TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN HỌC MÁY VÀ ỨNG DỤNG

DỰ ĐOÁN BỆNH BẰNG CÁCH SỬ DỤNG MÁY HỌC

Giảng viên hướng dẫn: **VÕ THỊ HỒNG THẮM**
Sinh viên thực hiện: **CHÂU HÙNG ANH**
MSSV: **2000005789**
Khoá: **2020**
Ngành/ chuyên ngành: **CÔNG NGHỆ THÔNG TIN/ KHOA HỌC DỮ LIỆU**

Tp HCM, tháng 08 năm 2023

TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN HỌC MÁY VÀ ỨNG DỤNG

DỰ ĐOÁN BỆNH BẰNG CÁCH SỬ DỤNG MÁY HỌC

Giảng viên hướng dẫn: **VÕ THỊ HỒNG THẮM**
Sinh viên thực hiện: **CHÂU HÙNG ANH**
MSSV: **2000005789**
Sinh viên thực hiện: **CHÂU THIÊN BẢO**
MSSV: **2000005640**
Sinh viên thực hiện: **VÕ THÁI HIẾN**
MSSV: **2000005690**
Khoá: **2020**
Ngành/ chuyên ngành: **CÔNG NGHỆ THÔNG TIN/ KHOA HỌC DỮ LIỆU**

TPHCM, tháng 08 năm 2023

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Cô về môn học "Học máy và ứng dụng" trong học kỳ vừa qua tại Trường Đại học Nguyễn Tất Thành. Cô đã tạo ra một môi trường học tập đầy thú vị và bổ ích, giúp em hiểu rõ hơn về lĩnh vực quan trọng này.

Em thực sự trân trọng cách Cô trình bày kiến thức một cách rõ ràng, dễ hiểu và thú vị. Nhờ vào sự hướng dẫn tận tâm của Cô, em đã có cơ hội tiếp cận những khái niệm phức tạp một cách dễ dàng hơn. Những ví dụ thực tế và bài tập thực hành trong khóa học đã giúp em áp dụng kiến thức vào thực tế và phát triển kỹ năng thực sự cần thiết.

Không chỉ giới hạn trong việc truyền đạt kiến thức, Cô còn tạo điều kiện cho chúng em thảo luận, trao đổi ý kiến và học hỏi từ nhau. Điều này thực sự đã tạo ra một không gian học tập tích cực và khuyến khích sự phát triển cá nhân của từng học viên.

Em cảm ơn Cô vì sự kiên nhẫn và lòng nhiệt tình trong việc giải đáp các thắc mắc của em. Dù là trong lớp học hay ngoài giờ, Cô luôn sẵn sàng hỗ trợ và động viên em vượt qua những khó khăn trong quá trình học tập.

Cuối cùng, em xin chân thành cảm ơn Cô vì sự dạy dỗ tận tâm và những kiến thức bổ ích mà Cô đã truyền đạt cho chúng em. Môn học này không chỉ giúp em nắm vững kiến thức về Học máy và ứng dụng, mà còn giúp em phát triển tư duy phân tích, giải quyết vấn đề và học hỏi cách tiếp cận các thách thức trong tương lai.

Một lần nữa, em xin bày tỏ lòng biết ơn chân thành đến Cô Võ Thị Hồng Thắm. Mong rằng Cô sẽ tiếp tục có những đóng góp quý báu trong việc truyền đạt kiến thức cho thế hệ học viên tương lai.

Ký tên

Châu Thiên Bảo

BM-ChT-

TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
TRUNG TÂM KHẢO THÍ

KỲ THI KẾT THÚC HỌC PHẦN
HỌC KỲ III NĂM HỌC 2022 - 2023

PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN

Môn thi: Học máy và ứng dụng ..... Lớp học phần: 20DTH1D.....

Nhóm sinh viên thực hiện : .....

1. Châu Hùng Anh ..... Tham gia đóng góp: 35% .....
2. Châu Thiên Bảo ..... Tham gia đóng góp: 35% .....
3. Võ Thái Hiển..... Tham gia đóng góp: 30% .....
4. ..... Tham gia đóng góp:.....
5. ..... Tham gia đóng góp:.....
6. ..... Tham gia đóng góp:.....
7. ..... Tham gia đóng góp:.....
8. ..... Tham gia đóng góp:.....

Ngày thi: 25/08/2023..... Phòng thi: L.604.....

Đề tài tiểu luận/báo cáo của sinh viên : Dự đoán bệnh bằng cách sử dụng máy học.....

.....

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

Tiêu chí (theo CĐR HP)	**Đánh giá của GV**	**Điểm tối đa**	**Điểm đạt được**
Cấu trúc của báo cáo	....		
Nội dung			
- Các nội dung thành phần	....		
- Lập luận	....		
- Kết luận	.....		
Trình bày	.....		
TỔNG ĐIỂM			

Giảng viên chấm thi
(ký, ghi rõ họ tên)

MỤC LỤC
(Bold, size 14)

DANH MỤC HÌNH

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
CSV	Comma-Separated Values
NLP	Natural Language Processing
SVC	Support Vector Classification
SVM	Support Vector Machine

CHƯƠNG 1: XÁC ĐỊNH ĐỀ TÀI

1.1. Lý do chọn đề tài

Sức khỏe con người là một vấn đề quan trọng và việc dự đoán bệnh một cách sớm có thể giúp cải thiện chất lượng cuộc sống và tối ưu hóa quá trình điều trị. Một trong những phương pháp hiện đại để tiếp cận việc này là sử dụng học máy, đặc biệt là mô hình SVM. Chọn đề tài này nhằm khám phá khả năng ứng dụng của học máy trong dự đoán bệnh và đánh giá hiệu suất của mô hình SVM trong việc này.

1.2. Phương pháp nghiên cứu

Thu thập dữ liệu: Sẽ thu thập dữ liệu liên quan đến các yếu tố có thể ảnh hưởng đến bệnh, chẳng hạn như thông tin về lối sống, di truyền, chỉ số sức khỏe. Dữ liệu này có thể được thu thập từ các nguồn y tế, nghiên cứu khoa học và cơ sở dữ liệu liên quan.

Tiền xử lý dữ liệu: Dữ liệu thu thập được thường không hoàn hảo và chứa nhiễu. Sẽ thực hiện các bước tiền xử lý như loại bỏ dữ liệu trùng lắp, điền giá trị bị thiếu, chuẩn hóa dữ liệu để đảm bảo tính nhất quán và chính xác.

Xây dựng mô hình SVM: Sử dụng thư viện Scikit-learn, sẽ xây dựng mô hình SVM để dự đoán khả năng mắc bệnh dựa trên dữ liệu đã tiền xử lý. Các tham số như loại Kernel, tham số C và gamma sẽ được tinh chỉnh để đạt được hiệu suất tốt nhất.

Huấn luyện và đánh giá mô hình: Dữ liệu sẽ được chia thành tập huấn luyện và tập kiểm tra. Mô hình SVM sẽ được huấn luyện trên tập huấn luyện và sau đó được đánh giá trên tập kiểm tra bằng các độ đo như độ chính xác, độ phủ và độ đo F1-score.

Phân tích kết quả: Sẽ phân tích kết quả dự đoán của mô hình SVM để hiểu rõ hơn về khả năng dự đoán bệnh và đối chiếu với các thông tin y tế thực tế.

1.3. Giới hạn phạm vi nghiên cứu

Phạm vi bệnh: Nghiên cứu sẽ tập trung vào một số bệnh cụ thể, không bao gồm toàn bộ danh sách bệnh. Điều này giúp tập trung nghiên cứu và đảm bảo tính xác thực của kết quả.

Dữ liệu: Nghiên cứu sẽ sử dụng các nguồn dữ liệu sẵn có hoặc mô phỏng dữ liệu. Dựa vào tính khả thi và tài nguyên, nghiên cứu có thể giới hạn mẫu dữ liệu.

Hiệu suất mô hình: Sẽ xem xét hiệu suất của mô hình SVM trong một tình huống cụ thể. Hiệu suất có thể thay đổi tùy thuộc vào dữ liệu và tham số mô hình được chọn.

Khả năng dự đoán: Mô hình dự đoán chỉ là một phần trong quá trình chẩn đoán. Sẽ không thực hiện chẩn đoán hoàn chỉnh mà tập trung vào khả năng dự đoán cơ bản.

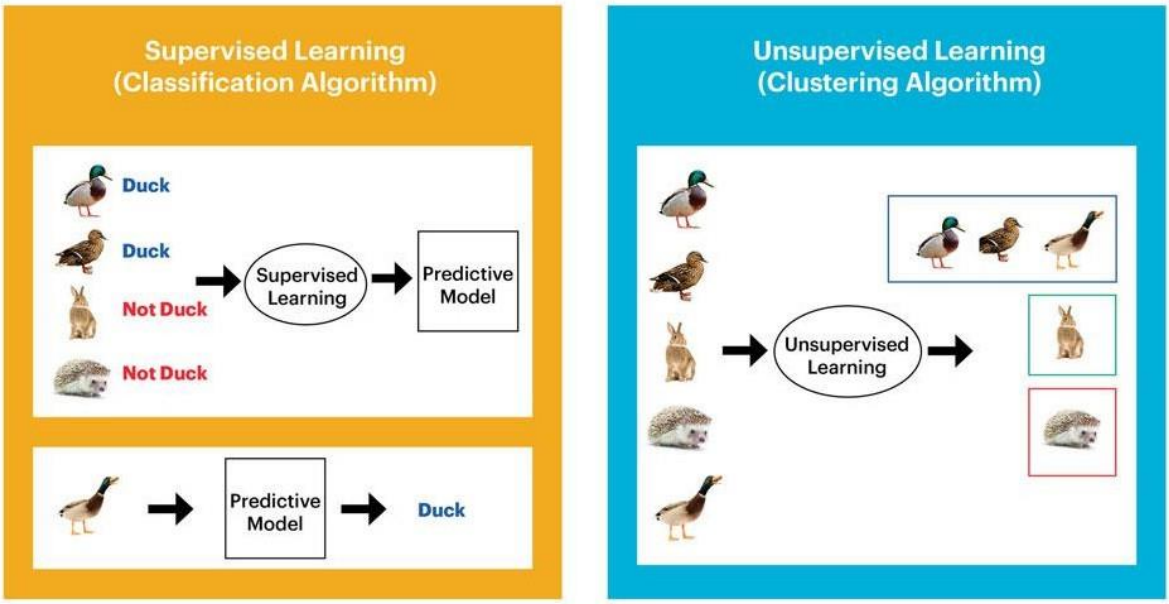
CHƯƠNG 2: HỌC MÁY VÀ ỨNG DỤNG, NGÔN NGỮ PYTHON VÀ MÔ HÌNH SVM

2.1. Học máy và ứng dụng

2.1.1. Lý thuyết

- Học máy là một lĩnh vực của trí tuệ nhân tạo tập trung vào việc phát triển các mô hình và thuật toán để máy tính có thể học từ dữ liệu và cải thiện hiệu suất theo thời gian.

- Học máy có hai loại chính: học có giám sát (supervised learning) và học không giám sát (unsupervised learning). Trong học có giám sát, mô hình được huấn luyện trên dữ liệu đã được gán nhãn, trong khi trong học không giám sát, mô hình tìm hiểu cấu trúc dữ liệu mà không có nhãn.



Hình 2.1: Học máy có giám sát (Supervised Learning) và học máy không giám sát (Unsupervised Learning)

- Có nhiều thuật toán học máy khác nhau, bao gồm cây quyết định, Random Forest, Naive Bayes, Support Vector Machine (SVM), Neural Networks, và nhiều thuật toán khác.

- Học máy có thể được sử dụng để phân loại các đối tượng vào các lớp khác nhau (classification) hoặc dự đoán một giá trị cụ thể (prediction), dựa trên dữ liệu đầu vào và mô hình đã học.

2.1.2. Ứng dụng trong thực tế

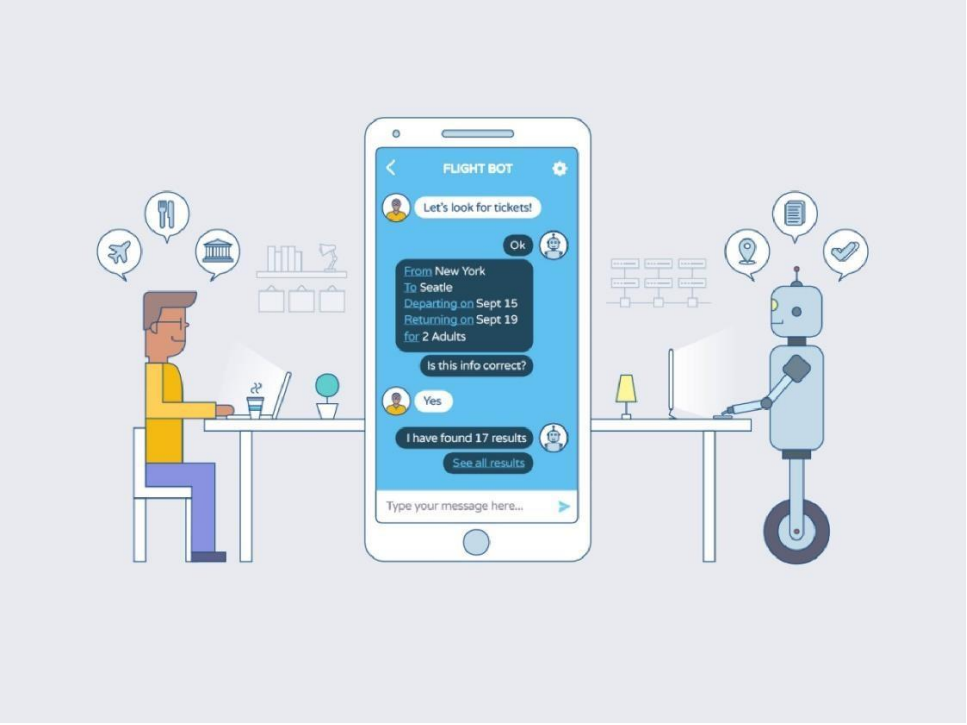
- Xử lý ngôn ngữ tự nhiên (NLP): Học máy có thể giúp xử lý văn bản, dự đoán ý kiến người dùng, dịch thuật tự động và phân tích tình cảm.

- Phân tích hình ảnh và video: Học máy được sử dụng để nhận diện đối tượng trong hình ảnh, phân loại nội dung video, và thậm chí tạo ra nội dung đồ họa mới.

- Dự đoán tài chính: Học máy có thể dự đoán giá cổ phiếu, biến động thị trường, đánh giá rủi ro tín dụng và phát hiện gian lận tài chính.

- Y học và chăm sóc sức khỏe: Học máy có thể giúp dự đoán và phát hiện bệnh, tối ưu hóa kế hoạch điều trị, phân tích hình ảnh y khoa và đưa ra quyết định y học.

- Ô tô tự hành: Học máy được sử dụng để phát hiện vật cản, xử lý dữ liệu từ cảm biến và điều khiển xe tự động.



Hình 2.2: Chat bot hỗ trợ giải quyết tự động các vấn đề thường gặp của khách hàng

2.2. Ngôn ngữ Python Mô hình SVM và ứng dụng trọng học máy

2.2.1. Ngôn ngữ Python

Python là ngôn ngữ lập trình mạnh mẽ và linh hoạt, đặc biệt phù hợp cho việc thực hiện các tác vụ trong lĩnh vực Học máy và Ứng dụng. Dưới đây là một số khía cạnh quan trọng về việc sử dụng Python trong đề tài "Dự đoán bệnh bằng cách sử dụng máy học":

- **Thư viện hỗ trợ Học máy:** Python cung cấp nhiều thư viện mạnh mẽ như Scikit-learn (Sklearn) để triển khai các thuật toán Học máy, bao gồm cả SVM. Sklearn cung cấp một tập hợp các công cụ để tiền xử lý dữ liệu, xây dựng mô hình và đánh giá hiệu suất.

- **Đọc và xử lý dữ liệu:** Python cho phép đọc và xử lý dữ liệu dễ dàng thông qua các thư viện như Pandas và NumPy. Bạn có thể nạp dữ liệu từ các nguồn khác nhau, làm sạch dữ liệu, và chuẩn bị chúng để huấn luyện mô hình.

- **Trực quan hóa dữ liệu:** Các thư viện như Matplotlib và Seaborn giúp bạn trực quan hóa dữ liệu để hiểu rõ hơn về mô hình và kết quả dự đoán.

2.2.2. Mô hình SVM

Support Vector Machine (SVM) là một trong những mô hình phân loại mạnh mẽ, đặc biệt trong trường hợp dự đoán bệnh. Dưới đây là một số khái niệm quan trọng về việc sử dụng SVM trong đề tài này:

- **Phân loại và dự đoán:** SVM được sử dụng rộng rãi để phân loại dữ liệu vào các lớp khác nhau dựa trên các đặc trưng. Trong đề tài dự đoán bệnh, SVM có thể học từ các dữ liệu đã biết để dự đoán xác suất một người có khả năng mắc bệnh.

- **Hyperplane và Margin:** SVM xây dựng siêu phẳng (hyperplane) để tách hai lớp dữ liệu. Margin là khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất. Mục tiêu là tìm siêu phẳng có margin lớn nhất.

- **Hàm Kernel:** Đôi khi dữ liệu không thể phân loại tuyến tính trong không gian hiện tại. SVM sử dụng hàm Kernel để ánh xạ dữ liệu vào một không gian khác, nơi chúng có thể phân loại tuyến tính.

2.2.3. Ứng dụng vào đề tài môn học

Dự đoán bệnh bằng cách sử dụng SVM có nhiều ứng dụng thực tế:

- Dự đoán sức khỏe: SVM có thể dự đoán nguy cơ mắc bệnh dựa trên dữ liệu về lối sống, di truyền và các chỉ số sức khỏe.

- Chẩn đoán sớm: SVM có khả năng dự đoán các biểu hiện sớm của bệnh dựa trên dữ liệu đầu vào, giúp chẩn đoán sớm và tăng khả năng điều trị hiệu quả.

- Quản lý dịch tễ học: SVM có thể được sử dụng để dự đoán sự lây lan của các bệnh truyền nhiễm trong cộng đồng, hỗ trợ việc quản lý dịch tễ học.

CHƯƠNG 3: HỌC MÁY: THUẬT TOÁN ALGORITHM, ỨNG DỤNG THỰC TẾ VÀ NGHIÊN CỨU

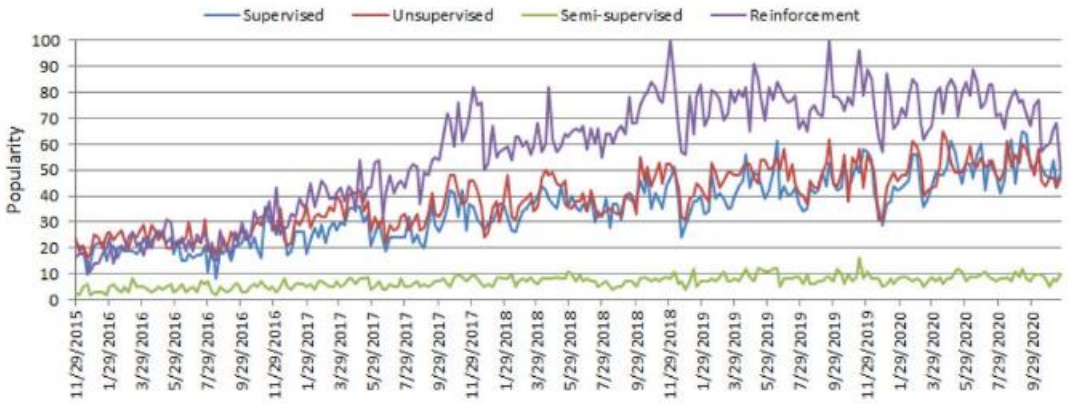
3.1. Tóm tắt.

- Trong thời đại hiện nay của Cách mạng Công nghiệp Thứ Tư (4IR hoặc Công nghiệp 4.0), thế giới kỹ thuật số đang có một lượng lớn dữ liệu, chẳng hạn như dữ liệu Internet vạn vật (IoT), dữ liệu an ninh mạng, dữ liệu di động, dữ liệu kinh doanh, dữ liệu phương tiện truyền thông xã hội, dữ liệu y tế, v.v. Để phân tích thông minh những dữ liệu này và phát triển các ứng dụng thông minh và tự động tương ứng, kiến thức về trí tuệ nhân tạo (AI), đặc biệt là học máy (ML) là yếu tố quan trọng. Các loại thuật toán học máy khác nhau như học có giám sát, học không có giám sát, học bán giám sát và học tăng cường tồn tại trong lĩnh vực này. Ngoài ra, học sâu, một phần của một họ phương pháp học máy rộng hơn, có khả năng phân tích thông minh dữ liệu trên quy mô lớn. Trong bài viết này, chúng tôi trình bày một cái nhìn toàn diện về những thuật toán học máy này có thể được áp dụng để nâng cao thông minh và khả năng của một ứng dụng. Do đó, đóng góp chính của nghiên cứu này là giải thích nguyên tắc của các kỹ thuật học máy khác nhau và khả năng áp dụng của chúng trong các lĩnh vực ứng dụng thực tế khác nhau, chẳng hạn như hệ thống an ninh mạng, thành phố thông minh, chăm sóc sức khỏe, thương mại điện tử, nông nghiệp và nhiều lĩnh vực khác. Chúng tôi cũng nhấn mạnh những thách thức và hướng nghiên cứu tiềm năng dựa trên nghiên cứu của chúng tôi. Tổng cộng, bài viết này nhằm mục đích phục vụ như một điểm tham khảo cho cả học thuật và các chuyên gia trong ngành công nghiệp cũng như người quyết định trong các tình huống và lĩnh vực ứng dụng thực tế khác nhau, đặc biệt từ góc độ kỹ thuật.

- Từ khóa: Học máy · Học sâu · Trí tuệ nhân tạo · Khoa học dữ liệu · Quyết định dựa trên dữ liệu Phân tích dự đoán · Các ứng dụng thông minh.

3.2. Giới thiệu.

Trong thời đại số hóa và công nghệ thông tin phát triển vượt bậc, học máy đã nổi lên như một trụ cột quan trọng, định hình cách chúng ta tiếp cận với dữ liệu và thông tin. Lĩnh vực này tập trung vào việc phát triển thuật toán và mô hình giúp máy tính tự học hỏi từ dữ liệu và áp dụng kiến thức đó để dự đoán, phân loại và giải quyết các vấn đề phức tạp. Báo cáo này sẽ đi sâu vào các khía cạnh của học máy, bao gồm thuật toán, ứng dụng thực tế và những hướng nghiên cứu tiềm năng trong lĩnh vực này.



Hình 11. Điểm phổ biến trên toàn thế giới của các loại thuật toán Học máy (học có giám sát, học không giám sát, học bán giám sát và học tăng cường) trong khoảng từ 0 (tối thiểu) đến 100 (tối đa) theo thời gian, trong đó trục x biểu thị thông tin thời gian và trục y biểu thị điểm phổ biến tương ứng.

- Khoa học máy tính SN :

+ Thảo luận về khả năng áp dụng của các giải pháp dựa trên học máy trong các lĩnh vực ứng dụng thực tế khác nhau.

+ Đề cập và tóm tắt các hướng nghiên cứu tiềm năng trong phạm vi nghiên cứu của chúng tôi về phân tích dữ liệu thông minh và dịch vụ.

- Phần còn lại của bài báo được tổ chức như sau. Phần tiếp theo trình bày các loại dữ liệu và thuật toán học máy trong một khía cạnh rộng hơn và xác định phạm vi nghiên cứu của chúng tôi. Chúng tôi sẽ tóm tắt và giải thích các thuật toán học máy khác nhau trong phần tiếp theo, sau đó trình bày và tóm tắt các lĩnh vực ứng dụng thực tế khác nhau dựa trên các thuật toán học máy.

- Trong phần cuối cùng, chúng tôi sẽ nhấn mạnh một số vấn đề nghiên cứu và hướng phát triển tiềm năng, và phần kết luận sẽ đưa ra tổng kết cho bài báo này.

3.3. Các loại dữ liệu thực tế và kỹ thuật học máy.

- Các thuật toán học máy thường tiêu thụ và xử lý dữ liệu để học các mẫu liên quan đến cá nhân, quy trình kinh doanh, giao dịch, sự kiện, và nhiều hơn nữa. Trong phần tiếp theo, chúng tôi sẽ thảo luận về các loại dữ liệu thực tế khác nhau cũng như các loại thuật toán học máy.

3.3.1. Các loại dữ liệu trong thế giới thực.

- Thường thì, tính sẵn có của dữ liệu được xem xét là yếu tố quan trọng để xây dựng một mô hình học máy hoặc các hệ thống thực tế dựa trên dữ liệu . Dữ liệu có thể có các hình thức khác nhau, chẳng hạn như có cấu trúc, bán cấu trúc hoặc không có cấu trúc. Bên cạnh đó, "siêu dữ liệu" là một loại khác thường biểu thị dữ liệu về dữ liệu. Dưới đây, chúng tôi ngắn gọn thảo luận về các loại dữ liệu này.

+ Cấu trúc: Dữ liệu có cấu trúc có một cấu trúc xác định rõ, tuân theo mô hình dữ liệu theo thứ tự tiêu chuẩn, được tổ chức một cách có trật tự và dễ dàng truy cập, và được sử dụng bởi một thực thể hoặc một chương trình máy tính. Trong các hệ thống có cấu trúc, như cơ sở dữ liệu quan hệ, dữ liệu có cấu trúc thường được lưu trữ dưới định dạng bảng. Ví dụ, tên, ngày, địa chỉ, số thẻ tín dụng, thông tin cổ phiếu, vị trí địa lý, v.v. là các ví dụ về dữ liệu có cấu trúc.

+ Không có cấu trúc: Ngược lại, không có định dạng hoặc tổ chức xác định trước cho dữ liệu không có cấu trúc, làm cho việc thu thập, xử lý và phân tích dữ liệu này khó khăn hơn nhiều, thường chứa văn bản và tài liệu đa phương tiện. Ví dụ, dữ liệu cảm biến, email, bài đăng trên blog, wikis và tài liệu xử lý văn bản, tệp PDF, tệp âm thanh, video, hình ảnh, bài thuyết trình, trang web và nhiều loại tài liệu kinh doanh khác có thể được coi là dữ liệu không có cấu trúc.

+ Bán cấu trúc: Dữ liệu bán cấu trúc không được lưu trữ trong cơ sở dữ liệu quan hệ như dữ liệu có cấu trúc đã được đề cập ở trên, nhưng nó có một số tính chất tổ chức cụ thể làm cho việc phân tích dễ dàng hơn. Tài liệu HTML, XML, JSON, cơ sở dữ liệu NoSQL, v.v., là một số ví dụ về dữ liệu bán cấu trúc.

+ Siêu dữ liệu: Đây không phải là dạng bình thường của dữ liệu, mà là "dữ liệu về dữ liệu". Sự khác biệt chính giữa "dữ liệu" và "siêu dữ liệu" là dữ liệu đơn giản chỉ là tài liệu có thể phân loại, đo lường hoặc thậm chí là tài liệu tài liệu liên quan đến các thuộc tính dữ liệu của tổ chức. Trong khi đó, siêu dữ liệu mô tả thông tin dữ liệu liên quan, tạo cho nó ý nghĩa quan trọng hơn đối với người dùng dữ liệu. Một ví dụ cơ bản về siêu dữ liệu của tài liệu có thể là tác giả, kích thước tệp, ngày tạo tài liệu, từ khóa xác định tài

liệu, v.v.

- Trong lĩnh vực học máy và khoa học dữ liệu, các nhà nghiên cứu sử dụng các tập dữ liệu phổ biến cho các mục đích khác nhau. Ví dụ, các tập dữ liệu về an ninh mạng như NSL-KDD, UNSW-NB15 , ISCX'12 , CIC-DDoS2019, Bot-IoT , vv., tập dữ liệu điện thoại thông minh như nhật ký cuộc gọi, nhật ký tin nhắn SMS , nhật ký sử dụng ứng dụng di động, nhật ký thông báo điện thoại di động , vv., dữ liệu IoT, dữ liệu nông nghiệp và thương mại, dữ liệu sức khỏe như bệnh tim, bệnh tiểu đường, COVID-19, vv., và nhiều loại dữ liệu khác trong các lĩnh vực ứng dụng khác nhau. Dữ liệu có thể thuộc các loại khác nhau được thảo luận ở trên, và điều này có thể thay đổi từ ứng dụng này sang ứng dụng khác trong thế giới thực. Để phân tích dữ liệu như vậy trong một lĩnh vực vấn đề cụ thể và để trích xuất cái nhìn hoặc kiến thức hữu ích từ dữ liệu để xây dựng các ứng dụng thông minh thực tế, các loại kỹ thuật học máy khác nhau có thể được sử dụng theo khả năng học của chúng, được thảo luận trong phần sau.

3.3.2. Các loại kỹ thuật học máy.

- Các thuật toán Học máy chủ yếu được chia thành bốn loại: Học có giám sát, Học không giám sát, Học bán giám sát và Học tăng cường , như được hiển thị trong Hình 12. Dưới đây, chúng tôi sẽ tóm tắt mỗi loại kỹ thuật học với phạm vi áp dụng của chúng để giải quyết các vấn đề thực tế.

+ Học Có Giám Sát: Trong học có giám sát, mô hình học từ dữ liệu đào tạo có sẵn với các cặp dữ liệu đầu vào và đầu ra tương ứng. Mục tiêu là xây dựng một mô hình có khả năng dự đoán đầu ra cho các dữ liệu mới. Ví dụ, dự đoán nhãn lớp hoặc cảm xúc của một đoạn văn bản, như một tweet hoặc một đánh giá sản phẩm, tức là phân loại văn bản, là một ví dụ về học có giám sát.

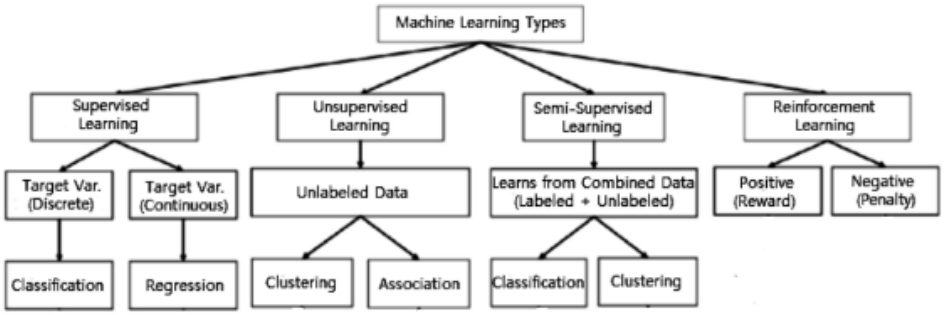
+ Học Không Giám Sát: Học không giám sát tập trung vào việc phân loại, phân cụm và hiểu cấu trúc của dữ liệu mà không cần biết đầu ra mong muốn trước. Điều này giúp phát hiện ra thông tin tiềm ẩn trong dữ liệu mà chúng ta chưa biết trước.

+ Học bán giám sát (Semi-supervised Learning) là một phương pháp học máy kết

hợp giữa học có giám sát và học không giám sát. Trong lĩnh vực này, chúng ta sử dụng một tập dữ liệu lớn, trong đó chỉ một phần nhỏ các ví dụ có nhãn (đầu ra mong muốn), còn lại không có nhãn. Mục tiêu là sử dụng thông tin từ cả dữ liệu có nhãn và dữ liệu không nhãn để cải thiện hiệu suất dự đoán.

+ Học Tăng Cường: Trong học tăng cường, mô hình tương tác với môi trường và học từ các phản hồi sau mỗi hành động. Mục tiêu là tối ưu hóa chính sách để đạt được phần thưởng lớn nhất trong môi trường cho trước.

- Do đó, để xây dựng các mô hình hiệu quả trong các lĩnh vực ứng dụng khác nhau, các kỹ thuật học máy khác nhau có thể đóng vai trò quan trọng tùy thuộc vào khả năng học của chúng, phụ thuộc vào tính chất của dữ liệu đã được thảo luận trước đó và mục tiêu đạt được. Trong Bảng 1, chúng tôi tóm tắt các kỹ thuật học máy khác nhau với các ví dụ. Dưới đây, chúng tôi cung cấp một cái nhìn toàn diện về các thuật toán học máy có thể được áp dụng để nâng cao thông minh và khả năng của một ứng dụng dựa trên dữ liệu.



Hình 12. Các loại kỹ thuật học máy khác nhau.

Loại học máy	Mô tả	Ví dụ
Học có giám sát	Sừ dụng dữ liệu được gắn nhãn để tạo mô hình học máy. Mô hình học cách thực hiện một nhiệm vụ cụ thể, chẳng hạn như phân loiạ các hồi quy	Phân loại email là spam hay không spam, dự đoán giá

		nhà dựa trên các đặc điểm nhận dạng đối tượng,…vv
Học không giám sát	Sử dụng dữ liệu chưa được gắn nhãn để đào tạo mô hình học máy. Mô hình học cách tìm các cấu trúc hoặc patterns trong dữ liệu.	Tìm nhóm khách hàng có đặc điểm tương tự nhau, phân cluster các văn bản,..vv
Học bán giám sát	Sử dụng cả dữ liệu được gắn nhãn và dữ liệu chưa được gắn nhãn để đào tạo mô hình học máy. Mô hình học cách thực hiện một nhiệm vụ cụ thể, nhưng không cần phải có nhãn cho tất cả các dữ liệu.	Phân loại hình ảnh các đối tượng khách nhau, nhưng chỉ có một số ít hình ảnh được gắn nhãn,..
Học tăng cường	Sử dụng thử nghiệm và sai sót để học cách thực hiện một	Một robot học

	nhiệm vụ. Mô hình được thưởng khi thực hiện đúng nhiệm vụ và bị phạt khi thực hiện sai.	cách chơi một trò chơi điện tử, một chương trình học cách lái xe tự động,…

Bảng 1. Các loại kỹ thuuạt học máy khác nhau cùng với ví dụ.

3.4. Nhiệm vụ và thuật toán học máy.

- Trong phần này, chúng ta sẽ thảo luận về các thuật toán máy học khác nhau bao gồm phân loại, hồi quy, gom cụm dữ liệu, học luật kết hợp, kỹ thuật kỹ thuật đặc trưng để giảm chiều dữ liệu, cũng như các phương pháp học sâu. Một cấu trúc tổng quát của một mô hình dự đoán dựa trên máy học đã được hiển thị trong Hình 13, trong đó mô hình được huấn luyện từ dữ liệu lịch sử trong giai đoạn 1 và kết quả được tạo ra trong giai đoạn 2 cho dữ liệu kiểm tra mới.

3.4.1. Phân tích phân loại.

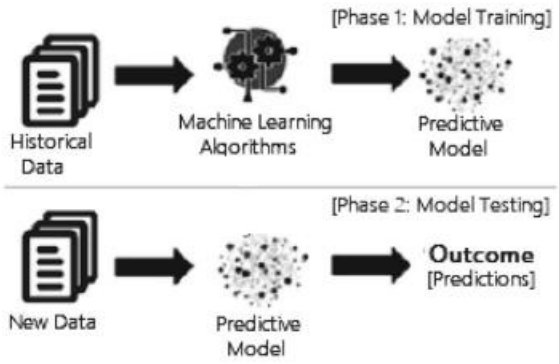
- Phân loại (Classification) được coi là một phương pháp học máy theo hướng dẫn giám sát, cũng liên quan đến mô hình dự đoán, trong đó một nhãn lớp được dự đoán cho một ví dụ đã cho . Toán học, nó ánh xạ một hàm (f) từ biến đầu vào (X) đến biến đầu ra (Y) như mục tiêu, nhãn hoặc các danh mục. Để dự đoán lớp của các điểm dữ liệu đã cho, nó có thể được thực hiện trên dữ liệu có cấu trúc hoặc không có cấu trúc. Ví dụ, việc phát hiện thư rác như "thư rác" và "không phải thư rác" trong dịch vụ email có thể là một vấn đề phân loại. Dưới đây, chúng tôi tóm tắt các vấn đề phân loại phổ biến.

+ Phân loại nhị phân (Binary Classification) là một loại bài toán trong học máy, trong đó mục tiêu là dự đoán một đối tượng, mẫu, hoặc dữ liệu mới thuộc vào một trong hai lớp hoặc nhãn khác nhau. Mục tiêu của phân loại nhị phân là tạo ra một mô hình dự đoán chính xác những điều kiện tương ứng với từng lớp.

+ Phân loại đa lớp (Multiclass Classification) là một loại bài toán trong học máy,

trong đó mục tiêu là phân loại một đối tượng, mẫu hoặc dữ liệu mới vào một trong nhiều lớp hoặc nhãn có sẵn. Khác với phân loại nhị phân chỉ có hai lớp, phân loại đa lớp có thể có ba lớp trở lên.

+ Phân loại đa nhãn (Multi-label Classification) là một loại bài toán trong học máy, trong đó mỗi mẫu hoặc đối tượng có thể thuộc vào một hoặc nhiều nhãn cùng một lúc. Điều này khác với phân loại đa lớp, trong đó mỗi mẫu chỉ có thể thuộc vào một lớp duy nhất.



Hình 13. Một cấu trúc chung của một dự báo dựa trên học máy mô hình xem xét cả giai đoạn đào tạo và thử nghiệm.

- Nhiều thuật toán phân loại đã được đề xuất trong văn bản học máy và khoa học dữ liệu. Dưới đây, chúng tôi tóm tắt các phương pháp phổ biến và phổ biến nhất được sử dụng rộng rãi trong nhiều lĩnh vực ứng dụng khác nhau.

+ Naive Bayes (NB): Thuật toán Naive Bayes dựa trên định lý Bayes với giả định về sự độc lập giữa mỗi cặp đặc trưng. Nó hoạt động tốt và có thể được sử dụng cho cả các loại nhị phân và đa lớp trong nhiều tình huống thực tế, như phân loại văn bản hoặc văn bản, lọc thư rác, v.v. Để phân loại hiệu quả các trường hợp nhiễu trong dữ liệu và xây dựng một mô hình dự đoán mạnh mẽ, có thể sử dụng bộ phân loại NB. Lợi ích chính là so với các phương pháp phức tạp hơn, nó cần một lượng nhỏ dữ liệu huấn luyện để ước tính các tham số cần thiết và nhanh chóng. Tuy nhiên, hiệu suất của nó có thể bị ảnh hưởng do giả định mạnh về tính độc lập của các đặc trưng. Gaussian, Multinomial, Complement, Bernoulli và Categorical là các biến thể phổ biến của bộ phân loại NB.

+ Phân tích Lực lượng Tuyến tính (Linear Discriminant Analysis - LDA) là một phương pháp trong thống kê và học máy được sử dụng để giảm chiều dữ liệu và tìm ra các thành phần tạo nên sự khác biệt giữa các lớp hoặc nhãn khác nhau. Mục tiêu chính của LDA là tìm ra các vectơ hệ số sao cho sự tách biệt giữa các lớp là lớn nhất.

+ Hồi quy Logistic (Logistic Regression - LR) là một trong những thuật toán quan trọng trong lĩnh vực học máy, được sử dụng chủ yếu cho các bài toán phân loại. Mặc dù có tên là "hồi quy," nhưng LR thực chất là một thuật toán phân loại, không phải là hồi quy dự đoán giá trị liên tục..

+ K-nearest neighbors (KNN) là một thuật toán học máy trong lĩnh vực phân loại và dự đoán. KNN là một trong những thuật toán đơn giản nhưng mạnh mẽ và dễ hiểu. Nó được sử dụng cho cả bài toán phân loại và dự đoán giá trị liên tục dựa trên các ví dụ gần nhất trong tập dữ liệu.

$$g(z) = \frac{1}{1 + \exp(-z)}. \qquad (1)$$

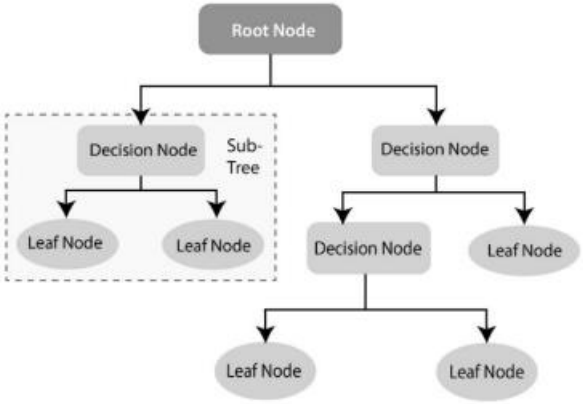
+ Máy Vector Hỗ Trợ (Support Vector Machine - SVM) là một thuật toán học máy phân loại và hồi quy. Nó được sử dụng rộng rãi trong nhiều ứng dụng khác nhau như phân loại ảnh, nhận dạng chữ viết tay, dự đoán giá chứng khoán, và nhiều lĩnh vực khác. SVM hoạt động dựa trên nguyên tắc tìm ra một đường ranh giới (hyperplane) tốt nhất để phân tách hai lớp dữ liệu.

+ Cây quyết định (DT): Cây quyết định (DT) là một phương pháp học có giám sát không tham số nổi tiếng. Phương pháp học DT được sử dụng cho cả nhiệm vụ phân loại và hồi quy . ID3, C4.5, và CART là những thuật toán DT nổi tiếng. Ngoài ra, BehavDT và IntrudTree được đề xuất gần đây bởi Sarker và đồng nghiệp là hiệu quả trong các lĩnh vực ứng dụng liên quan, chẳng hạn như phân tích hành vi người dùng và phân tích an ninh mạng. Bằng cách đi từ gốc cây đến một số nút lá, như được hiển thị trong Hình 14, DT phân loại các trường hợp. Các trường hợp được phân loại bằng cách kiểm tra thuộc tính được xác định bởi nút đó, bắt đầu từ gốc của cây, và sau đó di chuyển xuống nhánh cây tương ứng với giá trị thuộc tính. Đối với việc chia nhánh, các tiêu chí phổ biến nhất là "gini" cho độ không thuần khiết Gini và "entropy" cho lợi ích thông tin có thể được biểu

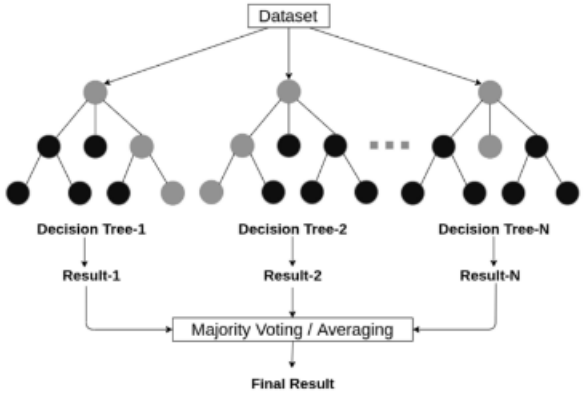
diễn toán học như.

$$\text{Entropy} : H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad (2)$$

$$\text{Gini}(E) = 1 - \sum_{i=1}^{c} p_i^2. \quad (3)$$



Hình 14. Ví dụ về cấu trúc cây quyết định.



Hình 15. Một ví dụ về cấu trúc rừng ngẫu nhiên xem xét nhiều cây quyết định.

+ Random Forest (RF) là một thuật toán học máy được sử dụng cho các nhiệm vụ phân loại, hồi quy và các tác vụ khác liên quan đến dữ liệu. Nó là một dạng của Ensemble Learning, tức là kết hợp nhiều mô hình đơn lẻ để tạo ra một mô hình mạnh hơn và ổn định hơn.

- Để xây dựng một loạt cây quyết định với sự biến đổi được kiểm soát, phương pháp này kết hợp phương pháp bootstrap aggregation (bagging) và lựa chọn đặc trưng ngẫu

nhiên. Nó có thể thích ứng với cả bài toán phân loại và hồi quy, và phù hợp cho cả giá trị rời rạc và liên tục.

+ AdaBoost (Adaptive Boosting) là một thuật toán học máy thuộc loại Ensemble Learning, được sử dụng chủ yếu trong các tác vụ phân loại. Nó tập trung vào việc kết hợp nhiều mô hình yếu thành một mô hình mạnh hơn. Ý tưởng chính của AdaBoost là tập trung vào việc cải thiện hiệu suất bằng cách tập trung vào các điểm dữ liệu được phân loại sai trong các lần huấn luyện sau.

+ Extreme Gradient Boosting (XGBoost) là một thuật toán học máy nổi tiếng thuộc loại Gradient Boosting, được sử dụng cho các tác vụ phân loại, hồi quy và xếp hạng. XGBoost được phát triển bởi Tianqi Chen và phải được xem xét là một phiên bản cải tiến của thuật toán Gradient Boosting.

+ Stochastic gradient descent (SGD): Stochastic gradient descent (SGD) là một phương pháp lặp để tối ưu hóa một hàm mục tiêu với tính chất mịn phù hợp, trong đó từ "stochastic" chỉ sự ngẫu nhiên. Điều này giúp giảm gánh nặng tính toán, đặc biệt là trong các bài toán tối ưu hóa có số chiều cao, cho phép các lặp nhanh hơn đổi lấy tốc độ hội tụ thấp hơn. Đạo hàm là độ dốc của một hàm tính toán mức độ thay đổi của một biến phụ thuộc đối với sự thay đổi của một biến khác. Toán học, Gradient Descent là một hàm lồi có đầu ra là đạo hàm riêng của một tập các tham số đầu vào. Giả sử, α là tốc độ học, và Ji là chi phí của ví dụ huấn luyện thứ i, thì Công thức (4) biểu thị phương pháp cập nhật trọng số gradient descent ngẫu nhiên ở lần lặp thứ j. Trong học máy quy mô lớn và thưa thớt, SGD đã được áp dụng thành công cho các vấn đề thường gặp trong phân loại văn bản và xử lý ngôn ngữ tự nhiên. Tuy nhiên, SGD nhạy cảm với việc tỷ lệ đặc trưng và cần một loạt các siêu tham số, chẳng hạn như tham số điều chuẩn và số lần lặp.

$$w_j := w_j - \alpha \frac{\partial J_i}{\partial w_j}. \qquad (4)$$

+ Phân loại dựa trên quy tắc: Thuật ngữ phân loại dựa trên quy tắc có thể được sử dụng để chỉ bất kỳ hệ thống phân loại nào sử dụng quy tắc IF-THEN để dự đoán lớp. Có nhiều thuật toán phân loại như Zero-R , One-R , cây quyết định , DTNB , Ripple Down Rule learner (RIDOR) , Repeated Incremental Pruning to Produce Error Reduction

(RIPPER) có khả năng tạo ra quy tắc. Cây quyết định là một trong những thuật toán phân loại dựa trên quy tắc phổ biến nhất trong các kỹ thuật này vì nó có nhiều lợi ích, như dễ hiểu hơn, khả năng xử lý dữ liệu có số chiều cao, đơn giản và nhanh chóng, độ chính xác tốt và khả năng tạo ra quy tắc dễ hiểu cho việc phân loại mà con người có thể hiểu được . Các quy tắc dựa trên cây quyết định cũng cung cấp độ chính xác đáng kể trong mô hình dự đoán cho các trường hợp kiểm tra chưa được nhìn thấy. Vì các quy tắc dễ hiểu, các bộ phân loại dựa trên quy tắc này thường được sử dụng để tạo ra các mô hình mô tả có thể mô tả một hệ thống bao gồm các thực thể và mối quan hệ của chúng.

3.4.2. Phân tích hồi quy.

- Phân tích hồi quy bao gồm một số phương pháp học máy cho phép dự đoán một biến kết quả liên tục (y) dựa trên giá trị của một hoặc nhiều biến dự đoán (x) . Sự khác biệt quan trọng nhất giữa phân loại và hồi quy là phân loại dự đoán các nhãn lớp riêng biệt, trong khi hồi quy dự đoán một lượng liên tục. Hình 6 cho thấy một ví dụ về sự khác biệt giữa phân loại và mô hình hồi quy. Thường có sự chồng chéo giữa hai loại thuật toán học máy này. Mô hình hồi quy hiện được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm dự báo tài chính, ước tính chi phí, phân tích xu hướng, tiếp thị, ước tính chuỗi thời gian, mô hình phản ứng thuốc, và nhiều lĩnh vực khác. Một số loại thuật toán hồi quy phổ biến bao gồm hồi quy tuyến tính, hồi quy đa thức, hồi quy lasso và ridge, v.v., được giải thích ngắn gọn như sau:

+ Hồi quy tuyến tính đơn và đa biến: Đây là một trong những kỹ thuật mô hình học máy phổ biến nhất cũng như là một kỹ thuật hồi quy nổi tiếng. Trong kỹ thuật này, biến phụ thuộc là liên tục, biến độc lập có thể là liên tục hoặc rời rạc, và dạng của đường hồi quy là tuyến tính. Hồi quy tuyến tính tạo ra một mối quan hệ giữa biến phụ thuộc (Y) và một hoặc nhiều biến độc lập (X) (còn được gọi là đường hồi quy) bằng cách sử dụng đường thẳng phù hợp nhất . Nó được xác định bởi các phương trình sau:

$$y = a + bx + e \quad (5)$$

$$y = a + b_1x_1 + b_2x_2 + \cdots + b_nx_n + e, \quad (6)$$

- Phương trình (5) và (6) được sử dụng để dự đoán giá trị của biến mục tiêu dựa trên các biến dự đoán đã cho. Hồi quy tuyến tính đa biến là một phương pháp mở rộng của hồi

quy tuyến tính đơn biến, cho phép hai hoặc nhiều biến dự đoán được sử dụng để mô hình một biến phản hồi y như một hàm tuyến tính được định nghĩa trong phương trình (6), trong khi hồi quy tuyến tính đơn biến chỉ có 1 biến độc lập được định nghĩa trong phương trình (5).

+ Hồi quy đa thức: Hồi quy đa thức là một phương pháp phân tích hồi quy trong đó mối quan hệ giữa biến độc lập x và biến phụ thuộc y không phải là tuyến tính, mà là một đa thức bậc n trong x . Phương trình cho hồi quy đa thức cũng được dẫn xuất từ phương trình hồi quy tuyến tính (hồi quy đa thức bậc 1), được định nghĩa như sau:

$$y = b_0 + b_1x + b_2x^2 + b_3x^3 + \cdots + b_nx^n + e. \qquad (7)$$

- Ở đây, y là đầu ra dự đoán/mục tiêu, b0, b1, ... bn là các hệ số hồi quy, x là một biến độc lập/đầu vào. Đơn giản mà nói, chúng ta có thể nói rằng nếu dữ liệu không được phân phối theo đường thẳng, mà thay vào đó là đa thức bậc n, thì chúng ta sử dụng hồi quy đa thức để có được đầu ra mong muốn.

+ LASSO và ridge regression: LASSO và Ridge regression được biết đến như là các kỹ thuật mạnh mẽ thường được sử dụng để xây dựng các mô hình học trong trường hợp có một số lượng lớn đặc trưng, nhờ khả năng ngăn chặn việc quá khớp và giảm độ phức tạp của mô hình. Mô hình hồi quy LASSO (least absolute shrinkage and selection operator) sử dụng kỹ thuật chính quy hóa L1 , sử dụng thu nhỏ, làm trừng phạt "giá trị tuyệt đối của hệ số" (L1 penalty). Kết quả là, LASSO có vẻ như làm cho các hệ số trở thành giá trị tuyệt đối bằng không. Do đó, hồi quy LASSO nhằm tìm tập con của các biến dự đoán mà giảm thiểu sai số dự đoán cho một biến phản hồi định lượng. Trong khi đó, ridge regression sử dụng chính quy hóa L2 , là "giá trị bình phương của hệ số" (L2 penalty). Do đó, ridge regression buộc các trọng số phải nhỏ nhưng không bao giờ đặt giá trị hệ số bằng không, và tạo ra một giải pháp không thưa thớt. Nhìn chung, hồi quy LASSO hữu ích để thu được một tập con của các biến dự đoán bằng cách loại bỏ các đặc trưng ít quan trọng, và ridge regression hữu ích khi tập dữ liệu có "đa tuyến" (multicollinearity), tức là các biến dự đoán có mối tương quan với các biến dự đoán khác.

3.4.3. Phân tích Cluster.

- Phân tích cụm, còn được gọi là phân cụm, là một kỹ thuật học máy không giám sát để xác định và nhóm các điểm dữ liệu liên quan trong các tập dữ liệu lớn mà không quan tâm đến kết quả cụ thể. Nó nhóm một tập hợp các đối tượng sao cho các đối tượng trong cùng một nhóm, gọi là một cụm, có một mức độ tương đồng lớn hơn so với các đối tượng trong các nhóm khác. Phân tích cụm thường được sử dụng như một kỹ thuật phân tích dữ liệu để khám phá các xu hướng hoặc mô hình thú vị trong dữ liệu, ví dụ như nhóm người tiêu dùng dựa trên hành vi của họ. Trong một loạt các lĩnh vực ứng dụng rộng, chẳng hạn như an ninh mạng, thương mại điện tử, xử lý dữ liệu di động, phân tích sức khỏe, mô hình người dùng và phân tích hành vi, phân cụm có thể được sử dụng. Trong phần sau, chúng tôi sẽ tóm tắt và trình bày ngắn gọn về các phương pháp phân cụm khác nhau.

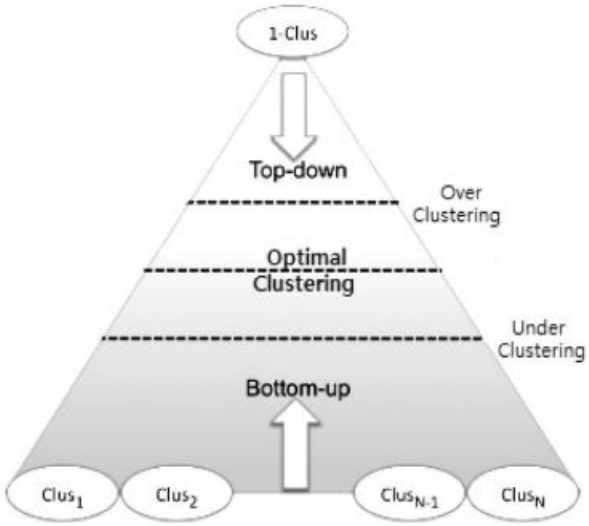
+ Phương pháp phân vùng: Dựa trên các đặc trưng và sự tương đồng trong dữ liệu, phương pháp gom cụm này phân loại dữ liệu thành nhiều nhóm hoặc cụm. Thông thường, các nhà khoa học dữ liệu hoặc nhà phân tích xác định số lượng cụm một cách động hoặc tĩnh tùy thuộc vào tính chất của ứng dụng mục tiêu, để tạo ra các phương pháp gom cụm. Các thuật toán gom cụm phổ biến nhất dựa trên phương pháp phân vùng là K-means, K-Medoids, CLARA, v.v.

+ Phương pháp dựa trên mật độ: Để xác định các nhóm hoặc cụm riêng biệt, phương pháp này sử dụng khái niệm rằng một cụm trong không gian dữ liệu là một vùng liền kề có mật độ điểm cao và cách biệt với các cụm khác bởi các vùng liền kề có mật độ điểm thấp. Các điểm không thuộc về một cụm được coi là nhiễu. Các thuật toán phân cụm thông thường dựa trên mật độ bao gồm DBSCAN, OPTICS, v.v.

-Phương pháp dựa trên mật độ thường gặp khó khăn khi xử lý các cụm có mật độ tương tự và dữ liệu có số chiều cao.

+ Phương pháp dựa trên phân cấp: Phân cụm phân cấp thường cố gắng xây dựng một cấu trúc cây của các cụm. Các chiến lược phân cụm phân cấp thường thuộc hai loại: (i) Phương pháp gom nhóm - một phương pháp "từ dưới lên" trong đó mỗi quan sát bắt đầu trong cụm của nó và các cặp cụm được kết hợp thành một cụm khi di chuyển lên cây phân cấp, và (ii) Phương pháp chia nhỏ - một phương pháp "từ trên xuống" trong đó tất cả các quan sát bắt đầu trong một cụm và các phân chia được thực hiện đệ quy khi di chuyển

xuống cây phân cấp, như được thể hiện trong Hình 16. Kỹ thuật BOTS mà chúng tôi đã đề xuất trước đây, Sarker et al., là một ví dụ về thuật toán phân cụm phân cấp, đặc biệt là phương pháp "từ dưới lên".



Hình 16. Giải thích bằng đồ họa về cụm phân cấp được sử dụng rộng rãi kỹ thuật tering(từ dưới lên và từ trên xuống)

+ Phương pháp dựa trên lưới: Để xử lý các tập dữ liệu lớn, phương pháp phân cụm dựa trên lưới rất phù hợp. Để thu được các cụm, nguyên tắc đầu tiên là tóm tắt tập dữ liệu bằng một biểu diễn lưới và sau đó kết hợp các ô lưới. STING, CLIQUE, v.v. là các thuật toán tiêu chuẩn của phương pháp phân cụm dựa trên lưới.

+ Phương pháp dựa trên mô hình: Có chủ yếu hai loại thuật toán phân cụm dựa trên mô hình: một loại sử dụng học thống kê và một loại dựa trên phương pháp học mạng neural. Ví dụ, GMM là một ví dụ về phương pháp học thống kê, và SOM là một ví dụ về phương pháp học mạng neural.

– Phương pháp dựa trên ràng buộc: Clustering dựa trên ràng buộc là một phương pháp bán giám sát để phân cụm dữ liệu sử dụng ràng buộc để tích hợp kiến thức lĩnh vực. Các ràng buộc liên quan đến ứng dụng hoặc người dùng được tích hợp để thực hiện việc phân cụm. Các thuật toán điển hình của loại phân cụm này là COP K-means , CMWK-Means, v.v.

– Có nhiều thuật toán phân cụm đã được đề xuất có khả năng nhóm dữ liệu trong lĩnh vực học máy và khoa học dữ liệu. Dưới đây, chúng tôi tóm tắt những phương pháp phổ biến được sử dụng rộng rãi trong các lĩnh vực ứng dụng khác nhau.

+ Phân cụm K-means là một thuật toán trong lĩnh vực học không giám sát được sử dụng để phân nhóm các điểm dữ liệu thành các cụm dựa trên sự tương đồng giữa chúng. Mục tiêu của thuật toán là tìm cách tối ưu hóa vị trí của các điểm trung tâm cụm để giảm thiểu tổng sai số bình phương giữa các điểm dữ liệu và điểm trung tâm của cụm mà chúng thuộc về.

+ Phân cụm Mean Shift là một phương pháp phân cụm không giám sát được sử dụng để phân nhóm các điểm dữ liệu trong không gian đa chiều dựa trên cơ chế dịch chuyển trung bình. Điểm đặc biệt của phân cụm Mean Shift là khả năng tìm ra số lượng cụm mà không cần biết trước, và có khả năng xử lý các cụm có hình dáng và kích thước không đều.

+ DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán phân cụm không giám sát sử dụng mật độ không gian để phân nhóm các điểm dữ liệu vào các cụm có mật độ tương tự nhau. Điểm đặc biệt của DBSCAN là khả năng xác định được các cụm có hình dáng và kích thước không đều, và có khả năng xử lý dữ liệu chứa nhiễu.

+ Phân cụm GMM (Gaussian Mixture Model) là một phương pháp phân cụm không giám sát dựa trên mô hình hỗn hợp các phân phối Gaussian. GMM giả định rằng dữ liệu trong mỗi cụm được sinh ra từ một phân phối Gaussian riêng biệt và tất cả các cụm được kết hợp lại từ các phân phối Gaussian này.

+ Phân cụm phân cấp gom nhóm (Hierarchical Clustering) là một phương pháp phân cụm không giám sát được sử dụng để tạo một cây phân cấp của các cụm dữ liệu. Trong phương pháp này, mỗi điểm dữ liệu ban đầu được coi là một cụm độc lập, sau đó các cụm này được kết hợp dần dựa trên sự tương đồng của chúng để tạo ra một cây phân cấp.

3.5. Giảm kích thước và học tính năng.

-Trong machine learning và data science, việc xử lý dữ liệu có số chiều cao là một nhiệm vụ khó khăn đối với cả nhà nghiên cứu và nhà phát triển ứng dụng. Do đó, việc

giảm chiều dữ liệu, một kỹ thuật học không giám sát, là rất quan trọng vì nó dẫn đến sự hiểu rõ tốt hơn từ con người, giảm chi phí tính toán và tránh tình trạng quá khớp và dư thừa bằng cách đơn giản hóa mô hình. Cả quá trình lựa chọn đặc trưng và trích xuất đặc trưng đều có thể được sử dụng để giảm chiều dữ liệu. Sự khác biệt chính giữa lựa chọn và trích xuất đặc trưng là "lựa chọn đặc trưng" giữ lại một tập con của các đặc trưng ban đầu, trong khi "trích xuất đặc trưng" tạo ra các đặc trưng hoàn toàn mới. Dưới đây, chúng ta sẽ tóm tắt ngắn gọn về các kỹ thuật này.

+ Lựa chọn đặc trưng: Lựa chọn đặc trưng, còn được gọi là lựa chọn biến hoặc thuộc tính trong dữ liệu, là quá trình chọn một tập con các đặc trưng duy nhất (biến, dự đoán) để sử dụng trong việc xây dựng mô hình học máy và khoa học dữ liệu. Nó giảm độ phức tạp của mô hình bằng cách loại bỏ các đặc trưng không liên quan hoặc ít quan trọng và cho phép huấn luyện nhanh hơn các thuật toán học máy. Một tập con đúng và tối ưu của các đặc trưng được lựa chọn trong một lĩnh vực vấn đề có khả năng giảm thiểu vấn đề quá khớp bằng cách đơn giản hóa và tổng quát hóa mô hình cũng như tăng độ chính xác của mô hình. Do đó, "lựa chọn đặc trưng" được coi là một trong những khái niệm chính trong machine learning ảnh hưởng đáng kể đến hiệu quả và hiệu suất của mô hình học máy mục tiêu. Một số kỹ thuật phổ biến để lựa chọn đặc trưng bao gồm kiểm định chi bình phương, kiểm định phân tích phương sai (ANOVA), hệ số tương quan Pearson, loại bỏ đặc trưng đệ quy.

+ Trích xuất đặc trưng: Trong một mô hình hoặc hệ thống dựa trên machine learning, các kỹ thuật trích xuất đặc trưng thường cung cấp một hiểu biết tốt hơn về dữ liệu, cách cải thiện độ chính xác dự đoán và giảm chi phí tính toán hoặc thời gian huấn luyện. Mục tiêu của "trích xuất đặc trưng" là giảm số lượng đặc trưng trong một tập dữ liệu bằng cách tạo ra các đặc trưng mới từ các đặc trưng hiện có và sau đó loại bỏ các đặc trưng gốc. Đa số thông tin được tìm thấy trong tập hợp đặc trưng gốc có thể được tóm tắt bằng cách sử dụng tập hợp đặc trưng mới đã giảm này. Ví dụ, phân tích thành phần chính (PCA) thường được sử dụng như một kỹ thuật giảm chiều dữ liệu để trích xuất không gian chiều thấp tạo ra các thành phần mới từ các đặc trưng hiện có trong tập dữ liệu.

- Có nhiều thuật toán đã được đề xuất để giảm số chiều dữ liệu trong các tài liệu về máy học và khoa học dữ liệu. Dưới đây, chúng tôi tóm tắt các phương pháp phổ biến

được sử dụng rộng rãi trong các lĩnh vực ứng dụng khác nhau.

+ Ngưỡng phương sai: Một phương pháp đơn giản để lựa chọn đặc trưng là ngưỡng phương sai . Phương pháp này loại bỏ tất cả các đặc trưng có phương sai thấp, tức là tất cả các đặc trưng có phương sai không vượt quá ngưỡng. Nó loại bỏ tất cả các đặc trưng có phương sai bằng không theo mặc định, tức là các đặc trưng có cùng giá trị trong tất cả các mẫu. Thuật toán lựa chọn đặc trưng này chỉ xem xét các đặc trưng (X), không phải đầu ra (y) cần thiết, và do đó có thể được sử dụng cho việc học không giám sát.

+ Tương quan Pearson: Tương quan Pearson là một phương pháp khác để hiểu mối quan hệ giữa một đặc trưng và biến phản hồi và có thể được sử dụng cho việc lựa chọn đặc trưng. Phương pháp này cũng được sử dụng để tìm hiểu mối liên hệ giữa các đặc trưng trong một tập dữ liệu. Giá trị kết quả nằm trong khoảng, trong đó -1 có nghĩa là tương quan âm hoàn hảo, +1 có nghĩa là tương quan dương hoàn hảo và 0 có nghĩa là hai biến không có tương quan tuyến tính. Nếu hai biến ngẫu nhiên đại diện cho X và Y, thì hệ số tương quan giữa X và Y được xác định như sau .

$$r(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \qquad (8)$$

+ ANOVA (Analysis of Variance) là một phương pháp thống kê được sử dụng để kiểm tra sự khác biệt giữa các trung bình của ba hoặc nhiều nhóm dữ liệu. ANOVA kiểm tra xem có sự khác biệt có ý nghĩa nào đó giữa các nhóm hay không, thay vì chỉ kiểm tra sự khác biệt giữa hai nhóm như trong t-test.

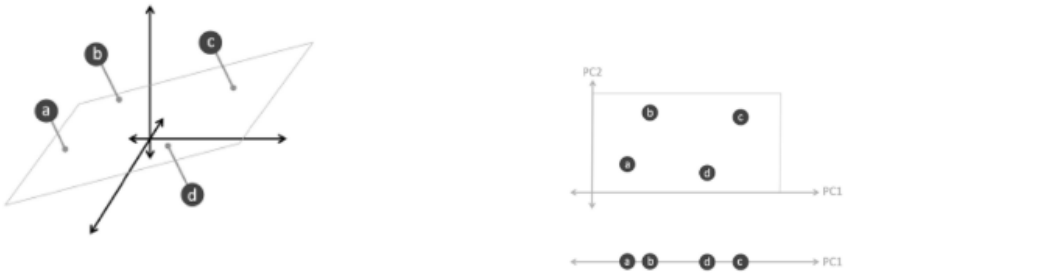
+ X-square (X2): Thống kê X-square (X2) là một ước lượng về sự khác biệt giữa các tác động của một loạt các sự kiện hoặc biến số được quan sát và tần số dự kiến. Độ lớn của sự khác biệt giữa các giá trị thực tế và quan sát, số độ tự do và kích thước mẫu phụ thuộc vào X2. Thống kê X-square thường được sử dụng để kiểm tra mối quan hệ giữa các biến phân loại. Nếu Oi đại diện cho giá trị quan sát và Ei đại diện cho giá trị dự kiến, thì công thức tính X2 như sau.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}. \qquad (9)$$

+ Recursive Feature Elimination (RFE) là một kỹ thuật lựa chọn đặc trưng trong quá trình xây dựng mô hình học máy. Nó giúp tối ưu hóa tập dữ liệu đầu vào bằng cách loại bỏ lần lượt các đặc trưng không quan trọng, dựa trên hiệu suất của mô hình trong quá trình loại bỏ. Mục tiêu của RFE là tạo ra tập dữ liệu đặc trưng có thể cung cấp hiệu suất tốt nhất cho mô hình học máy.

+ Lựa chọn dựa trên mô hình: Để giảm số chiều của dữ liệu, có thể sử dụng các mô hình tuyến tính được trừ phạt bằng việc sử dụng đặc trưng L1 regularization. Hồi quy Lasso (Lasso regression) là một loại hồi quy tuyến tính có tính chất thu hẹp một số hệ số về zero . Do đó, đặc trưng đó có thể được loại bỏ khỏi mô hình. Do đó, phương pháp hồi quy Lasso được trừ phạt thường được sử dụng trong học máy để lựa chọn tập con các biến. Extra Trees Classifier là một ví dụ về một bộ ước lượng dựa trên cây có thể được sử dụng để tính toán tính quan trọng dựa trên độ tạp chất, từ đó có thể loại bỏ các đặc trưng không liên quan.

+ Phân tích thành phần chính (PCA - Principal Component Analysis) là một kỹ thuật phân tích dữ liệu thống kê được sử dụng để giảm chiều dữ liệu, tìm ra các thành phần chính quan trọng nhất của dữ liệu và biểu diễn dữ liệu dưới dạng các thành phần này. PCA giúp giảm sự phức tạp của dữ liệu, loại bỏ đặc trưng không cần thiết, và tạo ra một biểu diễn mới dựa trên các chiều thành phần chính.



(a) An example of the original features in a 3D space. (b) Principal components in 2D and 1D space.

Hình 17. Một ví dụ về phân tích thành phần chúnh (PCA) và tạo ra các thành phần chính PC1 và PC2 trong không gian có chiều khác nhau.

3.6. Học luật kết hợp.

- Học luật kết hợp (Ensemble Learning) là một kỹ thuật trong lĩnh vực học máy, mà ở đó nhiều mô hình học máy đơn giản được kết hợp lại để tạo thành một mô hình mạnh hơn. Mục tiêu của học luật kết hợp là cải thiện hiệu suất dự đoán của mô hình bằng cách tận dụng sự đa dạng và sự thông tin của các mô hình nhỏ.

- Trong văn bản khai thác dữ liệu, đã được đề xuất nhiều phương pháp học luật kết hợp, chẳng hạn như phụ thuộc logic , dựa trên mẫu phổ biến và dựa trên cây. Các thuật toán học luật kết hợp phổ biến nhất được tóm tắt dưới đây.

+ AIS (Artificial Immune System) và SETM (Self-Evolving Transformational Machine) là hai thuật toán trong lĩnh vực Trí tuệ nhân tạo (AI) có liên quan đến hệ thống tự tổ chức và tiến hóa

+ Apriori là một thuật toán quan trọng trong lĩnh vực khai phá dữ liệu và phân tích tỷ lệ chéo. Thuật toán Apriori được sử dụng để tìm các mẫu thường xuyên trong tập dữ liệu, như các luật kết hợp giữa các mục, ví dụ như "nếu A xảy ra, thì B cũng xảy ra".

Apriori là một phương pháp tìm kiếm qua tất cả các luật có sự tương quan với tần suất xuất hiện trên ngưỡng đã định sẵn (được gọi là "tần số ngưỡng" hoặc "min-support threshold"). Thuật toán Apriori hoạt động dựa trên nguyên tắc "có cái trước mới có cái sau" (a priori property), tức là nếu một tập con không thỏa mãn ngưỡng, thì tất cả các tập con lớn hơn cũng không thỏa mãn.

+ ECLAT (Equivalence Class Transformation) là một thuật toán khai phá dữ liệu được sử dụng để tìm kiếm các mẫu thường xuyên trong tập dữ liệu. Tương tự như thuật toán Apriori, ECLAT cũng tập trung vào việc tìm các tập ứng viên thường xuyên (frequent itemsets) trong tập dữ liệu.

+ FP-Growth (Frequent Pattern Growth) là một thuật toán quan trọng trong khai phá dữ liệu được sử dụng để tìm các mẫu thường xuyên trong tập dữ liệu. Giống như Apriori và ECLAT, FP-Growth cũng tập trung vào việc tìm các tập ứng viên thường xuyên (frequent itemsets) trong dữ liệu.

Tuy nhiên, FP-Growth sử dụng một cách tiếp cận khác để tìm các tập ứng viên thường xuyên. Nó sử dụng cấu trúc dữ liệu gọi là "FP-Tree" để lưu trữ thông tin về các mẫu thường xuyên mà không cần phải tạo ra tất cả các tập ứng viên trong quá trình tìm

kiếm..

+ ABC-RuleMiner: Một phương pháp học máy dựa trên quy tắc, được đề xuất gần đây trong bài báo trước của chúng tôi, bởi Sarker và cộng sự , để khám phá các quy tắc không trùng lặp thú vị để cung cấp các dịch vụ thông minh thực tế. Thuật toán này hiệu quả xác định sự trùng lặp trong các mối quan hệ bằng cách xem xét tác động hoặc ưu tiên của các đặc trưng ngữ cảnh liên quan và khám phá một tập hợp các quy tắc kết hợp không trùng lặp. Thuật toán này trước tiên xây dựng một cây tạo ra quy tắc kết hợp (AGT), một phương pháp từ trên xuống, và sau đó trích xuất các quy tắc kết hợp thông qua việc duyệt cây. Do đó, ABC-RuleMiner mạnh mẽ hơn các phương pháp dựa trên quy tắc truyền thống về cả việc tạo ra quy tắc không trùng lặp và ra quyết định thông minh, đặc biệt trong một môi trường tính toán thông minh phù hợp với ngữ cảnh, nơi sở thích của con người hoặc người dùng được liên quan. Trong số các kỹ thuật học luật kết hợp được thảo luận ở trên, Apriori là thuật toán được sử dụng rộng rãi nhất để khám phá các quy tắc kết hợp từ một tập dữ liệu cho trước. Điểm mạnh chính của kỹ thuật học luật kết hợp là tính toàn diện của nó, vì nó tạo ra tất cả các mối quan hệ thỏa mãn các ràng buộc được chỉ định bởi người dùng, chẳng hạn như giá trị hỗ trợ tối thiểu và độ tin cậy. Phương pháp ABC-RuleMiner đã được thảo luận trước đây có thể mang lại kết quả đáng kể về việc tạo ra các quy tắc không trùng lặp và ra quyết định thông minh cho các lĩnh vực ứng dụng liên quan trong thế giới thực.

3.7. Học tăng cường

- Học tăng cường (Reinforcement Learning - RL) là một kỹ thuật học máy cho phép một tác nhân học thông qua thử và sai trong một môi trường tương tác bằng cách sử dụng thông tin từ các hành động và kinh nghiệm của nó. Khác với học có giám sát dựa trên dữ liệu mẫu hoặc ví dụ đã cho, phương pháp RL dựa trên việc tương tác với môi trường. Vấn đề được giải quyết trong học tăng cường (RL) được xác định là một Quyết định Markov (Markov Decision Process - MDP), tức là liên quan đến việc đưa ra quyết định theo trình tự. Một vấn đề RL thông thường bao gồm bốn yếu tố: Tác nhân (Agent), Môi trường (Environment), Phần thưởng (Rewards) và Chính sách (Policy).

- RL có thể được chia thành hai kỹ thuật chính là dựa trên mô hình (Model-based) và không dựa trên mô hình (Model-free). RL dựa trên mô hình là quá trình suy luận hành

vi tối ưu từ một mô hình của môi trường bằng cách thực hiện hành động và quan sát kết quả, bao gồm trạng thái tiếp theo và phần thưởng ngay lập tức. AlphaZero, AlphaGo là những ví dụ về phương pháp dựa trên mô hình. Trong khi đó, phương pháp không dựa trên mô hình không sử dụng phân phối xác suất chuyển tiếp và hàm phần thưởng liên quan đến MDP. Q-learning, Deep Q Network, Monte Carlo Control, SARSA là một số ví dụ về thuật toán không dựa trên mô hình. Mạng chính sách (policy network) là yếu tố khác biệt chính giữa học không dựa trên mô hình và học dựa trên mô hình. Dưới đây, chúng ta sẽ thảo luận về các thuật toán RL phổ biến.

+ Phương pháp Monte Carlo: Phương pháp Monte Carlo là một loại thuật toán tính toán dựa trên việc lấy mẫu ngẫu nhiên lặp lại để thu được kết quả số học. Ý tưởng cơ bản là sử dụng sự ngẫu nhiên để giải quyết các vấn đề mà cơ bản là xác định. Tối ưu hóa, tích phân số học và lấy mẫu từ phân phối xác suất là ba lớp vấn đề mà phương pháp Monte Carlo thường được sử dụng.

+ Q-learning: Q-learning là một thuật toán học tăng cường không dựa trên mô hình để học chất lượng hành vi, cho biết cho một tác nhân hành động nào được thực hiện trong điều kiện nào. Nó không cần một mô hình của môi trường (do đó được gọi là "không dựa trên mô hình"), và có thể xử lý các chuyển tiếp và phần thưởng ngẫu nhiên mà không cần sự thích nghi. 'Q' trong Q-learning thường đại diện cho chất lượng, vì thuật toán tính toán phần thưởng kỳ vọng tối đa cho một hành vi cụ thể trong một trạng thái cụ thể.

+ Deep Q-learning: Bước làm cơ bản trong Deep Q-Learning là đưa trạng thái ban đầu vào mạng nơ-ron, mạng nơ-ron sẽ trả về giá trị Q của tất cả các hành động có thể làm được. Khi chúng ta chỉ có một môi trường đơn giản để vượt qua, Q-learning hoạt động tốt. Tuy nhiên, khi số lượng trạng thái và hành động trở nên phức tạp hơn, deep learning có thể được sử dụng như một bộ xấp xỉ hàm. Học tăng cường, cùng với học có giám sát và học không giám sát, là một trong những mô hình học máy cơ bản. RL có thể được sử dụng để giải quyết nhiều vấn đề thực tế trong các lĩnh vực khác nhau như lý thuyết trò chơi, lý thuyết điều khiển, phân tích hoạt động, lý thuyết thông tin, tối ưu hóa dựa trên mô phỏng, sản xuất, quản lý chuỗi cung ứng, hệ thống đa tác nhân, trí tuệ đàn đồng, điều khiển máy bay, điều khiển chuyển động robot và nhiều lĩnh vực khác.

3.8. Mạng Nơ ron nhân tạo và học sâu.

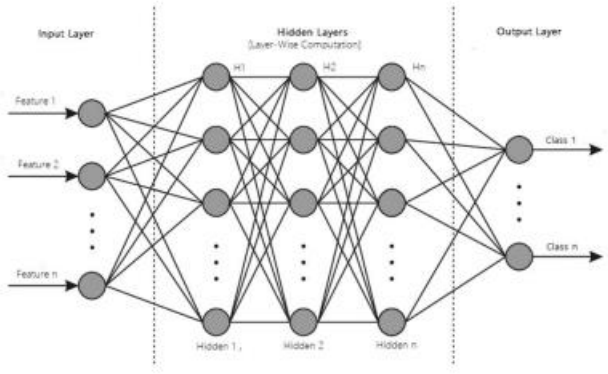
- Học sâu là một phần của gia đình rộng hơn của các phương pháp học máy dựa trên mạng thần kinh nhân tạo (ANN) với việc học biểu diễn. Học sâu cung cấp một kiến trúc tính toán bằng cách kết hợp nhiều lớp xử lý, chẳng hạn như lớp đầu vào, lớp ẩn và lớp đầu ra, để học từ dữ liệu. Lợi ích chính của học sâu so với các phương pháp học máy truyền thống là hiệu suất tốt hơn trong một số trường hợp, đặc biệt là học từ các bộ dữ liệu lớn. Hình 18 cho thấy hiệu suất tổng quát của học sâu so với học máy xem xét lượng dữ liệu tăng lên. Tuy nhiên, điều này có thể thay đổi tùy thuộc vào đặc điểm dữ liệu và cài đặt thực nghiệm.

Hình 18. Hiệu suất học máy và học sâu nói chung theo lượng dữ liệu.

- Các thuật toán học sâu phổ biến nhất là: Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN, hoặc ConvNet), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN). Trong phần tiếp theo, chúng ta sẽ thảo luận về các loại phương pháp học sâu khác nhau có thể được sử dụng để xây dựng các mô hình dựa trên dữ liệu hiệu quả cho các mục đích khác nhau.

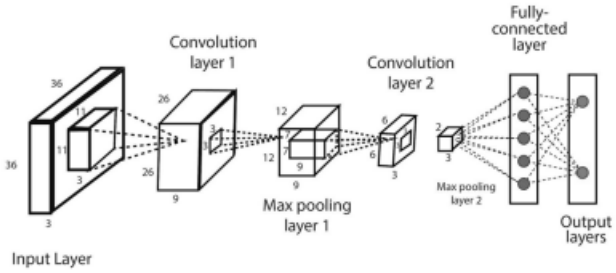
+ MLP: Kiến trúc cơ bản của học sâu, còn được gọi là mạng nơ-ron nhân tạo tiến lùi, được gọi là multilayer perceptron (MLP). Một MLP điển hình là một mạng kết nối đầy đủ bao gồm một lớp đầu vào, một hoặc nhiều lớp ẩn và một lớp đầu ra, như được hiển thị trong Hình 19. Mỗi nút trong một lớp kết nối với mỗi nút trong lớp tiếp theo với một trọng số nhất định. MLP sử dụng kỹ thuật "Backpropagation", khối xây dựng cơ bản nhất trong mạng nơ-ron, để điều chỉnh giá trị trọng số bên trong khi xây dựng mô hình. MLP nhạy cảm với việc tỷ lệ các đặc trưng và cho phép điều chỉnh nhiều siêu tham số, chẳng hạn như số lớp ẩn, số nơ-ron và số vòng lặp, điều này có thể dẫn đến một mô hình tốn

nhiều tính toán.



Hình 19. Cấu trúc của mô hình mạng Nơ-ron nhân tạo với nhiều lớp xử lý.

+ CNN hoặc ConvNet: Mạng nơ-ron tích chập (CNN) cải tiến thiết kế của ANN tiêu chuẩn, bao gồm các lớp tích chập, lớp gộp, cũng như các lớp kết nối đầy đủ, như được hiển thị trong Hình 20. Vì nó tận dụng cấu trúc hai chiều (2D) của dữ liệu đầu vào, nó thường được sử dụng rộng rãi trong nhiều lĩnh vực như nhận dạng hình ảnh và video, xử lý và phân loại hình ảnh, phân tích hình ảnh y tế, xử lý ngôn ngữ tự nhiên, v.v. Mặc dù CNN có gánh nặng tính toán lớn hơn, nhưng mà không cần can thiệp thủ công, nó có lợi thế tự động phát hiện các đặc trưng quan trọng, và do đó CNN được coi là mạnh mẽ hơn so với ANN truyền thống. Một số mô hình học sâu tiên tiến dựa trên CNN có thể được sử dụng trong lĩnh vực, chẳng hạn như AlexNet, Xception, Inception, Visual Geometry Group (VGG), ResNet, v.v.



Hình 20. Một ví dụ về mạng Nơ-ron tích chập (CNN hoặc ConNet) bao gồm nhiều lớp tích chập và lớp gộp.

+ LSTM-RNN: Long short-term memory (LSTM) là một kiến trúc mạng nơ-ron hồi quy nhân tạo được sử dụng trong lĩnh vực học sâu. LSTM có các liên kết phản hồi, khác

với các mạng nơ-ron tiến lùi thông thường. Mạng LSTM rất phù hợp để phân tích và học dữ liệu tuần tự, chẳng hạn như phân loại, xử lý và dự đoán dữ liệu dựa trên chuỗi thời gian, điều này làm nó khác biệt so với các mạng thông thường khác. Do đó, LSTM có thể được sử dụng khi dữ liệu có định dạng tuần tự, chẳng hạn như thời gian, câu văn, v.v. và thường được áp dụng trong lĩnh vực phân tích chuỗi thời gian, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, v.v.

- Ngoài các phương pháp học sâu thông thường đã được thảo luận ở trên, còn tồn tại một số phương pháp học sâu khác trong lĩnh vực này cho các mục đích khác nhau. Ví dụ, bản đồ tự tổ chức (SOM) sử dụng học không giám sát để biểu diễn dữ liệu chiều cao bằng một bản đồ lưới 2D, từ đó đạt được giảm chiều dữ liệu. Máy mã tự động (AE) là một kỹ thuật học khác được sử dụng rộng rãi để giảm chiều dữ liệu và trích xuất đặc trưng trong các nhiệm vụ học không giám sát. Máy Boltzmann hạn chế (RBM) có thể được sử dụng cho việc giảm chiều dữ liệu, phân loại, hồi quy, lọc cộng tác, học đặc trưng và mô hình chủ đề. Mạng niềm tin sâu (DBN) thường được tạo thành từ các mạng không giám sát đơn giản như máy Boltzmann hạn chế (RBM) hoặc máy mã tự động, và một mạng nơ-ron lan truyền ngược (BPNN). Mạng đối địch sinh (GAN) là một dạng mạng cho học sâu có thể tạo ra dữ liệu có đặc điểm gần giống với dữ liệu thực tế đầu vào. Học chuyển giao hiện nay rất phổ biến vì nó có thể huấn luyện mạng nơ-ron sâu với số lượng dữ liệu tương đối thấp, thường là việc tái sử dụng một mô hình đã được huấn luyện trước cho một vấn đề mới. Một thảo luận ngắn về các mô hình mạng nơ-ron nhân tạo (ANN) và học sâu (DL) này đã được tóm tắt trong bài báo trước của chúng tôi (Sarker et al.).

- Tổng quát, dựa trên các kỹ thuật học đã được thảo luận ở trên, chúng ta có thể kết luận rằng các loại kỹ thuật học máy khác nhau, chẳng hạn như phân loại, hồi quy, phân cụm dữ liệu, lựa chọn và trích xuất đặc trưng, giảm chiều dữ liệu, học luật kết hợp, học tăng cường hoặc các kỹ thuật học sâu, có thể đóng vai trò quan trọng cho các mục đích khác nhau dựa trên khả năng của chúng. Trong phần tiếp theo, chúng tôi sẽ thảo luận về một số lĩnh vực ứng dụng dựa trên các thuật toán học máy.

3.9. Ứng dụng của học máy.

Học máy (Machine Learning) đã có những ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ kỹ thuật và khoa học dữ liệu đến y tế, tài chính, giáo dục và nhiều lĩnh vực khác.

Dưới đây là một số ví dụ về các ứng dụng của học máy:

Dự đoán và phân loại: Học máy có thể được sử dụng để dự đoán và phân loại dữ liệu. Ví dụ, trong y tế, nó có thể được sử dụng để dự đoán nguy cơ bệnh, phân loại các loại bệnh, hoặc dự đoán phản ứng của bệnh nhân với các liệu pháp.

Xử lý ngôn ngữ tự nhiên (NLP): Học máy có thể giúp máy tính hiểu và xử lý ngôn ngữ tự nhiên. Ứng dụng NLP bao gồm dịch máy, phân tích tình cảm từ văn bản, tạo tổng hợp văn bản, và nhiều hơn nữa.

Tư vấn và hỗ trợ quyết định: Học máy có thể tạo ra các hệ thống tư vấn dựa trên dữ liệu, giúp người dùng đưa ra quyết định thông minh. Ví dụ, trong tài chính, nó có thể giúp đề xuất các giao dịch đầu tư dựa trên dữ liệu lịch sử và phân tích thị trường.

Xử lý hình ảnh và video: Trong xử lý hình ảnh, học máy có thể được sử dụng để nhận dạng đối tượng, khuôn mặt, phát hiện gian lận hình ảnh, và nhiều ứng dụng khác. Trong video, nó có thể được sử dụng để theo dõi chuyển động, phát hiện hành vi bất thường.

Tự động lái xe và robotics: Học máy đóng vai trò quan trọng trong phát triển xe tự động lái và robot tự động. Nó giúp xe tự động lái "học" cách nhận biết và phản ứng với tình huống giao thông.

Khai thác dữ liệu và dự báo: Học máy có thể giúp khai thác thông tin quý báu từ dữ liệu lớn, từ đó giúp dự báo kết quả tương lai. Ví dụ, trong kinh doanh, nó có thể được sử dụng để dự đoán xu hướng tiêu dùng và thị trường.

Phát hiện gian lận và bảo mật: Học máy có thể giúp phát hiện gian lận tài chính, gian lận thẻ tín dụng, và các hoạt động bất thường khác trong hệ thống.

Tạo nội dung và tạo âm nhạc: Học máy có thể tạo ra nội dung như bài viết, tóm tắt, hoặc thậm chí bài thơ. Nó cũng có thể được sử dụng để tạo âm nhạc và âm thanh dựa trên dữ liệu đầu vào.

Những ứng dụng này chỉ là một phần nhỏ trong loạt các ứng dụng của học máy. Cùng với sự phát triển của công nghệ, học máy ngày càng trở nên quan trọng và phổ biến trong nhiều ngành và lĩnh vực khác nhau.

3.10. Những thách thức và hướng nghiên cứu.

Học máy và trí tuệ nhân tạo đang đối mặt với nhiều thách thức hấp dẫn và còn nhiều hướng nghiên cứu tiềm năng để khám phá và giải quyết. Dưới đây là một số thách thức quan trọng và hướng nghiên cứu trong lĩnh vực này:

Thách thức:

Giải thích và minh bạch: Mô hình học máy phức tạp như Deep Learning thường khó giải thích. Việc hiểu tại sao một dự đoán cụ thể được thực hiện hoặc làm thế nào một mô hình đưa ra quyết định có thể rất quan trọng trong các lĩnh vực như y tế và tài chính.

Bảo mật và quyền riêng tư: Vấn đề liên quan đến bảo mật và quyền riêng tư trong việc sử dụng dữ liệu cá nhân để huấn luyện mô hình là một thách thức quan trọng. Cần phải phát triển các phương pháp để bảo vệ thông tin cá nhân trong quá trình học máy.

Áp dụng trong dữ liệu thiếu: Đối với nhiều ứng dụng thực tế, dữ liệu không luôn đầy đủ và đôi khi thiếu thông tin. Phải tìm cách giải quyết việc làm sao để mô hình có thể học và làm việc tốt trong những tình huống này.

Đa nhiệm và chuyển đổi kiến thức: Làm thế nào để mô hình có thể học từ nhiều nhiệm vụ khác nhau hoặc làm thế nào để chuyển đổi kiến thức từ một nhiệm vụ sang nhiệm vụ khác đang là một thách thức quan trọng.

Hướng nghiên cứu:

Học tăng cường và học tương tác: Nghiên cứu về cách để cho máy tính học từ tương tác với môi trường, thay vì chỉ từ dữ liệu tĩnh, đang ngày càng trở nên quan trọng. Học tăng cường liên quan đến việc máy tính học từ phản hồi sau các hành động mà nó thực hiện.

Học không giám sát tiến xa hơn: Các phương pháp học không giám sát, như học theo lớp, phân cụm, và học sâu không giám sát, đang ngày càng được quan tâm. Điều này liên quan đến việc học từ dữ liệu mà không cần có nhãn.

Học dựa trên biểu cảm và hiểu biết: Nghiên cứu về cách để máy tính có thể hiểu và đáp ứng vào biểu cảm của con người, cả trong ngôn ngữ tự nhiên và dạng khác, là một hướng tiềm năng.

Học đa ngôn ngữ và đa văn bản: Trong thế giới toàn cầu hóa, nghiên cứu về cách để máy

tính có thể học và làm việc với nhiều ngôn ngữ và văn bản khác nhau là rất quan trọng.

Xử lý dữ liệu không cân bằng: Nghiên cứu cách để xử lý dữ liệu không cân bằng, khi một lớp hoặc mẫu dữ liệu có số lượng ít hơn, để đảm bảo rằng mô hình không bị thiên vị.

Học máy có sự tham gia của con người: Nghiên cứu cách để tích hợp kiến thức và phản hồi từ con người vào quá trình học máy, giúp cải thiện hiệu suất và tạo ra mô hình có tính nhận thức cao hơn.

Những thách thức và hướng nghiên cứu này chỉ là một phần trong một loạt các vấn đề quan trọng trong lĩnh vực học máy và trí tuệ nhân tạo. Sự phát triển liên tục trong công nghệ và sự tìm kiếm của cộng đồng nghiên cứu sẽ tiếp tục định hình hướng phát triển của lĩnh vực này trong tương lai.

3.11. Phần kết luận.

- Trong bài báo này, chúng tôi đã tiến hành một tổng quan toàn diện về các thuật toán học máy cho phân tích dữ liệu thông minh và ứng dụng. Theo mục tiêu của chúng tôi, chúng tôi đã tóm tắt cách các loại phương pháp học máy khác nhau có thể được sử dụng để tạo ra các giải pháp cho các vấn đề thực tế khác nhau. Một mô hình học máy thành công phụ thuộc vào cả dữ liệu và hiệu suất của các thuật toán học. Sau đó, các thuật toán học phức tạp cần được huấn luyện thông qua dữ liệu thực tế đã thu thập và kiến thức liên quan đến ứng dụng mục tiêu trước khi hệ thống có thể hỗ trợ trong việc ra quyết định thông minh. Chúng tôi cũng đã thảo luận về một số lĩnh vực ứng dụng phổ biến dựa trên kỹ thuật học máy để nhấn mạnh tính ứng dụng của chúng trong các vấn đề thực tế khác nhau. Cuối cùng, chúng tôi đã tóm tắt và thảo luận về các thách thức và cơ hội nghiên cứu tiềm năng và hướng phát triển trong lĩnh vực này. Do đó, các thách thức được xác định tạo ra cơ hội nghiên cứu hứa hẹn trong lĩnh vực này, cần được giải quyết bằng các giải pháp hiệu quả trong các lĩnh vực ứng dụng khác nhau. Tổng thể, chúng tôi tin rằng nghiên cứu của chúng tôi về các giải pháp dựa trên học máy mở ra một hướng phát triển hứa hẹn và có thể được sử dụng như một hướng dẫn tham khảo cho các nghiên cứu và ứng dụng tiềm năng cho cả cán bộ giảng dạy và công nghiệp cũng như các nhà quyết định, từ một góc độ kỹ thuật.

CHƯƠNG 4: DỰ ĐOÁN BỆNH BẰNG CÁCH SỬ DỤNG HỌC MÁY

4.1. Các bước thực hiện mô hình học máy

import các thư viện và các bước chuẩn bị dữ liệu, xây dựng và đánh giá các mô hình học máy. Dưới đây là giải thích từng phần trong mã nguồn:

```
[ ]  # Importing libraries
     import numpy as np
     import pandas as pd
     from scipy.stats import mode
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.preprocessing import LabelEncoder
     from sklearn.model_selection import train_test_split, cross_val_score
     from sklearn.svm import SVC
     from sklearn.naive_bayes import GaussianNB
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score, confusion_matrix

     %matplotlib inline
```

Hình 4.1: Import các thư viện cần thiết cho mô hình

- numpy: Thư viện cho tính toán khoa học và số học trong Python.
- pandas: Thư viện dùng để làm việc với dữ liệu dưới dạng DataFrame.
- scipy.stats: Thư viện cho các phân phối thống kê và chức năng thống kê.
- matplotlib.pyplot và seaborn: Thư viện để tạo đồ thị và biểu đồ cho việc trực quan hóa dữ liệu.
- LabelEncoder từ sklearn.preprocessing: Dùng để chuyển đổi các nhãn văn bản thành số để sử dụng trong mô hình học máy.
- train_test_split từ sklearn.model_selection: Dùng để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
- SVC (Support Vector Classifier) từ sklearn.svm: Mô hình SVM dùng cho việc phân loại.
- GaussianNB từ sklearn.naive_bayes: Mô hình Naive Bayes dùng cho phân loại.

- RandomForestClassifier từ sklearn.ensemble: Mô hình Random Forest dùng cho phân loại.
- accuracy_score, confusion_matrix từ sklearn.metrics: Dùng để tính toán độ chính xác và ma trận nhầm lẫn của mô hình.
- %matplotlib inline: Đây là một lệnh magic trong Jupyter Notebook để hiển thị đồ thị trực tiếp trong giao diện notebook.

```
[ ]  # Reading the train.csv by removing the
     # last column since it's an empty column
     DATA_PATH = "/content/drive/MyDrive/HOC/data/Training.csv"
     data = pd.read_csv(DATA_PATH).dropna(axis = 1)

     # Checking whether the dataset is balanced or not
     disease_counts = data["prognosis"].value_counts()
     temp_df = pd.DataFrame({
       "Disease": disease_counts.index,
       "Counts": disease_counts.values
     })

     plt.figure(figsize = (18,8))
     sns.barplot(x = "Disease", y = "Counts", data = temp_df)
     plt.xticks(rotation=90)
     plt.show()
```

Hình 4.2: Mã nguồn 1

Mã nguồn trên thực hiện các bước sau:

Đọc dữ liệu từ tập tin CSV:

DATA_PATH: Đường dẫn đến tập tin "Training.csv".

pd.read_csv(DATA_PATH): Sử dụng pandas để đọc dữ liệu từ tập tin CSV và tạo DataFrame.

.dropna(axis=1): Loại bỏ cột có giá trị NaN (rỗng) từ DataFrame. Điều này thường được thực hiện để loại bỏ các cột không có dữ liệu.

Kiểm tra tính cân bằng của dữ liệu:

data["prognosis"].value_counts(): Đếm số lượng mỗi loại trong cột "prognosis", đây có thể là cột chứa thông tin về dự đoán/điều trị bệnh.

temp_df: Tạo một DataFrame tạm thời chứa thông tin về số lượng mỗi loại bệnh và số lần xuất hiện của chúng.

Vẽ biểu đồ cột thể hiện sự cân bằng của dữ liệu:

plt.figure(figsize=(18,8)): Tạo một kích thước mới cho biểu đồ cột.

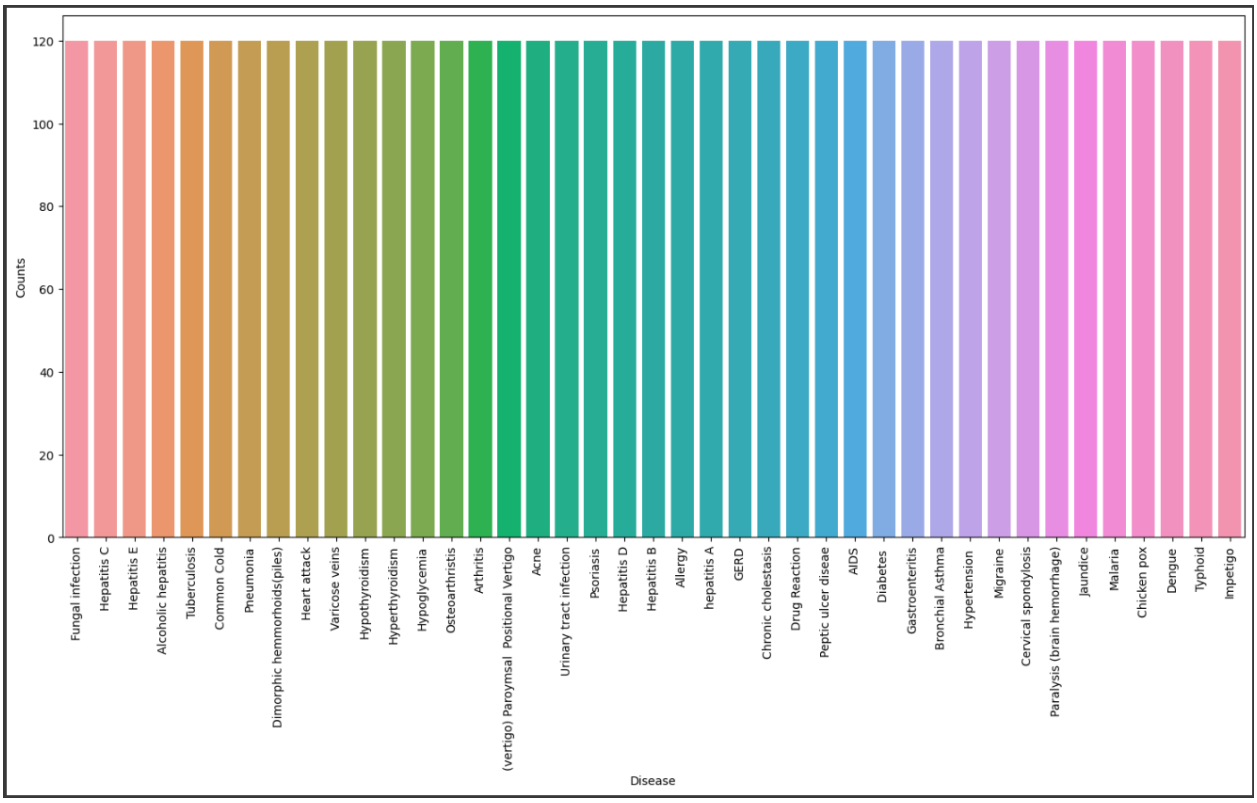
sns.barplot(x="Disease", y="Counts", data=temp_df): Sử dụng thư viện Seaborn để tạo biểu đồ cột, trục x là tên các bệnh và trục y là số lượng.

plt.xticks(rotation=90): Xoay tên các loại bệnh trên trục x để tránh trùng lắp.

plt.show(): Hiển thị biểu đồ cột.

Mục tiêu của đoạn mã này là để đọc dữ liệu từ tập tin "Training.csv", loại bỏ cột có giá trị NaN và sau đó kiểm tra tính cân bằng của dữ liệu bằng cách vẽ biểu đồ cột thể hiện số lượng của mỗi loại bệnh.

Kết quả:



Hình 4.3: Kết quả biểu đồ cột thể hiên số lượng các loại bệnh

```
[ ]  # Encoding the target value into numerical
     # value using LabelEncoder
     encoder = LabelEncoder()
     data["prognosis"] = encoder.fit_transform(data["prognosis"])
```

Hình 4.4: Mã nguồn 2

Mã nguồn trên thực hiện việc chuyển đổi nhãn của biến mục tiêu từ dạng văn bản thành dạng số bằng cách sử dụng LabelEncoder.

Import thư viện LabelEncoder:

encoder = LabelEncoder(): Tạo một đối tượng LabelEncoder để thực hiện việc chuyển đổi.

Encoding của biến mục tiêu:

data["prognosis"] = encoder.fit_transform(data["prognosis"]): Áp dụng fit_transform để chuyển đổi các giá trị trong cột "prognosis" từ dạng văn bản thành dạng số. Sau khi thực hiện, cột "prognosis" trong DataFrame data sẽ chứa các giá trị số thay vì nhãn ban đầu.

Quá trình chuyển đổi này thường được thực hiện để cho phép mô hình học máy xử lý các biến mục tiêu có dạng số thay vì dạng văn bản.

```
[ ]  X = data.iloc[:,:-1]
     y = data.iloc[:, -1]
     X_train, X_test, y_train, y_test =train_test_split(
     X, y, test_size = 0.2, random_state = 24)

     print(f"Train: {X_train.shape}, {y_train.shape}")
     print(f"Test: {X_test.shape}, {y_test.shape}")
```

Hình 4.5: Mã nguồn 3

Mã nguồn trên thực hiện việc chia dữ liệu thành các tập huấn luyện và kiểm tra bằng cách sử dụng hàm train_test_split từ thư viện sklearn.model_selection.

Chia dữ liệu:

X = data.iloc[:,:-1]: Lấy tất cả các cột trừ cột cuối cùng (biến mục tiêu) trong DataFrame data là đặt X, là tập dữ liệu đặc trưng.

y = data.iloc[:, -1]: Lấy cột cuối cùng (biến mục tiêu) trong DataFrame data là đặt y, là nhãn tương ứng với mỗi mẫu.

Sử dụng hàm train_test_split:

train_test_split(X, y, test_size=0.2, random_state=24): Chia tập dữ liệu thành các tập huấn luyện và kiểm tra. test_size=0.2 chỉ định rằng 20% dữ liệu sẽ được sử dụng cho tập kiểm tra và 80% cho tập huấn luyện. random_state=24 là để đảm bảo việc chia dữ liệu là nhất quán trong các lần thực hiện.

In kích thước các tập dữ liệu:

print(f"Train: {X_train.shape}, {y_train.shape}"): In ra kích thước của tập huấn luyện (đặc trưng và nhãn).

print(f"Test: {X_test.shape}, {y_test.shape}"): In ra kích thước của tập kiểm tra (đặc trưng và nhãn).

Quá trình này chia dữ liệu thành hai tập: tập huấn luyện (để huấn luyện mô hình) và tập kiểm tra (để đánh giá mô hình).

Kết quả:

```
Train: (3936, 132), (3936,)
Test: (984, 132), (984,)
```

Hình 4.6: Kết quả Train và Test

```python
# Defining scoring metric for k-fold cross validation
def cv_scoring(estimator, X, y):
  return accuracy_score(y, estimator.predict(X))

# Initializing Models
models = {
  "SVC":SVC(),
  "Gaussian NB":GaussianNB(),
  "Random Forest":RandomForestClassifier(random_state=18)
}

# Producing cross validation score for the models
for model_name in models:
  model = models[model_name]
  scores = cross_val_score(model, X, y, cv = 10,
             n_jobs = -1,
             scoring = cv_scoring)
  print("=="*30)
  print(model_name)
  print(f"Scores: {scores}")
  print(f"Mean Score: {np.mean(scores)}")
```

Hình 4.7: Mã nguồn 4

Mã nguồn trên thực hiện việc định nghĩa metric để đánh giá cho quá trình cross-validation (chạy k-fold cross-validation) và đánh giá hiệu suất của mô hình học máy sử dụng các loại mô hình khác nhau.

Định nghĩa hàm cv_scoring:

cv_scoring(estimator, X, y): Hàm này tính toán độ chính xác của mô hình estimator khi dự đoán trên dữ liệu X và so sánh với nhãn thực tế y.

Khởi tạo các mô hình:

models = {...}: Một từ điển chứa các mô hình khác nhau như SVC, Gaussian NB và Random Forest, cùng với các tham số cần thiết để khởi tạo mô hình.

Chạy cross-validation và tính điểm:

Dùng vòng lặp để duyệt qua các mô hình trong từ điển models.

cross_val_score(model, X, y, cv=10, n_jobs=-1, scoring=cv_scoring): Sử dụng hàm cross_val_score để chạy k-fold cross-validation trên mô hình model. cv=10 cho biết sẽ chia dữ liệu thành 10 phần (10-fold cross-validation), n_jobs=-1 để sử dụng tất cả các CPU, scoring=cv_scoring để sử dụng hàm cv_scoring để đánh giá mô hình.

In kết quả:

Dùng vòng lặp để in tên của mỗi mô hình và các điểm cross-validation tương ứng.

print("=="*30): In dấu gạch ngang để phân tách giữa các kết quả.

print(model_name): In tên của mô hình.

print(f"Scores: {scores}"): In điểm cross-validation của mô hình.

print(f"Mean Score: {np.mean(scores)}"): In điểm trung bình của cross-validation cho mô hình.

Mục tiêu của mã nguồn này là để đánh giá hiệu suất của các mô hình học máy sử dụng cross-validation và tính toán các điểm đánh giá, bao gồm cả điểm trung bình của cross-validation cho từng mô hình.

Kết quả:

```
============================================================
SVC
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
============================================================
Gaussian NB
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
============================================================
Random Forest
Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Mean Score: 1.0
```

Hình 4.8: Tính toán và đánh giá hiệu suất mô hình học máy

```
[ ]  # Training and testing SVM Classifier
     svm_model = SVC()
     svm_model.fit(X_train, y_train)
     preds = svm_model.predict(X_test)

     print(f"Accuracy on train data by SVM Classifier\
     : {accuracy_score(y_train, svm_model.predict(X_train))*100}")

     print(f"Accuracy on test data by SVM Classifier\
     : {accuracy_score(y_test, preds)*100}")
     cf_matrix = confusion_matrix(y_test, preds)
     plt.figure(figsize=(12,8))
     sns.heatmap(cf_matrix, annot=True)
     plt.title("Confusion Matrix for SVM Classifier on Test Data")
     plt.show()

     # Training and testing Naive Bayes Classifier
     nb_model = GaussianNB()
     nb_model.fit(X_train, y_train)
     preds = nb_model.predict(X_test)
     print(f"Accuracy on train data by Naive Bayes Classifier\
     : {accuracy_score(y_train, nb_model.predict(X_train))*100}")

     print(f"Accuracy on test data by Naive Bayes Classifier\
     : {accuracy_score(y_test, preds)*100}")
     cf_matrix = confusion_matrix(y_test, preds)
     plt.figure(figsize=(12,8))
     sns.heatmap(cf_matrix, annot=True)
     plt.title("Confusion Matrix for Naive Bayes Classifier on Test Data")
     plt.show()

     # Training and testing Random Forest Classifier
     rf_model = RandomForestClassifier(random_state=18)
     rf_model.fit(X_train, y_train)
     preds = rf_model.predict(X_test)
     print(f"Accuracy on train data by Random Forest Classifier\
     : {accuracy_score(y_train, rf_model.predict(X_train))*100}")

     print(f"Accuracy on test data by Random Forest Classifier\
     : {accuracy_score(y_test, preds)*100}")

     cf_matrix = confusion_matrix(y_test, preds)
     plt.figure(figsize=(12,8))
     sns.heatmap(cf_matrix, annot=True)
     plt.title("Confusion Matrix for Random Forest Classifier on Test Data")
     plt.show()
```

Hình 4.9: Mã nguồn 5

Mã nguồn trên thực hiện việc huấn luyện và kiểm tra các mô hình học máy (SVM Classifier, Naive Bayes Classifier, và Random Forest Classifier) trên tập dữ liệu huấn luyện và kiểm tra. Đồng thời, nó cũng hiển thị độ chính xác của mỗi mô hình trên tập huấn luyện và tập kiểm tra, và vẽ ma trận nhầm lẫn (confusion matrix) để đánh giá kết quả phân loại.

Huấn luyện và kiểm tra SVM Classifier:

svm_model = SVC(): Khởi tạo mô hình Support Vector Machine (SVM).

svm_model.fit(X_train, y_train): Huấn luyện mô hình trên tập huấn luyện.

preds = svm_model.predict(X_test): Dự đoán kết quả trên tập kiểm tra.

In ra độ chính xác trên tập huấn luyện và tập kiểm tra của SVM Classifier.

Vẽ ma trận nhầm lẫn (confusion matrix) để đánh giá kết quả phân loại.

Huấn luyện và kiểm tra Naive Bayes Classifier:

nb_model = GaussianNB(): Khởi tạo mô hình Naive Bayes.

nb_model.fit(X_train, y_train): Huấn luyện mô hình trên tập huấn luyện.

preds = nb_model.predict(X_test): Dự đoán kết quả trên tập kiểm tra.

In ra độ chính xác trên tập huấn luyện và tập kiểm tra của Naive Bayes Classifier.

Vẽ ma trận nhầm lẫn (confusion matrix) để đánh giá kết quả phân loại.

Huấn luyện và kiểm tra Random Forest Classifier:

rf_model = RandomForestClassifier(random_state=18): Khởi tạo mô hình Random Forest.

rf_model.fit(X_train, y_train): Huấn luyện mô hình trên tập huấn luyện.

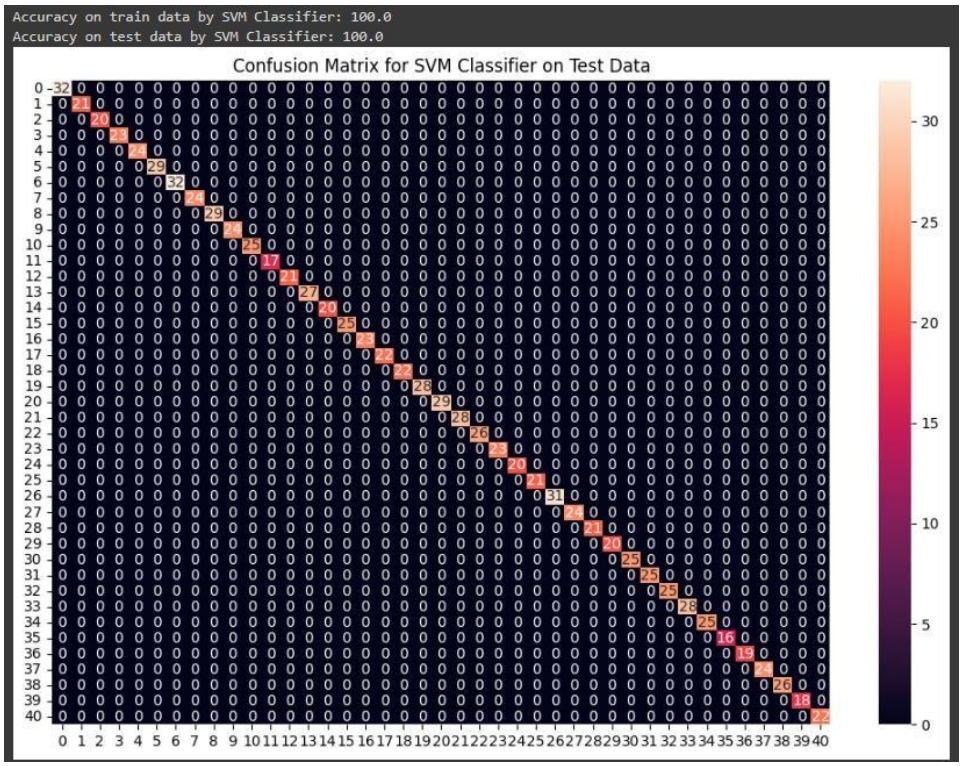
preds = rf_model.predict(X_test): Dự đoán kết quả trên tập kiểm tra.

In ra độ chính xác trên tập huấn luyện và tập kiểm tra của Random Forest Classifier.

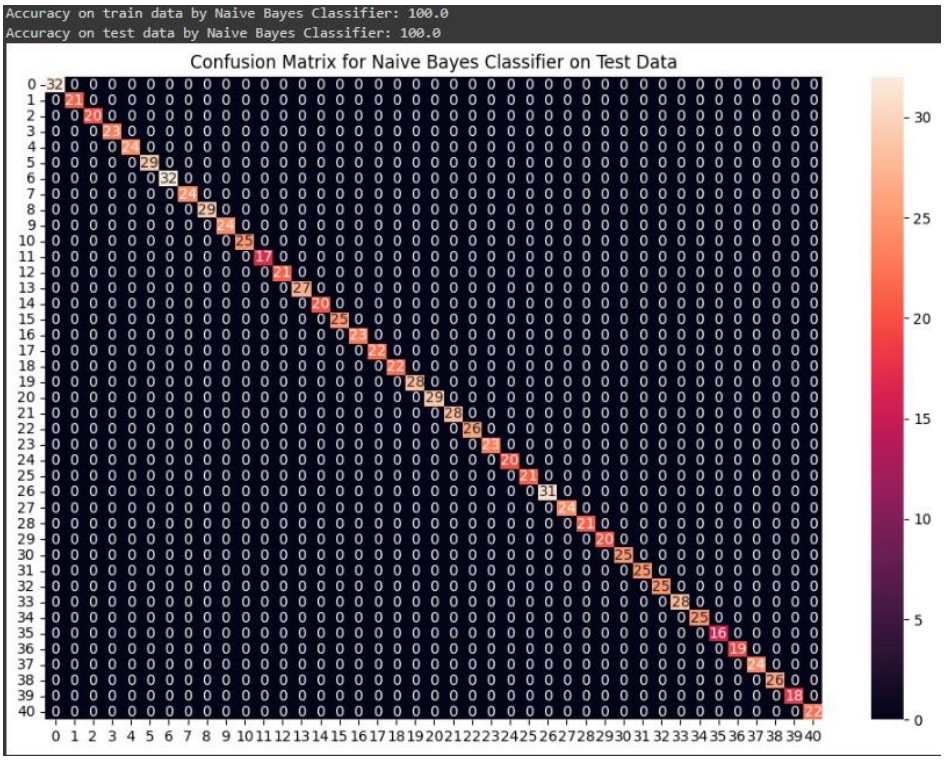
Vẽ ma trận nhầm lẫn (confusion matrix) để đánh giá kết quả phân loại.

Mục tiêu của mã nguồn này là huấn luyện, dự đoán và đánh giá hiệu suất của các mô hình học máy trên tập dữ liệu kiểm tra và hiển thị ma trận nhầm lẫn để biết được mô hình phân loại như thế nào trên các lớp khác nhau.

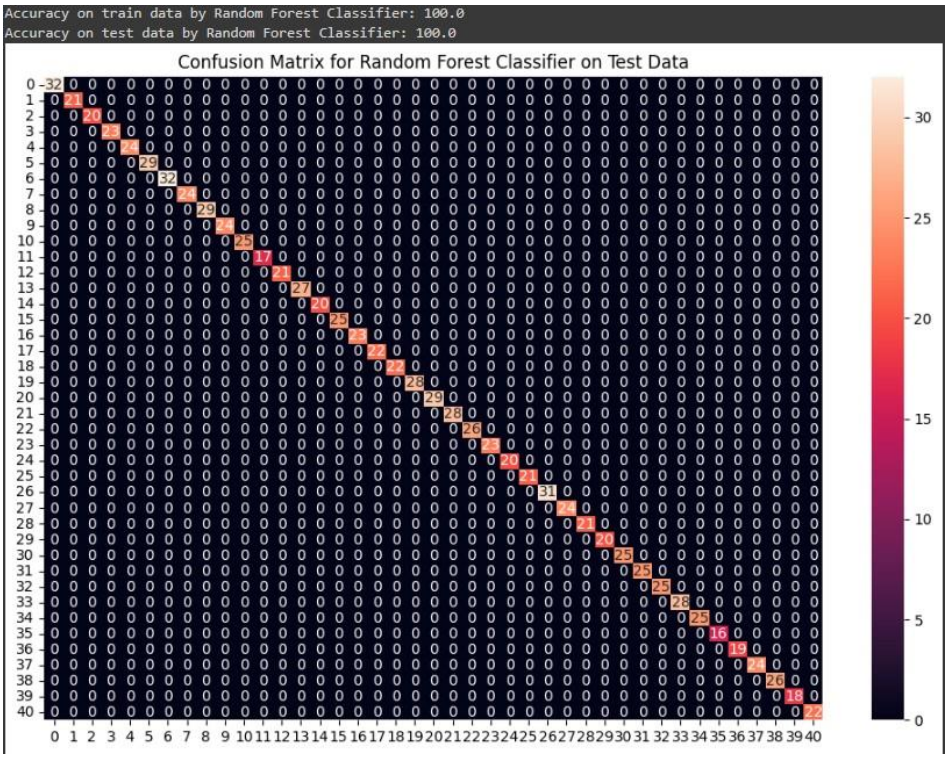
Các kết quả:



Hình 4.10: Mô hình phân loại lớp 1



Hình 4.11: Mô hình phân loại lớp 2



Hình 4.12: Mô hình phân loại lớp 3

```
[ ]  # Training the models on whole data
     final_svm_model = SVC()
     final_nb_model = GaussianNB()
     final_rf_model = RandomForestClassifier(random_state=18)
     final_svm_model.fit(X, y)
     final_nb_model.fit(X, y)
     final_rf_model.fit(X, y)

     # Reading the test data
     test_data = pd.read_csv("/content/drive/MyDrive/HOC/data/Testing.csv").dropna(axis=1)

     test_X = test_data.iloc[:, :-1]
     test_Y = encoder.transform(test_data.iloc[:, -1])

     # Making prediction by take mode of predictions
     # made by all the classifiers
     svm_preds = final_svm_model.predict(test_X)
     nb_preds = final_nb_model.predict(test_X)
     rf_preds = final_rf_model.predict(test_X)

     final_preds = [mode([i,j,k])[0][0] for i,j,
           k in zip(svm_preds, nb_preds, rf_preds)]

     print(f"Accuracy on Test dataset by the combined model\
     : {accuracy_score(test_Y, final_preds)*100}")

     cf_matrix = confusion_matrix(test_Y, final_preds)
     plt.figure(figsize=(12,8))

     sns.heatmap(cf_matrix, annot = True)
     plt.title("Confusion Matrix for Combined Model on Test Dataset")
     plt.show()
```

Hình 4.13: Mã nguồn 6

Mã nguồn trên thực hiện việc huấn luyện các mô hình cuối cùng trên toàn bộ dữ liệu, sau đó sử dụng các mô hình đã huấn luyện để dự đoán trên dữ liệu kiểm tra. Kết quả dự đoán từ các mô hình được kết hợp bằng cách lấy mode của các dự đoán.

Huấn luyện mô hình cuối cùng trên toàn bộ dữ liệu:

final_svm_model = SVC(): Khởi tạo mô hình SVM cuối cùng.

final_nb_model = GaussianNB(): Khởi tạo mô hình Naive Bayes cuối cùng.

final_rf_model = RandomForestClassifier(random_state=18): Khởi tạo mô hình Random Forest cuối cùng.

final_svm_model.fit(X, y): Huấn luyện mô hình SVM trên toàn bộ dữ liệu.

final_nb_model.fit(X, y): Huấn luyện mô hình Naive Bayes trên toàn bộ dữ liệu.

final_rf_model.fit(X, y): Huấn luyện mô hình Random Forest trên toàn bộ dữ liệu.

Đọc dữ liệu kiểm tra và chuẩn bị dữ liệu kiểm tra:

Đọc dữ liệu kiểm tra từ tập tin "Testing.csv".

test_X: Lấy tất cả các cột trừ cột cuối cùng (biến mục tiêu) trong dữ liệu kiểm tra.

test_Y: Chuyển đổi nhãn cuối cùng (biến mục tiêu) từ dạng văn bản thành dạng số sử dụng encoder.

Dự đoán và kết hợp dự đoán từ các mô hình:

Dự đoán kết quả trên dữ liệu kiểm tra sử dụng mô hình SVM cuối cùng, Naive Bayes cuối cùng và Random Forest cuối cùng.

Sử dụng mode để lấy giá trị xuất hiện nhiều nhất của các dự đoán từ các mô hình, tạo thành kết quả cuối cùng.

Đánh giá kết quả trên dữ liệu kiểm tra:

In ra độ chính xác của mô hình kết hợp trên dữ liệu kiểm tra.

Vẽ ma trận nhầm lẫn (confusion matrix) để đánh giá kết quả phân loại bằng mô hình kết hợp trên dữ liệu kiểm tra.

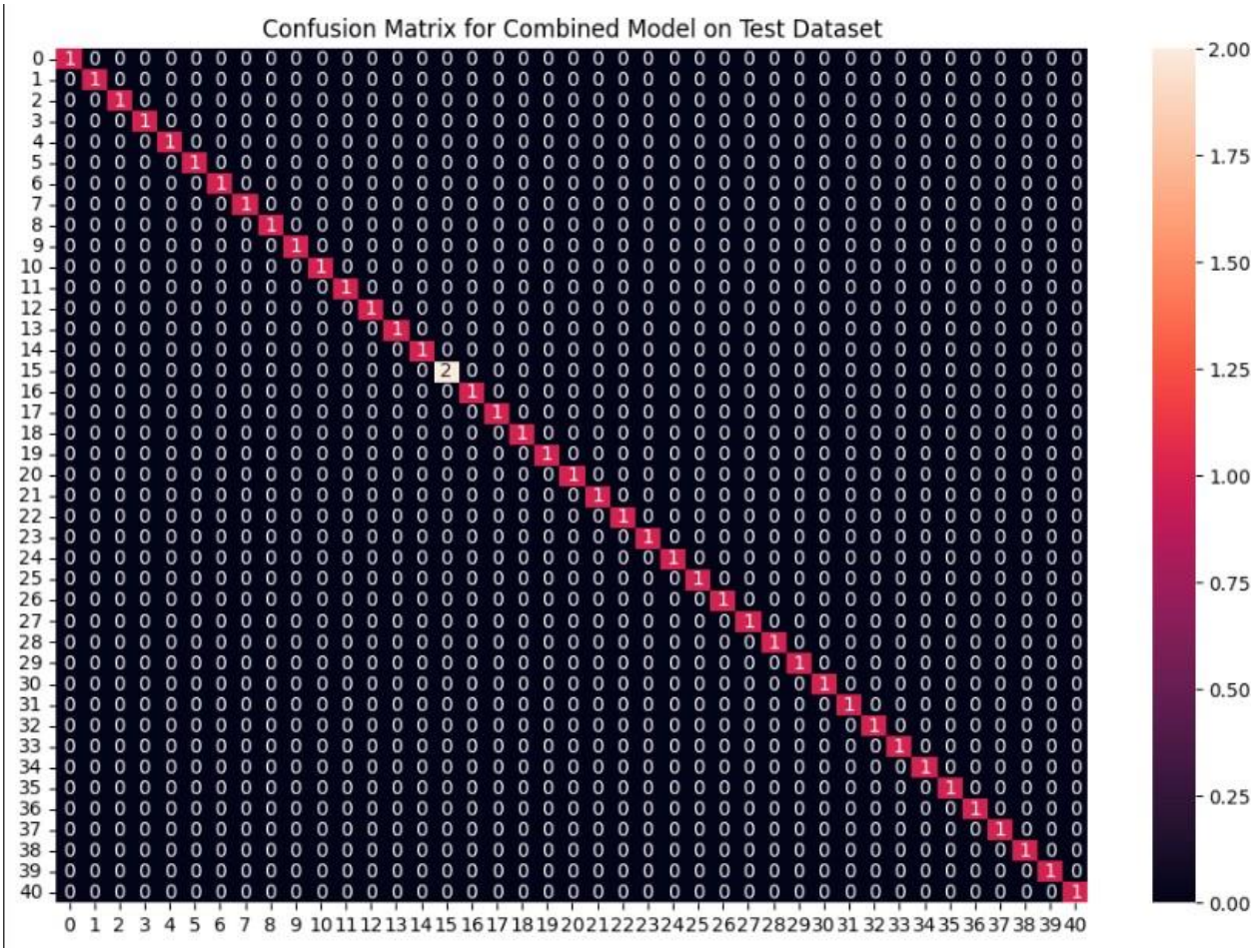
Mục tiêu của mã nguồn này là dự đoán và đánh giá hiệu suất của mô hình kết hợp trên dữ liệu kiểm tra, sau khi đã huấn luyện các mô hình cuối cùng trên toàn bộ dữ liệu huấn luyện.

Kết quả:

<ipython-input-9-1f5d90bc9249>:21: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.

final_preds = [mode([i,j,k])[0][0] for i,j,

Accuracy on Test dataset by the combined model: 100.0



Hình 4.14: Mô hình cuối cùng dựa trên các mô hình kiểm tra ở trên

```
[ ] symptoms = X.columns.values

    # Creating a symptom index dictionary to encode the
    # input symptoms into numerical form
    symptom_index = {}
    for index, value in enumerate(symptoms):
      symptom = " ".join([i.capitalize() for i in value.split("_")])
      symptom_index[symptom] = index

    data_dict = {
      "symptom_index":symptom_index,
      "predictions_classes":encoder.classes_
    }

    # Defining the Function
    # Input: string containing symptoms separated by commas
    # Output: Generated predictions by models
    def predictDisease(symptoms):
      symptoms = symptoms.split(",")

      # creating input data for the models
      input_data = [0] * len(data_dict["symptom_index"])
      for symptom in symptoms:
        index = data_dict["symptom_index"][symptom]
        input_data[index] = 1

      # reshaping the input data and converting it
      # into suitable format for model predictions
      input_data = np.array(input_data).reshape(1,-1)

      # generating individual outputs
      rf_prediction = data_dict["predictions_classes"][final_rf_model.predict(input_data)[0]]
      nb_prediction = data_dict["predictions_classes"][final_nb_model.predict(input_data)[0]]
      svm_prediction = data_dict["predictions_classes"][final_svm_model.predict(input_data)[0]]

      # making final prediction by taking mode of all predictions
      final_prediction = mode([rf_prediction, nb_prediction, svm_prediction])[0][0]
      predictions = {
        "rf_model_prediction": rf_prediction,
        "naive_bayes_prediction": nb_prediction,
        "svm_model_prediction": svm_prediction,
        "final_prediction":final_prediction
      }
      return predictions

    # Testing the function
    print(predictDisease("Itching,Skin Rash,Nodal Skin Eruptions"))
```

Hình 4.15: Mã nguồn 7

Mã nguồn trên định nghĩa một hàm predictDisease(symptoms) để dự đoán bệnh dựa trên các triệu chứng nhập vào. Dưới đây là giải thích từng phần của mã nguồn:

Xác định các triệu chứng và chỉ số cho chúng:

symptoms = X.columns.values: Lấy tên của các cột trong tập dữ liệu đặc trưng, đây là tên của các triệu chứng.

symptom_index = {}: Khởi tạo từ điển để ánh xạ tên triệu chứng sang chỉ số tương ứng.

Tạo từ điển dùng cho việc dự đoán:

Dùng vòng lặp để tạo một từ điển chứa thông tin về chỉ số của triệu chứng và các lớp dự đoán từ encoder.

Định nghĩa hàm predictDisease:

Hàm này nhận vào một chuỗi chứa các triệu chứng được phân tách bằng dấu phẩy.

Tạo dữ liệu đầu vào cho các mô hình dự đoán bằng cách thay 1 vào chỉ số của triệu chứng trong danh sách input_data.

Thực hiện dự đoán của các mô hình cuối cùng (SVM, Naive Bayes, Random Forest) trên dữ liệu đầu vào và chuyển đổi kết quả thành dạng văn bản bằng cách sử dụng từ điển data_dict.

Sử dụng mode để lấy giá trị xuất hiện nhiều nhất của các dự đoán từ các mô hình, tạo thành kết quả dự đoán cuối cùng.

Kiểm tra hàm bằng cách dự đoán một tập hợp cụ thể của triệu chứng:

Gọi hàm predictDisease với chuỗi triệu chứng "Itching,Skin Rash,Nodal Skin Eruptions".

In kết quả dự đoán từ tập hợp các mô hình và dự đoán cuối cùng.

Mục tiêu của mã nguồn này là định nghĩa một hàm dự đoán bệnh dựa trên các triệu chứng được nhập vào. Hàm này sử dụng các mô hình đã huấn luyện để dự đoán và kết hợp kết quả từ các mô hình để đưa ra dự đoán cuối cùng về bệnh.

4.2 Kết luận, Ưu – Nhược điểm và cách khắc phục mô hình

4.2.1 Kết luận

Mô hình trên mô tả quá trình hoàn chỉnh từ việc tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá hiệu suất, tạo dự đoán và cuối cùng là việc kết hợp các dự đoán từ các mô hình khác nhau để đưa ra dự đoán cuối cùng về bệnh dựa trên các triệu chứng. Dưới đây là các bước chính trong quá trình này:

Tiền xử lý dữ liệu:

Đọc dữ liệu từ tập tin CSV và loại bỏ cột trống.

Thực hiện chuyển đổi LabelEncoder để chuyển đổi biến mục tiêu từ dạng văn bản thành dạng số.

Chia dữ liệu thành tập huấn luyện và tập kiểm tra.

Huấn luyện và đánh giá mô hình:

Khởi tạo và huấn luyện các mô hình học máy (SVM, Naive Bayes, Random Forest) trên tập huấn luyện.

Sử dụng cross-validation để đánh giá hiệu suất của mỗi mô hình.

In kết quả độ chính xác của mô hình trên tập kiểm tra và vẽ ma trận nhầm lẫn.

Dự đoán bệnh dựa trên triệu chứng:

Xác định chỉ số cho các triệu chứng và tạo từ điển để ánh xạ tên triệu chứng sang chỉ số tương ứng.

Định nghĩa hàm predictDisease để dự đoán bệnh dựa trên triệu chứng nhập vào.

Dự đoán bệnh bằng cách sử dụng các mô hình cuối cùng và kết hợp kết quả từ chúng.

Dự đoán và đánh giá trên dữ liệu kiểm tra:

Huấn luyện các mô hình cuối cùng trên toàn bộ dữ liệu huấn luyện.

Dự đoán và kết hợp kết quả từ các mô hình cuối cùng trên dữ liệu kiểm tra.

Đánh giá hiệu suất của mô hình kết hợp trên dữ liệu kiểm tra bằng cách in độ chính xác và vẽ ma trận nhầm lẫn.

Tóm lại, các mã nguồn này thực hiện một quy trình hoàn chỉnh từ tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá và dự đoán trên cả dữ liệu huấn luyện và dữ liệu kiểm tra. Cách làm này cho phép đưa ra dự đoán về bệnh dựa trên các triệu chứng và đánh giá hiệu suất của mô hình phân loại.

4.2.2 Ưu điểm

Tiện ích ứng dụng: cung cấp một ứng dụng thực tế để dự đoán bệnh dựa trên các triệu chứng, có thể hữu ích trong việc hỗ trợ chẩn đoán sơ bộ trong lĩnh vực y tế.

Tiền xử lý dữ liệu: nó cho thấy cách tiền xử lý dữ liệu bằng cách loại bỏ cột trống và chuyển đổi biến mục tiêu thành dạng số, đảm bảo dữ liệu sẵn sàng để huấn luyện và dự đoán.

Huấn luyện và đánh giá mô hình: việc huấn luyện và đánh giá mô hình học máy, sử dụng các mô hình khác nhau như SVM, Naive Bayes và Random Forest. Cách sử dụng cross-validation giúp đảm bảo độ chính xác tổng quát của mô hình.

Dự đoán dựa trên triệu chứng: định nghĩa hàm để dự đoán bệnh dựa trên triệu chứng nhập vào, cho phép người dùng đưa ra dự đoán sơ bộ về bệnh dựa trên thông tin triệu chứng.

4.2.3 Nhược điểm

Số lượng mẫu: Để đạt được hiệu suất tốt, mô hình thường yêu cầu một lượng lớn dữ liệu huấn luyện. Nếu tập dữ liệu không đủ lớn, mô hình có thể bị overfitting hoặc underfitting.

Chất lượng dữ liệu: Hiệu suất của mô hình phụ thuộc nhiều vào chất lượng dữ liệu. Dữ liệu không chính xác, thiếu sót hoặc nhiễu có thể ảnh hưởng đến hiệu suất của mô hình.

Tùy chỉnh tham số: Các mô hình có thể cần tinh chỉnh tham số để đạt được hiệu suất tốt nhất. Việc tìm kiếm tham số tốt yêu cầu kiến thức sâu về mô hình và thực nghiệm.

4.2.4 Hướng khắc phục

Tăng số lượng mẫu: Nếu có khả năng, cố gắng tăng kích thước tập dữ liệu huấn luyện để cải thiện khả năng tổng quát hóa của mô hình.

Kiểm tra chất lượng dữ liệu: Thực hiện kiểm tra và làm sạch dữ liệu để loại bỏ dữ liệu nhiễu hoặc không chính xác. Sử dụng phương pháp khôi phục dữ liệu thiếu nếu cần.

Tinh chỉnh tham số: Sử dụng phương pháp tìm kiếm tham số tốt nhất như Grid Search hoặc Random Search để tìm các tham số tối ưu cho mô hình.

Thêm tính đa dạng vào dữ liệu: Đảm bảo rằng tập dữ liệu bao gồm đủ nhiều loại triệu chứng và bệnh khác nhau để đảm bảo mô hình phân loại chính xác trên mọi tình huống.

Sử dụng mô hình phức tạp hơn: Nếu dữ liệu và yêu cầu thực tế cho phép, bạn có thể sử dụng các mô hình phức tạp hơn như mạng neural để cải thiện hiệu suất dự đoán.

4.2.5 Đánh giá thuật toán

Đánh giá một thuật toán học máy và ứng dụng của nó phải dựa trên nhiều yếu tố như hiệu suất, tính khả dụng, ứng dụng thực tế và sự thể hiện trong điều kiện khác nhau:

Cross-validation Scores: Đánh giá độ chính xác của các mô hình thông qua cross-validation. Điều này cho biết khả năng tổng quát hóa của mô hình trên các dữ liệu không nhìn thấy trước.

Độ chính xác trên dữ liệu kiểm tra: Hiệu suất của mô hình trên tập dữ liệu kiểm tra có phù hợp với mục tiêu không. Độ chính xác càng cao, càng tốt, nhưng cần xem xét kỹ càng để tránh overfitting.

Dự đoán chính xác bệnh: Mục tiêu chính của thuật toán là dự đoán chính xác bệnh dựa trên triệu chứng. Cách dự đoán có thể đem lại giá trị thực sự cho ứng dụng trong lĩnh vực y tế.

Thời gian huấn luyện và dự đoán: Thuật toán có thể chạy nhanh chóng và hiệu quả trên dữ liệu thực tế không.

Tính tương thích: Thuật toán có thể hoạt động trên các nền tảng và môi trường khác nhau một cách dễ dàng không.

Tinh chỉnh tham số: Có khó khăn trong việc tinh chỉnh các tham số của thuật toán không.

Yêu cầu kiến thức: Yêu cầu kiến thức cao về cách điều chỉnh tham số và lựa chọn thuật toán.

Dữ liệu: Độ lớn của tập dữ liệu có đủ lớn để huấn luyện mô hình không.

Chất lượng dữ liệu: Dữ liệu có chất lượng tốt không, không bị nhiễu hay thiếu sót nhiều.

Tính đa dạng: Dữ liệu có đại diện cho nhiều trường hợp, triệu chứng và bệnh khác nhau hay không.

4.2.6 Ứng dụng trong thế giới thực:

Dự đoán bệnh bằng cách sử dụng máy học có thể áp dụng trong nhiều lĩnh vực y học, chẳng hạn như dự đoán ung thư, bệnh tim mạch, tiểu đường và nhiều bệnh khác. Thông qua việc thu thập và phân tích dữ liệu từ bệnh nhân, máy móc có thể học hỏi các mẫu và quy luật ẩn đằng sau các triệu chứng bệnh để dự đoán khả năng mắc bệnh của một người.

4.2.7 Hướng nghiên cứu:

Tối ưu hóa mô hình SVM: Nghiên cứu về cách tối ưu hóa mô hình SVM để đạt được độ chính xác cao hơn và tăng tính ổn định của dự đoán.

Kết hợp dữ liệu đa nguồn: Kết hợp dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như dữ liệu y tế, dữ liệu thể dục và dinh dưỡng để cải thiện khả năng dự đoán bệnh.

Phát triển giao diện ứng dụng: Xây dựng giao diện ứng dụng thân thiện với người dùng để họ có thể dễ dàng sử dụng và kiểm tra khả năng mắc bệnh của mình.

Xử lý dữ liệu không đầy đủ và nhiễu: Nghiên cứu về cách xử lý dữ liệu không đầy đủ hoặc bị nhiễu để đảm bảo tính chính xác của mô hình. Các phương pháp tiền xử lý như loại bỏ nhiễu, điền giá trị bị thiếu hoặc chuẩn hóa dữ liệu có thể được khám phá để cải thiện hiệu suất của mô hình.

Tích hợp dữ liệu định tính: Mở rộng phạm vi dự đoán bằng cách tích hợp dữ liệu định tính như dữ liệu về di truyền, môi trường sống, hoặc các yếu tố khác có thể ảnh hưởng đến sức khỏe. Điều này đòi hỏi phải áp dụng các phương pháp chuyển đổi dữ liệu và xây dựng mô hình phù hợp.

Nhận dạng yếu tố quan trọng: Sử dụng các kỹ thuật như feature selection hoặc feature importance để xác định những yếu tố có tác động lớn đến việc dự đoán bệnh. Việc này giúp tối ưu hóa mô hình và giảm thiểu tác động của các yếu tố không quan trọng.

Mở rộng ứng dụng cho dự đoán theo thời gian: Áp dụng học máy để dự đoán sự tiến triển của bệnh trong tương lai dựa trên dữ liệu thời gian. Điều này có thể hỗ trợ việc đưa ra các quyết định điều trị và kiểm tra định kỳ.

KẾT LUẬN

Dự đoán bệnh bằng cách sử dụng máy học, đặc biệt là mô hình SVM, đang mở ra một tương lai hứa hẹn trong lĩnh vực y học. Việc áp dụng công nghệ này có thể cách mạng hóa quá trình chẩn đoán và điều trị bệnh, giúp tăng khả năng phát hiện sớm và cải thiện chất lượng cuộc sống của con người. Tuy nhiên, việc nghiên cứu và phát triển trong lĩnh vực này còn đặt ra nhiều thách thức hấp dẫn, đòi hỏi sự hợp tác giữa các chuyên gia y học và các nhà khoa học máy.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Thị Hồng Thắm, *Tài liệu, Slide bài giảng*, Bộ môn Học máy và ứng dụng Trường Đại học Nguyễn Tất Thành, Tp. Hồ Chí Minh.

2. NTC Team (2017), *Học máy có giám sát và Học máy không giám sát*, https://www.thegioimaychu.vn/blog/ai-hpc/hoc-may-co-giam-sat-va-hoc-may-khong-giam-sat-p470/, thời gian truy cập: 22/07/2023.

3. Elcom (2022), *Máy học (Machine Learning) là gì? Ứng dụng công nghệ máy học trong thực tiễn,* https://www.elcom.com.vn/may-hoc-machine-learning-la-gi-ung-dung-cong-nghe-may-hoc-trong-thuc-tien-1666003970, thời gian truy cập: 25/07/2023.

Dữ liệu thực hiện đề tài:

QR Code data:

QR Code code: