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

HỌ VÀ TÊN : LƯƠNG QUANG TRƯỜNG

DỰ ĐOÁN TỬ VONG CỦA BỆNH NHÂN SUY TIM BẰNG MỘT SỐ MÔ HÌNH HỌC MÁY

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

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

HỌ VÀ TÊN : LƯƠNG QUANG TRƯỜNG

DỰ ĐOÁN TỬ VONG CỦA BỆNH NHÂN SUY TIM BẰNG MỘT SỐ MÔ HÌNH HỌC MÁY

Ngành : Công nghệ thông tin
Mã số: 7480201

NGƯỜI HƯỚNG DẪN THS. TRƯƠNG XUÂN NAM

HÀ NỘI, NĂM 2024

GÁY BÌA ĐỒ ÁN TỐT NGHIỆP, KHÓA LUẬN TỐT NGHIỆP

HỌ VÀ TÊN: LƯƠNG QUANG TRƯỜNG

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

	CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM **Độc lập - Tự do - Hạnh phúc** ----------★---------- **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Lương Quang Trường Hệ đào tạo : Đại học chính quy

Lớp: 62TH-VA Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1- TÊN ĐỀ TÀI:

DỰ ĐOÁN TỬ VONG CỦA BỆNH NHÂN SUY TIM BẰNG MỘT SỐ MÔ HÌNH HỌC MÁY

2- CÁC TÀI LIỆU CƠ BẢN:

3- NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung các phần	Tỉ lệ %
Chương 1: Giới thiệu về bài toán	10%
Chương 2: Tiếp cận cơ sở lý thuyết • Tìm hiểu các khái niệm về học máy • Tìm hiểu các phương pháp đánh giá mô hình • Tìm hiểu về Rừng ngẫu nhiên – Random Forest • Tìm hiểu về Logistic Regression - hồi quy Logistic • Tìm hiểu về cây quyết định – Decision Tree • Tìm hiểu về K – Nearest Neightbors	20%
Chương 3: Xây dựng mô hình học máy • Mô hình giải quyết bài toán	60%

• Các thuật toán, thư viện sử dụng • Thực nghiệm	
Chương 4: Kết quả và đánh giá • Kết quả thực nghiệm • Đánh giá	10%

4- GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Phần	Họ và tên giảng viên hướng dẫn
Chương 1: Giới thiệu về bài toán	Ths Trương Xuân Nam
Chương 2: Tiếp cận cơ sở lý thuyết	Ths Trương Xuân Nam
Chương 3: Xây dựng mô hình học máy	Ths Trương Xuân Nam
Chương 4: Kết luận	Ths Trương Xuân Nam

5- NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày .... tháng …. năm 2024.

Trưởng Bộ môn
(Ký và ghi rõ Họ tên)

Giáo viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

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

Ngày …. tháng …. năm 2023.
Chủ tịch Hội đồng
(Ký và ghi rõ Họ tên)

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

Ngày .... tháng …. năm 2023.
Sinh viên làm Đồ án tốt nghiệp
(Ký và ghi rõ Họ tên)

	TRƯỜNG ĐẠI HỌC THUỶ LỢI **KHOA CÔNG NGHỆ THÔNG TIN** **BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP**

TÊN ĐỀ TÀI:

DỰ ĐOÁN TỬ VONG CỦA BỆNH NHÂN SUY TIM BẰNG MỘT SỐ MÔ HÌNH HỌC MÁY.

Sinh viên thực hiện: Lương Quang Trường

Lớp: 62TH – VA

Giáo viên hướng dẫn: ThS. Trương Xuân Nam

TÓM TẮT ĐỀ TÀI

Suy tim là một tình trạng y tế nghiêm trọng, trong đó tim không hoạt động hiệu quả để cung cấp đủ máu cho cơ thể. Hiện nay, bệnh suy tim là một vấn đề y tế quan trọng trên khắp thế giới. Dù đã có sự tiến bộ trong chuẩn đoán và điều trị, nhưng suy tim vẫn là nguyên nhân hàng đầu gây ra tử vong và suy giảm chất lượng cuộc sống cho nhiều người.

Sự ra đời của các thuật toán học máy đã giúp cho lĩnh vực y tế có một bước tiến nhảy vọt trong việc dự đoán và điều trị bệnh. Bài toán đặt ra là liệu có thể sử dụng các thuật toán học máy dựa vào dữ liệu đã có, từ đó đưa ra dự đoán về khả năng tử vong của của người bệnh. Sau đó đưa ra các phương pháp điều trị một cách hiệu quả, và giảm tỉ lệ tử vong cho bệnh nhân. Vì vậy em chọn đề tài "Dự đoán tỉ lệ tử vong của bệnh nhân suy tim bằng một số mô hình học máy".

Để thực hiện đề tài, em sẽ nghiên cứu các mô hình học máy như Logistic Regresstion, KnearestNeighbours, Decision Tree và Random Forest. Ngoài ra, em có sử dụng các phương pháp phân tích và tiền xử lý dữ liệu, các phương pháp đánh giá mô hình, ...

CÁC MỤC TIÊU CHÍNH

- Mục tiêu 1: Tìm hiểu bộ dữ liệu cho bài toán.
- Mục tiêu 2: Tìm hiểu, nghiên cứu các mô hình học máy.
- Mục tiêu 3: Giải quyết các bài toán nghiệm vụ như tiền xử lý dữ liệu, phân tích dữ liệu, ...
- Mục tiêu 4: Áp dụng các mô hình học máy để dự đoán.
- Mục tiêu 5: Đánh giá mô hình.
- Mục tiêu 6: Kết luận.

KẾT QUẢ DỰ KIẾN

- Hoàn thành các mục tiêu đã đề ra.
- Hiểu và xây dựng các mô hình học máy áp dụng cho bài toán.
- Thực nghiệm mô hình bằng Python.
- Báo cáo tổng kết.

LỜI CAM ĐOAN

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

Tác giả ĐATN

Lương Quang Trường

LỜI CÁM ƠN

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Thủy Lợi đã tạo điều kiện cho em thực hiện Đồ án tốt nghiệp.

Em xin gửi lời cảm ơn chân thành đến tất cả các thầy, cô đã giảng dạy chúng em trong suốt thời gian qua. Cảm ơn thầy Trương Xuân Nam - người đã hướng dẫn em thực hiện đồ án này. Em cảm ơn thầy đã giúp đỡ, bổ sung cho em những kiến thức và cho em những lời khuyên, gợi ý để em có thể hoàn thành đồ án một cách nhanh chóng và hiệu quả nhất.

Trong quá trình học tập và thực hiện đồ án em đã may mắn được sự chỉ bảo, hướng dẫn tận tình của các thầy cô giáo và được gia đình, bạn bè quan tâm, động viên, luôn ở bên và tạo mọi điều kiện thuận lợi để hoàn thành tốt đồ án này. Trong suốt quá trình làm đồ án với đề tài "Dự đoán tử vong của bệnh nhân suy tim bằng một số mô hình học máy", em đã nỗ lực hết sức để xây dựng và hoàn thiện đồ án một cách tốt nhất, nhưng do kiến thức còn hạn chế và thiếu kinh nghiệm thực tế nên không thể tránh những sai sót. Một lần nữa, em xin chân thành cảm ơn thầy cô giáo, bạn bè và gia đình, những người đã giúp đỡ, ủng hộ em trong thời gian vừa qua.

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

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

TH: trường hợp

ML – Machine Learing: Học máy.

LR – Logistic Regression: Thuật toán hồi quy Logistic.

KNN – K-NearestNeightbors: Thuật toán K-NearestNeightbors.

DT – Decision Tree: Thuật toán cây quyết định.

RF – Random Forest: Thuật toán rừng ngẫu nhiên.

TP – True Positive: Mẫu thực sự dương và được mô hình dự đoán là dương.

FP – False Positive: Mẫu thực sự âm và được mô hình dự đoán là dương.

TN – True Negative: Mẫu thực sự âm và được mô hình dự đoán là âm.

FN – False Negative: Mẫu thực sự dương và được mô hình dự đoán là âm.

ACC – Accuracy: Độ chính xác accuracy.

CHƯƠNG 1 GIỚI THIỆU

1.1 Lý do chọn đề tài

Trong những năm gần đây, các biến chứng về sức khỏe nói chung và suy tim nói riêng đã trở thành một ván đề sức khỏe cộng đồng đáng quan ngại trên toàn cầu. Tỉ lệ mắc bệnh và tử vong do suy tim ngày càng gia tăng, gây áp lực lớn lên hệ thống y tế và ảnh hưởng nghiêm trọng đến chất lượng cuộc sống của người bệnh. Suy tim không chỉ là gánh nặng của bản thân người bệnh và gia đình, nó còn tác động tiêu cực đến kinh tế xã hội do chi phí điều trị cao và giảm năng suất lao động. Vì vậy, việc tìm ra giải pháp hiệu ủa để ngăn ngừa, chuẩn đoán sớm và điều trị suy tim là một ưu tiên hàng đầu trong lĩnh vực y tế hiện nay.

Suy tim là một hội chứng lâm sàng phức tạp, xảy ra khi tim không thể bơm đủ máu để đáp ứng nhu cầu của cơ thể. Tình trạng này có thể ảnh hưởng đến cả hai bên tim hoặc một trong hai và bệnh có thể tiến triển từ từ hoặc đột ngột. Có rất nhiều nguyên nhân dẫn đến suy tim như bệnh mạch vành, béo phì, hút thuốc lá, hay như lạm dụng rượu bia.

Việc chuẩn đoán và điều trị suy tim sớm là rất quan trọng để cải thiện tiên lượng bệnh và giảm nguy cơ tử vong. Tuy nhiên, việc chuẩn đoán suy tim vẫn còn nhiều thách thức lớn như triệu chứng không đặc hiệu và dễ gây nhầm lẫn với các bệnh lý khác như mệt mỏi, khó thở,... Do đó, việc nghiên cứu và phát triển các phương pháp chuẩn đoán và dự đoán suy tim mới, chính xác và hiệu quả hơn chẳng hạn như sử dụng công cụ chuẩn đoán hình ảnh tiên tiến, xét nghiệm sinh học mới hoặc ứng dụng công nghệ như áp dụng trí tuệ nhân tạo là điều rất cần thiết để cải thiện khả năng phát hiện sớm và điều trị kịp thời cho các bệnh nhân suy tim.

Đồ án này nhằm mục tiêu đóng góp vào nỗ lực chung trong việc giải quyết vấn đề suy tim bằng cách phân tích các yếu tố nguy cơ và xây dựng mô hình dự đoán nguy cơ tử vong do suy tim dựa trên các đặc điểm lâm sàng và cận lâm sàng của bệnh nhân. Em hy vọng rằng kết quả này sẽ cung cấp thông tin hữu ích cho các bác sĩ trong việc chuẩn đoán, đánh giá nguy cơ, lựa chọn phương pháp điều trị và quản lý bệnh nhân suy tim, từ đó cải thiện chất lượng cuộc sống và giảm tỉ lệ tử vong của bệnh.

1.2 Mục tiêu

Suy tim đang nổi lên như một vấn nạn sức khỏe đáng báo động trên toàn cầu, không chỉ riêng tại Việt Nam. Với hàng triệu người mắc phải và con số tử vong ngày càng tăng cao, suy tim không chỉ ảnh hưởng nặng nề đến sức khỏe và chất lượng cuộc sống của người bệnh mà còn tạo ra gánh nặng kinh tế to lớn cho gia đình bệnh nhân và xã hội. Tại Việt Nam, tình hình càng trở nên cấp bách hơn do dân số đang già hóa nhanh chóng và các yếu tố nguy cơ như tăng huyết áp, tiểu đường, béo phì đang ngày càng phổ biến. Do đó, việc nghiên cứu và phát triển các giải pháp chuẩn đoán, dự đoán và điều trị suy tim hiệu quả là vô cùng cần thiết để giảm thiểu những tác động tiêu cực này và cải thiện sức khỏe cộng đồng, nâng cao tuổi thọ người dân.

Mô hình dự đoán nguy cơ tử vong do suy tim không chỉ dừng lại ở việc hỗ trợ chuẩn đoán và tiên lượng bệnh, mà còn mở ra nhiều tiềm năng ứng dụng trong thực tế lâm sàng. Mô hình này có thể giúp các bác sĩ đưa ra những đánh giá chính xác hơn về tình trạng bệnh, xác định các yếu tố nguy cơ tiềm ẩn và tiên lượng khả năng tử vong của bệnh nhân. Từ đó, việc cá nhân hóa phác đồ, liệu trình điều trị trở nên khả thi, tối ưu hóa hiệu quả điều trị và giảm thiểu tác dụng phụ không mong muốn. Hơn nữa, mô hình này còn có thể được sử dụng để theo dõi và quản lý bệnh nhân suy tim ngoại trú, giúp phát hiện sớm các dấu hiệu xấu đi và can thiệp kịp thời, giảm nguy cơ nhập viện và tử vong. Cuối cùng, dữ liệu từ mô hình dự đoán còn có thể đóng vai trò quan trọng trong việc đánh giá hiệu quả của các phương pháp điều trị mới, từ đó thúc đẩy nghiên cứu và phát triển các loại thuốc và liệu pháp điều trị suy tim hiệu quả hơn.

Nghiên cứu này hoàn toàn khả thi nhờ vào sự sẵn của dữ liệu bệnh nhân suy tim. Việc ứng dụng các kỹ thuật học máy tiên tiến vào phân tích và trích xuất thông tin hữu ích từ nguồn dữ liệu này sẽ giúp xây dựng một mô hình dự doán chính xác. Bên cạnh đó, sự phát triển vượt bậc của công nghệ thông tin và trí tuệ nhân tạo đã đã cung cấp những công cụ mạnh mẽ để xử lý và phân tích dữ liệu, xây dựng và tối ưu các mô hình học máy phức tạp.

Đề tài này hoàn toàn phù hợp với chuyên ngành được đào tạo. Nó giúp áp dụng những kiến thức và kỹ năng đã được học vào việc giải quyết một vấn đề thực tiễn có ý nghĩa xã hội. Cụ thể, tôi sẽ sử dụng các kỹ năng phân tích dữ liệu như làm sạch dữ liệu, khám

phá dữ liệu và trực quan hóa dữ liệu để hiểu rõ hơn về đặc điểm của bệnh nhân suy tim và các yếu tố nguy cơ. Đồng thời, áp dụng các kỹ năng xây dựng mô hình học máy như lựa chọn đặc trưng, huấn luyện mô hình và đánh giá mô hình để xây dựng một mô hình dự đoán chính xác và hiệu quả. Qua đó, tôi không chỉ củng cố kiến thức chuyên môn mà còn có cơ hội đóng góp vào việc cải thiện sức khỏe cộng đồng.

1.3 Đối tượng, phạm vi nghiên cứu

1.3.1 Đối tượng nghiên cứu

Bài toán "Dự đoán tử vong của bệnh nhân suy tim bằng một số mô hình học máy" dựa trên thông tin cá nhân (tuổi, giới tính), tình trạng sức khỏe (thiếu máu, bệnh tiểu đường, huyết áp cao, hút thuốc), chỉ số y tế, thời gian theo dõi của từng người bệnh.

1.3.2 Phạm vi nghiên cứu

Trong đề tài này, em sử dụng ngôn ngữ lập trình Python kết hợp với các thư viện hỗ trợ như Pandas, Numpy, Scikit-Learn để cài đặt một vài mô hình học máy nhằm dự đoán tỉ lệ tử vong của bệnh nhân suy tim. Dữ liệu được cung cấp bởi các bác sĩ bao gồm các thông số y tế như tuổi, bệnh thiếu máy, hàm lượng creatinine phosphokinase, bệnh tiểu đường, phân suất tống máu, huyết áp cao, số lượng tiểu cầu,… và sự kiện tử vong.

Sau khi xem xét tập dữ liệu, em sẽ sử dụng các thuật toán học máy như hồi quy logistic, KNN, cây quyết định để triển khai mô hình dự đoán tỉ lệ tử vong của các bệnh nhân.

Kết quả của mô hình sẽ được đánh giá dựa trên các chỉ số hiệu suất như độ chính xác, độ nhạy,…

1.4 Phương pháp nghiên cứu

Trong bài toán, em sẽ triển khai theo các bước sau:

- Tiền xử lý dữ liệu: bao gồm việc làm sạch dữ liệu như xử lý giá trị thiếu và dữ liệu không hợp lệ, biến đổi và chuẩn hóa dữ liệu như đưa về một phạm vi chung của để đảm bảo cho các đặc trưng có tầm quan trọng tương đương và cuối cùng là mã hóa dữ liệu
- Chia tập dữ liệu thành các tập huấn luyện, tập kiểm tra.
- Xây dựng các mô hình học máy: Mô hình được đề xuất như LR, KNN, DT, RR.

- Đánh giá hiệu quả của mô hình dựa trên các chỉ số như Accuracy, F1 Score, Recall, Precision.
- Chọn ra mô hình tối ưu nhất đối với bài toán đề ra.

CHƯƠNG 2 TIẾP CẬN CƠ SỞ LÝ THUYẾT

2.1 Học máy

2.1.1 Tổng quan về học máy

Học máy (machine learing) là một lĩnh vực của trí tuệ nhân tạo (AI – Artificial Intelligence) tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính tự học từ dữ liệu và cải thiện hiệu suất của chúng trong việc thực hiện các nhiệm vụ cụ thể mà không cần được lập trình rõ ràng. Khái niệm học máy bắt nguồn từ ý tưởng rằng máy tính có thể tự học và cải thiện từ kinh nghiệm, tương tự như cách con người học từ thực tế và tích lũy kiến thức.

Học máy có ba loại chính: học có giám sát (supervised learning), học không giám sát (unsupervised learning) và học tăng cường (reinforcement learning).

Trong học có giám sát, mô hình được huấn luyện trên một tập dữ liệu bao gồm các đầu vào và các đầu ra mong muốn, từ đó học cách ánh xạ các đầu vào đến các đầu ra một cách chính xác. Các thuật toán phổ biến trong học có giám sát bao gồm hồi quy tuyến tính, hồi quy logistic, cây quyết định, rừng ngẫu nhiên, …

Trong học máy không giám sát, mô hình được huấn luyện trên một tập dữ liệu chỉ bao gồm các đầu vào mà không có các đầu ra mong muốn. Mục tiêu của học máy không giám sát là khám phá ra các cấu trúc hoặc mẫu ẩn trong dữ liệu. Các thuật toán phổ biến đối với cách học này có thể kể đến như phân cụ K-means, phân tích thành phần chính PCA, …

Học tăng cường là một phương pháp học mà trong đó môt tác nhân (agent) học cách hành động trong một môi trường bằng cách thực hiện các hành động và nhận phần thưởng hoặc hình phạt. Mục tiêu của tác nhân là tối đa hóa tổng phần thưởng nhận được theo thời gian. Học tăng cường thường được áp dụng trong các lĩnh vực thực như chơi game, điều kiểu robot, …

Một phần quan trọng của học máy là tiền xử lý dữ liệu, nơi mà dữ liệu được đi qua các quá trình như làm sạch dữ liệu, xử lý các giá trị thiếu, chuẩn hóa các biến số hoặc có thể

là tạo ra các đặc trưng mới từ dữ liệu gốc. Việc xử lý dữ liệu đúng cách có thể cải thiện đáng kể hiệu xuất của các mô hình học máy này.

Sau khi dữ liệu được xử lý, một bước quan trọng sẽ là lựa chọn và huấn luyện các mô hình học. Quá trình này có thể bao gồm việc chia tập dữ liệu thành các tập huấn luyện và kiểm thử (hoặc có thể cần thêm tập xác thực). Sau đó sử dụng tập huấn luyện để đào tạo mô hình và dùng tập kiểm thử để đánh giá hiệu suất. Việc đánh giá này thường dựa trên các chỉ số như đánh giá về độ chính xác, độ nhạy, …

Những năm gần đây, học máy nói riêng và trí tuệ nhân tạo nói chung đã được ứng dụng nhiều vào các lĩnh vực khác nhau của đời sống. Trong y tế, học máy được sử dụng để chuẩn đoán bệnh, dự đoán quá trình phát triển của bệnh hoặc là cá nhân hóa việc điều trị cho bệnh nhân. Trong tài chính, học máy có thể được sử dụng để phát hiện các sai phạm, dự báo thị trường, hoặc quản lý rủi ro. Trong thương mại, nó có thể được ứng dụng trong việc cải thiện trải nghiệm người dung qua các chức năng như đề xuất sản phẩm, phân tích hành vi khác hang, tối ưu hóa chiến dịch tiếp thị.

Ngoài những nổi bật ở trên, học máy cũng đối diện với các thách thức lớn như việc đảm bảo minh bạch dữ của dữ liệu, duy trì bảo mật, quyền riêng tư của dữ liệu hoặc việc xử lý các dữ liệu lớn. Ngoài ra, việc huấn luyện các mô hình lớn đòi hỏi nhiều tài nguyên tính toán hoặc có thể tiêu tốn nhiều thời gian.

Tổng kết lại, học máy là một lĩnh vực quan trọng và phát triển rất nhanh trong lĩnh vực trí tuệ nhân tạo, với tiềm năng to lớn để thay đổi nhiều ngành công nghiệp và cải thiện cuộc sống hàng ngày của con người. Với sự tiến bộ không ngừng của công nghệ và khả năng tiếp cận dữ liệu, học máy hứa hẹn sẽ tiếp tục mang lại những phát minh và ứng dụng đột phá trong tương lai.

2.1.2 Một số phương pháp học máy

2.1.2.1 Học máy có giám sát (Supervised Learning)

Học máy có giám sát (Supervised Learning) là một trong những phương pháp chính trong lĩnh vực học máy và được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Phương pháp này dựa trên việc sử dụng các dữ liệu đầu vào (training dataset) đã được gán nhãn tức là với mỗi dữ liệu đầu vào, đều có một kết quả mong muốn tương ứng

nhằm giúp máy tính có thể học cách ánh xạ từ các đặc điểm của dữ liệu đó đến đầu ra kết quả bằng cách tối ưu hóa một hàm mục tiêu (hàm mất mát – loss function). Hàm mục tiêu này đo lường sự khác biệt giữa đầu ra của mô hình và kết quả thực tế. Mô hình sẽ tự điều chỉnh các tham số để giảm thiểu giá trị của hàm mục tiêu này.

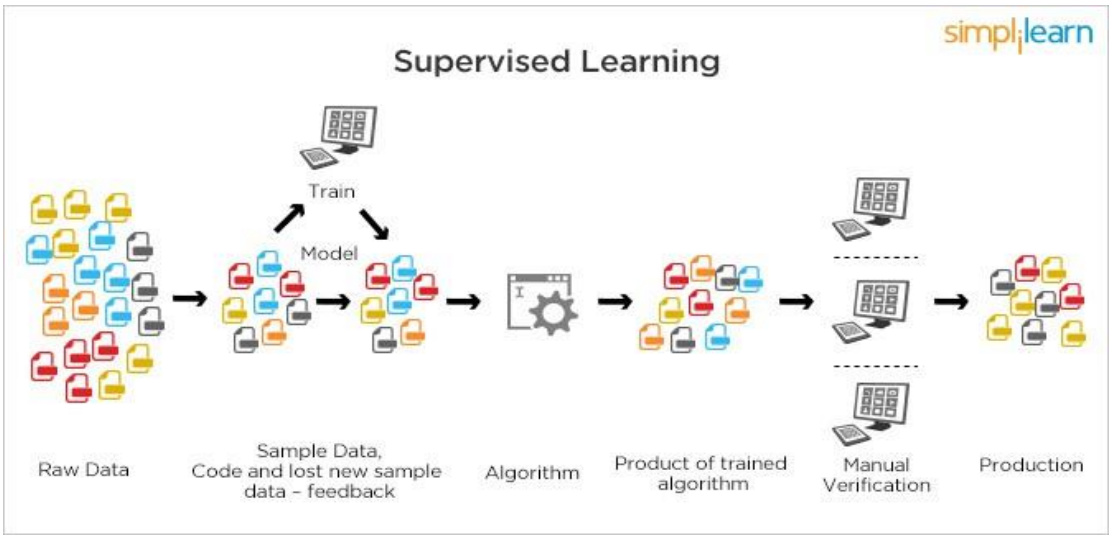
Học máy có giám sát (Supervised Learning) là một kỹ thuật trong lĩnh vực trí tuệ nhân tạo, nơi máy tính được huấn luyện để đưa ra dự đoán hoặc quyết định dựa trên dữ liệu đã được gắn nhãn. Dữ liệu gắn nhãn này gao gồm các cặp đầu vào và đầu ra mong muốn, giúp máy tính học cách ánh xạ từ đầu vào đến đầu ra chính xác.

Sau khi mô hình được huấn luyện, nó sẽ được đưa vào quá trình kiểm thử dựa trên một tập dữ liệu chưa từng được đưa vào trong quá trình huấn luyện. Điều này giúp đánh giá khả năng của mô hình trong việc tổng quá hóa và dự đoán chính xác các dữ liệu mới. Các chỉ số đánh giá như độ chính xác (accuracy), độ nhạy (recall),… thường được sử dụng để đánh giá hiệu suất của mô hình.

Các thuật toán học máy có giám sát phổ biến:

- Hồi quy tuyến tính (Linear Regression): Mô hình này học cách tìm ra mối quan hệ tuyến tính giữa các biến đầu vào và đầu ra bằng cách tối hiểu hóa hàm mất mát, mô hình dùng trong bài toán dự đoán các giá trị liên tục như dự đoán giá nhà, doanh số bán hàng, nhiệt độ, lượng mưa,…
- Cây quyết định (Decision Tree): Mô hình này có thể sử dụng với các bài toán phân loại hoặc hồi quy, về cơ bản, mô hình sử dụng cấu trúc cây để đưa ra quyết định dựa trên đặc trưng của dữ liệu. Mỗi nút trên cây đại diện cho một đặc trưng và các nhánh đại diện cho một giá trị hoặc một khoảng giá trị nào đó. Mô hình này được ứng dụng nhiều trong các bài toán về y tế, phân loại khách hàng, phân tích rủi ro.
- Hồi quy Logistic (Logistic Regression): Mô hình sử dụng kỹ thuật thống kê dùng để dự đoán khả năng xảy ra của một biến nhị phân đựa trên một hoặc nhiều biến độc lập. Nó thường được sử dụng trong các lĩnh vực như y tế, tài chính và nghiên cứu xã hội với mục đích dự đoán xác suất xảy ra của một sự kiện cụ thể nào đó như có bệnh hay không hoặc khả năng phá sản của một công ty,...

Ngoài ra còn rất nhiều thuật toán khác.



Hình 2.1: Mô hình học máy có giám sát

2.1.2.2 Học máy không giám sát (Unsupervised Learing)

Học không giám sát là một nhánh quan trọng trong lĩnh vực học máy, đóng vai trò then chốt trong việc khám phá tri thức ẩn trong dữ liệu. Đặc điểm của phương pháp này là máy tính được cung cấp một tập dữ liệu không có nhãn (không có kết quả mong muốn được gắn sẵn), điều này hữu ích trong các tính huống mà việc gắn nhãn là tốn kém, mất thời gian hoặc không khả thi. Trong phương pháp học này, máy tính sẽ tự tìm cách đưa ra cấu trúc, mô hình hoặc các mối quan hệ ẩn trong dữ liệu đó.

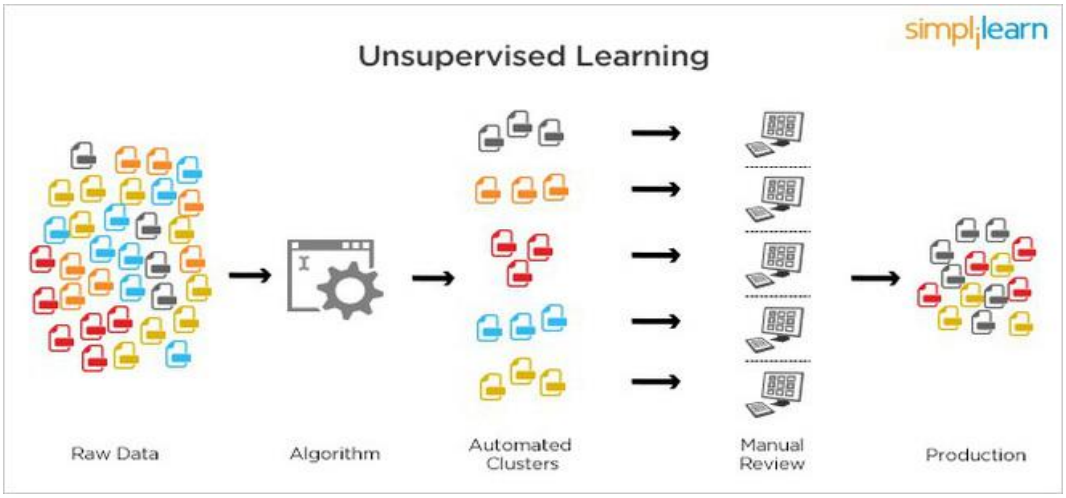
Với cách học này, mô hình nhận vào dữ liệu thô và cố gắng tìm ra đặc trưng của dữ liệu. Không giống như học có giám sát, mô hình không có một hàm mục tiêu cụ thể để tối ưu, thay vào đó thuật toán sẽ tìm các cách hiệu quả nhất để nhóm dữ liệu hoặc giảm chiều của dữ liệu.

Đối với học không giám sát, các thuật toán học thường được sử dụng như:

- Phân cụm: Nhóm các điểm dữ liệu thành các cụm sao cho các điểm trong cụm có các đặc điểm giống nhau hơn so với cụm khác. Các thuật toán chính thường thấy như K-means, DBSCAN, …
- Giảm chiều dữ liệu.
- Phát hiện bất thường.

Học máy không giám sát cũng có thách thức và hạn chế nhất định, ví dụ như việc xác định số lượng cụm là rất cần thiết. Điều này có thể gây khó khăn và có thể cần đến sự

can thiệp của con người. Ngoài ra việc đánh giá hiệu suất của mô hình có thể phức tạp hơn so với học có giám sát vì không có nhãn để so sánh trực tiếp.



Hình 2.2: Mô hình học máy không giám sát

2.1.2.3 Học tăng cường (Reinforcement Learning)

Học tăng cường là một lĩnh vực con của học máy, tập trung vào việc đào tạo các tác nhân thông minh học các tương tác với môi trường xung quanh để đạt được một mục tiêu cụ thể nào đó. Quá trình học này không dựa trên một hướng dẫn rõ ràng nào mà thông qua việc thử nghiệm và sai sót, giống như cách con người học hỏi từ kinh nghiệm.

Tác nhân trong học tăng cường được thiết kế để đưa ra quyết định dựa trên trạng thái hiện tại của môi trường xung quanh và nhận được phần thưởng hoặc hình phạt tương ứng với một hoặc các hành động đã chọn. Mục tiêu của tác nhân trong môi trường là tìm ra chiến lược tối ưu để tối đa hóa mục thành tích có thể tích lũy trong quá trình tương tác.

Học tăng cường đã chứng minh được sức mạnh của mình thông qua các thành tựu đáng kinh ngạc trong lĩnh vực trò chơi. Có một số chương trình AI sử dụng học tăng cường đã vượt qua con người trong các trò chơi phức tạp như cờ vây, cờ vua hoặc các trò chơi điện tử khác. Ngày nay, ứng dụng của học tăng cường được sử dụng để giải quyết nhiều vấn đề thực tế như robot, điều khiển tự động.

Hình 2.3: Tuyển thủ Garry Kasparov thi đấu với Deep Blue.

2.1.2.4 Các bước xây dựng một mô hình học máy

Để xây dựng được một mô hình học máy chính xác và được đưa vào ứng dụng trong thực tế, chúng ta cần phải trải qua các quy trình nghiêm ngặt và chi tiết. Bước đầu tiên là thu thập dữ liệu chất lượng và đủ lớn, đủ tổng quát để mô hình có thể học được từ các mẫu dữ liệu. Tuy nhiên trong thực tế, khi thu thập dữ liệu không thể tránh khỏi việc thiếu dữ liệu, dữ liệu có nhiễu hoặc dữ liệu không phù hợp. Do đó, quá trình tiền xử lý dữ liệu là vô cùng quan trong. Quá trình này bao gồm các việc như làm sạch dữ liệu, xử lý giá trị thiếu, chuẩn hóa dữ liệu,…

Sau khi qua bước tiền xử lý, dữ liệu cần được phân chia thành các tập như tập huấn luyện, tập kiểm tra và tập xác thực. Thông thường, khoảng 70 – 80% dữ liệu được đưa vào huấn luyện và phần còn lại được sử dụng để kiểm tra và xác thực. Việc phân chia này giúp đảm bảo mô hình không bị quá khớp (overfitting) và có khả năng tổng quá hóa tốt trên dữ liệu mới. Tiếp theo đó là việc lựa chọn mô hình học máy phù hợp với bài toán cần giải quyết. Các mô hình phổ biến đều tùy thuộc vào tính chất dữ liệu và yêu cầu cụ thể của bài toán.

Quá trình huấn luyện mô hình là việc sử dụng mô tập huấn luyện để dạy mô hình cách ánh xạ dữ liệu đầu vào với kết quả tương ứng (đối với mô hình học có giám sát) và tìm ra các đặc trưng của dữ liệu (đối với mô hình học không giám sát). Với mô hình học có giám sát, quá trình này bao là việc tối ưu hóa các tham số của mô hình để giảm thiểu sai

số cho việc dự đoán. Sau khi huấn luyện, mô hình được đánh giá bằng cách sử dụng tập kiểm tra để xác định hiệu suất. Các chỉ số đánh giá ở đây có thể là độ chính xác, độ nhạy, F1-score,… giúp chúng ta có thể có cái nhìn toàn diện về khả năng dự đoán của mô hình.

Nếu mô hình chưa đạt yêu cầu, cần phải điều chỉnh các tham số và thử nghiệm các kỹ thuật cải thiện mới để tối ưu hóa về hiệu suất. Khi mô hình đạt được kết quả mong muốn, bước tiếp theo là đưa vào thực tế sử dụng.

Sau khi triển khai, việc giám sát và bảo trì mô hình là không thể thiếu. Cần phải theo dõi hiệu suất của mô hình một cách thường xuyên và đưa ra cập nhật cần thiết để đảm bảo được độ chính xác của mô hình.

2.1.3 Một số khái niệm trong học máy

Dữ liệu (Data): Là thông tin được thu thập từ các quan sát, đo lường hoặc các nguồn khác nhau, có thể được sử dụng để phân tích, diễn giải,… Dữ liệu có thể ở nhiều dạng khác nhau, bao gồm số, văn bản, hình ảnh, âm thanh,… và có thể đến từ nhiều nguồn như cảm biến, thiết bị báo đài, cơ sở dữ liệu,... Trong học máy, dữ liệu là nền tảng và là yếu tố cốt lõi giúp các mô hình học máy có thể học và đưa ra dự đoán.

Dữ liệu có thể được phân thành hai loại chính: dữ liệu có cấu trúc và dữ liệu phi cấu trúc. Dữ liệu có cấu trúc là dữ liệu được tổ chức theo một cấu trúc rõ ràng và có thể lưu trữ và truy xuất trong các hệ thống cơ sở dữ liệu. Ngược lại, dữ liệu phi cấu trúc không có cấu trúc rõ ràng và không dễ dàng để lưu trữ trong các cơ sở dữ liệu truyền thống. Ví dụ về dữ liệu phi cấu trúc có thể là văn bản tự do, video, ảnh, email.

Dữ liệu không chỉ là nền tảng mà còn đóng vai trò quan trọng trong học máy. Nó có thể quyết định đến sự thành công của một mô hình.

Để có được dữ liệu tốt cho học máy, cần phải đảm bảo dữ liệu đầu vào là đầy đủ, chính xác và liên quan đến các vấn đề đang được giải quyết. Ngoài ra, cần chuẩn bị dữ liệu phù hợp với bài toán và định dạng dữ liệu đúng cách.

Để đảm bảo tính đầy đủ của dữ liệu, chúng ta cần đảm bảo dữ liệu đầu vào có thể bao quát tất cả các giá trị có thể xảy ra, không bị thiếu, không có lỗi hoặc sai sót và liên quan đến vấn đề đang được xử lý.

Dữ liệu là yếu tố quan trọng trong học máy, đóng góp từ việc huấn luyện mô hình, giám sát mô hình và cải thiện mô hình.

	PhiXSectContin	PixelColor	NeighbColorGrad	Betw2Amplify	Lable
0	0	251	64	0	Solid
1	1	78	19	1	thraot
2	0	138	29	0	NC_Vugs
3	0	133	35	1	NC_Vugs
4	1	185	45	0	Solid
5	1	96	84	0	Pore
6	0	238	43	1	Solid
7	0	155	51	1	Solid
8	0	213	67	1	Solid
9	1	185	67	1	thraot
10	0	65	81	0	NC_Vugs
11	0	129	30	0	NC_Vugs
12	1	176	66	0	Solid
13	0	10	47	0	NC_Vugs
14	0	137	45	1	NC_Vugs
15	0	85	12	0	NC_Vugs
16	0	260	26	1	Solid
17	1	155	22	0	Solid
18	1	206	53	0	Solid
19	1	187	51	1	thraot

Hình 2.4: Ví dụ về dữ liệu trong học máy

Học máy là quá trình máy tính tự "học" bằng cách thay đổi các tham số của mô hình từ dữ liệu (học từ kinh nghiệm). Do vậy việc tìm hiểu dữ liệu là rất cần thiết. Trong đồ án này, dữ liệu là như một bảng gồm các hàng và cột được lấy ra từ tập excel. Đây là một cấu trúc dữ liệu cơ bản trong Học máy.

Mẫu (sample): Là một điểm dữ liệu riêng lẻ trong tập dữ liệu, như một dòng trong bảng dữ liệu Excel hoặc một hình ảnh trong bộ sưu tập hình hay như là một email, ... Mỗi mẫu này gồm nhiều đặc trưng riêng về các thuộc tính hoặc đặc điểm của dữ liệu. Có thể kèm thêm nhãn của dữ liệu đối với các bài toán học máy có giám sát. Các mẫu này có thể biểu diễn bằng các dạng dữ liệu khác nhau như số, chuỗi, ảnh, âm thanh,… Một số bài toán học máy yêu cầu người dung phải định dạng lại dữ liệu đầu vào thành một định dạng cụ thể trước khi đưa vào mô hình.

Đặc trưng (feature): Là một thuộc tính hoặc đặc điểm của dữ liệu. Mỗi mẫu có thể có một hoặc nhiều đặc trưng. Ví dụ như về tập dữ liệu về khách hàng có thể bao gồm tuổi, giới tính, nghề nghiệp,… đây là một số đặc trưng là dữ liệu quan sát được và có thể bao gồm một số đặc trưng cần được dự đoán. Đặc trưng là yếu tố quan trọng mà mô hình học máy dựa vào để học hỏi và đưa ra dự đoán. Mỗi đặc trưng cần đại diện cho một khía cạnh hoặc đặc điểm cụ thể của dữ liệu và có thể ảnh hưởng trực tiếp đến kết quả của một mô hình.

Kiểu dữ liệu (data type): Các đặc trưng thường có một kiểu dữ liệu xác định, có thể là số nguyên, số thực, hoặc kiểu rời rạc. Ngoài ra trong thực tế cũng xuất hiện các kiểu dữ liệu như ngày, chuỗi ký tự. Việc hiểu rõ về kiểu dữ liệu là cần thiết để đảm bảo hiệu quả làm việc với dữ liệu. Đôi khi chúng ta cần chuyển đổi định dạng của dữ liệu phù hợp với yêu cầu của bài toán.

Việc lựa chọn kiểu dữ liệu phù hợp là rất quan trọng trong học máy vì nó có thể ảnh hưởng đến hiệu xuất của mô hình học máy.

Tập dữ liệu (Datasets): là một tập hợp các thể hiện dữ liệu. Một vài tập dữ liệu sẽ được dùng cho các mục đích khác nhau hoặc từ một tập dữ liệu chúng ta có thể chia ra thành các tập nhỏ hơn dành cho từng mục đích riêng trong việc xây dựng mô hình như tập train, tập test và tập validation. Với các tập dữ liệu nhỏ, chúng ta có thể sử dụng kỹ thuật như k-fold để có thể đào tạo được mô hình tốt hơn.

Để có một tập dữ liệu tốt, khi dữ liệu được lấy từ nhiều nguồn khác nhau cần phải qua bước tiền xử lý dữ liệu như chuẩn hóa dữ liệu, đưa dữ liệu về cùng một phạm vi nhất định, xử lý dữ liệu thiếu, và mã hóa dữ liệu phân loại từ dạng chữ về dạng số.

Tập dữ liệu huấn luyện (Training dataset): Tập dữ liệu huấn luyện là tập dữ liệu được sử dụng với mục tiêu đào tạo cho mô hình học máy. Nó là một phần rất quan trọng cho việc mô hình học máy do nó giúp mô hình học các quan hệ giữa các đặc trưng và nhãn của dữ liệu để đưa ra một bộ tham số tối ưu. Nói một cách khác, mô hình/thuật toán Học máy sẽ học từ tập dữ liệu này.

Tập huấn luyện phải có đủ mức độ đa dạng để giúp mô hình học có khả năng khái quát tốt cho các dữ liệu mới. Nếu tập huấn luyện không đủ đa dạng, mô hình có thể dễ dàng bị overfitting (quá khớp) và không có khả năng khái quát tốt cho các dữ liệu mới.

Tập huấn luyện cũng phải được chọn một cách cân bằng giữa các nhãn lớp sao cho tránh trường hợp một số nhãn quá ít hoặc quá nhiều so với các nhãn khác. Nếu số lượng các mẫu dữ liệu của một nhãn quá ít, mô hình có thể không có khả năng học được các quan hệ giữa các đặc trưng và nhãn cho nhãn đó từ đó dẫn đến việc mô hình dự đoán kết quả của nhãn nhiều hơn với tỉ lệ đúng cao hơn so với nhãn lớp có số lượng ít hơn.

Để chọn một tập tập huấn luyện tốt, người dùng có thể thực hiện các bước sau:

- Tìm hiểu về bài toán: Trước khi chọn tập huấn luyện, người dùng nên tìm hiểu kỹ về bài toán mà mình muốn giải quyết, để có thể chọn được tập dữ liệu phù hợp nhất về các yếu tố như số lượng điểm dữ liệu, số lượng điểm dữ liệu trên từng nhãn.
- Chọn nguồn dữ liệu: Người dùng có thể chọn dữ liệu từ các nguồn khác nhau như các cơ sở dữ liệu, các tập dữ liệu mở.
- Đối chiếu với yêu cầu của bài toán: Người dùng nên đối chiếu tập dữ liệu với yêu cầu của bài toán để xác định xem tập dữ liệu có đủ đa dạng hay không, khái quát được tất cả các đặc trưng về bài toán của mình cần hướng đến hay không.
- Đánh giá tính toàn vẹn của tập dữ liệu: Người dùng cần đưa ra đánh giá xem tập dữ liệu có chứa đủ số lượng và loại dữ liệu khác nhau hay không.

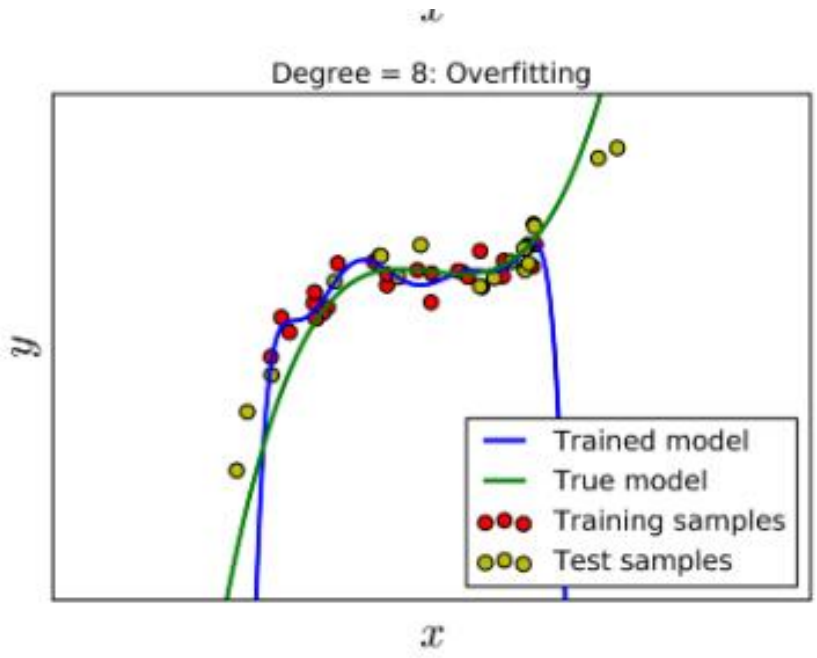
Tập dữ liệu kiểm tra (Testing dataset): Testing dataset là tập dữ liệu được sử dụng để đánh giá hiệu xuất của mô hình học máy sau khi đã được huấn luyện bằng tập dữ liệu dành cho việc đào tạo mô hình. Tập dữ liệu kiểm tra được sử dụng để xác định xem mô hình có khả năng khái quát hóa tốt cho các dữ liệu mới hay không.

Tập dữ liệu kiểm tra phải đạt đủ mức độ đa dạng để giúp mô hình được đánh giá một cách chính xác về khả năng khái quát hóa. Nếu tập dữ liệu kiểm tra không đủ đa dạng, mô hình có thể dễ dàng bị underfitting (quá ít khớp) và không được đánh giá một cách chính xác về khả năng khái quát hóa.

Các thước đo đánh giá mô hình:

- Accuracy (độ chính xác): Được tính bằng tỉ lệ số lượng dự đoán đúng so với tổng số lượng dự đoán. Nó được coi là phương pháp cơ bản nhất và phù hợp khi các lớp có số lượng mẫu cân bằng.
- Precision: Được tính toán bằng tỉ lệ số lượng dự đoán đúng tích cực so với tổng số dự đoán tích cực.
- Recall: Tỉ lệ số lượng dự đoán đúng tích cực so với tổng số mẫu thực sự tích cực trong tập dự đoán.
- F1-Score: Trung bình điều hòa của Precision và Recall. Giá trị này cao khi cả hai đều cao. Đây là một độ đo hữu ích khi cần cân bằng giữa Precision và Recall.
- Confusion Matrix: Đây là bảng biểu diễn số lượng dự đoán đúng và sai cho mỗi lớp. Nó giúp hiểu rõ hơn về hiệu suất mô hình ở mỗi lớp.

Quá khớp (Overfitting): Là một vấn đề phổ biến trong học máy, có thể xảy ra khi mô hình học máy học quá kỹ lưỡng về chi tiết và nhiễu của tập huấn luyện. Khi một mô hình bị quá khớp, nó hoạt động rất tốt trên mô hình huấn luyện nhưng lại có kết quả thấp khi áp dụng với dữ liệu mới (dữ liệu không được đưa vào trong quá trình đào tạo). Quá khớp có thể có nhiều nguyên nhân gây ra như mô hình phức tạp, dữ liệu huấn luyện không đầy đủ hoặc có nhiều nhiễu.



Hình 2.5: Minh họa cho quá khớp trong học máy

2.1.4 Mô hình K – Nearest Neightbors

2.1.4.1 Thuật toán KNN

Thuật toán KNN là một trong những phương pháp học máy có giám sát (supervised learing) không tham số, nghĩa là nó không đưa ra giả định nào về dữ liệu. Phương pháp này đơn giản nhưng vô cùng hiệu quả, nó được ứng dụng rộng rãi trong nhiều lĩnh vực như phân loại, dự đoán và khai phá dữ liệu.

Ý tưởng cốt lõi của KNN dựa trên nguyên lý "những vật thể gần nhau thường có xu hướng tương đồng". Khi cần phân loại một điểm dữ liệu mới, KNN sẽ xem xét K điểm dữ liệu gần nhất trong tập huấn luyện và dựa vào đó để đưa ra quyết định. Nếu K điểm dữ liệu đó đa số thuộc về lớp A thì điểm dữ liệu mới cũng sẽ được gán vào lớp A.

Ưu điểm lớn nhất của KNN là đơn giản và dễ hiểu. Thuật toán không yêu cầu xây dựng mô hình phức tạp hay ước lượng tham số, mọi tính toán đều được thực hiện trực tiếp khi cần dự đoán. Điều này giúp KNN trở thành lựa chọn tốt cho những bài toán có dữ liệu nhỏ hoặc yêu cầu thời gian xử lý nhanh.

Ngoài ra, KNN có nhược điểm như khá nhạy cảm với nhiễu, nhất là khi giá trị K nhỏ. Hơn nữa, việc tính toán khoảng cách từ tất cả các điểm dữ liệu trong tập huấn luyện có thể tốn kém về mặt tính toán, nhất là khi số lượng điểm dữ liệu lớn hoặc số chiều của một điểm dữ liệu cao.

Mặc dù vậy, với những ưu nhược điểm vượt trội và khả năng ứng dụng linh hoat, KNN vẫn là một công cụ mạnh mẽ trong học máy. Từ việc phân loại thư rác, nhận dạng chữ viết tay, dự đoán giá nhà đất…

Hồi quy tuyến tính là một mô hình học máy đơn giản dựa vào thống kê để hồi quy dữ liệu với các biến phụ thuộc với các giá trị liên tiếp trong khi các biến độc lập có thể là một trong hai giá trị liên tục hoặc đơn lẻ.

2.1.4.2 Xây dựng công thức

Thuật toán KNN không có một công thức cụ thể như các thuật toán khác, mà nó dựa trên công thức tính khoảng cách và phân loại dựa trên số đông. Tuy nhiên các bước của thuật toán KNN có thể tóm tắt lại như sau

- Tính khoảng cách: Chọn một độ đo khoảng cách như Euclidean, Manhattan, Minkowski,…
- Chọn K láng giềng gần nhất: Sau khi sắp xếp các điểm trong tập dữ liệu theo thứ tự tăng dần của khoảng cách đã tính, thuật toán chọn ra K điểm dữ liệu gần nhất với điểm cần xét.
- Phân loại: Đếm số lượng các điểm dữ liệu thuộc mỗi lớp trong K láng giếng đó, gán nhãn của điểm dữ liệu cần dự đoán cho lớp có số lượng điểm dữ liệu là nhiều nhất.

Giả sử điểm x $(x_1,\ x_2,\ \ldots, x_n)$ và y $(y_1,\ y_2, \ldots, y_n)$, chúng ta có thể áp dụng các công thức tính khoảng cách thường sau tùy thuộc vào từng yêu cầu của bài toán:

- Euclidean:

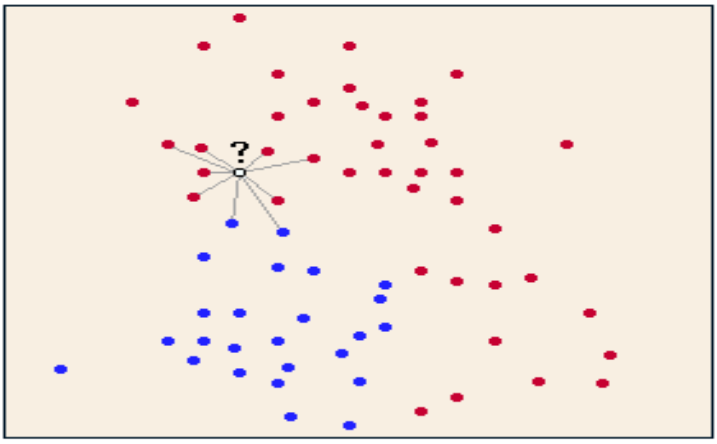
$$d(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \qquad 2\text{-}1$$

- Manhattan:

$$d(x,y) = \sum_{i=1}^{k}|x_i - y_i| \qquad 2\text{-}2$$

- Minkowski:

$$d(x,y) = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}} \qquad 2\text{-}3$$



Hình 2.6: Minh họa về thuật toán KNN.

Tại ví dụ trong ảnh, với K = 9, do số điểm màu đỏ là 7 và số điểm màu xanh là 2 nên điểm được dự đoán sẽ được gắn cho lớp màu đỏ.

2.1.4.3 Nhược điểm của thuật toán KNN

Một số nhược điểm của thuật toán KNN:

- Nhạy cảm với nhiễu
- Tốn kém về tính toán
- Hiệu suất giảm khi số chiều tăng
- Không xử lý tốt dữ liệu mất cân bằng

2.1.5 Mô hình Logistic Regression

2.1.5.1 Giới thiệu về thuật toán

Hồi quy logistic là một phương pháp thống kê phổ biến để phân loại nhị phân. Nó được sử dụng rộng rãi trong nhiều lĩnh vực như y học, kinh tế, sinh học và khoa học xã hội để dự đoán khả năng xảy ra của một sự kiện. Mô hình hồi quy logistic dựa trên việc sử dụng hàm logistic (hàm sigmoid) để biến đổi kết quả tuyến tính thành xác suất. giúp dự đoán các kết quả nhị phân.

Ý tưởng cơ bản của hồi quy logistic là thay vì dự đoán trực tiếp giá trị của biến phụ thuộc (thường là 0 hoặc 1), mô hình sẽ dự đoán xác suất của sự kiện đó xảy ra. Sau đó, xác suất này có thể được sử dụng để đưa ra quyết định phân loại.

Hàm sigmoid được định nghĩa như sau:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad 2\text{-}4$$

Hàm này khi được biểu diễn dưới dạng đồ thị có dạng chữ S, các giá trị của x đều đưa ra kết quả f(x) có giá trị từ 0 đến 1. Điều này rất phù hợp với việc mô hình hóa xác suất.

Với bài toán hồi quy tuyến tính, kết quả của biến phụ thuộc y được đưa ra bởi công thức:

$$\hat{y} = w_1 x_1 + w_2 x_2 + \dots w_k x_k + b \qquad 2\text{-}5$$

Trong đó:

- $\hat{y}$: biến phụ thuộc cần dự đoán.

- $x_1, x_1, \ldots, x_k$ là các biến độc lập (biến đặc trưng).
- $w_1, w_2, \ldots, w_k$ là các hệ số hồi quy.
- b là sai số.

Công thức trên có thể viết gọi lại thành:

$$\hat{y} = w^T.x + b \qquad 2\text{-}6$$

Trong hồi quy logistic, $\hat{y}$ sẽ được đưa qua hàm sigmoid để tính được xác suất xảy ra.

2.1.5.2 Xây dựng thuật toán Logistic Regresison

Logistic regression là một thuật toán học máy có giám sát với nhiệm vụ phân loại nhị phân bằng cách dự đoán xác suất của một kết quả, sự kiện hoặc quan sát. Mô hình đưa ra kết quả nhị phân hoặc phân đôi được giới hạn ở hai kết quả có thể xảy: có – không, 0 – 1 hoặc đúng – sai

Logistic regression sử dụng hàm logistic được gọi là hàm sigmoid để ánh xạ các dự đoán và xác suất của chúng. Hàm sigmoid đề cập đến một đường cong chữ S chuyển đổi bất kỳ giá trị thực nào thành phạm vi từ 0 đến 1.

Hàm sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad 2\text{-}7$$

Trong đó:

- f(x) là xác xuất đầu ra (có giá trị giữa 0 và 1)
- x = $w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$ là một tổ hợp tuyến tính của các biến đầu vào x_1, x_2, x_n với các trọng số tương ứng w_1, w_2, w_n và bias b.

Công thức cross-entropy của hồi quy logistic được cho bởi:

$$L(y, \hat{y}) = -[y * \log \hat{y} + (1 - y) * \log 1 - \hat{y}] \qquad 2\text{-}8$$

Hàm mất mát (loss function) trong logistic regression là hàm cross-entropy. Hàm này đo lường sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế. Công thức của hàm mất mát trong bài toán này là này là:

$$L(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log \hat{y} + (1 - y_i) log(1 - \hat{y}_i)] \quad 2\text{-}9$$

Trong đó:

- M là số lượng mẫu trong tập dữ liệu
- y_i là giá trị thực tế của nhãn (0 hoặc 1)
- $\hat{y}_i$ là giá trị dự đoán (xác xuất) của mô hình

Để thuật toán đạt độ chính xác cao nhất, việc của chúng ta là tối ưu hàm mất mát này sao cho giá trị của hàm là nhỏ nhất. Một cách phổ biến có thể nhắc tới là Gradient Descent.

Quá trình tối ưu hóa: để tìm ra các trọng số tối ưu **w** và bias **b**, ta sử dụng phương pháp tối ưu hóa như Gradient Decesnt. Quá trình này bao gồm các bước sau:

1. Khởi tạo các trọng số và bias ngẫu nhiên hoặc bằng 0
2. Tính toán giá trị dự đoán $\hat{y}$ cho mỗi mẫu dữ liệu sử dụng hàm sigmoid.
3. Tính toán hàm mất mát sử dụng công thức cross-entropy.
4. Cập nhật trọng số và bias bằng cách giảm thiểu hàm mất mát thông qua Gradient Descent, Quá trình này lặp lại cho đến khi hàm mất mát hội tụ đến giá trị tối thiểu hoặc đạt đến số lượng vòng lặp tối đa.

2.1.5.3 Nhược điểm của logistic regression

Một số nhược điểm của logistic regression:

- Dễ bị ảnh hưởng bởi các điểm dữ liệu ngoại lai
- Yêu cầu chuẩn bị dữ liệu cẩn thận
- Không xử lý tốt dữ liệu mất cân bằng

2.1.6 Thuật toán cây quyết định (Decision Tree)

2.1.6.1 Giới thiệu về thuật toán

Cây quyết định (Decision Tree) là một trong những kỹ thuật học máy phổ biến được sử dụng trong cả phân loại và hồi quy. Một cây quyết định mô phỏng quá trình đưa ra quyết định của con người bằng cách chia dữ liệu thành các nhóm con dựa trên các thuộc tính, từ đó xây dựng một cấu trúc dạng cây với các nút quyết định và lá.

Ý tưởng chính của cây quyết định là xây dựng một mô hình dạng cây để dự đoán giá trị của các biến mục tiêu dựa trên các thuộc tính của dữ liệu đầu vào. Mỗi nút trong cây đại diện cho một thuộc tính của dữ liệu, mỗi nhánh đại diện cho một giá trị của thuộc tính đó, và mỗi lá đại diện cho một giá trị dự đoán của biến mục tiêu. Cây quyết định phân chia dữ liệu dựa trên các thuộc tính sao cho sự không đồng nhất của dữ liệu trong nhóm con là thấp nhất có thể.

Cây quyết định là một phương pháp thông dụng trong khai phá dữ liệu. Nó quyết định mô tả cấu trúc của một cây, trong đó các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó. Một quyết định có thể được học bằng cách chia tập hợp nguồn thành các tập con dựa theo một kiểm tra giá trị thuộc tính. Quá trình kiểm tra này được lặp lại đệ quy cho mỗi tập con dẫn xuất.

Thuật toán cây quyết định là một trong những thuật toán học máy phổ biến và dễ hiểu nhất. Nó được sử dụng cho cả bài toán phân loại về hồi quy. Mô hình cây quyết định là một cấu trúc phân cấp gồm các nút và nhánh. Trong đó bao gồm:

- Nút gốc (root node): Đại diện hco toàn bộ tập dữ liệu
- Nút quyết định (decision node): Đại diện cho một câu hỏi hoặc điều kiện kiểm tra trên một thuộc tính
- Nhánh (branch): Đại diện cho kết quả của câu hỏi hoặc điều kiện kiểm tra
- Nút lá (leaf node): Đại diện cho lớp hoặc giá trị dự đoán.

Thuật toán bắt đầu bằng cách chọn thuộc tính có khả năng phân chia tập dữ liệu tốt nhất dựa trên một tiêu chí đo lường như entropy, information gain hoặc gini index. Sau đó chia tập dữ liệu thành các tập con dựa trên giá trị của thuộc tính đã chọn cho đến khi tất cả các nút lá đều thuần nhất (chỉ chứa một lớp) hoặc đạt đến một điều kiện dừng (ví dụ: độ sâu tối đa của cây).

2.1.6.2 Xây dựng thuật toán cây quyết định ID3

Cây được xây dựng đệ quy từ trên xuống và theo cách chia để trị. Ban đầu tất cả mẫu học đều nằm ở gốc, sau đó phân loại các thuộc tính và chọn ra một thuộc tính làm gốc, sau đó tiếp tục phân chia các thuộc tính còn lại.

Thông tin mong đợi để phân lớp một mẫu trong D theo nhãn lớp:

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \quad \textit{2-10}$$

Trong đó:

- p(i) là xác suất của lớp i trong tập S.
- Information Gain: Đo lường độ giảm entropy sau khi chia tập dữ luệu S dựa trên một thuộc tính nào đó.

Thông tin cần thiết để phân chia D theo thuộc tính A:

$$Entropy_D(\mathrm{A}) = \sum_{j=i}^{v} \frac{|D_j|}{|D|} Entropy(D_j) \quad \textit{2-11}$$

Độ lợi thông tin của sự phân chia dựa trên thuộc tính A:

$$Gain(A) = Entropy(D) - Entropy_A(D) \quad \textit{2-12}$$

Tiếp tục áp dụng cho mỗi nút con của nút gốc cho đến khi đạt đến nút lá hoặc nút có entropy = 0 ta được cây quyết định.

Các bước xây dựng cây quyết định:

1. Tính Entropy hoặc Gini index cho tập dữ liệu gốc: đầu tiên, tính toán entropy hoặc gini index cho tập dữ liệu ban đầu để đo lường độ hỗn loạn hoặc thuần khiết của dữ liệu.
2. Tính toán Information gain hoặc Gain ratio cho mỗi thuộc tính để xác định mức giảm entropy hoặc gini index khi chia nhỏ dữ liệu theo thuộc tính đó.
3. Chọn thuộc tính có Information gain hoặc Gain ratio cao nhất. Thuộc tính này sẽ trở thành nút quyết định trong cây.
4. Chia nhỏ dữ liệu dựa trên các giá trị của thuộc tính được chọn. Mỗi giá trị của thuộc tính sẽ tạo thành một nhánh mới trong cây.
5. Áp dụng lại các bước từ 1 đến 4 cho từng nhánh con cho đến khi một trong các điều kiện như tất cả các mẫu trong nhánh con thuộc cùng một lớp hoặc không còn thuộc tính nào để chia nhỏ dữ liệu hoặc độ sâu của cây đạt đến giới hạn.

6. Khi quá trình chia nhỏ kết thúc, mỗi lá trong cây đại diện cho một giá trị dự đoán của biến mục tiêu. Đối với bài toán phân loại, giá trị dự đoán là lớp chiếm đa số trong lá đó. Đối với bài toán hồi quy, giá trị dự đoán là trung bình hoặc trung bị của các giá trị trong lá đó.

2.1.6.3 Nhược điểm của thuật toán

- Dễ bị overfitting: Cây quyết định có xu hướng học quá chi tiết từ dữ liệu huấn luyện, dẫn đến có thể quá khớp với dữ liệu được đưa vào và kém hiệu quả khi áp dụng cho dữ liệu mới. Điều này đặc biệt xảy ra khi cây quá sâu hoặc có quá nhiều nhánh.
- Nhạy cảm với dữ liệu nhiễu: Các điểm dữ liệu nhiễu hoặc ngoại lai có thể ảnh hưởng đáng kể đến cấu trúc của cây quyết định, dẫn đến việc tạo ra các nhánh không cần thiết hoặc sai lệch.
- Thiên vị với các thuộc tính có nhiều giá trị: Các thuộc tính có quá nhiều giá trị thường được ưu tiên hơn trong quá trình chia tách, mặc dù có thể chúng không có ý nghĩa trong việc dự đoán kết quả.
- Chi phí tính toán cao: Việc xây dựng cây quyết định tối ưu có thể gây tốn kém về mặt tính toán, đặc biệt với dữ liệu có nhiều chiều hoặc có nhiều lớp.

Giải pháp khắc phục:

- Cắt tỉa cây: Áp dụng các kỹ thuật cắt tỉa để giảm độ phức tạp của cây và tránh overfitting.
- Xử lý dữ liệu nhiễu: Loại bỏ hoặc giảm thiểu ảnh hưởng của nhiễu bằng các kỹ thuật làm sạch dữ liệu.

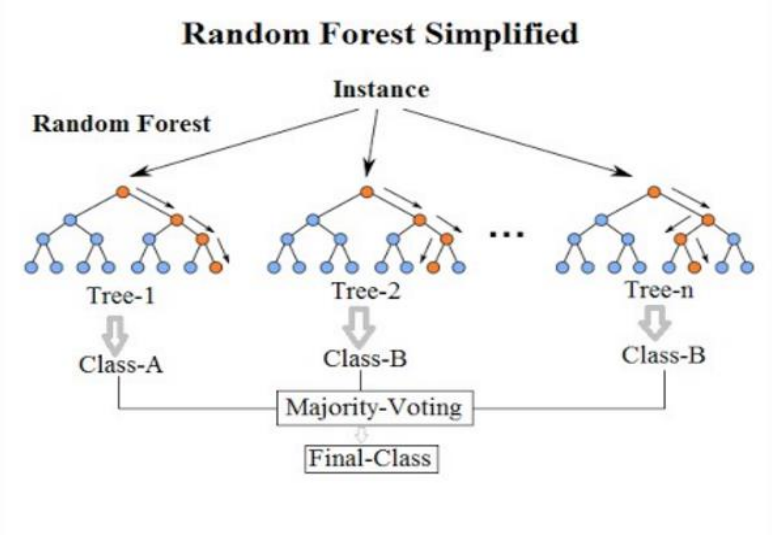
2.1.7 Thuật toán rừng ngẫu nhiên (Random Forest)

2.1.7.1 Giới thiệu về thuật toán

Rừng ngẫu nhiên (Random Forest) là một trong những thuật toán học máy có giám sát được sử dụng rộng rãi. Nó là một kỹ thuật Ensemble, tức là nó kết hợp nhiều mô hình nhỏ để tạo ra một mô hình dự đoán mạnh mẽ hơn. Thuật toán này chủ yếu dựa trên cây quyết định (Decision Tree) và bổ sung thêm các yếu tố ngẫu nhiên vào quá trình huấn luyện, giúp giảm thiểu hiện tượng quá khớp (overfitting) mà thường thấy ở một cây quyết định duy nhất. Sau đó ở bước dự đoán với một dữ liệu mới, mỗi cây quyết định sẽ

đưa ra kết quả của mình, kết quả của thuật toán này sẽ được tổng kết dựa trên kết quả của từng cây quyết định.

Rừng ngẫu nhiên có thể giải quyết cả các bài toán hồi quy và phân loại. Nó coi mỗi cây quyết định là một cử tri riêng, ở cuối cuộc bầu cử, câu trả lời nhận được nhiều bầu chọn nhiều nhất sẽ là kết quả của thuật toán.



Hình 2.7 Mô hình Random Forest

Random Forest được coi là một phương pháp mạnh mẽ và chính xác vì có nhiều cây quyết định tham gia vào quá trình này. Thuật toán không bị vấn đề quá khớp do lý do chính nó dựa vào kết quả của số đông. Nhưng thuật toán này đưa ra dự đoán chậm vì nó có nhiều cây quyết định. Trước khi nó đưa ra quyết định cho bài toán thì phải chờ tất các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và thực hiện bỏ phiếu trên đó. Toàn bộ quá trình nói trên có thể tốn thời gian. Mô hình này khó hiểu hơn so với cây quyết định.

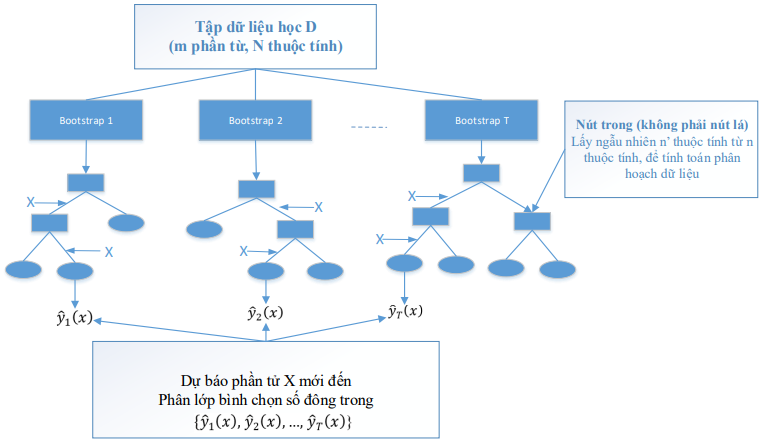
2.1.7.2 Xây dựng thuật toán Random Forest

Ý tưởng chính của thuật toán là kết hợp nhiều cây quyết định để giảm thiểu sai lệch, tăng cường độ chính xác của mô hình. Mỗi cây trong rừng được xây dựng từ một vài mẫu ngẫu nhiên của dữ liệu và sử dụng tập con ngẫu nhiêu của các đặc trưng để chia các nút. Từ đó đảm bảo rằng các cây quyết định bên trong là khác nhau và giảm bớt được nguy cơ quá khớp.

Giả sử bộ dữ liệu của mình có n dữ liệu (sample) và mỗi dữ liệu có d thuộc tính.

Để xây dựng mỗi cây quyết định ta làm những bước sau:

1. Chọn mẫu (Bootstrap): với một tập dữ liệu có N mẫu, chúng ta chọn n mẫu một cách ngẫu nhiên để tạo thành một tập dữ liệu mới cho mỗi cây.
2. Xây dựng cây: dựa trên tập dữ liệu mới này, cây quyết định được đưa ra. Trong quá trình phân chia nút, ở bước này, không phải tất cả các đặc trưng đều được xem xét, thay vào đó, một tập con ngẫu nhiên các đặc trưng được chọn.
3. Lặp lại quá trình: quá trình lặp lại là để xây dựng nhiều cây, thường là vài tram hoặc vài nghìn cây, tùy thuộc vào yêu cầu của bài toán.
4. Dự đoán: khi thực hiện dự đoán, mỗi cây trong rừng sẽ đưa ra một dự đoán riêng. Dự đoán cuối cùng của rừng là dự đoán phổ biến nhất cho bài toán phân loại hoặc trung bình của các dự đoán đối với bài toán hồi quy.



Hình 2.8: Giải thích chi tiết về mô hình RF

2.1.7.3 Nhược điểm của Random Forest

Một số nhược điểm của thuật toán Random Forest:

- Tính phức tạp và thời gian tính toán:
- Yêu cầu bộ nhớ lớn
- Thiên vị với các đặc trưng có nhiều giá trị:
- Không hiệu quả với dữ liệu thưa

2.2 Các phương pháp đánh giá mô hình dự báo

Độ chính xác là một trong những chỉ số đơn giản và phổ biến nhất được sử dụng để đánh giá các mô hình hồi quy và phân loại. Mặc dù là một chỉ số hữu ích, nhưng nó có một số hạn chế đáng kể trong các trường hợp dữ liệu không cân bằng.

Chúng ta cùng đi vào tìm hiểu các phương pháp đánh giá độ chính xác bằng một ví dụ: Bảng dữ liệu mô phỏng kết quả dự đoán và thực tế của bài toán đự đoán email có phải là spam hay không.

Bảng 2.1 Ví dụ về bảng dữ liệu email

Email ID	Thực tế	Dự đoán
1	Không spam	Không spam
2	Không spam	Spam
3	Spam	Spam
4	Spam	Spam
5	Không spam	Spam
6	Spam	Không spam
7	Spam	Spam

2.2.1 Accuracy

Độ chính xác (Accuracy) được tính toán bằng cách dựa trên số lượng dự đoán đúng trên tổng số lượng dự đoán của mô hình. Nói cách khác nó cho biết tỉ lệ mẫu trong tập dữ liệu được mô hình dự đoán chính xác.

Công thức:

$$Accuracy = \frac{Total\ of\ correct\ Predictions}{Total\ number\ of\ predictions} \qquad 2\text{-}13$$

Trong đó:

- Total of correct predictions: tổng số mẫu dự đoán đúng của mô hình.
- Total number of predictions: tổng số mẫu được đưa vào mô hình để dự đoán.

Đối với bảng dữ liệu về email spam, ta có thể đưa ra kết quả:

$$Accuracy = \frac{4}{7} = 57\%. \quad \textit{2-14}$$

Độ chính xác là một chỉ số quan trọng vì nó cung cấp một cái nhìn tổng quan và nhanh chóng về hiệu suất mô hình. Tuy nhiên nó có một số hạn chế đáng kể như không dự đoán chính xác hiệu suất mô hình khi số lượng mẫu của các lớp không cân bằng.

2.2.2 Precision

Độ chính xác Precision là một chỉ số đánh giá quan trọng trong các bài toán phân loại của học máy, đặc biệt là đánh giá mức độ chính xác của các dự đoán tích cực mà mô hình tạo ra. Precision đo lường tỉ lệ số lượng dự đoán đúng là tích cực. Nói cách khác precision cho biết một mô hình dự đoán một mẫu là tích cực thì mẫu nó có thực sự là tích cực hay không.

Công thức tính Precision là:

$$Precision = \frac{TP}{TP + FP} \quad \textit{2-15}$$

Precision cao cho thấy khi mô hình phân loại một mẫu là tích cực, khả năng cao mẫu đó thực sự là tích cực. Điều này rất quan trọng trong các tình huống mà hậu quả của một dự đoán sai tích cực là nghiêm trọng.

2.2.3 Recall

Recall còn được gọi là độ nhạy, là một chỉ số đánh giá hiệu suất mô hình phân loại trong học máy. Nó đo lường khả năng của mô hình phát hiện ra tất cả các trường hợp tích cực thực sự trong dữ liệu. Nói cách khác, recall cho chúng ta biết mô hình tìm ra bao nhiêu phần trăm các trường hợp tích cực thực tế trong dữ liệu.

Công thức tính Recall là:

$$Recall = \frac{TP}{TP + FN} \quad 2\text{-}16$$

Recall cao cho thấy ý nghĩa quan trọng trong nhiều tính huống, đặc biệt là khi hậu quả của việc bỏ xót trong trường hợp dương tính là nghiêm trọng. Ngoài ra, recall cao có thể dẫn đến một số điểm yếu nhất định như nhận diện sai các giá trị âm tính. Do đó cần cân nhắc kỹ giữa việc sử dụng Recall.

2.2.4 F1 Score

F1 Score là chỉ số đánh giá hiệu xuất của các mô hình phân loại, đặc biệt trong các bài toán phân loại nhị phân hoặc đa lớp với các dữ liệu không cân bằng. F1 Score cung cấp một độ đo cân bằng giữa recall và precision, hai chỉ số quan trọng trong việc đánh giá các mô hình phân loại.

F1 score là trung bình điều hòa của recall và precision. Điều này có nghĩa là F1 score lấy hai tỷ lệ này một cách đồng đều và cung cấp một độ đo tổng hợp.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad 2\text{-}17$$

F1 score hữu ích khi cần một độ đo duy nhất để đánh giá hiệu suất của mô hình, đặc biệt khi các lớp dữ liệu không cân bằng hoặc khi chúng ta muốn cân bằng giữa precision và recall.

CHƯƠNG 3 XÂY DỰNG MÔ HÌNH HỌC MÁY DỰ BÁO LƯU LƯỢNG

3.1 Giới thiệu bài toán

3.1.1 Bài toán xây dựng mô hình học máy dự đoán tỉ lệ tử vong của bệnh nhân suy tim

a. Mục đích của bài toán:

Xây dựng mô hình học máy dự đoán tỉ lệ tử vong của bệnh nhân suy tim mang lại nhiều lợi ích quan trọng cho cả bệnh nhân và hệ thống chăm sóc sức khỏe. Mục đích chính là xác định những bệnh nhân có nguy cơ tử vong cao, từ đó cho phép can thiệp sớm và điều chỉnh kế hoạch điều trị một cách cá nhân hóa. Mô hình này sử dụng các yếu tố như chức năng thận, chức năng tim,… để đưa ra dự đoán.

Thông tin này không chỉ giúp bác sĩ đưa ra quyết định điều trị tốt hơn mà còn giúp phân bổ nguồn lực y tế một cách hiệu quả, ưu tiên cho những bệnh nhân có nguy cơ cao hơn. Hơn nữa, mô hình này có thể giúp chúng ta hiểu rõ hơn về các yếu tố nguy cơ và phát triển các phương pháp điều trị mới. Tuy nhiên đây chỉ là một công cụ hỗ trợ và không thể thay thế hoàn toàn đánh giá lâm sàng của bác sĩ.

b. Ý nghĩa của bài toán

Đối với bệnh nhân, mô hình này giúp đưa ra dự đoán để các bác sĩ có thể căn cứ vào đó để đưa ra được liệu trình phù hợp, cá nhân hóa dựa trên tình trạng của từng bệnh nhân.

3.1.2 Bộ dữ tình trạng sức khỏe

Tập dữ liệu

Bảng 3.1 Thông tin bộ dữ liệu

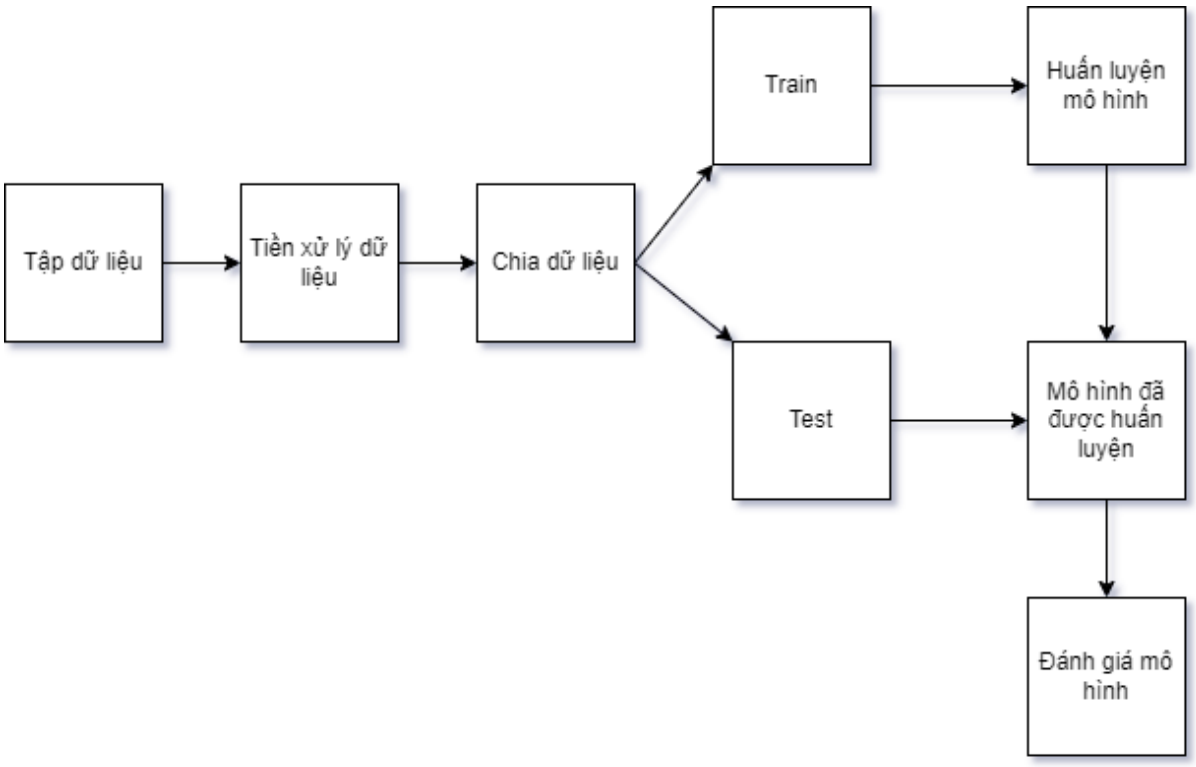
Tên trường	Nội dung
age	Tuổi bệnh nhân
anaemia	Thiếu máu
creatinine_phosphokinase	Mức độ enzyme CPK trong máu
diabetes	Tiểu đường
ejection_fraction	Phần trăm máu ra khỏi tim mỗi nhịp đập

high_blood_pressure	Huyết áp cao
platelets	Số lượng tiểu cầu trong máu
serum_creatinine	Mức độ creatinine trong huyết thanh
serum_sodium	Mức độ natri trong máu
sex	Giới tính
Smoking	Hút thuốc
Time	Thời gian theo dõi
DEATH_EVENT	Có tử vong trong thời gian nghiên cứu

3.2 Mô hình giải quyết bài toán

Bài toán đặt ra trong đồ án là đưa ra mô hình dự đoán tỉ lệ tử vong của bệnh nhân dựa trên các thông tin cá nhân như tuổi, giới tính và các thông tin sức khỏe như… để đưa ra dự đoán về khả năng tử vong của bệnh nhân.

Các thuật toán được sử dụng như Logistic Regression, KNN, Decision Tree và Random Forest để huấn luyện mô hình và chọn ra một mô hình tốt nhất để giải quyết bài toán.



Hình 3.1 Các bước giải quyết bài toán

Quá trình xây dựng mô hình gồm các bước sau:

1. Tiền xử lý dữ liệu:

- Làm sạch dữ liệu: Sử dụng các phương pháp loại bỏ các bản ghi chứa giá trị thiếu hoặc điền các giá trị thiều bằng giá trị trung bình, trung vị hoặc giá trị gần nhất.
- Xử lý các giá trị ngoại lai để giảm ảnh hưởng tiêu cực của chúng.
- Xử lý dữ liệu không nhất quán.
- Chuyển đổi dữ liệu sao cho các giá trị nằm trên cùng một phạm vi cụ thể.
- Chuẩn hóa dữ liệu.
- Phân chia dữ liệu thành tập train và tập test.

2. Xây dựng mô hình: Xây dựng các mô hình giải quyết bài toán, các mô hình được đề xuất là LR, KNN, DT và RF.
3. Huấn luyện mô hình: mô hình được huấn luyện bằng cách đưa dữ liệu huấn luyện vào và điều chỉnh các thông số của mô hình để đạt được dự đoán chính xác nhất.
4. Đánh giá và tinh chỉnh: mô hình được đánh giá bằng cách sử dụng dữ liệu kiểm tra độc lập (chưa từng được sử dụng trong quá trình training) để đo lường hiệu suất và độ chính xác của dự đoán. Nếu cần thiết, mô hình có thể được điều chỉnh và tinh chỉnh để cải thiện kết quả dự đoán.
5. Dự đoán tử vong của người bệnh: sau khi mô hình đã được huấn luyện và đánh giá, nó có thể được sử dụng để dự đoán tỉ lệ tử vong của người bệnh trong tương lai dựa trên các tham số đầu vào.

3.3 Công cụ và thư viện

3.3.1 Ngôn ngữ Python

3.3.1.1 Khái niệm Python

Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở và đa nền tảng. Python được Guido van Rossum giới thiệu vào năm 1991 và đã trải qua 3 giai đoạn phát triển khác nhau tương ứng với các version, mới nhất hiện nay là Python version 3x.

3.3.1.2 Đặc điểm của Python

Python được thiết kế với tư tưởng giúp người học dễ đọc, dễ hiểu và dễ nhớ, vì thế ngôn ngữ Python có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học.

Cấu trúc của Python cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, nói cách khác thì so với các ngôn ngữ lập trình khác, chúng ta có thể sử dụng ít dòng code hơn để viết ra một chương trình trong Python.

Ban đầu, Python được phát triển để chạy trên nền Unix, vì thế nó là mã nguồn mở. Sau này qua thời gian phát triển, Python mở rộng và hiện nay đã hỗ trợ hầu hết các nền tảng khác như Window hay MacOS.

Python là một ngôn ngữ lập trình đa mẫu hình, do nó hỗ trợ hoàn toán lập trình hướng đối tượng và lập trình cấu trúc. Nhờ vậy, Python có thể làm được rất nhiều việc và được sử dụng trong nhiều lĩnh vực khác nhau.

3.3.1.3 Ứng dụng của Python

Python là ngôn ngữ được ứng dụng đa dạng trong các lĩnh vực:

Phát triển ứng dụng Web với các Framework của Python: Django và Flask là 2 framework phổ biến hiện nay dành cho các lập trình viên Python để tạo ra các website.

Tool tự động hóa: các ứng dụng như từ điển, crawl dữ liệu từ website, tool giúp tự động hóa công việc được các lập trình viên ưu tiên lựa chọn Python để viết nhờ tốc độ code nhanh của nó.

Khoa học máy tính: Trong Python có rất nhiều thư viện quan trọng phục vụ cho ngành khoa học máy tính như: OpenCV cho xử lý ảnh và machine learning, Scipy và Numpys cho lĩnh vực toán học, đại số tuyến tính, Pandas cho việc phân tích dữ liệu,…

Lĩnh vực Internet: Python có thể viết được các ứng dụng cho nền tảng nhúng, đồng thời cũng được lựa chọn cho việc xử lý dữ liệu lớn. Vì thế Python là một ngôn ngữ quen thuộc trong lĩnh vực Internet kết nối vạn vật.

Làm game: Pygame là một bộ module Python cross-platform được thiết kế để viết game cho cả máy tính và các thiết bị di động.

3.3.2 Các thư viện hỗ trợ

3.3.2.1 Thư viện numpy

Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng "core Python" đơn thuần.

Dưới đây là một số chức năng quan trọng của thư viện NumPy:

- Tạo mảng: Sử dụng hàm ' np.array() ' hoặc các hàm tạo mảng khác như 'np.zeros()', ' np.ones() ', ' np.arange() ', ' np.linspace() '.
- Thao tác trên mảng: NumPy cung cấp các phép toán số học và logic trên mảng, như cộng, trừ, nhân, chia, lũy thừa, căn bậc hai, tích vô hướng, tích vô hướng, phép so sánh, v.v.
- Truy cập và cắt mảng: Sử dụng chỉ số và cắt (slicing) để truy cập và thay đổi giá trị của mảng.
- Ma trận và phép toán ma trận: NumPy hỗ trợ các phép toán ma trận như nhân ma trận, chuyển vị ma trận, tính định thức, nghịch đảo, giải hệ phương trình tuyến tính, v.v.
- Xử lý dữ liệu số: NumPy cung cấp các chức năng để thực hiện các phép toán thống kê, tính toán tổng, trung bình, phương sai, tích chập, biến đổi Fourier, v.v.
- Tích hợp với thư viện khác: NumPy có khả năng tích hợp tốt với các thư viện khác như SciPy, Matplotlib và pandas để thực hiện các tác vụ phức tạp như tối ưu hóa, xử lý dữ liệu, vẽ đồ thị, v.v.
- Thư viện NumPy cần được cài đặt nó trước bằng cách chạy lệnh 'pip install numpy'. Sau đó, ta có thể import thư viện và sử dụng các chức năng của nó trong mã Python của mình bằng câu lệnh 'import numpy as np'.

3.3.2.2 Thư viện Pandas

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu.

Dưới đây là một số tính năng quan trọng của thư viện Pandas:

- Đọc và ghi dữ liệu: Pandas cho phép đọc dữ liệu từ nhiều nguồn khác nhau như tệp CSV, Excel, SQL, v.v. Ta cũng có thể ghi dữ liệu thành các định dạng tương tự.
- Tạo DataFrame: DataFrame là một cấu trúc dữ liệu hai chiều giống bảng, mà mỗi cột có thể chứa dữ liệu khác nhau. Pandas cho phép tạo DataFrame từ các cấu trúc dữ liệu khác nhau như danh sách, mảng NumPy, và cả từ các tệp dữ liệu.
- Xử lý và truy xuất dữ liệu: Pandas cung cấp nhiều công cụ để xử lý và truy xuất dữ liệu trong DataFrame. Người lập trình có thể thực hiện các hoạt động như lọc, chọn, sắp xếp, và thay đổi giá trị của dữ liệu.
- Xử lý dữ liệu thiếu: Pandas cung cấp các phương thức để xử lý dữ liệu thiếu trong DataFrame, như xóa các hàng hoặc cột có dữ liệu thiếu, điền giá trị thiếu bằng giá trị khác, hoặc thực hiện các phương pháp khác phù hợp với từng tình huống.
- Tính toán và thống kê: Pandas cung cấp các công cụ tính toán và thống kê trên dữ liệu, bao gồm tính tổng, trung bình, phương sai, độ tương quan, và nhiều phương pháp thống kê khác.
- Trực quan hóa dữ liệu: Pandas tích hợp với thư viện Matplotlib để trực quan hóa dữ liệu trong DataFrame. Thư viện này cho phép vẽ biểu đồ đường, biểu đồ cột, biểu đồ scatter, và nhiều loại biểu đồ khác.

3.3.2.3 Thư viện Seaborn

Seaborn là một trong những thư viện Python được đánh giá cao nhất thế giới được xây dựng nhằm mục đích tạo ra các hình ảnh trực quan đẹp mắt. Nó có thể được gọi là một phần mở rộng của một thư viện khác có tên là Matplotlib vì nó được xây dựng trên đó.

Dưới đây là một số tính năng quan trọng của thư viện Seaborn:

- Tạo biểu đồ: Seaborn cung cấp các hàm để tạo các biểu đồ phổ biến như biểu đồ đường, biểu đồ cột, biểu đồ phân phối, biểu đồ scatter, biểu đồ hộp, biểu đồ violin, v.v. Các biểu đồ này có thể được tùy chỉnh màu sắc, phong cách và thiết kế để tạo ra các biểu đồ trực quan và hấp dẫn.
- Xử lý dữ liệu: Seaborn cung cấp các công cụ để xử lý và trực quan hóa dữ liệu có cấu trúc, bao gồm việc xử lý dữ liệu thiếu, biến đổi dữ liệu, và làm sạch dữ liệu để chuẩn bị cho trực quan hóa.

- Tương quan và phân tích: Seaborn cung cấp các công cụ để thực hiện phân tích tương quan, tìm hiểu mối quan hệ giữa các biến, và trực quan hóa mô hình tương quan như biểu đồ heatmap và pairplot.
- Tương tác: Seaborn hỗ trợ tích hợp với thư viện Ipython và Jupyter Notebook để tạo ra các biểu đồ tương tác, cho phép bạn tương tác với dữ liệu và thay đổi biểu đồ trong thời gian thực.
- Tích hợp với Pandas: Seaborn tích hợp tốt với thư viện Pandas, cho phép bạn trực tiếp trực quan hóa dữ liệu từ DataFrame của Pandas.

3.3.2.4 Thư viện Matplotlib

Thư viện Matplotlib là một trong những thư viện trực quan hóa dữ liệu phổ biến nhất trong Python. Nó cung cấp các công cụ mạnh mẽ để tạo ra các biểu đồ, đồ thị và hình ảnh đẹp mắt để hiển thị dữ liệu.

Dưới đây là một số tính năng quan trọng của thư viện Matplotlib:

- Tạo biểu đồ cơ bản: Matplotlib cho phép bạn tạo các biểu đồ cơ bản như đồ thị đường, đồ thị cột, đồ thị scatter, đồ thị hình tròn và nhiều loại biểu đồ khác. Bạn có thể tùy chỉnh các yếu tố như màu sắc, kích thước, chú thích và tiêu đề để tạo ra các biểu đồ tùy chỉnh.
- Trực quan hóa dữ liệu 2D và 3D: Matplotlib hỗ trợ trực quan hóa dữ liệu 2D và 3D. Bạn có thể tạo các biểu đồ dạng đường, đám mây điểm, bề mặt 3D và nhiều loại biểu đồ khác để khám phá và hiển thị dữ liệu phức tạp.
- Phong cách và thiết kế: Matplotlib cung cấp các phong cách và thiết kế tùy chỉnh để tạo ra các biểu đồ chuyên nghiệp và thẩm mỹ. Bạn có thể tạo các bản đồ màu, đặt các đường viền, điều chỉnh các font chữ và tạo ra các hình ảnh chất lượng cao.
- Tích hợp với NumPy và Pandas: Matplotlib tích hợp tốt với thư viện NumPy và Pandas, cho phép bạn trực tiếp trực quan hóa dữ liệu từ các mảng NumPy và DataFrame của Pandas.
- Tùy chỉnh đồ họa: Matplotlib cho phép bạn tùy chỉnh mọi khía cạnh của đồ họa, từ trục x và y, nhãn trục, chú thích, lưới đồ thị, vùng màu, đường viền, hình dạng và nhiều hơn nữa.

3.4 Thực nghiệm

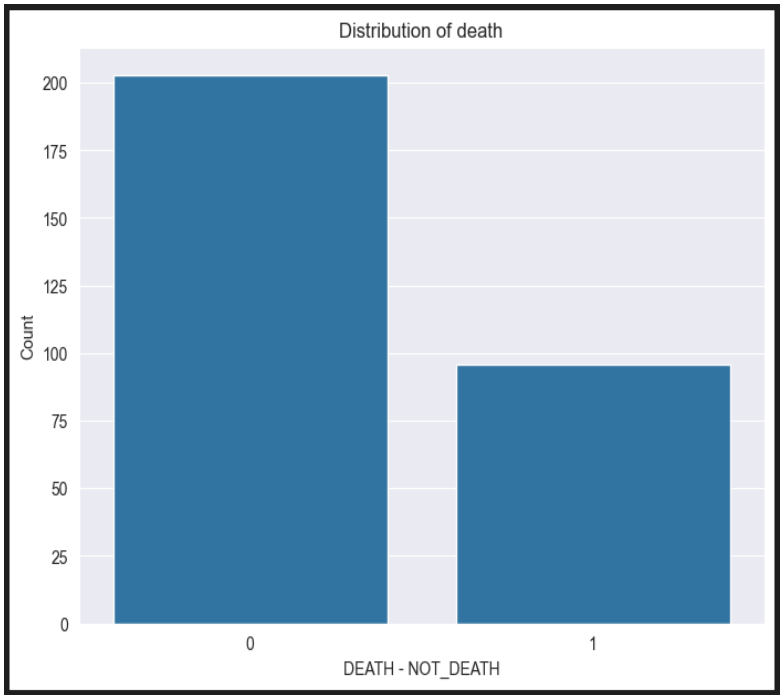
3.4.1 Trực quan hóa và tiền xử lý dữ liệu

3.4.1.1 Tiền xử lý dữ liệu

- Bộ dữ liệu không có giá trị thiếu.
- Do dữ liệu được lấy từ một nguồn nên dữ liệu này không cần phải chuẩn hóa.

3.4.1.2 Trực quan hóa dữ liệu

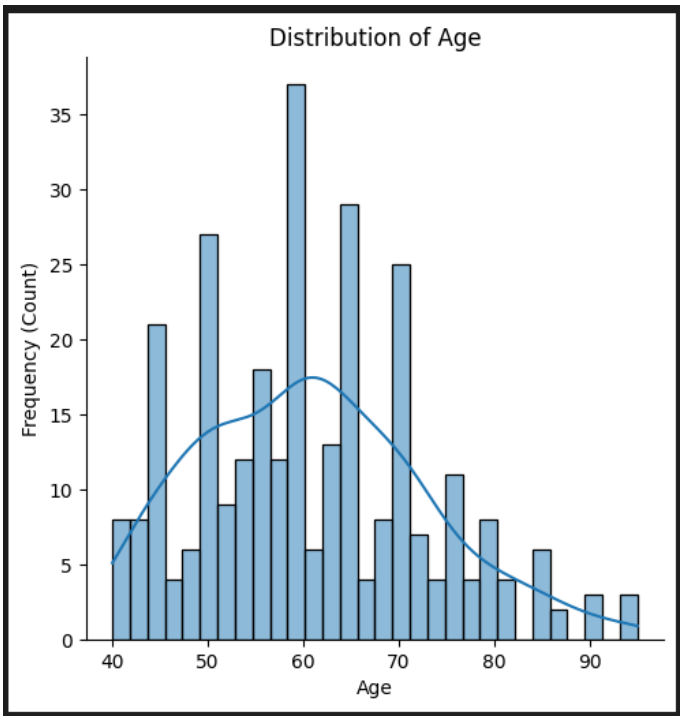
a. Biểu đồ về số lượng bệnh nhân sống sót và tử vong



Hình 3.2: Biểu đồ về số lượng bệnh nhân sống sót và tử vong

Từ biểu đồ, ta có thể thấy sự mất cân bằng về số lượng bệnh nhân sống sót và tử vong. Điều này có thể ảnh hưởng đến kết quả dự đoán của mô hình.

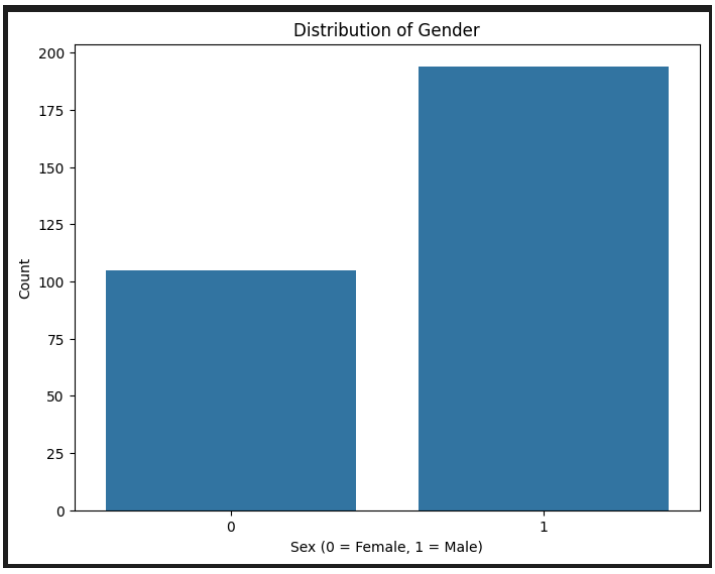
b. Biểu đồ phân phối tuổi của bệnh nhân



Hình 3.3: Biểu đồ phân bố tuổi của bệnh nhân

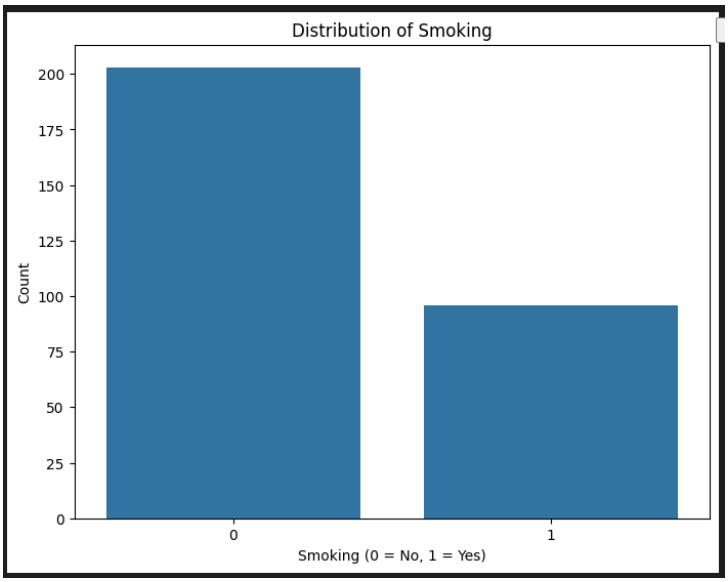
Từ biểu đồ trên, ta thấy được tuổi của bệnh nhân trải dài từ 40 đến 90, có một số trường hợp lớn hơn 90 tuổi.

c. Biểu đồ phân phối giới tính



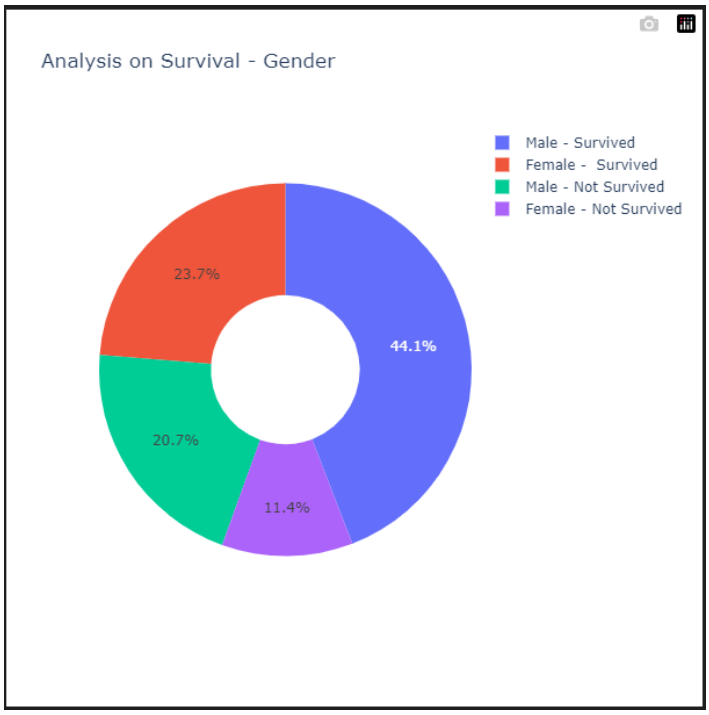
Hình 3.4: Biểu đồ phân bố giới tính của bệnh nhân

d. Biểu đồ phân phối bệnh nhân hút thuốc



Hình 3.5: Biểu đồ phân phối bệnh nhân hút thuốc.

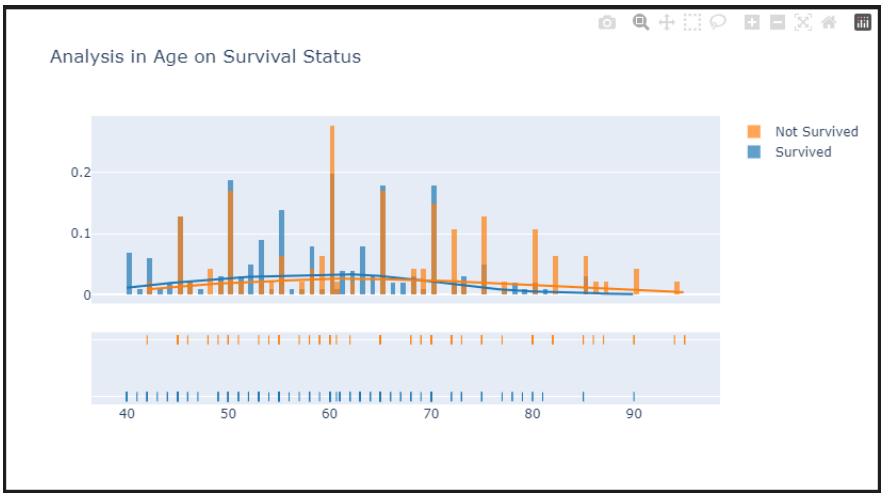
e. Biểu đồ phân tích tỉ lệ sống sót theo giới tính



Hình 3.6: Biểu đồ phân bố tỉ lệ sống sót và tử vong theo giới tính

Từ biểu đồ, ta thấy được tỉ lệ sống sót của nam giới (44.1%) cao hơn tỉ lệ sống sót của nữ giới (23.7%) và tỉ lệ tử vong của nam giới (20.7%) cao hơn tỉ lệ tử vong của nữ giới (11.4%). Tổng tỉ lệ sống sót (67.8%) cao hơn tổng tỉ lệ tử vong (20.7%). Điều này cho thấy trong tập dữ liệu, có nhiều bệnh nhân sống sót hơn là tử vong. Và tỉ lệ sống sót có thể coi như gấp đôi tỉ lệ tử vong.

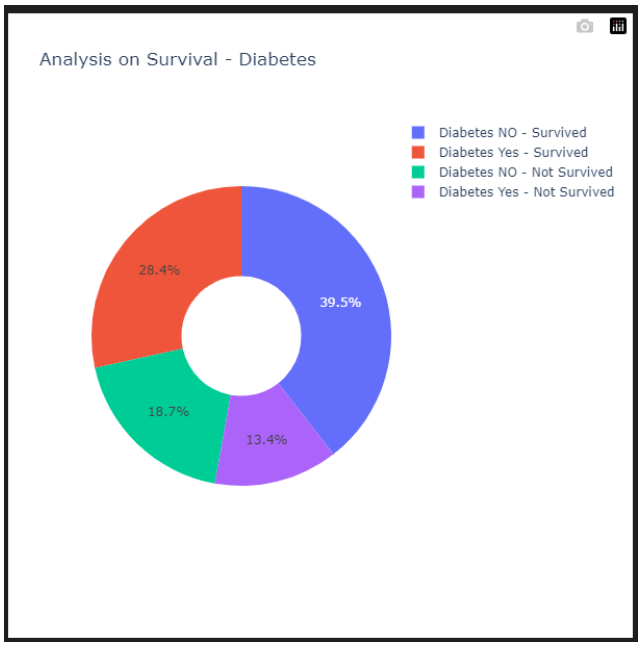
f. Biểu đồ quan hệ giữa tuổi và trạng thái sống sót của bệnh nhân



Hình 3.7: Biểu đồ quan hệ giữa tuổi và trạng thái sống sót của bệnh nhân

Biểu đồ này cho thấy mối quan hệ giữa độ tuổi và trạng thái sống sót của bệnh nhân. Bệnh nhân ở độ tuổi cao hơn có xu hướng tử vong nhiều hơn trong khi bệnh nhân trẻ có tỉ lệ sống sót cao hơn. Điều này cho thấy tuổi tác là một yếu tố quan trọng cần được xem xét khi đánh giá nguy cơ tử vong của bệnh nhân suy tim.

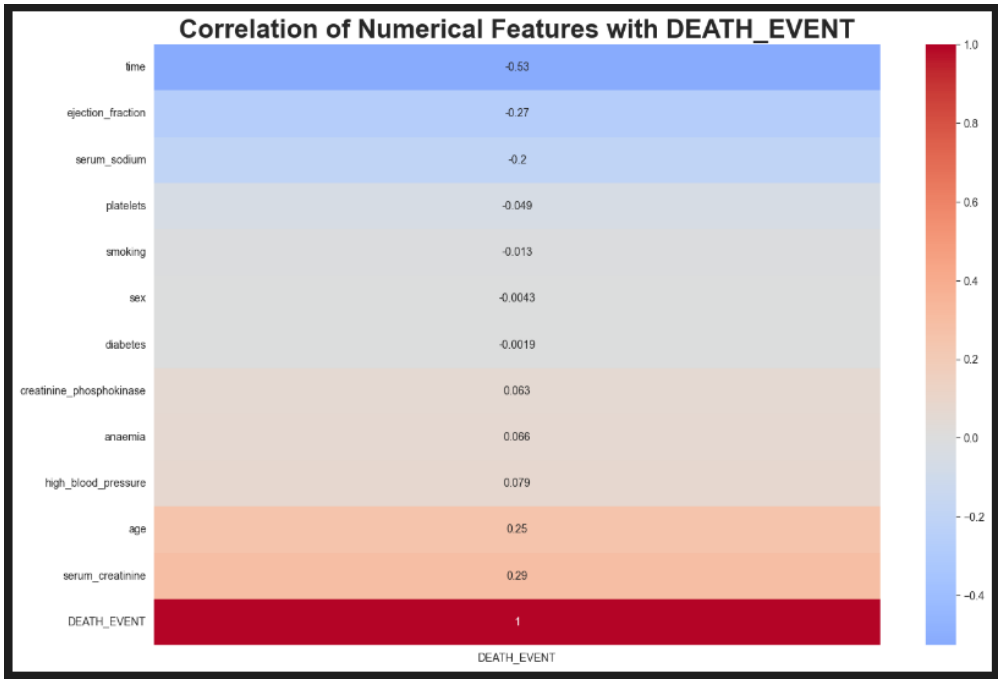
g. Biểu đồ tỉ lệ sống sót và tử vong giữa bệnh nhân tiểu đường và không bị tiểu đường



Hình 3.8: Biểu đồ phân bố tỉ lệ tử vong và bệnh tiểu đường của bệnh nhân

Trong biểu đồ trên, tỉ lệ bệnh nhân không bị tiểu đường và sống sót là 39.5%, tỉ lệ bệnh nhân tiểu đường và không sống sót là 28.4%, không tiểu đường nhưng tử vong là 18.7%, cuối cùng là tiểu đường, tử vong là 13.4%.

h. Biểu đồ tương quan giữa các đặc trưng với biến phụ thuộc DEATH_EVENT



Hình 3.9: Biểu đồ tương quan giữa các thuộc tính với DEATH_EVENT

Từ biểu đồ trên, ta có thể đưa ra nhận xét:

- Thời gian (time) có hệ số tương quan âm mạnh nhất với DEATH_EVENT (-0.53). Điều này có nghĩa thời gian theo dõi dài hơn có thể ảnh hưởng đến DEATH_EVENT giảm.
- Tỉ lệ phần trăm máu được bơm ra khỏi tim trong mỗi lần đập (ejection_fraction): có hệ số tương quan âm là -0.27. Điều này cho thấy giá trị ejection_ fraction cao hơn có liên quan đến tỉ lệ tử vong thấp hơn.
- Tỉ lệ natri trong máu (serum_sodium) có tương quan -0.2 cho thấy giá trị này cao hơn thì tỉ lệ tử vong thấp hơn.
- Tuổi (age) có tương quan dương 0.25, cho thấy tuổi tác của bệnh nhân cao hơn thì tỉ lệ tử vong của bệnh nhân cũng cao hơn.
- serum_creatinine có tương quan dương cao nhất là 0.29. Điều này cho thấy giá trị của nó tăng thì tỉ lệ tử vong của bệnh nhân tăng lên.

Tóm lại qua các biểu đồ trong bài toán, đặc biệt là biểu đồ tương quan. Ta có thể chọn các thuộc tính dựa có hệ số tương quan mạnh (dương hoặc âm) với biến mục tiêu là DEATH_EVENT để cung cấp các thông tin quan trọng cho mô hình. Các thuộc tính được chọn sẽ là:

- serum_creatinine (0.29).
- age (0.25)
- ejection_fraction (-0.27)
- time (-0.53)
- serum_sodium (-0.2)

3.4.2 Xây dựng mô hình và đào tạo mô hình

Xây dựng mô hình LR, KNN, DT, RF cho với tập dữ liệu đã có.

Tiến hành chia tập dữ liệu thành hai phần:

- Trainning set: 80%
- Testing set: 20%

Chạy code với bốn mô hình: LR, KNN, DT, RF.

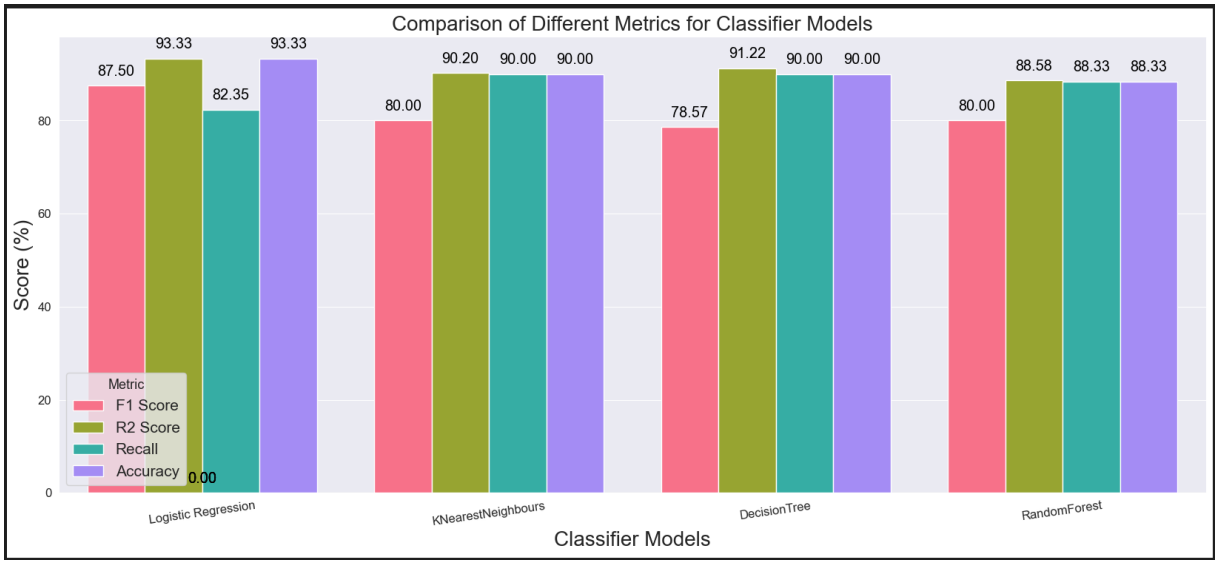
CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ

Sau quá trình đào tạo các mô hình, kết quả được đưa ra trong bảng sau:

Bảng 4.1 Kết quả của các mô hình

Model	Accuracy	F1 score	Precision score	Recall score
LR	0.933	0.875	0.933	0.824
KNN	0.9	0.8	0.902	0.9
DT	0.9	0.786	0.912	0.9
RR	0.883	0.8	0.886	0.883

Kết quả được mô hình đưa ra với tập test:



Hình 4.1: Biểu đồ các độ đo theo từng mô hình

Như đã thấy trong biểu đồ hình 3.2, tập dữ liệu huấn luyện đang gặp phải vấn đề mất cân bằng giữa hai lớp bệnh nhân sống sót và tử vong. Số lượng bệnh nhân sống sót đang áp đảo hơn hẳn so với số lượng bệnh nhân tử vong. Điều này có thể dẫn đến việc mô hình học máy có thể dẫn đến xu hướng dự đoán thiên vị về phía lớp đa số (lớp bệnh nhân sống sót).

Để đánh giá hiệu suất của các mô hình được đào tạo trên tập dữ liệu đã có, em sử dụng F1 score vì nó kết hợp cả Precision và Recall để đánh giá toàn diện hơn về khả năng dự đoán của mô hình do F1 score là độ đo phù hợp nhất với bài toán mà dữ liệu đầu vào mất cân bằng giữa các nhãn lớp.

Từ kết quả của bảng và biểu đồ, ta có thể thấy được mô hình LR có F1 score cao nhất và các chỉ số còn lại cũng rất cao. Nó cho thấy khả năng dự đoán đúng cho cả hai nhãn lớp sống và chết đều cao, giảm rủi ro dự đoán nhầm tỉ lệ tử vong của bệnh nhân. Phù hợp với tập dữ liệu ban đầu đã bị mất cân bằng từ trước.

KẾT LUẬN

Trong đồ án, em đã nghiên cứu, tìm hiểu bài toán về dự đoán tử vong của bệnh nhân suy tim và xây dựng một số mô hình học máy. Trong quá trình tìm hiểu, em thấy được tầm quan trọng của việc dự đoán sớm khả năng tử vong của người bệnh là rất quan trọng.

Kết quả đạt được

Về mặt lý thuyết:

- Tìm hiểu các phương pháp LR, KNN, DT, RF.
- Tìm hiểu được tổng quan các bước xây dựng mô hình Học máy.

Về mặt ứng dụng:

- Xây dựng và cài đặt các mô hình LR, KNN, DT, RF trên môi trường Python và đánh giá kết quả thực hiện của các mô hình dựa trên các tiêu chí: ACC, F1, Recall, Precision.
- Lựa chọn được mô hình LR là phù hợp cho bài toán dựa trên các tiêu chí ACC, F1, Recall, Precision.

Hạn chế

- Tập dữ liệu không có tính tổng quát cao do mất cân bằng về nhãn lớp.
- Độ chính xác chưa được cao.
- Nghiên cứu, tìm hiểu thêm các kĩ thuật phân tích và xử lý cũng như áp dụng thêm các mô hình học máy khác để cải thiện kết quả, tăng tính chính xác của bài toán.

Hướng phát triển

- Nâng cao chất lượng dữ liệu: thu thập dữ liệu từ nhiều nguồn khác nhau, tăng tính đa dạng của dữ liệu, giảm việc mất cân bằng của dữ liệu đó. Bổ sung thêm các chi tiết khác ảnh hưởng đến suy tim.
- Phát triển và tối ưu hóa mô hình: thử nghiệm các mô hình học máy hoặc học sâu khác. Có thể kết hợp nhiều loại mô hình để có thể giải quyết bài toán tốt hơn.

- Nghiên cứu mở rộng: mở rộng mô hình với một số biến cố khác như tái nhập viện, áp lực công việc,...
- Xây dựng giao diện cho phép người dùng có thể nhập dữ liệu và trả kết quả cho người dùng.

TÀI LIỆU THAM KHẢO

[1] N. T. Hop, "Viblo," [Online]. Available: https://viblo.asia/p/knn-k-nearest-neighbors-1-djeZ14ejKWz.

[2] H. X. Huấn, Giáo trình Học máy, Hà Nội: Nhà xuất bản Đại học Quốc gia Hà Nội, 2016.

[3] N. H. Nam, N. T. Thành and H. Q. Thụy, Giáo trình Khai phá dữ liệu, Hà Nội: Nhà xuất bản Đại học Quốc gia Hà Nội, 2016.

[4] N. T. K. Ngân, Bài giảng Học máy, Hà Nội, 2021.

[5] V. H. Tiệp, "Machine Learning cơ bản," [Online]. Available: https://machinelearningcoban.com/.

[6] N. T. Trung, Bài giảng Khai phá dữ liệu, Hà Nội, 2021.

[7] scikit-learn, "scikit-learn: machine learing in Python," [Online]. Available: https://scikit-learn.org/.

[8] Wikipedia, "Logistic regression," [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression.