. Lựa chọn đặc trưng giúp:
  + Tạo mô hình đơn giản, dễ hiểu.
  + Rút ngắn thời gian huấn luyện.
  + Giảm nguy cơ quá khớp, tăng khả năng dự đoán trên dữ liệu mới.

- Giới thiệu SelectKBest SelectKBest là một công cụ trong thư viện `Scikit-learn`, cho phép tự động chọn ra k đặc trưng tốt nhất dựa trên các phép kiểm định thống kê. Trong bài toán này, nhóm sử

dụng $k = 7$ như yêu cầu đề bài. SelectKBest được ví như "người chọn nguyên liệu", giúp tìm ra những đặc trưng quan trọng nhất cho nhãn `DiseaseLevel`.

- Cơ chế tính điểm: `f_classif` và ANOVA F-test SelectKBest sử dụng hàm `f_classif`, dựa trên kiểm định ANOVA F-test, để chấm điểm các đặc trưng trong bài toán phân loại. ANOVA kiểm tra sự khác biệt trung bình của một đặc trưng (chẳng hạn `BP_Cholesterol`) giữa các mức của biến mục tiêu `DiseaseLevel` (Không Bệnh, Thấp, Trung Bình, Cao).

  Một đặc trưng có F-score cao thể hiện khả năng phân tách tốt giữa các nhóm vì giá trị trung bình khác biệt rõ rệt. Ngược lại, F-score thấp cho thấy đặc trưng đó không đóng góp nhiều cho việc phân biệt các mức bệnh. Tuy nhiên, nhóm cũng nhận thức rằng `f_classif` chỉ đánh giá từng đặc trưng độc lập và không xem xét tương tác giữa các đặc trưng.

- Hạn chế và lưu ý SelectKBest với `f_classif` có thể bỏ sót đặc trưng yếu riêng lẻ nhưng mạnh khi kết hợp, hoặc chọn đặc trưng dư thừa nếu chúng tương quan cao với mục tiêu. Ví dụ, nếu BP và Cholesterol đều liên quan đến DiseaseLevel `f_classif` có thể chọn cả hai dù một cái là đủ. Ngoài ra, số k=7 là cố định trong bài, nhưng thực tế, nhóm cần thử nghiệm qua kiểm định chéo để tìm k tối ưu, tránh thiếu hoặc dư đặc trưng, gây thiếu khớp hoặc quá khớp. ANOVA cũng giả định dữ liệu chuẩn, nên nhóm cần kiểm tra kỹ.

- Tầm quan trọng Lựa chọn đặc trưng là bước quan trọng giúp cải thiện hiệu suất mô hình, đặc biệt với dữ liệu bệnh tim có tính chất phức tạp. Nhóm kỳ vọng rằng với SelectKBest, các đặc trưng như `BP_Cholesterol`, `Age_BP`, … sẽ được lựa chọn một cách hợp lý. Kết hợp với kiểm tra thực nghiệm, mô hình có thể đưa ra dự đoán chính xác hơn, hỗ trợ bác sĩ trong việc chẩn đoán và điều trị bệnh tim. Đây là bước cân bằng giữa việc đơn giản hóa mô hình và giữ lại thông tin quan trọng, đòi hỏi sự phối hợp giữa kỹ thuật và hiểu biết y tế.

Kỹ Thuật 4:Chuẩn Hóa(Standardization)

- Chuẩn hóa dữ liệu bằng StandardScaler

  Chuẩn hóa (Standardization) là một kỹ thuật tiền xử lý dữ liệu, đưa các đặc trưng về cùng một thang đo với trung bình bằng 0 và độ lệch chuẩn bằng 1. Kỹ thuật này được thực hiện dễ dàng bằng công cụ `StandardScaler` trong thư viện `Scikit-learn`. Công thức chuẩn hóa được định nghĩa như sau:

$$\text{standardized\_value} = \frac{\text{value} - \mu}{\sigma} \tag{2.1}$$

  Trong đó, μ là giá trị trung bình và σ là độ lệch chuẩn của đặc trưng.

- Tại sao cần chuẩn hóa?

  - **Độ nhạy của thuật toán:** Các thuật toán như K-Nearest Neighbors (KNN), Support Vector Machine (SVM), hoặc hồi quy logistic phụ thuộc vào khoảng cách hoặc quá trình gradient descent, do đó dễ bị ảnh hưởng nếu các đặc trưng có thang đo khác nhau. Ví dụ, đặc trưng *cholesterol* (100–500) có thể chi phối *tuổi* (20–80) nếu không chuẩn hóa.

- **Ngăn áp đảo:** Chuẩn hóa đặt các đặc trưng về cùng một thang đo, đảm bảo không đặc trưng nào chi phối mô hình chỉ vì giá trị số lớn hơn.
- **Tăng tốc hội tụ:** Giúp quá trình gradient descent hội tụ nhanh hơn, từ đó cải thiện hiệu suất huấn luyện mô hình.

Chuẩn hóa đặc biệt hữu ích khi sử dụng các thuật toán như PCA, KNN, SVM. Tuy nhiên, với các mô hình như cây quyết định hoặc rừng ngẫu nhiên, chuẩn hóa thường không cần thiết vì các mô hình này không phụ thuộc vào khoảng cách giữa các điểm dữ liệu. Nếu dữ liệu có độ lệch lớn, nhóm cần cân nhắc áp dụng biến đổi log trước khi chuẩn hóa, vì StandardScaler giả định dữ liệu phân phối gần chuẩn.

- Áp dụng StandardScaler trong bài toán dự đoán bệnh tim

  Sau khi lựa chọn 7 đặc trưng đầu vào (ví dụ: `BP_Cholesterol`, `Age_BP`), nhóm áp dụng quy trình chuẩn hóa như sau:

  1. Tính trung bình μ và độ lệch chuẩn σ từ tập dữ liệu huấn luyện.
  2. Biến đổi dữ liệu huấn luyện và kiểm tra sử dụng các giá trị μ và σ đã tính.

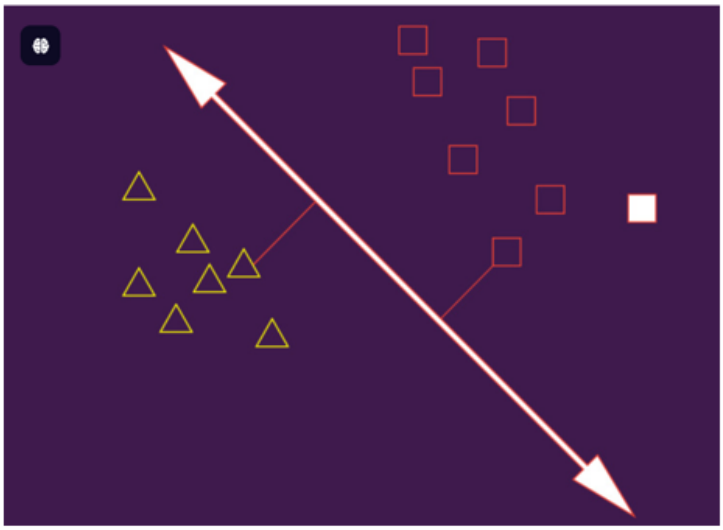
  Lưu ý quan trọng là chỉ được `fit StandardScaler` trên tập huấn luyện nhằm tránh rò rỉ thông tin từ tập kiểm tra, đảm bảo đánh giá mô hình một cách trung thực. Bước chuẩn hóa giúp các đặc trưng như huyết áp hay cholesterol có cùng thang đo, hỗ trợ mô hình dự đoán chính xác hơn mức độ bệnh tim (`DiseaseLevel`).

- Tầm quan trọng Chuẩn hóa là một công cụ giúp cân bằng dữ liệu, đảm bảo mô hình học một cách công bằng và hiệu quả. Nhóm hy vọng rằng việc sử dụng `StandardScaler` sẽ giúp cải thiện hiệu suất dự đoán bệnh tim, đặc biệt là với các thuật toán nhạy cảm như SVM, đồng thời đảm bảo tính nhất quán khi xử lý dữ liệu mới.

2.2 Mục tiêu và Nguyên lý hoạt động của SVM

Máy Vector Hỗ trợ (SVM) là một trong những thuật toán học máy có giám sát mạnh mẽ và linh hoạt nhất, được sử dụng rộng rãi cho cả bài toán phân loại và hồi quy, mặc dù nó nổi tiếng hơn với các ứng dụng phân loại.

- **Mục tiêu chính:** Mục tiêu cốt lõi của SVM là tìm ra một siêu phẳng (hyperplane) trong không gian đặc trưng N chiều để phân tách các điểm dữ liệu thuộc các lớp khác nhau một cách tối ưu. Điều làm cho SVM trở nên đặc biệt là nó không chỉ tìm một siêu phẳng bất kỳ có thể phân chia dữ liệu, mà nó tìm kiếm một siêu phẳng "tốt nhất"– được định nghĩa là siêu phẳng có khoảng cách (margin - hay còn gọi là "lề") lớn nhất đến các điểm dữ liệu gần nhất của mỗi lớp.



Hình 2.2: https://www.iostream.co/article/gioi-thieu-ve-mo-hinh-svm-D15jcg/

Nguyên lý cốt lõi của SVM

- **Siêu phẳng phân cách (Separating Hyperplane):** Trong một không gian N chiều, một siêu phẳng là một không gian con có số chiều $N-1$. Ví dụ, trong không gian 2 chiều, siêu phẳng là một đường thẳng; trong không gian 3 chiều, nó là một mặt phẳng. Siêu phẳng này chia không gian đặc trưng thành hai nửa riêng biệt, mỗi nửa tương ứng với một lớp dữ liệu (trong bài toán phân loại nhị phân).

- **Lề (Margin):** Lề là khoảng không gian nằm giữa hai siêu phẳng song song với siêu phẳng phân cách chính, và hai siêu phẳng này đi qua các điểm dữ liệu gần nhất của mỗi lớp. SVM cố gắng tối đa hóa độ rộng của lề này. Ý tưởng là một lề rộng hơn sẽ giúp mô hình có khả năng tổng quát hóa tốt hơn trên dữ liệu mới chưa từng thấy, đồng thời giảm nguy cơ bị quá khớp (overfitting).

- **Vector hỗ trợ (Support Vectors):** Đây là các điểm dữ liệu nằm chính xác trên các siêu phẳng lề (tức là những điểm gần nhất với siêu phẳng phân cách chính). Các điểm này được gọi là "vector hỗ trợ"bởi vì chúng "hỗ trợ"hoặc quyết định hoàn toàn vị trí và hướng của siêu phẳng phân cách tối ưu. Nếu các vector hỗ trợ này di chuyển, siêu phẳng phân cách cũng sẽ thay đổi. Các điểm dữ liệu khác, nằm xa hơn lề, không có ảnh hưởng trực tiếp đến việc xác định siêu phẳng.

Mục tiêu tối đa hóa lề của SVM giúp SVM đang ngầm tìm kiếm một mô hình có độ phức tạp thấp nhất trong số tất cả các mô hình có thể phân tách được dữ liệu huấn luyện, điều này theo lý thuyết sẽ dẫn đến một giới hạn trên chặt chẽ hơn cho lỗi tổng quát hóa.

Một đặc điểm quan trọng khác của SVM xuất phát từ việc chỉ các vector hỗ trợ mới quyết định siêu phẳng là tính thưa của nghiệm (sparsity). Vector trọng số $\mathbf{w}$ xác định siêu phẳng (sẽ được trình bày chi tiết hơn ở phần sau) là một tổ hợp tuyến tính của các vector hỗ trợ. Điều này có nghĩa là nhiều nhân tử Lagrange α_i (liên quan đến các điểm dữ liệu không phải là vector hỗ trợ) sẽ bằng không. Do đó, nghiệm của SVM thường là "thưa"theo nghĩa là nó chỉ phụ thuộc vào một tập con nhỏ các điểm dữ liệu huấn luyện (chính là các vector hỗ trợ). Tính chất này mang lại lợi ích về mặt hiệu quả tính toán và sử dụng bộ nhớ, đặc biệt khi dự đoán cho các điểm dữ liệu mới, vì chỉ cần tính toán dựa trên các vector hỗ trợ.

2.3 Trường hợp Dữ liệu Phân tách Tuyến tính (Linearly Separable Case - Hard Margin SVM)

Khi dữ liệu huấn luyện có thể được phân tách hoàn toàn bằng một siêu phẳng mà không có điểm nào bị phân loại sai, chúng ta có trường hợp phân tách tuyến tính. SVM trong trường hợp này được gọi là **Hard Margin SVM**.

Phương trình Siêu phẳng Một siêu phẳng trong không gian d chiều có thể được biểu diễn bằng phương trình:

$$w^T x + b = 0$$

Trong đó:

- $w = (w_1, w_2, \ldots, w_d)^T$ là vector trọng số (weight vector), còn được gọi là vector pháp tuyến (normal vector) của siêu phẳng, vì nó vuông góc với siêu phẳng.
- $x = (x_1, x_2, \ldots, x_d)^T$ là vector đặc trưng của một điểm dữ liệu.
- b là một đại lượng vô hướng, được gọi là độ lệch (bias) hoặc hệ số chặn (intercept). Nó xác định vị trí của siêu phẳng so với gốc tọa độ.

Định nghĩa Lề (Margin) Đối với một bài toán phân loại nhị phân, giả sử nhãn lớp là $y_i \in \{-1, +1\}$. Chúng ta muốn tìm siêu phẳng (w, b) sao cho tất cả các điểm dữ liệu thuộc lớp +1 nằm ở một phía của siêu phẳng (ví dụ: $w^T x_i + b \geq +1$) và tất cả các điểm dữ liệu thuộc lớp -1 nằm ở phía còn lại (ví dụ: $w^T x_i + b \leq -1$). Hai điều kiện này có thể được gộp lại thành một ràng buộc duy nhất:

$$y_i(w^T x_i + b) \geq 1, \forall i = 1, \ldots, N$$

Trong đó N là tổng số điểm dữ liệu.

Các điểm dữ liệu mà tại đó $y_i(w^T x_i + b) = 1$ được gọi là các **vector hỗ trợ** (support vectors). Chúng nằm trên hai siêu phẳng song song với siêu phẳng phân cách chính $w^T x + b = 0$, đó là $w^T x + b = +1$ và $w^T x + b = -1$. Khoảng cách từ siêu phẳng phân cách chính đến mỗi siêu phẳng lề này là $\frac{1}{\|w\|}$. Do đó, tổng độ rộng của lề (margin) giữa hai lớp là $\frac{2}{\|w\|}$.

Bài toán Tối ưu (Primal Problem) Để tối đa hóa lề $\frac{2}{\|w\|}$, chúng ta cần tối thiểu hóa $\|w\|$. Để thuận tiện cho việc giải bài toán tối ưu (đặc biệt là khi lấy đạo hàm và đảm bảo tính lồi của hàm mục tiêu), người ta thường tối thiểu hóa $\frac{1}{2}\|w\|^2$ (bình phương chuẩn Euclidean của w).

Vậy, bài toán tối ưu hóa cho Hard Margin SVM (dạng nguyên bản - primal problem) được phát biểu như sau:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1, \forall i = 1, \ldots, N$$

Đây là một bài toán quy hoạch toàn phương (Quadratic Programming - QP) lồi, vì hàm mục tiêu là lồi (do $\|w\|^2$ là lồi) và các ràng buộc là tuyến tính.

Điều kiện Karush-Kuhn-Tucker (KKT) Để giải bài toán QP có ràng buộc bất đẳng thức này, người ta thường sử dụng phương pháp nhân tử Lagrange và các điều kiện KKT. Lagrangian của bài toán được

xây dựng như sau:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(w^T x_i + b) - 1\right]$$

Trong đó, $\alpha_i \geq 0$ là các nhân tử Lagrange tương ứng với mỗi ràng buộc $y_i(w^T x_i + b) \geq 1$.

Các điều kiện KKT cần và đủ cho nghiệm tối ưu (w^*, b^*, α^*) bao gồm:

- Điều kiện không âm: $\alpha_i \geq 0$.
- Điều kiện tối ưu của Lagrangian: $\frac{\partial L}{\partial w} = 0$, $\frac{\partial L}{\partial b} = 0$.
- Điều kiện ràng buộc: $\alpha_i \left[y_i(w^T x_i + b) - 1\right] = 0$, tức là α_i chỉ khác không khi $y_i(w^T x_i + b) = 1$, tức các vector hỗ trợ.

Điều kiện tối ưu trong SVM

Bài toán tối ưu của SVM tuân theo các điều kiện sau:

1. **Điều kiện dừng (Stationarity)**: Đạo hàm của hàm mục tiêu L theo các tham số w và b phải bằng 0 tại điểm tối ưu:

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{N} \alpha_i^* y_i x_i = 0 \quad \Rightarrow \quad w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i^* y_i = 0$$

2. **Tính khả thi của ràng buộc nguyên bản (Primal feasibility)**: Điều kiện này yêu cầu:

$$y_i(w^T x_i + b^*) \geq 1, \quad \forall i$$

3. **Tính khả thi của ràng buộc đối ngẫu (Dual feasibility)**: Điều kiện này yêu cầu:

$$\alpha_i^* \geq 0, \quad \forall i$$

4. **Điều kiện bù (Complementary slackness)**: Điều kiện này yêu cầu:

$$\alpha_i^* = 0, \quad \forall i$$

Vai trò của vector hỗ trợ

Điều kiện bù (complementary slackness) là rất quan trọng và nói rằng:

- Nếu $\alpha_i^* > 0$, thì yêu cầu:

$$y_i(w^T x_i + b^*) - 1 = 0, \quad \text{hay} \quad y_i(w^T x_i + b^*) = 1$$

Điều này có nghĩa là điểm dữ liệu x_i nằm chính xác trên một trong hai siêu phẳng lề, và những điểm này chính là các *vector hỗ trợ*.

- Nếu $y_i(w^T x_i + b^*) - 1 > 0$ (tức là $y_i(w^T x_i + b^*) > 1$), thì $\alpha_i^* = 0$. Điều này có nghĩa là các điểm dữ liệu nằm đúng phía và cách xa siêu phẳng lề, và chúng không có vai trò gì trong việc xác định siêu phẳng tối ưu.

Bài toán tối ưu của Hard Margin SVM là một bài toán quy hoạch toàn phương lồi với các ràng buộc tuyến tính, và nó có một nghiệm tối ưu toàn cục duy nhất (nếu tồn tại một siêu phẳng phân tách). Tính duy nhất này đảm bảo rằng kết quả huấn luyện SVM không phụ thuộc vào điểm khởi tạo của thuật toán tối ưu, điều này khác với một số thuật toán khác như mạng nơ-ron có thể bị kẹt ở các cực tiểu địa phương. Tính lồi của hàm mục tiêu và miền ràng buộc (được tạo bởi các bất đẳng thức tuyến tính) là chìa khóa cho tính chất này.

Từ điều kiện KKT:

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

chúng ta thấy rằng vector pháp tuyến w^* của siêu phẳng tối ưu là một tổ hợp tuyến tính của các vector đặc trưng x_i của các điểm dữ liệu, trong đó các hệ số α_i^* chỉ khác không đối với các vector hỗ trợ. Điều này có nghĩa là vector w^* nằm trong không gian con được sinh bởi các vector hỗ trợ. Về mặt hình học, điều này cho thấy hướng của siêu phẳng phân cách được quyết định hoàn toàn bởi những điểm dữ liệu "khó phân loại nhất"– những điểm nằm sát lề. Điều này lý giải tại sao các điểm dữ liệu khác, nằm xa lề, lại không đóng vai trò trong việc xác định siêu phẳng.

2.4 Mục tiêu và Nguyên lý hoạt động của LDA

2.4.1 Mục tiêu chính

Linear Discriminant Analysis (LDA) là một kỹ thuật học máy có giám sát, được sử dụng để giảm chiều dữ liệu và phân loại. Mục tiêu cốt lõi của LDA là tìm một không gian con (*subspace*) có số chiều thấp hơn không gian ban đầu, sao cho các lớp dữ liệu được phân tách tối ưu. Cụ thể, LDA xác định các trục chiếu (*directions*) sao cho:

- **Tối đa hóa sự phân tách giữa các lớp:** Đẩy các trung bình (*centroids*) của các lớp cách xa nhau nhất trong không gian chiếu.
- **Tối thiểu hóa sự phân tán trong mỗi lớp:** Làm cho các điểm dữ liệu trong cùng một lớp co cụm chặt chẽ quanh trung bình của lớp đó.

2.4.2 Nguyên lý cốt lõi

LDA hoạt động dựa trên việc tối đa hóa phương sai giữa các lớp (*between-class variance*) và tối thiểu hóa phương sai trong mỗi lớp (*within-class variance*). Điều này đảm bảo các lớp tách biệt rõ ràng và các điểm trong cùng lớp tập trung gần nhau.

- **Khác với PCA:** PCA là phương pháp giảm chiều không có giám sát, tập trung vào phương sai tổng thể của dữ liệu mà không xét đến thông tin nhãn lớp. Trong khi đó, LDA là phương pháp có giám sát, sử dụng thông tin nhãn lớp để tìm các phép chiếu tối ưu, gọi là các thành phần phân biệt tuyến tính (*linear discriminants*).

- **Thành phần phân biệt tuyến tính:** Đây là các tổ hợp tuyến tính của các đặc trưng ban đầu, được thiết kế để tối ưu hóa khả năng phân lớp giữa các nhãn.

- **Liên kết với định lý Bayes:** Khi dữ liệu trong mỗi lớp tuân theo phân phối Gaussian đa biến và có chung ma trận hiệp phương sai, LDA tìm ra ranh giới quyết định tuyến tính tối ưu theo định lý Bayes, từ đó giúp phân tách các lớp một cách hiệu quả.

2.4.3 Cơ sở toán của LDA

LDA dựa trên hai ma trận chính để đạt mục tiêu phân tách lớp:

Ma trận tán xạ nội lớp (Within-class Scatter Matrix - S_W): Đo lường sự phân tán của các điểm dữ liệu trong mỗi lớp, với mục tiêu làm cho S_W sau khi chiếu càng nhỏ càng tốt.

$$S_W = \sum_{j=1}^{c} S_j$$

trong đó, S_j là ma trận tán xạ của lớp C_j:

$$S_j = \sum_{x_i \in C_j} (x_i - m_j)(x_i - m_j)^T$$

với x_i: vector đặc trưng của mẫu trong lớp C_j, m_j: vector trung bình của lớp C_j, c: số lớp. S_W nhỏ khi các điểm trong mỗi lớp có cụm chặt chẽ quanh trung bình.

Ma trận tán xạ giữa các lớp (Between-class Scatter Matrix - S_B): Đo lường sự phân tán của các trung bình lớp so với trung bình toàn cục, với mục tiêu làm cho S_B sau khi chiếu càng lớn càng tốt.

$$S_B = \sum_{j=1}^{c} N_j (m_j - m)(m_j - m)^T$$

với N_j: số mẫu trong lớp C_j, m_j: trung bình lớp C_j, m: trung bình toàn cục. S_B lớn khi các trung bình lớp cách xa nhau.

Hàm mục tiêu J(W) của Fisher: LDA tìm ma trận chiếu W để tối đa hóa tỷ lệ giữa phương sai giữa các lớp và phương sai trong mỗi lớp:

$$J(W) = \frac{W^T S_B W}{W^T S_W W}$$

Trong trường hợp chiếu xuống một chiều (vector chiếu w):

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Mục tiêu là tìm W hoặc w sao cho J đạt giá trị lớn nhất.

Tìm ma trận chiếu tối ưu W: Bài toán được chuyển thành bài toán trị riêng tổng quát:

$$S_B w = \lambda S_W w$$

Nếu S_W khả nghịch, bài toán tương đương với:

$$(S_W^{-1} S_B)w = \lambda w$$

Các vector riêng $w_1, w_2, \ldots, w_k$ ứng với k trị riêng lớn nhất ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$) tạo thành các cột của ma trận chiếu

$$W = [w_1, w_2, \ldots, w_k].$$

- **Hạn chế:** S_W có thể không khả nghịch trong trường hợp số đặc trưng lớn hơn số mẫu hoặc có đa cộng tuyến. Để khắc phục, sử dụng kỹ thuật điều chuẩn (regularization) hoặc áp dụng PCA trước LDA.
- **Số chiều tối đa:** Số chiều của không gian chiếu mới là $k \leq c - 1$, do hạng của S_B tối đa là $c - 1$.

2.4.4 LDA trong Giảm chiều Dữ liệu

Ứng dụng chính: LDA được sử dụng để giảm chiều dữ liệu, chuẩn bị cho các thuật toán phân loại.

- Giảm số chiều từ p (số đặc trưng ban đầu) xuống $k \leq c - 1$, với c là số lớp.
- Các chiều mới (thành phần phân biệt tuyến tính) được sắp xếp theo trị riêng giảm dần, trong đó các chiều đầu tiên giữ nhiều thông tin phân biệt nhất.
- **Ưu điểm so với PCA**: LDA sử dụng thông tin nhãn lớp, tối ưu hóa sự phân tách lớp, do đó phù hợp hơn cho các nhiệm vụ phân loại.

Hạn chế:

- LDA giả định rằng các lớp tuân theo phân phối Gaussian và có ma trận hiệp phương sai chung. Nếu giả định này không đúng, hiệu quả của LDA sẽ giảm.
- LDA không hiệu quả nếu các lớp có cùng trung bình nhưng khác nhau về phương sai, hoặc có cấu trúc phi tuyến tính (ví dụ, dữ liệu hình chữ "X" lồng nhau). Trong trường hợp này, PCA hoặc các phương pháp phi tuyến tính có thể phù hợp hơn.
- **Lựa chọn số chiều** k: Cần đánh giá số chiều giữ lại (dựa trên tỷ lệ phương sai giữa các lớp hoặc sử dụng kiểm tra chéo) để tránh mất thông tin quan trọng hoặc giữ lại nhiễu không cần thiết.

Lưu ý: Việc chọn $k < c - 1$ là hợp lệ nhưng có thể làm mất thông tin phân biệt, cần cân nhắc kỹ để đảm bảo hiệu suất phân loại tối ưu.

2.5 Sử dụng kỹ thuật blending kết hợp 2 mô hình catboost lightgbm để dự đoán mức độ bệnh tim

Kỹ thuật Stacking trong Học máy

Khái niệm: Stacking (Stacked Generalization – tổng quát hóa xếp chồng) là một kỹ thuật học máy thuộc nhóm Ensemble Learning, trong đó nhiều mô hình con (base models) được huấn luyện song song, và kết

quả dự đoán của chúng được sử dụng làm đặc trưng đầu vào cho một mô hình tổng hợp (meta model) nhằm đưa ra dự đoán cuối cùng.

Điểm nổi bật của Stacking so với Blending là:

• Stacking sử dụng Cross-validation (ví dụ K-fold CV) để giảm overfitting và đảm bảo rằng meta model được huấn luyện trên dự đoán hợp lệ (out-of-fold predictions), chứ không phải trên dự đoán của các mô hình đã học trên toàn bộ dữ liệu.

+ Cách hoạt động

1. Chia dữ liệu huấn luyện thành K phần (cross-validation, ví dụ: $K = 5$).
2. Với mỗi mô hình con (A, B, …), lặp qua từng fold:
   - Huấn luyện mô hình đó trên $K - 1$ phần.
   - Dự đoán trên phần còn lại (fold chưa dùng).
   - Tích lũy các dự đoán trên toàn bộ K fold để tạo **Out-of-Fold predictions** cho mô hình meta.
3. Sau khi hoàn tất, ta có một tập dữ liệu mới, với các đặc trưng là các dự đoán của mô hình con → được dùng để huấn luyện mô hình meta.
4. Khi dự đoán trên dữ liệu mới (tập test):
   - Các mô hình con được huấn luyện lại trên toàn bộ tập train.
   - Dự đoán của chúng sẽ là input cho mô hình meta để đưa ra dự đoán cuối cùng.

+Mục tiêu của Stacking

- Tận dụng điểm mạnh của từng mô hình con: mỗi mô hình học một khía cạnh khác nhau của dữ liệu.
- Tăng độ chính xác, ổn định và khả năng khái quát.
- Giảm rủi ro *overfitting* nhờ *cross-validation*.
- *Stacking* thường giúp cải thiện kết quả đáng kể so với từng mô hình đơn lẻ và thậm chí là *Blending*, nhất là khi số lượng dữ liệu không quá nhỏ.

2.5.1 Thuật toán CatBoost

+ Giải thích tổng quan: CatBoost (viết tắt của *Categorical Boosting*) là một thư viện học máy mã nguồn mở được phát triển bởi Yandex, thuộc nhóm các thuật toán tăng cường độ dốc (Gradient Boosting) trên cây quyết định. Điểm nổi bật của CatBoost là khả năng xử lý hiệu quả các biến phân loại (categorical features) mà không cần đến các phương pháp mã hóa phức tạp như *one-hot encoding* hoặc *label encoding*.

Thay vào đó, CatBoost sử dụng một kỹ thuật mã hóa đặc biệt gọi là *ordered target statistics* hoặc *permutation-driven encoding*, giúp tránh hiện tượng *target leakage* và nâng cao hiệu suất mô hình. Thuật toán cũng được tối ưu hóa để mang lại hiệu suất cao, độ chính xác tốt và giảm thiểu nhu cầu tinh chỉnh thủ công các siêu tham số.

CatBoost hoạt động hiệu quả trong cả các bài toán phân loại nhị phân, phân loại đa lớp và hồi quy. Nhờ khả năng xử lý trực tiếp các đặc trưng phân loại và cơ chế giảm thiểu *overfitting*, CatBoost thường cho kết quả tốt ngay cả khi áp dụng trên dữ liệu phức tạp và không cần tiền xử lý quá nhiều.

+ **Cơ chế hoạt động của CatBoost:** Về cơ bản, CatBoost hoạt động tương tự như các thuật toán tăng cường độ dốc (Gradient Boosting) truyền thống: nó xây dựng một chuỗi các cây quyết định để giảm dần sai số dự đoán của mô hình, thông qua việc tối thiểu hóa độ dốc của hàm mất mát.

Tuy nhiên, CatBoost có một số cải tiến quan trọng giúp cải thiện hiệu suất và hạn chế hiện tượng quá khớp (*overfitting*), đặc biệt là khi xử lý các biến phân loại:

- **Tăng cường theo thứ tự (Ordered Boosting)**: Để tránh hiện tượng rò rỉ thông tin (*target leakage*), CatBoost sử dụng một chiến lược đặc biệt gọi là *ordered boosting*. Trong đó, mỗi điểm dữ liệu chỉ được học từ những mẫu trước nó trong một hoán vị ngẫu nhiên, nhằm mô phỏng quá trình huấn luyện thực tế và giữ cho mô hình tổng quát hơn.

- **Mã hóa mục tiêu theo thứ tự (Ordered Target Encoding)**: Khi xử lý các biến phân loại, CatBoost không sử dụng *one-hot* hay *label encoding*, mà áp dụng mã hóa thống kê có thứ tự. Cụ thể, giá trị của một đặc trưng phân loại được thay thế bằng giá trị thống kê của nhãn (ví dụ: trung bình) dựa trên các mẫu xuất hiện trước đó trong tập huấn luyện, giúp tránh rò rỉ thông tin và cải thiện độ chính xác.

- **Cây đối xứng (Symmetric Trees)**: CatBoost xây dựng các cây đối xứng, trong đó tất cả các nút ở cùng một mức sử dụng cùng một điều kiện chia tách. Điều này giúp giảm độ phức tạp mô hình, tăng tốc độ dự đoán và ổn định hóa việc học.

Những kỹ thuật trên giúp CatBoost đạt được hiệu quả tốt cả về độ chính xác và thời gian huấn luyện, đồng thời giảm nhu cầu tiền xử lý dữ liệu và điều chỉnh siêu tham số.

+ **Ưu điểm chính của CatBoost** CatBoost mang lại nhiều lợi thế so với các thuật toán tăng cường độ dốc khác như XGBoost hay LightGBM, đặc biệt trong việc xử lý dữ liệu thực tế với nhiều biến phân loại. Các ưu điểm chính bao gồm:

- **Xử lý trực tiếp dữ liệu phân loại:** CatBoost nổi bật với khả năng xử lý trực tiếp các đặc trưng phân loại mà không cần mã hóa *one-hot* hoặc *label encoding* thủ công. Điều này giúp tiết kiệm thời gian xử lý và hạn chế mất thông tin từ biến phân loại.

- **Độ ổn định và khả năng giảm quá khớp:** Nhờ cây đối xứng và kỹ thuật *ordered boosting*, CatBoost kiểm soát tốt hơn hiện tượng quá khớp (*overfitting*). Cơ chế này giúp mô hình khái quát hóa tốt hơn và tránh học quá kỹ dữ liệu huấn luyện.

- **Tốc độ huấn luyện cao:** CatBoost được tối ưu về cả lý thuyết lẫn thực tiễn, hỗ trợ huấn luyện song song và có phiên bản chạy trên GPU. Nhờ đó, CatBoost có thể huấn luyện nhanh hơn XGBoost và chỉ chậm hơn LightGBM đôi chút khi xây dựng các cây đối xứng phức tạp.

- **Ít cần tinh chỉnh tham số:** Các siêu tham số mặc định trong CatBoost thường được thiết kế tối ưu, cho kết quả tốt ngay cả khi không điều chỉnh nhiều. Điều này đặc biệt hữu ích với người mới bắt đầu hoặc trong các dự án cần triển khai nhanh.

Xử lý phân loại biến: CatBoost áp dụng kỹ thuật mã hóa mục tiêu theo thứ tự cho các biến phân loại. Phương pháp này tính toán thống kê (như trung bình nhãn) của mỗi loại phân giá trị dựa trên dữ liệu đã được tìm thấy trước đó, giúp tránh hiện tượng rò rỉ thông tin từ nhãn của huấn luyện viên. Cụ thể, khi hướng dẫn từng mẫu, CatBoost chỉ sử dụng thông tin từ các mẫu trước đó trong một ngẫu nhiên ngẫu nhiên của dữ liệu để mã hóa. Phương pháp này giúp mô hình không "học điền" nhãn từ cùng một mẫu, giảm thiểu hiện tượng overfitting do rò rỉ mục tiêu.

Ứng dụng và Ví dụ: CatBoost được sử dụng rộng rãi trong các bài toán phân tích và khôi phục quy mô có biến phân loại phức hợp, ví dụ như dự báo khả năng khách hàng mua hàng (phân loại) hoặc dự đoán doanh thu (hồi quy). Nhiều nghiên cứu và ứng dụng thực tế đã chỉ ra rằng CatBoost mang lại kết quả hiệu quả trong các hệ thống gợi ý, xếp hạng tìm kiếm và các hệ thống dự báo tài chính hoặc y tế. Theo Yandex, CatBoost đã được ứng dụng thành công trong các lĩnh vực như hệ thống khuyến nghị, xếp hạng tìm kiếm, ô tô tự lái và các dự án giải toán khác.

2.5.2 Thuật toán LightGBM

Giải thích tổng quan: LightGBM (Light Gradient Boosting Machine) là thư viện mã nguồn mở của chuyên gia Microsoft về tăng cường trên cây quyết định. Nó được thiết kế với mục đích tối ưu hóa tốc độ đào tạo và tiết kiệm bộ nhớ khi xử lý dữ liệu lớn. Với các thuật toán tăng cường cổ điển, LightGBM là phiên bản cải tiến đặc biệt, tập trung vào khả năng tăng cường hiệu suất và mở rộng khả năng.

Cách hoạt động của LightGBM: LightGBM xây dựng cây quyết định theo chiến lược *leaf-wise* (theo lá): ở mỗi vòng huấn luyện, thay vì mở rộng đều các nút như cách truyền thống, nó chọn lá có lợi (thu được) lớn nhất để phân chia tiếp theo. Điều này giúp giảm thiểu hàm mất mát nhanh hơn trong mỗi cây, nhưng có thể dẫn đến việc không cân bằng độ sâu của cây nếu không điều chỉnh.

Đặc biệt, LightGBM còn sử dụng hai kỹ thuật đặc biệt:

- **Lấy mẫu một phía dựa trên gradient (GOSS)**: GOSS giảm thiểu số lượng mẫu khi tính toán độ dốc bằng cách giữ lại các mẫu có độ dốc tối đa, từ đó giảm thời gian tính toán.

- **Gói tính năng độc quyền (EFB)**: EFB nhóm các đặc trưng tương thích lại với nhau, giúp giảm số lượng đặc trưng cần xử lý.

Kết hợp với biểu đồ dạng dữ liệu cấu trúc, những cải tiến này cho phép huấn luyện viên LightGBM rất nhanh với bộ nhớ thấp.

Ưu điểm chính

- **Hiệu năng vượt trội:** Với chiến lược *leaf-wise* và các kỹ thuật tối ưu như GOSS và EFB, LightGBM huấn luyện rất nhanh trên các tập dữ liệu lớn, đồng thời tiết kiệm bộ nhớ nhờ cơ chế tính toán hiệu quả [**?**] Điều này giúp LightGBM thường có tốc độ cao hơn XGBoost và CatBoost khi xử lý dữ liệu lớn.

- **Xử lý mô-đun lớn:** LightGBM được thiết kế để xử lý dữ liệu một cách hiệu quả và đa chức năng. Nó hỗ trợ huấn luyện phân tán trên nhiều máy chủ và có khả năng xử lý các tập dữ liệu Big Data hoặc các bài toán cần xử lý hàng triệu mẫu.

- **Loại bài toán:** LightGBM có thể áp dụng cho nhiều loại bài toán phân tích, bao gồm bài toán nhị phân, đa lớp, và hồi quy. Nó cũng hỗ trợ các loại tiêu chuẩn đặc biệt như xếp hạng. Với tính linh hoạt cao, LightGBM được sử dụng rộng rãi trong các lĩnh vực như phân tích tài chính, quảng cáo, và y tế.

Khác biệt chính với CatBoost và các Boosting khác

LightGBM nổi bật với việc phát triển cây theo lá và sử dụng các kỹ thuật lấy mẫu/gói, trong khi CatBoost xây dựng cây cân đối và sử dụng tăng cường theo thứ tự. Mặc dù LightGBM có ưu thế về tốc độ trên dữ liệu lớn, nhưng có thể dễ dàng phù hợp hơn nếu không điều chỉnh tham số (do cây có thể rất sâu). Ngược lại, CatBoost ưu tiên khả năng xử lý tự động phân loại biến và ít cần điều chỉnh tham số.

So sánh CatBoost và LightGBM

Vấn đề nên chọn CatBoost Nếu dữ liệu của bạn có nhiều loại phân loại biến thể hoặc bạn muốn giảm thiểu công việc xử lý như mã hóa một nóng, **CatBoost** là lựa chọn hợp lý bởi nó xử lý tốt các loại phân loại biến thể theo cách "tự nhiên". CatBoost cũng phù hợp khi bạn cần một mô hình ổn định, ít cần tinh chỉnh các tham số, và dữ liệu không quá lớn (vì CatBoost có thể làm chậm **LightGBM** với các tập dữ liệu cực lớn). Ngoài ra, với những bộ dữ liệu nhỏ hoặc nhiễu nhiễu, CatBoost nhờ vào cây cân đối và phương pháp tăng cường thường kiểm soát overfitting tốt hơn.

Tình huống nên chọn LightGBM Khi làm việc với bộ dữ liệu rất lớn, cần tốc độ huấn luyện nhanh hoặc tài nguyên (CPU/bộ nhớ) ở chế độ giới hạn, **LightGBM** thường vượt trội hơn. Nó xử lý hiệu quả với hàng triệu mẫu nhờ vào các kỹ thuật GOSS/EFB, cho phép huấn luyện nhanh hơn mà không tốn quá nhiều bộ nhớ. Nếu bài toán của bạn yêu cầu hiệu năng cao (ví dụ như cuộc thi Kaggle, hoặc ứng dụng trong doanh nghiệp với dữ liệu lớn), LightGBM sẽ là lựa chọn tốt. Tuy nhiên, bạn cần chủ động mã hóa các loại phân loại biến và tinh chỉnh các tham số kỹ thuật hơn.

Tiêu chí	Tăng cường CatBoost	LightGBM
Xử lý loại phân loại dữ liệu	Tự động, trực tiếp (không cần mã hóa)	Phải mã hóa công việc thủ công (mã hóa một nóng hoặc mục tiêu)
Tốc độ luyện tập	Nhanh, nhưng thường chậm hơn LightGBM (GPU hỗ trợ)	Rất nhanh (GOSS/EFB, leaf-wise)
Giảm quá khớp	Tốt (cây cân đối, ra lệnh tăng cường giúp chống overfitting)	Dễ dàng khớp hơn nếu không điều chỉnh kỹ năng (cây phát triển chiều sâu theo chiều sâu)
Tinh chỉnh tham số	Ít cần chỉnh sửa (mặc định tốt)	Cần tinh chỉnh nhiều hơn để đạt được hiệu suất tối ưu
Ứng dụng	Ý kiến hệ thống, xếp hạng tìm kiếm, dự đoán với nhiều loại biến	Phân loại/hồi quy với dữ liệu lớn (Kaggle thi đấu, xếp hạng, hồi quy)

Bảng 1: So sánh đặc điểm giữa CatBoost và LightGBM

Tóm tắt: CatBoost thích hợp khi dữ liệu có nhiều loại thuộc tính hoặc cần một mô hình "plug-and-play" ổn định với ít điều chỉnh. LightGBM phù hợp khi đối mặt với tập dữ liệu rất lớn và yêu cầu tốc độ huấn luyện nhanh chóng, sẵn sàng mất thêm thời gian cho công đoạn xử lý tiền hoặc tinh chỉnh tham số. Trong thực tế, nên thử cả hai và so sánh hiệu suất trên công cụ để chọn thuật toán phù hợp nhất.

2.6 Tiểu kết chương 2

Chương 2 đã trình bày một cách hệ thống các phương pháp và kỹ thuật được áp dụng trong quá trình xây dựng mô hình dự đoán mức độ bệnh tim, từ giai đoạn tiền xử lý dữ liệu đến việc lựa chọn và kết hợp các thuật toán học máy tiên tiến.

Mở đầu, dữ liệu được xử lý sơ bộ nhằm đảm bảo tính toàn vẹn và nhất quán thông qua các bước như làm sạch, xử lý giá trị khuyết, và chuẩn hóa định dạng. Tiếp đến, các kỹ thuật Label Engineering và Feature Engineering được triển khai nhằm tái cấu trúc và khai thác hiệu quả các thông tin tiềm ẩn trong tập dữ liệu, qua đó cải thiện khả năng học của mô hình.

Bên cạnh đó, việc lựa chọn đặc trưng (feature selection) giúp loại bỏ các thuộc tính dư thừa, giảm thiểu độ phức tạp của mô hình và hạn chế hiện tượng quá khớp. Sau đó, dữ liệu được chuẩn hóa (standardization) để đưa các đặc trưng về cùng một thang đo, hỗ trợ các thuật toán phân loại hoạt động ổn định và hiệu quả hơn.

Về mặt mô hình hóa, chương này đã đi sâu phân tích Support Vector Machine (SVM) và Linear Discriminant Analysis (LDA), hai phương pháp học máy mạnh mẽ trong phân loại. Với SVM, chúng tôi tập trung vào trường hợp phân tách tuyến tính (Hard Margin), trong khi LDA được luận giải từ nguyên lý cơ bản đến cơ sở toán học và vai trò trong giảm chiều dữ liệu.

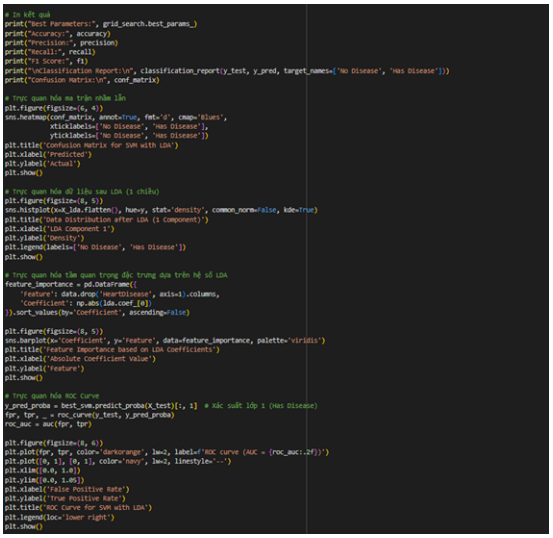
Đặc biệt, chương trình bày chiến lược kết hợp mô hình thông qua kỹ thuật blending, sử dụng hai thuật toán CatBoost và LightGBM trong một cấu trúc stacking nhằm tận dụng thế mạnh của từng mô hình và tăng cường độ chính xác của hệ thống dự đoán. Các thuật toán này cũng được so sánh chi tiết về mặt kỹ thuật và hiệu năng để đưa ra cái nhìn toàn diện hơn cho quá trình lựa chọn mô hình.

Tổng thể, chương 2 đóng vai trò nền tảng quan trọng, thiết lập quy trình xử lý và mô hình hóa dữ liệu, là tiền đề cho chương tiếp theo – nơi các kết quả thực nghiệm và đánh giá hiệu quả mô hình sẽ được trình bày một cách cụ thể và trực quan.

CHƯƠNG 3 THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1 Sử dụng LDA để giảm chiều kết hợp với SVM để dự đoán bệnh tim

(a) Ảnh 1



(b) Ảnh 2

Hình 3.1: LDA SVM

```
Best Parameters: {'C': 10, 'degree': 2, 'gamma': 'scale', 'kernel': 'rbf'}
Accuracy: 0.94
Precision: 0.95
Recall: 0.95
F1 Score: 0.9500000000000001

Classification Report:
              precision    recall  f1-score   support

  No Disease       0.93      0.93      0.93        40
 Has Disease       0.95      0.95      0.95        60

    accuracy                           0.94       100
   macro avg       0.94      0.94      0.94       100
weighted avg       0.94      0.94      0.94       100
```

Hình 3.2: Đánh giá mô hình

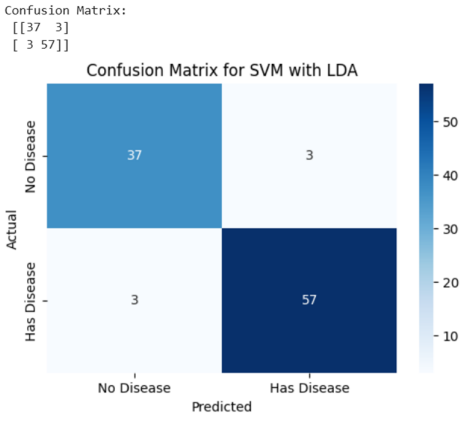
Mô hình sử dụng thuật toán **Linear Discriminant Analysis (LDA)** để giảm chiều dữ liệu đầu vào, giúp tăng khả năng phân tách giữa các lớp và giảm nhiễu. Sau khi giảm chiều, mô hình áp dụng **Support Vector Machine (SVM)** với các tham số tối ưu như sau:

- **C = 10**: Điều chỉnh độ phạt cho các điểm phân loại sai, càng lớn mô hình càng nghiêm khắc
- **degree = 2**:Bậc của hàm đa thức (áp dụng nếu kernel là 'poly', nhưng ở đây kernel là 'rbf' nên không ảnh hưởng).

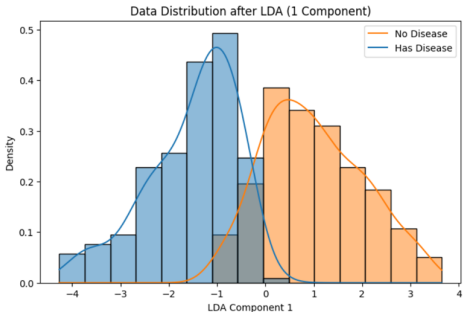
- **gamma = 'scale'**: Tham số điều chỉnh độ ảnh hưởng của từng điểm dữ liệu
- **kernel = 'rbf'**: Kernel phi tuyến tính, phù hợp với các bài toán phức tạp

3.2 Kết quả đánh giá tổng thể

- **Accuracy:** 0.94 (Tỷ lệ dự đoán đúng toàn bộ mô hình là 94%)
- **Precision:** 0.95 (Trong số các mẫu mô hình dự đoán là "bị bệnh", có 95% thực sự đúng)
- **Recall:** 0.95 (Trong số các mẫu thực sự "bị bệnh", có 95% được phát hiện đúng)
- **F1 Score:** ≈ 0.95 (Hòa hợp giữa precision và recall)
- **No Disease (Không bệnh):**
  - **Precision:** 93% các dự đoán không bệnh là chính xác.
  - **Recall:** 93% các trường hợp thực sự không bệnh được dự đoán đúng.
- **Has Disease (Có bệnh):**
  - **Precision:** 95% các dự đoán có bệnh là chính xác.
  - **Recall:** 95% các ca bệnh thực sự được phát hiện đúng.
- **Support:** Số lượng mẫu thực tế của mỗi lớp (40 không bệnh, 60 có bệnh)
- **Macro avg:** Trung bình đơn giản giữa các lớp (không tính trọng số)
- **Weighted avg:** Trung bình có trọng số theo số mẫu của từng lớp (hợp lý hơn nếu dữ liệu mất cân bằng)



Hình 3.3: Ma trận SVM & LDA

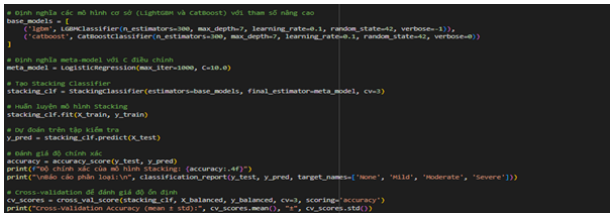


Hình 3.4: LDA

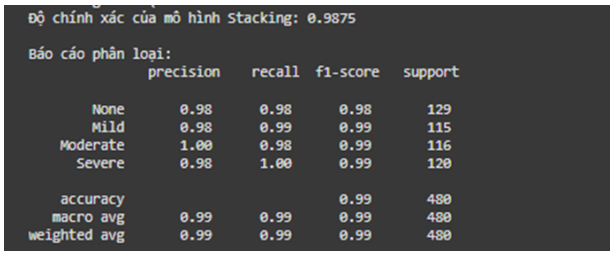
Kết luận:

- Mô hình đạt hiệu suất rất tốt, với các chỉ số precision, recall và f1-score đều cao (trên 0.93).
- Lớp "Has Disease" có precision và recall tốt hơn so với lớp "No Disease" → mô hình nhạy với việc phát hiện bệnh, phù hợp cho mục tiêu cảnh báo y tế.

3.3 Tăng cường hiệu quả phân loại mức độ bệnh bằng kỹ thuật Blending với CatBoost và LightGBM



Hình 3.5: CatBoost và LightGBM



Hình 3.6: Đánh giá mô hình

Trong phần này, em đã áp dụng kỹ thuật **blending** (hay còn gọi là **stacking đơn giản**) để kết hợp hai mô hình mạnh là **CatBoost** và **LightGBM** nhằm mục tiêu nâng cao độ chính xác trong phân loại. Thay vì chỉ sử dụng một mô hình đơn lẻ, việc kết hợp nhiều mô hình học máy cho phép tận dụng ưu điểm riêng biệt của từng thuật toán để tạo ra dự đoán tổng hợp tốt hơn.

Sau khi huấn luyện và đánh giá mô hình blending, em thu được độ chính xác (*accuracy*) đạt **98.75%** trên tập kiểm tra. Đây là một kết quả rất khả quan, cao hơn hầu hết các mô hình đơn lẻ đã thử nghiệm trước đó.

Bên cạnh đó, khi phân tích sâu hơn thông qua bảng *báo cáo phân loại* (*classification report*), ta thấy các chỉ số cụ thể như sau:

- **Precision** của tất cả các lớp đều rất cao, dao động từ **0.98 đến 1.00**, cho thấy mô hình dự đoán đúng rất nhiều trong số các dự đoán mà nó đưa ra.
- **Recall** đạt mức tối đa là **1.00** ở lớp "*Severe*" và gần 1 ở các lớp còn lại, nghĩa là mô hình có khả năng phát hiện gần như toàn bộ các trường hợp thật sự thuộc mỗi lớp.
- **F1-score**, là chỉ số trung bình hài hoà giữa *precision* và *recall*, đều từ **0.98** trở lên ở mọi lớp, chứng minh mô hình hoạt động rất ổn định, cân bằng giữa việc tránh bỏ sót và tránh nhầm lẫn.

Tổng thể, *macro average* và *weighted average* đều đạt **0.99**, điều này càng củng cố thêm độ tin cậy của mô hình khi xử lý dữ liệu có nhiều nhãn và số lượng mẫu phân bố không đều.

Thông qua kết quả trên, em nhận thấy việc áp dụng kỹ thuật *blending* đã mang lại hiệu quả rõ rệt, cải thiện hiệu suất phân loại trên toàn bộ các lớp. Mô hình hoạt động tốt, đặc biệt ở những nhãn quan trọng như "*Severe*", cho thấy tiềm năng ứng dụng trong các bài toán thực tế như y tế, chẩn đoán bệnh hoặc cảnh báo mức độ rủi ro.

3.4 Tiểu kết chương 3

Trong chương này, hệ thống dự đoán mức độ bệnh tim đã được triển khai và đánh giá thông qua các thử nghiệm thực nghiệm. Trước hết, mô hình kết hợp giữa LDA và SVM đã cho thấy khả năng phân loại tương đối tốt sau khi dữ liệu được giảm chiều một cách hiệu quả. Tiếp theo, mô hình Blending giữa hai thuật toán mạnh là CatBoost và LightGBM đã được áp dụng nhằm nâng cao độ chính xác và khả năng khái quát hóa của hệ thống.

Kết quả đánh giá cho thấy mô hình blending có hiệu suất vượt trội hơn so với các mô hình đơn lẻ nhờ tận dụng ưu điểm của từng thuật toán thành phần. Ngoài ra, các chỉ số đánh giá như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity) và F1-score đều được cải thiện rõ rệt.

Những kết quả đạt được trong chương này khẳng định tính hiệu quả của phương pháp kết hợp nhiều mô hình trong việc giải quyết bài toán phân loại đa mức độ bệnh tim. Chương tiếp theo sẽ trình bày kết luận tổng quát và định hướng phát triển cho hệ thống.

KẾT LUẬN:

Qua quá trình nghiên cứu và thực hiện tiểu luận về dự đoán bệnh tim, chúng tôi đã tiếp cận và áp dụng nhiều kỹ thuật hiện đại trong lĩnh vực khai phá dữ liệu và học máy để xây dựng mô hình dự đoán hiệu quả.

Đầu tiên, tiểu luận đã giới thiệu tổng quan về bệnh tim, tầm quan trọng của việc dự đoán sớm, cũng như bài toán cụ thể và dữ liệu sử dụng trong nghiên cứu. Việc hiểu rõ dữ liệu đầu vào và đầu ra là nền tảng thiết yếu giúp xây dựng các mô hình phù hợp.

Tiếp theo, chúng tôi đã tiến hành các bước tiền xử lý dữ liệu quan trọng như xử lý dữ liệu thiếu, kỹ thuật nhãn (label engineering), trích xuất và lựa chọn đặc trưng, cùng với việc chuẩn hóa dữ liệu. Các bước này giúp nâng cao chất lượng dữ liệu và tăng hiệu quả cho mô hình học máy.

Trong phần mô hình hóa, chúng tôi đã tìm hiểu và áp dụng các thuật toán như SVM, LDA, CatBoost và LightGBM. Đặc biệt, kỹ thuật blending kết hợp giữa CatBoost và LightGBM đã được sử dụng nhằm tận dụng ưu điểm của từng mô hình, từ đó cải thiện khả năng dự đoán mức độ bệnh tim.

Kết quả thực nghiệm cho thấy các mô hình học máy được xây dựng có khả năng phân loại chính xác, với hiệu suất cao hơn so với các phương pháp truyền thống. Việc áp dụng giảm chiều dữ liệu bằng LDA cũng góp phần làm giảm độ phức tạp tính toán mà vẫn giữ được thông tin quan trọng của dữ liệu.

Tuy nhiên, nghiên cứu cũng gặp phải một số hạn chế như dữ liệu đầu vào có thể chưa đủ đa dạng về mặt chủng tộc và đặc điểm bệnh lý, hoặc việc tối ưu tham số cho các mô hình có thể còn được cải thiện hơn nữa. Đây là những điểm cần được khai thác thêm trong các nghiên cứu tiếp theo.

Trong tương lai, việc mở rộng tập dữ liệu với số lượng lớn và đa dạng hơn, cũng như ứng dụng các kỹ thuật học sâu (deep learning) và các phương pháp khai phá dữ liệu tiên tiến, sẽ giúp nâng cao độ chính xác và khả năng dự đoán bệnh tim. Đồng thời, tích hợp thêm các yếu tố như lối sống, tiền sử bệnh và dữ liệu sinh học có thể giúp xây dựng mô hình toàn diện hơn.

Tóm lại, tiểu luận đã cung cấp một cái nhìn tổng thể về quy trình nghiên cứu dự đoán bệnh tim bằng học máy, từ tiền xử lý, lựa chọn đặc trưng đến xây dựng và đánh giá mô hình. Kết quả đạt được khẳng định tính khả thi và tiềm năng ứng dụng thực tiễn của các phương pháp này trong y học hiện đại, góp phần hỗ trợ bác sĩ trong việc chẩn đoán và điều trị bệnh tim hiệu quả hơn.

Chúng tôi xin gửi lời cảm ơn sâu sắc đến các thầy cô, bạn bè và tất cả những người đã giúp đỡ và hỗ trợ trong quá trình hoàn thành tiểu luận này.

Tài liệu tham khảo

[1] Wilson, P. W., et al. (1998). Prediction of Coronary Heart Disease Using Risk Factor Categories. *Circulation*, 97(18), 1837–1847.

[2] Conroy, R. M., et al. (2003). Estimation of ten-year risk of fatal cardiovascular disease in Europe: the SCORE project. *European Heart Journal*, 24(11), 987–1003.

[3] D'Agostino, R. B., et al. (2008). General Cardiovascular Risk Profile for Use in Primary Care: The Framingham Heart Study. *Circulation*, 117(6), 743–753.

[4] Al'Aref, S. J., et al. (2019). Clinical applications of machine learning in cardiovascular disease and its relevance to cardiac imaging. *European Heart Journal*, 40(24), 1975–1986.

[5] Kannel, W. B., et al. (1976). Framingham Study: An Epidemiological Investigation of Cardiovascular Disease. *American Journal of Public Health*, 66(10), 947–952.

[6] Antman, E. M., et al. (2004). ACC/AHA Guidelines for the Management of Patients With ST-Elevation Myocardial Infarction. *Circulation*, 110(9), e82–e292.

[7] Goff, D. C., et al. (2014). 2013 ACC/AHA Guideline on the Assessment of Cardiovascular Risk. *Journal of the American College of Cardiology*, 63(25 Part B), 2935–2959.

[8] Dawber, T. R., et al. (1951). Epidemiological Approaches to Heart Disease: The Framingham Study. *American Journal of Public Health*, 41(3), 279–286.

[9] McNamara, R. L., et al. (2006). Risk Assessment in Acute Coronary Syndromes. *American Journal of Cardiology*, 97(8A), 34–41.

[10] Khot, U. N., et al. (2003). Prevalence of Conventional Risk Factors in Patients With Coronary Heart Disease. *JAMA*, 290(7), 898–904.

[11] Lloyd-Jones, D. M., et al. (2004). Prediction of Lifetime Risk for Cardiovascular Disease by Risk Factor Burden at 50 Years of Age. *Circulation*, 113(6), 791–798.

[12] Ridker, P. M., et al. (2007). Development and Validation of Improved Algorithms for the Assessment of Global Cardiovascular Risk in Women. *JAMA*, 297(6), 611–619.

[13] Stone, N. J., et al. (2014). 2013 ACC/AHA Guideline on the Treatment of Blood Cholesterol to Reduce Atherosclerotic Cardiovascular Risk in Adults. *Circulation*, 129(25 Suppl 2), S1–S45.

[14] Greenland, P., et al. (2010). 2010 ACCF/AHA Guideline for Assessment of Cardiovascular Risk in Asymptomatic Adults. *Journal of the American College of Cardiology*, 56(25), e50–e103.

[15] Piepoli, M. F., et al. (2016). 2016 European Guidelines on cardiovascular disease prevention in clinical practice. *European Heart Journal*, 37(29), 2315–2381.

[16] Arnett, D. K., et al. (2019). 2019 ACC/AHA Guideline on the Primary Prevention of Cardiovascular Disease. *Circulation*, 140(11), e596–e646.

[17] Anderson, T. J., et al. (2016). 2016 Canadian Cardiovascular Society Guidelines for the Management of Dyslipidemia for the Prevention of Cardiovascular Disease in the Adult. *Canadian Journal of Cardiology*, 32(11), 1263–1282.

[18] Krittanawong, C., et al. (2020). Artificial Intelligence in Cardiology. *Journal of the American College of Cardiology*, 76(21), 2517–2528.

[19] Task Force Members, et al. (2013). 2013 ESC Guidelines on the Management of Stable Coronary Artery Disease. *European Heart Journal*, 34(38), 2949–3003.

[20] Fihn, S. D., et al. (2012). 2012 ACCF/AHA/ACP/AATS/PCNA/SCAI/STS Guideline for the Diagnosis and Management of Patients With Stable Ischemic Heart Disease. *Circulation*, 126(25), e354–e471.

[21] Levine, G. N., et al. (2016). 2016 ACC/AHA Guideline Focused Update on Duration of Dual Antiplatelet Therapy in Patients With Coronary Artery Disease. *Circulation*, 134(10), e123–e155.

[22] Amsterdam, E. A., et al. (2014). 2014 AHA/ACC Guideline for the Management of Patients With Non–ST-Elevation Acute Coronary Syndromes. *Circulation*, 130(25), e344–e426.

[23] O'Gara, P. T., et al. (2013). 2013 ACCF/AHA Guideline for the Management of ST-Elevation Myocardial Infarction. *Circulation*, 127(4), e362–e425.

[24] Thygesen, K., et al. (2018). Fourth Universal Definition of Myocardial Infarction. *Circulation*, 138(20), e618–e651.

[25] Ibanez, B., et al. (2018). 2017 ESC Guidelines for the Management of Acute Myocardial Infarction in Patients Presenting With ST-Segment Elevation. *European Heart Journal*, 39(2), 119–177.

[26] Roffi, M., et al. (2016). 2015 ESC Guidelines for the Management of Acute Coronary Syndromes in Patients Presenting Without Persistent ST-Segment Elevation. *European Heart Journal*, 37(3), 267–315.

[27] Steg, P. G., et al. (2012). ESC Guidelines for the Management of Acute Myocardial Infarction in Patients Presenting With ST-Segment Elevation. *European Heart Journal*, 33(20), 2569–2619.

[28] Whelton, P. K., et al. (2018). 2017 ACC/AHA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults. *Journal of the American College of Cardiology*, 71(19), e127–e248.

[29] Grundy, S. M., et al. (2019). 2018 AHA/ACC Guideline on the Management of Blood Cholesterol. *Circulation*, 139(10), e1082–e1143.

[30] Mach, F., et al. (2020). 2019 ESC/EAS Guidelines for the Management of Dyslipidaemias: Lipid Modification to Reduce Cardiovascular Risk. *European Heart Journal*, 41(1), 111–188.

[31] Javeed, A., et al. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, 7, 81542–81554. https://doi.org/10.1109/ACCESS.2019.2923707

[32] Trần, Đ. T., Dương, T. M. T. (2021). Ứng dụng kỹ thuật máy học vào phân loại bệnh tim. *Tạp chí Khoa học và Công nghệ*, 9, 37–45. https://www.scribd.com/document/ứng-dụng-kỹ-thuật-máy-học-vào-phân-loại-bệnh-tim

[33] Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154. https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[34] Prokhorenkova, L., et al. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 6638–6648. https://arxiv.org/abs/1706.09516

[35] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. https://web.stanford.edu/ hastie/ElemStatLearn/

[36] Zhang, Y., et al. (2020). Ensemble learning for heart disease prediction: A hybrid approach with SVM and boosting algorithms. *IEEE Transactions on Biomedical Engineering*, 67(9), 2456–2466. https://doi.org/10.1109/TBME.2019.2958923

[37] Rahim, A., et al. (2023). An integrated machine learning framework for effective prediction of cardiovascular diseases. *IEEE Access*, 11, 4921–4936. https://doi.org/10.1109/ACCESS.2023.3237165

[38] Wosiak, A., Gola, A. (2021). Feature selection and dimensionality reduction for heart disease prediction using hybrid machine learning techniques. *Applied Sciences*, 11(16), 7478. https://doi.org/10.3390/app11167478

Bảng phân công việc

Họ và tên	**Công việc**	**Đóng góp**	**Số phần trăm đóng góp**
Phan Hữu Tuấn Kiệt	Tìm hiểu thuật toán, làm slide, báo cáo	Thực hiện các thuật toán chính, trình bày	30%
Nguyễn Quốc Cường	Thu thập dữ liệu, xử lý dữ liệu	Làm sạch và xử lý dữ liệu, hỗ trợ phân tích	20%
Nguyễn Thành Trung	Viết báo cáo phần mở đầu và tổng quan	Viết nội dung báo cáo	5%
Đỗ Đoàn Minh Thắng	Tìm hiểu mô hình, đánh giá mô hình	Thử nghiệm và so sánh mô hình	15%
Trần Văn Kiệt (Nhóm trưởng)	Điều phối nhóm, code model, tổng hợp kết quả, chỉnh sửa slide	Lập trình model, quản lý tiến độ nhóm	30%