TRƯỜNG ĐẠI HỌC SÀI GÒN
Khoa Công Nghệ Thông Tin

Đề Cương Nghiên Cứu Khoa Học

Môn Học: Phương Pháp Nghiên Cứu Khoa Học

Giảng viên: Đỗ Như Tài

Nhóm
Phạm Tấn Khương – 3122410191
Hoàng Vũ - 3122560089
Huỳnh Thanh Bình - 3122410033
Nguyễn Minh Tú - 3120411167

Thành Phố Hồ Chí Minh, tháng 5 năm 2025

Hoàng Vũ		100%
Huỳnh Thanh Bình		100%
Phạm Tấn Khương		100%
Nguyễn Minh Tú		100%

Mục Lục

Tóm tắt **3**

Giới thiệu **3**

Khảo sát các bài báo nghiên cứu khoa học **4**

Phương pháp luận **5**

Tổng quan hệ thống **7**

Phân tích và mô hình hoá **10**

Trích dẫn nguồn **17**

Tóm tắt

Trái tim đóng vai trò quan trọng trong các sinh vật sống. Việc chẩn đoán và dự đoán các bệnh liên quan đến tim đòi hỏi độ chính xác, hoàn hảo và đúng đắn cao, vì chỉ một sai sót nhỏ cũng có thể gây ra vấn đề mệt mỏi hoặc thậm chí dẫn đến tử vong. Hiện nay, số ca tử vong do bệnh tim đang ngày càng gia tăng theo cấp số nhân. Để đối phó với vấn đề này, việc phát triển một hệ thống dự đoán nhằm nâng cao nhận thức về bệnh là rất cần thiết.

Học máy là một nhánh của Trí tuệ Nhân tạo (AI), cung cấp sự hỗ trợ đắc lực trong việc dự đoán bất kỳ sự kiện nào bằng cách huấn luyện từ các sự kiện tự nhiên. Trong bài báo này, chúng tôi tính toán độ chính xác của các thuật toán học máy trong việc dự đoán bệnh tim. Các thuật toán được sử dụng bao gồm hồi quy logistic, bộ phân loại rừng ngẫu nhiên (random forest classifier) và máy vector hỗ trợ (SVM), với tập dữ liệu từ Kaggle để huấn luyện và kiểm tra. Để triển khai bằng lập trình Python, Jupyter Notebook trong Anaconda là công cụ tốt nhất, vì nó cung cấp nhiều thư viện và tệp tiêu đề giúp công việc trở nên chính xác và hiệu quả hơn.

Giới thiệu

Trái tim là một trong những cơ quan quan trọng và thiết yếu nhất của cơ thể con người, do đó, việc chăm sóc tim là điều vô cùng cần thiết. Phần lớn các bệnh đều liên quan đến tim, vì vậy, việc dự đoán bệnh tim là rất quan trọng và cần có các nghiên cứu so sánh trong lĩnh vực này. Ngày nay, nhiều bệnh nhân tử vong vì bệnh của họ chỉ được phát hiện ở giai đoạn cuối do độ chính xác thấp của các thiết bị y tế. Vì vậy, cần có những thuật toán hiệu quả hơn trong việc dự đoán bệnh.

Học máy (Machine Learning) là một trong những công nghệ hiệu quả để kiểm tra, dựa trên quá trình huấn luyện và thử nghiệm. Đây là một nhánh của Trí tuệ Nhân tạo (AI), một lĩnh vực rộng lớn trong đó máy móc có thể mô phỏng khả năng của con người. Hệ thống học máy được huấn luyện để xử lý và sử dụng dữ liệu, do đó, sự kết hợp giữa hai công nghệ này còn được gọi là Trí tuệ Máy móc (Machine Intelligence).

Theo định nghĩa của học máy, nó học từ các hiện tượng và dữ kiện tự nhiên. Vì vậy, trong dự án này, chúng tôi sử dụng các thông số sinh học làm dữ liệu kiểm tra, chẳng hạn như mức cholesterol, huyết áp, giới tính, tuổi tác, v.v. Dựa trên những thông số này, chúng tôi thực hiện so sánh độ chính xác của các thuật toán, bao gồm hồi quy logistic, rừng ngẫu nhiên (Random Forest) và máy vector hỗ trợ (SVM). Trong bài báo này, chúng tôi tính toán độ chính xác của ba phương pháp học máy khác nhau và từ đó kết luận thuật toán nào hoạt động tốt nhất.

Trái tim là một cơ quan không thể thiếu đối với con người. Bệnh tim là nguyên nhân hàng đầu gây tử vong trên thế giới. Theo Tổ chức Y tế Thế giới (WHO), mỗi năm có khoảng 12 triệu ca tử vong do bệnh tim mạch. Bệnh tim được xem là một "kẻ giết người thầm lặng" vì nó có thể dẫn đến tử vong mà không có triệu chứng rõ ràng. Việc phát hiện bệnh sớm giúp phòng ngừa và giảm thiểu các biến chứng. Như câu nói "phòng bệnh hơn chữa bệnh", ngăn ngừa bệnh tim có thể giúp ngăn chặn nhiều ca tử vong sớm và giảm tỷ lệ tử vong.

Các bác sĩ không thể giám sát bệnh nhân suốt 24 giờ. Mặc dù có nhiều thiết bị y tế trên thị trường, nhưng chúng chưa thể phát hiện chính xác bệnh tim, và một số thiết bị còn rất đắt đỏ, yêu cầu chuyên môn cao để sử dụng. Học máy là một công nghệ tiên tiến, giúp máy móc cải thiện hiệu suất qua trải nghiệm. Công nghệ này cho phép hệ thống tự nhận diện các mẫu dữ liệu và đưa ra dự đoán.

Trong dự án này, chúng tôi sử dụng học máy để dự đoán liệu một người có mắc bệnh tim hay không. Chúng tôi xem xét nhiều thuộc tính của bệnh nhân như giới tính, loại đau ngực, huyết áp khi đói, mức cholesterol trong huyết thanh, bài kiểm tra gắng sức (exang), v.v. Chúng tôi áp dụng các thuật toán như SVM, Random Forest và hồi quy logistic. Dựa trên các thuộc tính này, chúng tôi tiến hành phân tích so sánh độ chính xác của các thuật toán, và thuật toán nào có độ chính xác cao nhất sẽ được sử dụng để dự đoán bệnh tim.

Khảo sát các bài báo nghiên cứu khoa học

Bài báo này mô tả việc dự đoán bệnh tim trong lĩnh vực y tế thông qua việc sử dụng khoa học dữ liệu. Do có rất nhiều nghiên cứu được thực hiện liên quan đến vấn đề này, nhưng độ chính xác của dự đoán vẫn chưa được cải thiện đáng kể. Vì vậy, nghiên cứu này tập trung vào các kỹ thuật lựa chọn đặc trưng và các thuật toán, trong đó nhiều tập dữ liệu về bệnh tim được sử dụng để phân tích thực nghiệm nhằm đạt được độ chính xác cao hơn.

Chúng tôi đề xuất một phương pháp mới nhằm tìm ra các đặc trưng quan trọng bằng cách áp dụng các kỹ thuật học máy, từ đó cải thiện độ chính xác trong việc dự đoán bệnh tim mạch. Mô hình dự đoán được xây dựng với các tổ hợp đặc trưng khác nhau cùng với nhiều kỹ thuật phân loại đã biết.

Trong bài báo này, chúng tôi phân tích các thuật toán phân loại phổ biến được sử dụng trong các tập dữ liệu y tế để hỗ trợ dự đoán bệnh tim – một trong những nguyên nhân hàng đầu gây tử vong trên toàn thế giới. Việc dự đoán cơn đau tim là một nhiệm vụ phức tạp đối với các bác sĩ, vì nó đòi hỏi nhiều kinh nghiệm và kiến thức chuyên môn. Ngành y tế hiện nay chứa đựng nhiều thông tin tiềm ẩn nhưng có giá trị, có thể được khai thác để đưa ra quyết định chính xác hơn. Các thí nghiệm được thực hiện trong nghiên cứu này đã cho thấy thuật toán hoạt động như mong đợi.

Bảng 1: Khảo sát các bài báo nghiên cứu khoa học 1

Tác giả	Năm	Hướng đi	Input	Output	Cách giải quyết	Liên kết bài báo
Shital D. Bhatt Himanshu B. Soni	2021	Phân loại bệnh tim bằng các thuật toán ML	Dữ liệu lâm sàng (huyết áp, cholesterol, nhịp tim, tuổi, giới tính)	Dự đoán có/không mắc bệnh tim	Sử dụng Random Forest, Decision Tree và tối ưu hóa bằng kỹ thuật lai (Hybrid Intelligent Techniques)	**Image Retrieval using Bag-of-Features for Lung Cancer Classification | IEEE Conference Publication | IEEE Xplore**
Nhóm người của khoa liên quan khoa học máy tính đến từ Đại học Khoa học và Công nghệ Điện tử Trung Quốc	2018	Phát hiện bệnh tim bằng học máy có giám sát	Dữ liệu từ Cleveland Heart Disease Dataset (tuổi, giới tính, ECG, v.v.)	Phân loại bệnh tim (0-4 mức độ)	Áp dụng SVM, KNN, và Logistic Regression với lựa chọn đặc trưng bằng PCA	**Heart Disease Prediction System Using Model Of Machine Learning and Sequential Backward Selection Algorithm for Features Selection | IEEE Conference Publication | IEEE Xplore**
Nhóm người đến từ học viện công nghệ Vellore	2020	Dự đoán bệnh tim bằng học sâu	Dữ liệu lâm sàng và hình ảnh (ECG, siêu âm tim)	Xác suất mắc bệnh tim	Sử dụng mạng nơ-ron sâu (Deep Neural Network) kết hợp CNN để phân tích dữ liệu đa chiều	**Prediction of Cardiovascular Disease Using Machine Learning Algorithms | IEEE Conference Publication | IEEE Xplore**
Dr.S.Subbulakshmil, Dr.G.Marimuthu, Mrs.N.Neelavathy	2018	Phân loại bệnh tim bằng kỹ thuật lai	Dữ liệu từ UCI Repository (13 thuộc tính: tuổi, cholesterol, v.v.)	Phân loại có/không mắc bệnh tim	Sử dụng Genetic Algorithm kết hợp Naive Bayes để tối ưu hóa hiệu suất	**L0808067077-libre.pdf**
Nhóm người đến từ Cao đẳng Kỹ thuật Sardar Patel	2018	Dự đoán bệnh tim bằng ML đơn giản	Dữ liệu lâm sàng cơ bản (tuổi, giới tính, nhịp tim, huyết áp)	Dự đoán nguy cơ bệnh tim	Áp dụng KNN và Decision Tree với tiền xử lý dữ liệu đơn giản	**Prediction of Heart Disease Using Machine Learning | IEEE Conference Publication | IEEE Xplore**

Phương pháp luận

1. Mục tiêu nghiên cứu

Mục tiêu của khảo sát là thu thập dữ liệu từ đối tượng tham gia để đánh giá các yếu tố nguy cơ liên quan đến bệnh tim, từ đó xây dựng cơ sở dữ liệu phục vụ việc dự đoán bệnh tim bằng các phương pháp phân tích hoặc mô hình học máy.

2. Thiết kế nghiên cứu

Nghiên cứu sử dụng phương pháp khảo sát cắt ngang (cross-sectional survey) để thu thập thông tin từ một nhóm dân số tại một thời điểm nhất định. Phương pháp này được chọn vì tính hiệu quả trong việc xác định mối liên hệ giữa các yếu tố nguy cơ và bệnh tim[3].

3. Dân số và mẫu nghiên cứu

Dân số mục tiêu: Người trưởng thành từ 18 tuổi trở lên, không phân biệt giới tính, có hoặc không có tiền sử bệnh tim.

- Cỡ mẫu: Cỡ mẫu được tính dựa trên công thức Slovin:

$$\boldsymbol{n = \frac{1 + Ne}{2N}}$$

Trong đó:

- n: cỡ mẫu
- N: tổng số dân số mục tiêu
- e: sai số cho phép (thường lấy 5%, tức 0.05).

Ví dụ, nếu dân số mục tiêu là 10.000 người, với sai số 5%, cỡ mẫu tối thiểu là khoảng 370 người.

- Phương pháp chọn mẫu: Chọn mẫu ngẫu nhiên phân tầng (stratified random sampling) để đảm bảo đại diện cho các nhóm tuổi, giới tính và khu vực địa lý[1].

4. Công cụ thu thập dữ liệu

- Bảng câu hỏi khảo sát: Được thiết kế dựa trên các yếu tố nguy cơ bệnh tim đã được xác định trong y văn, bao gồm: tuổi, giới tính, chỉ số khối cơ thể (BMI), tiền sử hút thuốc, mức độ hoạt động thể chất, chế độ ăn uống, tiền sử gia đình về bệnh tim, huyết áp, mức cholesterol và bệnh tiểu đường.

- Nguồn tham khảo: Bảng câu hỏi được xây dựng dựa trên hướng dẫn của Tổ chức Y tế Thế giới (WHO) về giám sát các yếu tố nguy cơ bệnh không lây nhiễm (WHO, 2013)[4].

- Hình thức khảo sát: Kết hợp khảo sát trực tuyến (qua Google Forms hoặc nền tảng tương tự) và phỏng vấn trực tiếp (nếu cần).

5. Quy trình thu thập dữ liệu

1. Chuẩn bị: Thiết kế và kiểm tra thử bảng câu hỏi trên một nhóm nhỏ (10-20 người) để đảm bảo tính rõ ràng và phù hợp.

2. Triển khai: Phát bảng câu hỏi đến đối tượng tham gia qua email, mạng xã hội hoặc tại các cơ sở y tế.

3. Thời gian: Thu thập dữ liệu trong vòng 4-6 tuần, tùy thuộc vào quy mô mẫu.

4. Kiểm soát chất lượng: Kiểm tra dữ liệu để loại bỏ các câu trả lời không đầy đủ hoặc không hợp lệ.

6. Phân tích dữ liệu

- Phương pháp phân tích: Sử dụng thống kê mô tả (tần suất, trung bình, độ lệch chuẩn) để tổng hợp đặc điểm mẫu. Áp dụng phân tích hồi quy logistic (logistic regression) để xác định mối quan hệ giữa các yếu tố nguy cơ và khả năng mắc bệnh tim[2].

- Công cụ phần mềm: Sử dụng SPSS, R hoặc Python để xử lý và phân tích dữ liệu.

7. Vấn đề đạo đức

- Thu thập sự đồng ý của người tham gia trước khi tiến hành khảo sát.

- Đảm bảo tính bảo mật thông tin cá nhân theo quy định của Luật Bảo vệ Dữ liệu Cá nhân (nếu áp dụng tại quốc gia của bạn).

- Nghiên cứu tuân thủ các nguyên tắc đạo đức trong nghiên cứu y sinh học của Tuyên bố Helsinki (WMA, 2013)[5].

8. Hạn chế của nghiên cứu

- Khảo sát có thể gặp thiên lệch hồi đáp (response bias) nếu người tham gia không trả lời trung thực.

- Thiết kế cắt ngang không thể xác định mối quan hệ nhân quả, chỉ có thể đánh giá sự liên quan.

Tổng quan hệ thống

Hệ thống dự đoán bệnh tim được xây dựng dựa trên một quy trình học máy tiêu chuẩn, bao gồm các bước từ thu thập dữ liệu đến triển khai giao diện người dùng. Dưới đây là các bước chính:

Thu thập và xử lý dữ liệu

- Đầu tiên, dữ liệu được thu thập từ các nguồn như hồ sơ y tế hoặc kho dữ liệu công cộng, chẳng hạn như [UCI Machine Learning Repository](http://archive.ics.uci.edu/ml).

- Dữ liệu sau đó được xử lý để làm sạch, chuẩn hóa, và chuyển đổi thành định dạng phù hợp cho mô hình, sử dụng các công cụ như thư viện scikit-learn.

Phân tích và huấn luyện

- Phân tích dữ liệu khám phá (EDA) được thực hiện để hiểu rõ hơn về dữ liệu, tìm kiếm mẫu và mối quan hệ giữa các biến số.

- Dữ liệu được chia thành tập huấn luyện và tập kiểm tra, thường theo tỷ lệ như 70-30 hoặc 80-20, để đánh giá hiệu quả mô hình.

- Các thuật toán học máy, như hồi quy logistic, cây quyết định, hoặc rừng ngẫu nhiên, được áp dụng để huấn luyện mô hình trên tập huấn luyện.

Đánh giá và triển khai

- Độ chính xác của mô hình được kiểm tra bằng các chỉ số như độ chính xác, độ chính xác chi tiết, độ nhạy, và F1-score, sau đó áp dụng các kỹ thuật đánh giá nâng cao như kiểm định chéo (cross-validation).

- Chọn hai mô hình tốt nhất và sử dụng nó để dự đoán trên dữ liệu mới, sau đó được lưu bằng thư viện Joblib để triển khai sau này.

- Cuối cùng, một giao diện người dùng (GUI) được phát triển, chẳng hạn bằng Tkinter hoặc Streamlit, để người dùng có thể tương tác dễ dàng.

Một chi tiết bất ngờ: việc lưu mô hình bằng Joblib không chỉ đơn giản là lưu trữ mà còn giúp giảm thời gian xử lý khi triển khai, đặc biệt hữu ích cho các hệ thống y tế cần phản hồi nhanh.

Ghi chú chi tiết về hệ thống dự đoán bệnh tim

Hệ thống dự đoán bệnh tim được thiết kế dựa trên một quy trình học máy tiêu chuẩn, với mục tiêu thu thập, xử lý, và phân tích dữ liệu để xây dựng mô hình dự đoán chính xác. Dưới đây là mô tả chi tiết từng bước, bao gồm các phương pháp và nguồn tham khảo, nhằm đảm bảo tính minh bạch và đáng tin cậy.

Mô tả hệ thống

Hệ thống bắt đầu bằng việc thu thập dữ liệu và chọn các thuộc tính quan trọng, sau đó xử lý dữ liệu thành định dạng phù hợp, chia thành tập huấn luyện và kiểm tra, áp dụng các thuật toán học máy, đánh giá độ chính xác, và cuối cùng triển khai giao diện người dùng (GUI). Quy trình này phản ánh một pipeline học máy điển hình, được điều chỉnh để phù hợp với dự đoán bệnh tim, một lĩnh vực nhạy cảm và quan trọng trong y tế.

Các mô-đun chi tiết

Dưới đây là danh sách các mô-đun và giải thích chi tiết, kèm theo các nguồn tham khảo phù hợp:

1. Thu thập tập dữ liệu (Collection of Dataset):

- Bước này liên quan đến việc thu thập dữ liệu từ các nguồn đáng tin cậy, chẳng hạn như hồ sơ y tế, khảo sát, hoặc kho dữ liệu công cộng. Một nguồn phổ biến là [UCI Machine Learning Repository](http://archive.ics.uci.edu/ml), nơi cung cấp tập dữ liệu về bệnh tim như tập Cleveland Heart Disease.

- Dữ liệu thường bao gồm các yếu tố như tuổi, giới tính, chỉ số khối cơ thể (BMI), tiền sử hút thuốc, huyết áp, mức cholesterol, và bệnh tiểu đường. Việc chọn các thuộc tính quan trọng có thể được tích hợp ở bước này hoặc sau khi phân tích dữ liệu[6].

2. Xử lý dữ liệu (Data Processing):

- Sau khi thu thập, dữ liệu cần được làm sạch để xử lý các giá trị thiếu, loại bỏ nhiễu, và chuẩn hóa để đảm bảo tính đồng nhất. Các kỹ thuật phổ biến bao gồm xử lý giá trị thiếu, chuẩn hóa (normalization), và mã hóa biến hạng (encoding).

- Thư viện như scikit-learn cung cấp các công cụ mạnh mẽ cho bước này, chẳng hạn như StandardScaler cho chuẩn hóa hoặc SimpleImputer cho xử lý giá trị thiếu[7].

3. Phân tích dữ liệu khám phá (EDA - Exploratory Data Analysis):

- EDA được thực hiện để hiểu rõ hơn về dữ liệu, bao gồm thống kê mô tả (tần suất, trung bình, độ lệch chuẩn), trực quan hóa (biểu đồ phân tán, biểu đồ hộp), và phân tích mối quan hệ giữa các biến (hệ số tương quan).

- Ví dụ, EDA có thể giúp xác định mối liên hệ giữa huyết áp cao và nguy cơ bệnh tim, hoặc phân tích sự phân bố của tuổi trong tập dữ liệu. Sách như "Exploratory Data Analysis" của John W. Tukey là nguồn tham khảo cổ điển, trong khi "Python Data Science Handbook" của Jake VanderPlas cung cấp hướng dẫn hiện đại[8].

4. Chia tập dữ liệu (Splitting of dataset):

- Dữ liệu được chia thành tập huấn luyện (training set) và tập kiểm tra (testing set), thường theo tỷ lệ như 70-30 hoặc 80-20, để đánh giá hiệu quả mô hình. Bước này đảm bảo mô hình không bị overfitting và có khả năng tổng quát hóa trên dữ liệu mới[9].

5. Áp dụng các thuật toán khác nhau (Applying different algorithms):

- Các thuật toán học máy, như hồi quy logistic, cây quyết định, rừng ngẫu nhiên (random forest), hoặc máy vector hỗ trợ (SVM), được áp dụng để huấn luyện mô hình trên tập huấn luyện. Việc chọn thuật toán phụ thuộc vào đặc điểm dữ liệu và mục tiêu dự đoán.

- Có nhiều nghiên cứu so sánh hiệu quả của các thuật toán này trong dự đoán bệnh tim, chẳng hạn như các bài báo về "Heart Disease Prediction Using Machine Learning Algorithms". Tuy nhiên, không có thuật toán nào là tốt nhất cho mọi trường hợp (no free lunch theorem)[10].

6. Kiểm tra điểm số độ chính xác của các thuật toán (Checking accuracy scores of algorithms):

- Sau khi huấn luyện, hiệu suất của từng mô hình được đánh giá bằng các chỉ số như độ chính xác (accuracy), độ chính xác chi tiết (precision), độ nhạy (recall), và F1-score. Các chỉ số này giúp so sánh hiệu quả của các mô hình[11].

7. Áp dụng các kỹ thuật đánh giá mô hình khác nhau (Applying different model evaluation techniques):

- Ngoài việc kiểm tra điểm số, các kỹ thuật đánh giá nâng cao như kiểm định chéo (k-fold cross-validation), đường cong ROC (Receiver Operating Characteristic), và ma trận nhầm lẫn (confusion matrix) được sử dụng để đánh giá độ tin cậy và khả năng tổng quát hóa của mô hình[12].

8. Dự đoán trên giá trị dữ liệu mới (Prediction on new data values):

- Chọn hai mô hình tốt nhất, hai mô hình được sử dụng để dự đoán trên dữ liệu mới, chẳng hạn như thông tin của bệnh nhân chưa được phân loại. Bước này là mục tiêu cuối cùng của hệ thống, kết hợp kết quả dự đoán của hai mô hình lại với nhau để đưa ra kết luận thống nhất hơn về dự đoán nguy cơ bị bệnh tim.

9. Lưu mô hình bằng JOBLIB (Save model using JOBLIB):

- Mô hình được lưu bằng thư viện Joblib, một công cụ Python giúp lưu và tải các đối tượng Python, bao gồm mô hình học máy. Điều này giúp giảm thời gian xử lý khi triển khai mô hình trong thực tế, đặc biệt hữu ích cho các hệ thống y tế cần phản hồi nhanh.

- Tài liệu của Joblib hoặc các bài viết về triển khai mô hình cung cấp hướng dẫn chi tiết[13].

10. Giao diện người dùng (GUI):

- Một giao diện đồ họa (GUI) được phát triển để người dùng, chẳng hạn như bác sĩ hoặc bệnh nhân, có thể tương tác với hệ thống mà không cần kiến thức lập trình. Các thư viện như Tkinter, PyQt, hoặc Streamlit thường được sử dụng.

- Việc phát triển GUI giúp tăng tính thân thiện với người dùng, đặc biệt trong bối cảnh y tế, nơi giao diện trực quan rất quan trọng[14].

Điểm đáng chú ý

Một chi tiết bất ngờ là việc lưu mô hình bằng Joblib không chỉ đơn giản là lưu trữ mà còn giúp giảm thời gian xử lý khi triển khai, đặc biệt hữu ích cho các hệ thống y tế cần phản hồi nhanh. Điều này có thể không được chú ý trong các nghiên cứu ban đầu, nhưng rất quan trọng trong thực tế.

Ngoài ra, có thể có sự khác biệt trong cách chọn thuật toán và đánh giá, tùy thuộc vào nguồn dữ liệu và mục tiêu nghiên cứu, phản ánh tính phức tạp của lĩnh vực này. Ví dụ, một số nghiên cứu có thể ưu tiên độ nhạy (recall) hơn độ chính xác (accuracy) để đảm bảo phát hiện tất cả các trường hợp bệnh tim, ngay cả khi có một số dự đoán sai.

Phân tích và mô hình hoá

Hồi quy Logistic, Máy vector hỗ trợ (SVM), và Rừng ngẫu nhiên (Random Forest)

Hồi quy Logistic, Máy vector hỗ trợ (SVM), và Rừng ngẫu nhiên (Random Forest) là ba thuật toán học máy được sử dụng phổ biến trong việc dự đoán bệnh tim, mỗi thuật toán có ưu và nhược điểm riêng.

Hồi quy Logistic

Hồi quy Logistic là một thuật toán phân loại học có giám sát, được sử dụng để giải quyết cả vấn đề phân loại và hồi quy. Trong các vấn đề phân loại, biến mục tiêu có thể ở định dạng nhị phân hoặc rời rạc, chẳng hạn như 0 hoặc 1. Thuật toán này hoạt động dựa trên hàm sigmoid, giúp đưa ra kết quả phân loại nhị phân như Có/Không, Đúng/Sai, v.v. Hàm sigmoid trả về giá trị giữa 0 và 1, với ngưỡng thường là 0,5: nếu giá trị nhỏ hơn 0,5 thì được coi là 0, và lớn hơn 0,5 thì được coi là 1.

Rừng ngẫu nhiên (Random Forest)

Rừng ngẫu nhiên là một kỹ thuật học có giám sát, có thể được sử dụng cho cả phân loại và hồi quy. Nó dựa trên việc kết hợp nhiều bộ phân loại (ensemble learning) để cải thiện hiệu suất mô hình, đặc biệt là trong việc xử lý các vấn đề phức tạp. Rừng ngẫu nhiên bao gồm nhiều cây quyết định được xây dựng trên các tập con khác nhau của tập dữ liệu, và kết quả cuối cùng được lấy dựa trên đa số phiếu bầu (majority voting) cho phân loại hoặc trung bình của các kết quả cho hồi quy. Số lượng cây quyết định càng lớn, độ chính xác càng cao và giảm nguy cơ overfitting.

Máy vector hỗ trợ (SVM)

Máy vector hỗ trợ (SVM) là một thuật toán học có giám sát, được sử dụng để phân tích dữ liệu và giải quyết các vấn đề phân loại và hồi quy. Mô hình SVM biểu diễn các ví dụ như các điểm trong không gian, được ánh xạ sao cho các ví dụ thuộc các hạng mục khác nhau được phân tách bởi một khoảng cách rõ ràng, gọi là siêu phẳng (hyperplane). Dựa trên tập huấn luyện, thuật toán SVM xây dựng mô hình để phân loại các ví dụ mới vào một trong hai hạng mục, làm cho nó trở thành một bộ phân loại nhị phân tuyến tính không xác suất.

So sánh và Lựa chọn Thuật toán

Trong hệ thống dự đoán bệnh tim, mục tiêu là chọn thuật toán có độ chính xác cao nhất giữa Hồi quy Logistic, SVM, và Rừng ngẫu nhiên. Dựa trên các nghiên cứu, Rừng ngẫu nhiên thường cho thấy hiệu suất tốt hơn nhờ khả năng xử lý các mối quan hệ phức tạp và giảm overfitting. Tuy nhiên, hiệu suất thực tế phụ thuộc vào tập dữ liệu cụ thể, bao gồm cách tiền xử lý dữ liệu và lựa chọn đặc trưng.

Ví dụ, một nghiên cứu trên tập dữ liệu bệnh tim Cleveland cho thấy Rừng ngẫu nhiên đạt độ chính xác 88,5%, trong khi Hồi quy Logistic và SVM chỉ đạt 80,32% (Implementation of a Heart Disease Risk Prediction Model Using Machine Learning). Tuy nhiên, một nghiên cứu khác trên cùng tập dữ liệu cho thấy Hồi quy Logistic đạt 90,16%, cao hơn so với các thuật toán khác, bao gồm Rừng ngẫu nhiên (Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization). Điều này cho thấy sự khác biệt có thể đến từ cách xử lý dữ liệu hoặc tối ưu hóa tham số.

Do đó, để đảm bảo chọn được thuật toán tốt nhất, bạn nên:

- Thử nghiệm cả ba thuật toán trên tập dữ liệu của mình.
- Sử dụng các kỹ thuật đánh giá như kiểm định chéo (cross-validation) và ma trận nhầm lẫn (confusion matrix) để so sánh độ chính xác, độ chính xác chi tiết (precision), độ nhạy (recall), và F1-score.
- Tối ưu hóa tham số bằng GridSearchCV hoặc các phương pháp tương tự để đạt hiệu suất cao nhất.

Một chi tiết bất ngờ: Rừng ngẫu nhiên không chỉ cung cấp độ chính xác cao mà còn cho phép đánh giá tầm quan trọng của các đặc trưng (feature importance), rất hữu ích cho việc xác định các yếu tố nguy cơ bệnh tim, chẳng hạn như tuổi, huyết áp, hoặc mức cholesterol, hỗ trợ giải thích lâm sàng.

Báo cáo chi tiết

Dưới đây là báo cáo chi tiết về việc so sánh và lựa chọn thuật toán cho hệ thống dự đoán bệnh tim, dựa trên các nghiên cứu và phân tích hiện có. Báo cáo này bao gồm mô tả chi tiết về từng thuật toán, các nghiên cứu so sánh, và khuyến nghị thực tiễn.

Mô tả chi tiết các thuật toán

1. **Hồi quy Logistic**
    - Hồi quy Logistic là một thuật toán phân loại học có giám sát, sử dụng hàm sigmoid để ánh xạ đầu vào thành xác suất thuộc một trong hai lớp (0 hoặc 1). Hàm sigmoid trả về giá trị giữa 0 và 1, với ngưỡng thường là 0,5 để quyết định lớp.
    - Ưu điểm: Đơn giản, dễ giải thích, nhanh chóng, phù hợp với tập dữ liệu nhỏ.
    - Nhược điểm: Giả định mối quan hệ tuyến tính, có thể không bắt được các mối quan hệ phức tạp.
    - **Nguồn tham khảo:** James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). New York: Springer (An introduction to statistical learning).
2. **Máy vector hỗ trợ (SVM)**
    - SVM là một thuật toán học có giám sát, tìm siêu phẳng (hyperplane) tối ưu để phân tách các lớp trong không gian đặc trưng. Với dữ liệu phi tuyến, SVM có thể sử dụng nhân (kernel) như RBF để ánh xạ vào không gian cao hơn.
    - Ưu điểm: Xử lý tốt dữ liệu phi tuyến với nhân phù hợp, hiệu quả trong không gian chiều cao.

- Nhược điểm: Tốn tài nguyên tính toán, nhạy cảm với việc chọn tham số, khó giải thích.
- **Nguồn tham khảo:** Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python*. Journal of machine learning research, 12, 2825-2830 (Scikit-learn: Machine learning in Python).

3. **Rừng ngẫu nhiên (Random Forest)**
   - Rừng ngẫu nhiên là một phương pháp ensemble, kết hợp nhiều cây quyết định (decision trees) trên các tập con ngẫu nhiên của dữ liệu. Kết quả cuối cùng dựa trên đa số phiếu bầu cho phân loại hoặc trung bình cho hồi quy.
   - Ưu điểm: Xử lý cả dữ liệu tuyến tính và phi tuyến, giảm overfitting, cung cấp tầm quan trọng của đặc trưng, phù hợp với dữ liệu phức tạp.
   - Nhược điểm: Ít dễ giải thích hơn Hồi quy Logistic, có thể chậm hơn với tập dữ liệu lớn.
   - **Nguồn tham khảo:** Breiman, L. (2001). *Random forests*. Machine learning, 45(1), 5-32 (Random forests).

So sánh hiệu suất dựa trên các nghiên cứu

Các nghiên cứu so sánh hiệu suất của Hồi quy Logistic, SVM, và Rừng ngẫu nhiên trên tập dữ liệu dự đoán bệnh tim, đặc biệt là tập dữ liệu Cleveland, cho thấy sự khác biệt đáng kể:

- Một nghiên cứu từ PMC (Implementation of a Heart Disease Risk Prediction Model Using Machine Learning) cho thấy:
  - Rừng ngẫu nhiên đạt độ chính xác 88,5%.
  - Hồi quy Logistic đạt 80,32%.
  - SVM đạt 80,32%.
  - Kết luận: Rừng ngẫu nhiên vượt trội hơn đáng kể.
- Một nghiên cứu khác từ MDPI (Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization) trên tập dữ liệu Cleveland cho thấy:
  - Hồi quy Logistic đạt 90,16%, cao nhất trong sáu thuật toán (bao gồm Rừng ngẫu nhiên, KNN, Naïve Bayes, Gradient Boosting, và AdaBoost).
  - Độ chính xác của Rừng ngẫu nhiên không được nêu cụ thể, nhưng thấp hơn Hồi quy Logistic.
  - Kết luận: Hồi quy Logistic có thể vượt trội trong một số trường hợp, tùy thuộc vào cách xử lý dữ liệu.
- Một bài viết trên Medium (Using KNN, Logistic Regression, and SVM to predict Heart Disease Dataset) so sánh KNN, Hồi quy Logistic, và SVM, với:
  - KNN đạt độ chính xác 0,83 và precision 0,79.
  - Hồi quy Logistic đạt độ chính xác 0,81 và precision 0,77.
  - SVM đạt độ chính xác 0,80 và precision 0,75.
  - Lưu ý: Rừng ngẫu nhiên không được so sánh, nhưng Hồi quy Logistic tốt hơn SVM trong trường hợp này.

Phân tích và Khuyến nghị

- **Hiệu suất chung:** Rừng ngẫu nhiên thường có lợi thế nhờ khả năng xử lý dữ liệu phức tạp và giảm overfitting, đặc biệt với tập dữ liệu có nhiều đặc trưng và mối quan hệ phi tuyến. Tuy nhiên, Hồi quy Logistic có thể cạnh tranh nếu dữ liệu có mối quan hệ tuyến tính mạnh, và SVM có thể hiệu quả với nhân phù hợp, nhưng thường yêu cầu tối ưu hóa tham số.
- **Chi tiết bất ngờ:** Rừng ngẫu nhiên không chỉ cung cấp độ chính xác cao mà còn cho phép đánh giá tầm quan trọng của đặc trưng, rất hữu ích cho việc xác định các yếu tố nguy cơ bệnh tim, chẳng hạn như tuổi, huyết áp, hoặc mức cholesterol, hỗ trợ giải thích lâm sàng.
- **Khuyến nghị thực tiễn:**
    - Thử nghiệm cả ba thuật toán trên tập dữ liệu của bạn, sử dụng các kỹ thuật đánh giá như kiểm định chéo (k-fold cross-validation), ma trận nhầm lẫn (confusion matrix), và các chỉ số như độ chính xác (accuracy), precision, recall, và F1-score.
    - Tối ưu hóa tham số bằng GridSearchCV hoặc các phương pháp tương tự để đạt hiệu suất cao nhất.
    - Xem xét Rừng ngẫu nhiên là lựa chọn ưu tiên, nhưng không bỏ qua Hồi quy Logistic nếu dữ liệu có xu hướng tuyến tính, và SVM nếu dữ liệu có không gian chiều cao.

Phương pháp luận và Thảo luận

Giới thiệu về Hệ thống Dự đoán Bệnh Tim

Hệ thống dự đoán bệnh tim được xây dựng dựa trên một quy trình học máy tiêu chuẩn, với mục tiêu thu thập, xử lý, và phân tích dữ liệu để xây dựng mô hình dự đoán chính xác. Dưới đây là mô tả chi tiết từng bước, kèm theo các nguồn tham khảo phù hợp, nhằm đảm bảo tính minh bạch và đáng tin cậy.

Các bước chi tiết

1. **Thu thập tập dữ liệu (Collection of Dataset):**
    - Dữ liệu được thu thập từ các nguồn đáng tin cậy, chẳng hạn như hồ sơ y tế, khảo sát, hoặc kho dữ liệu công cộng. Một nguồn phổ biến là UCI Machine Learning Repository, nơi cung cấp tập dữ liệu về bệnh tim như tập Cleveland Heart Disease.
    - Dữ liệu thường bao gồm các yếu tố như tuổi, giới tính, chỉ số khối cơ thể (BMI), tiền sử hút thuốc, huyết áp, mức cholesterol, và bệnh tiểu đường. Việc chọn các thuộc tính quan trọng có thể được tích hợp ở bước này hoặc sau khi phân tích dữ liệu.
    - Paper của cơ sở dữ liệu:
    - International application of a new probability algorithm for the diagnosis of coronary artery disease - PubMed
    - Dữ liệu ở đâu:
    - UCI Heart Disease Data

2. **Xử lý dữ liệu (Data Processing):**
    - Sau khi thu thập, dữ liệu cần được làm sạch để xử lý các giá trị thiếu, loại bỏ nhiễu, và chuẩn hóa để đảm bảo tính đồng nhất. Các kỹ thuật phổ biến bao

gồm xử lý giá trị thiếu, chuẩn hóa (normalization), và mã hóa biến hạng (encoding).

- Thư viện như scikit-learn cung cấp các công cụ mạnh mẽ cho bước này, chẳng hạn như StandardScaler cho chuẩn hóa hoặc SimpleImputer cho xử lý giá trị thiếu.
- **Nguồn tham khảo:** Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12, 2825-2830.

3. **Phân tích dữ liệu khám phá (EDA - Exploratory Data Analysis):**
   - EDA được thực hiện để hiểu rõ hơn về dữ liệu, bao gồm thống kê mô tả (tần suất, trung bình, độ lệch chuẩn), trực quan hóa (biểu đồ phân tán, biểu đồ hộp), và phân tích mối quan hệ giữa các biến (hệ số tương quan).
   - Ví dụ, EDA có thể giúp xác định mối liên hệ giữa huyết áp cao và nguy cơ bệnh tim, hoặc phân tích sự phân bố của tuổi trong tập dữ liệu. Sách như "Exploratory Data Analysis" của John W. Tukey là nguồn tham khảo cổ điển, trong khi "Python Data Science Handbook" của Jake VanderPlas cung cấp hướng dẫn hiện đại.
   - **Nguồn tham khảo:** Tukey, J. W. (1977). Exploratory data analysis. Reading, Mass.; VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.
4. **Chia tập dữ liệu (Splitting of dataset):**
   - Dữ liệu được chia thành tập huấn luyện (training set) và tập kiểm tra (testing set), thường theo tỷ lệ như 70-30 hoặc 80-20, để đánh giá hiệu quả mô hình. Bước này đảm bảo mô hình không bị overfitting và có khả năng tổng quát hóa trên dữ liệu mới.
   - Sách như "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow" của Aurélien Géron giải thích chi tiết về việc chia tập dữ liệu và tầm quan trọng của nó.
   - **Nguồn tham khảo:** Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
5. **Áp dụng các thuật toán khác nhau (Applying different algorithms):**
   - Các thuật toán học máy, như hồi quy logistic, cây quyết định, rừng ngẫu nhiên (random forest), hoặc máy vector hỗ trợ (SVM), được áp dụng để huấn luyện mô hình trên tập huấn luyện. Việc chọn thuật toán phụ thuộc vào đặc điểm dữ liệu và mục tiêu dự đoán.
   - Có nhiều nghiên cứu so sánh hiệu quả của các thuật toán này trong dự đoán bệnh tim, chẳng hạn như các bài báo về "Heart Disease Prediction Using Machine Learning Algorithms". Tuy nhiên, không có thuật toán nào là tốt nhất cho mọi trường hợp (no free lunch theorem).
   - **Nguồn tham khảo:** Các bài báo so sánh thuật toán, chẳng hạn "Heart Disease Prediction Using Machine Learning Algorithms" (giả định).

6. **Kiểm tra điểm số độ chính xác của các thuật toán (Checking accuracy scores of algorithms):**
    - Sau khi huấn luyện, hiệu suất của từng mô hình được đánh giá bằng các chỉ số như độ chính xác (accuracy), độ chính xác chi tiết (precision), độ nhạy (recall), và F1-score. Các chỉ số này giúp so sánh hiệu quả của các mô hình.
    - Sách như "An Introduction to Statistical Learning" cung cấp hướng dẫn chi tiết về cách tính và giải thích các chỉ số này.
    - **Nguồn tham khảo:** James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.
7. **Áp dụng các kỹ thuật đánh giá mô hình khác nhau (Applying different model evaluation techniques):**
    - Ngoài việc kiểm tra điểm số, các kỹ thuật đánh giá nâng cao như kiểm định chéo (k-fold cross-validation), đường cong ROC (Receiver Operating Characteristic), và ma trận nhầm lẫn (confusion matrix) được sử dụng để đánh giá độ tin cậy và khả năng tổng quát hóa của mô hình.
    - Sách "The Elements of Statistical Learning" cung cấp lý thuyết sâu về các kỹ thuật này.
    - **Nguồn tham khảo:** Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.
8. **Dự đoán trên giá trị dữ liệu mới (Prediction on new data values):**
    - Chọn hai mô hình tốt nhất, hai mô hình đó được sử dụng để dự đoán trên dữ liệu mới, chẳng hạn như thông tin của bệnh nhân chưa được phân loại. Bước này là mục tiêu cuối cùng của hệ thống, kết hợp kết quả dự đoán của hai mô hình lại với nhau để đưa ra kết luận thống nhất hơn về dự đoán nguy cơ bị bệnh tim.
    - Quy trình này được hỗ trợ bởi các thư viện như scikit-learn, và được mô tả trong nhiều tài liệu học máy tiêu chuẩn.
    - **Nguồn tham khảo:** Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
9. **Lưu mô hình bằng JOBLIB (Save model using JOBLIB):**
    - Mô hình được lưu bằng thư viện Joblib, một công cụ Python giúp lưu và tải các đối tượng Python, bao gồm mô hình học máy. Điều này giúp giảm thời gian xử lý khi triển khai mô hình trong thực tế, đặc biệt hữu ích cho các hệ thống y tế cần phản hồi nhanh.
    - Tài liệu của Joblib hoặc các bài viết về triển khai mô hình cung cấp hướng dẫn chi tiết.
    - **Nguồn tham khảo:** Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.
10. **Giao diện người dùng (GUI):**

- Một giao diện đồ họa (GUI) được phát triển để người dùng, chẳng hạn như bác sĩ hoặc bệnh nhân, có thể tương tác với hệ thống mà không cần kiến thức lập trình. Các thư viện như Tkinter, PyQt, hoặc Streamlit thường được sử dụng.
- Việc phát triển GUI giúp tăng tính thân thiện với người dùng, đặc biệt trong bối cảnh y tế, nơi giao diện trực quan rất quan trọng.
- **Nguồn tham khảo:** Tài liệu chung về thư viện GUI Python, chẳng hạn McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. "O'Reilly Media, Inc."

Kết luận Cuối cùng

Dự án này dự đoán bệnh tim bằng cách trích xuất lịch sử y tế của bệnh nhân, dẫn đến các bệnh tim nguy hiểm, từ một tập dữ liệu bao gồm lịch sử y tế như đau ngực, mức đường huyết, huyết áp, v.v. Tim là một cơ quan thiết yếu của cơ thể con người, và việc dự đoán bệnh tim là một mối quan tâm quan trọng. Độ chính xác của thuật toán là một thông số quan trọng để phân tích hiệu suất, và nó phụ thuộc vào tập dữ liệu được sử dụng cho huấn luyện và kiểm tra. Dựa trên các nghiên cứu, Rừng ngẫu nhiên thường có hiệu suất tốt hơn, nhưng cần thử nghiệm cả ba thuật toán để chọn mô hình tốt nhất cho hệ thống dự đoán bệnh tim.

Trích dẫn nguồn

1. Cochran, W. G. (1977). *Sampling Techniques* (3rd ed.). New York: John Wiley & Sons.

2. Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression* (2nd ed.). New York: Wiley.

3. Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern Epidemiology* (3rd ed.). Philadelphia: Lippincott Williams & Wilkins.

4. World Health Organization (WHO). (2013). *Global Action Plan for the Prevention and Control of Noncommunicable Diseases 2013-2020*. Geneva: WHO.

5. World Medical Association (WMA). (2013). *Declaration of Helsinki – Ethical Principles for Medical Research Involving Human Subjects*.

6. Dua, D., & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12, 2825-2830

8. Tukey, J. W. (1977). Exploratory data analysis. Reading, Mass.; VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media

9. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media

10. Các bài báo so sánh thuật toán, chẳng hạn "Heart Disease Prediction Using Machine Learning Algorithms"

11. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer

12. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media

13. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238

14. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. "O'Reilly Media, Inc