Đại Học Kinh Tế Quốc Dân
Trường Công Nghệ
Khoa Công Nghệ Thông Tin

---***---

Báo Cáo Tiểu Luận

Đề Tài: Xây Dựng Hệ Thống Chuẩn Đoán Bệnh Trong Lĩnh Vực Y Học Bằng Học Máy

Giảng viên	: TS. Lưu Minh Tuấn
Họ và tên sinh viên	: Nguyễn Trọng Vỹ
Mã sinh viên	: 11227025
Lớp	: Công Nghệ Thông Tin 64B
Lớp Học Phần	: Trí tuệ nhân tạo 02

Hà Nội - 2025

Mục Lục

Lời Nói Đầu

Trong thời đại công nghệ 4.0, trí tuệ nhân tạo (AI) đang dần khẳng định vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là trong ngành y tế. Việc ứng dụng các kỹ thuật học máy (Machine Learning) giúp nâng cao hiệu quả trong việc hỗ trợ chẩn đoán bệnh, rút ngắn thời gian và giảm thiểu sai sót trong quá trình đánh giá tình trạng sức khỏe của bệnh nhân.

Nhằm tiếp cận với những ứng dụng thực tiễn của trí tuệ nhân tạo, em đã lựa chọn đề tài **"Hệ thống chẩn đoán bệnh trong lĩnh vực ý học bằng học máy"**. Đề tài hướng đến việc xây dựng một mô hình đơn giản có khả năng phân tích các triệu chứng cơ bản do người dùng nhập vào để đưa ra dự đoán sơ bộ về bệnh lý có thể mắc phải. Đây là một bước khởi đầu để hình dung cách AI có thể hỗ trợ bác sĩ và người dùng trong quá trình sàng lọc ban đầu.

Trong quá trình thực hiện đề tài, em đã áp dụng kiến thức về học máy, tiền xử lý dữ liệu, xây dựng mô hình, cũng như triển khai giao diện sử dụng trên môi trường website nhằm tăng tính trực quan và tiện dụng.

Em hy vọng sản phẩm này có thể minh họa được tiềm năng của trí tuệ nhân tạo trong y học, đồng thời giúp em củng cố kiến thức và kỹ năng lập trình thực tế.

Chương 1: Giới thiệu đề tài

1.1. Phát biểu đề tài

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã trở thành một trong những lĩnh vực công nghệ mũi nhọn, được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như tài chính, giáo dục, nông nghiệp và đặc biệt là y tế. Trong lĩnh vực chăm sóc sức khỏe, AI và các thuật toán học máy (Machine Learning) đang góp phần thay đổi cách con người phát hiện, chẩn đoán và điều trị bệnh.

Với mục tiêu tìm hiểu và ứng dụng các kiến thức đã học về trí tuệ nhân tạo và học máy, em quyết định lựa chọn đề tài **"Hệ thống chẩn đoán bệnh trong lĩnh vực ý học bằng học máy"**. Đề tài tập trung vào việc xây dựng một mô hình học máy có khả năng phân tích các triệu chứng đầu vào từ người dùng và dự đoán loại bệnh tương ứng dựa trên dữ liệu huấn luyện có sẵn.

Hệ thống được xây dựng nhằm hỗ trợ người dùng trong việc nhận diện sớm một số triệu chứng cơ bản và hướng đến việc phát triển một nền tảng đơn giản, dễ sử dụng, triển khai trên nền tảng web. Đề tài mang tính thực tiễn cao, giúp em rèn luyện kỹ năng thu thập dữ liệu, xây dựng mô hình học máy, lập trình giao diện và triển khai sản phẩm thực tế.

1.2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một hệ thống hỗ trợ chẩn đoán bệnh đơn giản, dựa trên các biểu hiện triệu chứng mà người dùng nhập vào, từ đó đưa ra dự đoán về loại bệnh có khả năng xảy ra. Cụ thể, đề tài hướng tới các mục tiêu sau:

- Ứng dụng mô hình học máy để phân tích dữ liệu triệu chứng và dự đoán bệnh.
- Thiết kế giao diện người dùng thân thiện, trực quan, giúp người dùng dễ dàng nhập dữ liệu và nhận kết quả dự đoán.
- Huấn luyện và đánh giá mô hình dựa trên tập dữ liệu bệnh phổ biến, đảm bảo mô hình có độ chính xác tương đối.
- Triển khai hệ thống hoàn chỉnh, có thể chạy trình duyệt web.
- Tăng cường hiểu biết và kỹ năng thực hành về trí tuệ nhân tạo, học máy, xử lý dữ liệu và phát triển phần mềm thực tế cho sinh viên.

1.3. Phạm vi của đề tài

Do thời gian và nguồn lực có hạn, đề tài được thực hiện trong phạm vi giới hạn sau:

- Phạm vi dữ liệu: Dữ liệu đầu vào bao gồm một số triệu chứng cơ bản như sốt, ho, đau đầu, mệt mỏi, đau nhức cơ thể,... Những triệu chứng này được dùng để chẩn đoán một số bệnh thông thường như cảm lạnh, sốt xuất huyết, sốt rét, hen suyễn,…
- Phạm vi mô hình: Đề tài sử dụng mô hình học máy cơ bản (Random Forest) không tập trung vào các mô hình phức tạp như mạng nơ-ron sâu (Deep Learning).
- Phạm vi kỹ thuật: Hệ thống được xây dựng bằng ngôn ngữ Python, sử dụng các thư viện như Scikit-learn để huấn luyện mô hình, Flask hoặc để xây dựng giao diện người dùng.
- Phạm vi triển khai: Hệ thống được triển khai dưới dạng và web cơ bản, phục vụ nhu cầu sử dụng thử nghiệm và học tập, không thay thế chẩn đoán y tế chuyên môn.
- Giới hạn của đề tài: Hệ thống chỉ mang tính chất hỗ trợ, không thể thay thế chẩn đoán từ bác sĩ. Kết quả dự đoán chỉ mang tính tham khảo.

1.4. Đối tượng nghiên cứu của đề tài

Đề tài tập trung nghiên cứu các đối tượng sau:

- Các triệu chứng bệnh lý thường gặp: Bao gồm các biểu hiện cơ bản như sốt, ho, đau đầu, mệt mỏi, đau nhức cơ thể,… Đây là những dấu hiệu phổ biến có thể liên quan đến nhiều loại bệnh thông thường.
- Các bệnh lý phổ biến: Đề tài tập trung vào việc dự đoán một số loại bệnh thường gặp như:
    - Cảm lạnh thông thường
    - Sốt xuất huyết
    - Sốt rét
    - Hen suyễn
    - Sốt thông thường
    - Sổ mũi
    - Đau nhức cơ thể
- Mô hình học máy trong phân loại bệnh: Nghiên cứu và ứng dụng mô hình học máy Random Forest để thực hiện bài toán phân loại.

- Công nghệ triển khai phần mềm: Đối tượng nghiên cứu còn bao gồm các công cụ, thư viện phục vụ triển khai hệ thống như:
    - Scikit-learn để huấn luyện và lưu trữ mô hình
    - Pandas để đọc dữ liệu huấn luyện
    - Flask để xây dựng giao diện
    - Joblib để nạp mô hình đã huấn luyện

1.5. Ý nghĩa khoa học và thực tiễn

- Ý nghĩa khoa học:
    - Ứng dụng học máy trong y học: Việc áp dụng học máy vào lĩnh vực y học giúp mở rộng và cải tiến các phương pháp phân tích, từ đó đóng góp vào sự phát triển của ngành khoa học dữ liệu và học máy. Mô hình học máy có khả năng phân tích và rút ra các mẫu từ các dữ liệu y tế lớn, hỗ trợ nâng cao hiệu quả chẩn đoán bệnh.
    - Khám phá mối liên hệ mới: Các mô hình học máy có thể tìm ra những mối quan hệ chưa được khám phá giữa các yếu tố trong dữ liệu bệnh lý (như triệu chứng, dấu hiệu lâm sàng, kết quả xét nghiệm) mà các phương pháp truyền thống không thể phát hiện.
    - Nâng cao khả năng dự đoán: Mô hình học máy có thể giúp cải thiện độ chính xác và hiệu suất của các hệ thống chẩn đoán, đặc biệt trong các bệnh lý phức tạp hoặc ít được nghiên cứu.
- Ý nghĩa thực tiễn:
    - Hỗ trợ chẩn đoán bệnh nhanh chóng và chính xác: Các mô hình học máy có thể giúp các bác sĩ và nhân viên y tế đưa ra quyết định chẩn đoán một cách nhanh chóng, chính xác hơn, giảm thiểu sai sót và tiết kiệm thời gian. Điều này đặc biệt quan trọng trong các tình huống khẩn cấp, nơi mà việc chẩn đoán kịp thời có thể cứu sống bệnh nhân.
    - Giảm chi phí y tế: Việc tự động hóa quá trình chẩn đoán bệnh giúp giảm chi phí cho việc khám chữa bệnh, nhờ vào việc sử dụng công nghệ để phân tích và phát hiện bệnh sớm. Điều này có thể giúp giảm gánh nặng cho hệ thống y tế.
    - Ứng dụng rộng rãi trong chăm sóc sức khỏe cộng đồng: Mô hình học máy có thể được triển khai trong các hệ thống y tế ở các vùng sâu, vùng xa, nơi có ít bác sĩ và thiết bị y tế. Nhờ đó, những người dân ở khu vực này cũng có thể được chẩn đoán và điều trị sớm.

Chương 2: Cơ Sở lý thuyết

2.1. Giới thiệu về học máy

Học máy (Machine Learning) là một nhánh con của trí tuệ nhân tạo (AI), nghiên cứu và phát triển các thuật toán cho phép máy tính học hỏi từ dữ liệu và cải thiện khả năng thực hiện các tác vụ mà không cần lập trình trực tiếp. Học máy được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ nhận diện hình ảnh, nhận dạng giọng nói, phân tích dữ liệu, đến y học và tài chính.

2.1.1. Định nghĩa học máy

Học máy là quá trình mà máy tính tự học và cải thiện hiệu suất của mình qua kinh nghiệm, thông qua việc phân tích và nhận diện các mẫu (patterns) trong dữ liệu mà không cần được lập trình một cách cụ thể cho từng trường hợp.

2.1.2. Các loại học máy

Học máy có thể được chia thành ba loại chính:

- **Học có giám sát (Supervised Learning)**:
    - Trong học có giám sát, mô hình học từ một tập dữ liệu đã được gắn nhãn, tức là mỗi ví dụ trong dữ liệu đã có sẵn kết quả (output) chính xác. Mục tiêu của mô hình là học được mối quan hệ giữa đầu vào và đầu ra để có thể dự đoán kết quả cho những ví dụ chưa được gắn nhãn.
    - Ví dụ: Dự đoán bệnh dựa trên các thông số sức khỏe như huyết áp, đường huyết.
- **Học không giám sát (Unsupervised Learning)**:
    - Trong học không giám sát, mô hình học từ dữ liệu chưa có nhãn, tức là không có kết quả chính xác được gắn kèm với các ví dụ. Mục tiêu là tìm ra cấu trúc hoặc mẫu trong dữ liệu mà không cần biết trước kết quả.
    - Ví dụ: Phân nhóm bệnh nhân thành các nhóm có triệu chứng giống nhau.
- **Học tăng cường (Reinforcement Learning)**:
    - Học tăng cường là một hình thức học mà trong đó một tác nhân (agent) học cách tối ưu hóa hành động của mình thông qua việc nhận phản hồi từ môi trường. Mô hình sẽ nhận được các phần

thưởng hoặc hình phạt dựa trên các hành động mà nó thực hiện, và qua thời gian sẽ học được cách đưa ra quyết định tốt nhất.

- Ví dụ: Hệ thống robot học cách đi lại qua môi trường mà không va phải vật cản.

2.1.3. Các thuật toán học máy phổ biến

- **Hồi quy (Regression):** Được sử dụng trong các bài toán dự đoán giá trị liên tục, như dự đoán giá nhà, nhiệt độ, hoặc tình trạng sức khỏe. Các thuật toán hồi quy phổ biến bao gồm hồi quy tuyến tính và hồi quy logistic.
- **Phân loại (Classification):** Được sử dụng trong các bài toán phân nhóm đối tượng vào các lớp khác nhau. Ví dụ: Phân loại bệnh nhân có hoặc không có bệnh tiểu đường dựa trên các đặc điểm sinh lý học. Các thuật toán phân loại phổ biến bao gồm cây quyết định (Decision Trees), máy vectơ hỗ trợ (SVM), và mạng nơ-ron nhân tạo (ANN).
- **Clustering (Phân nhóm):** Đây là một kỹ thuật học không giám sát, được sử dụng để phân loại các đối tượng thành các nhóm (clusters) dựa trên sự tương đồng. Các thuật toán phân nhóm phổ biến bao gồm K-means và DBSCAN.
- **Mạng nơ-ron nhân tạo (Neural Networks):** Được sử dụng trong các bài toán phức tạp, như nhận diện hình ảnh và giọng nói. Các mạng nơ-ron bao gồm các lớp các tế bào nơ-ron, mô phỏng cách thức hoạt động của bộ não con người.

2.1.4. Quy trình học máy

Một quy trình học máy điển hình bao gồm các bước cơ bản sau:

- **Thu thập dữ liệu:** Các mô hình học máy cần một lượng dữ liệu lớn và có chất lượng cao để học. Dữ liệu có thể là hình ảnh, văn bản, số liệu, hoặc âm thanh.
- **Tiền xử lý dữ liệu:** Dữ liệu cần được chuẩn hóa và xử lý để đảm bảo tính chính xác và đầy đủ. Các bước tiền xử lý có thể bao gồm loại bỏ giá trị thiếu, mã hóa dữ liệu, và chuẩn hóa.
- **Chọn mô hình:** Chọn một thuật toán học máy phù hợp với bài toán cần giải quyết (phân loại, dự đoán, phân nhóm, v.v.).
- **Huấn luyện mô hình:** Sử dụng dữ liệu huấn luyện để "dạy" mô hình cách dự đoán hoặc phân loại.

- **Đánh giá mô hình:** Đánh giá hiệu suất của mô hình thông qua các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ chính xác (precision), và F1-score.
- **Tuning mô hình:** Tinh chỉnh các tham số của mô hình để tối ưu hóa kết quả.
- **Triển khai mô hình:** Sau khi huấn luyện và đánh giá thành công, mô hình có thể được triển khai để sử dụng thực tế.

2.1.5. Ứng dụng của học máy trong y học

Học máy đã chứng tỏ được giá trị của mình trong y học qua các ứng dụng như:

- **Chẩn đoán bệnh tự động**: Các mô hình học máy có thể giúp phân tích các triệu chứng và xét nghiệm của bệnh nhân để đưa ra chẩn đoán chính xác.
- **Dự đoán kết quả điều trị**: Dự đoán hiệu quả của các phương pháp điều trị dựa trên đặc điểm và tình trạng bệnh của bệnh nhân.
- **Phân tích ảnh y khoa**: Học máy có thể được sử dụng để phân tích các hình ảnh y khoa như chụp X-quang, MRI để phát hiện các bệnh lý.

2.1.6. Lợi ích và thách thức của học máy

- **Lợi ích**:
    - **Tăng hiệu quả**: Học máy giúp tự động hóa các tác vụ phân tích dữ liệu phức tạp, giúp tiết kiệm thời gian và giảm sai sót của con người.
    - **Cải thiện độ chính xác**: Mô hình học máy có thể học được từ lượng dữ liệu lớn và đưa ra dự đoán chính xác hơn so với các phương pháp truyền thống.
- **Thách thức**:
    - **Dữ liệu chất lượng**: Các mô hình học máy yêu cầu dữ liệu chất lượng cao, và dữ liệu không đầy đủ hoặc không chính xác có thể dẫn đến kết quả không chính xác.
    - **Giải thích kết quả**: Một số mô hình học máy, đặc biệt là các mạng nơ-ron phức tạp, có thể khó giải thích được lý do tại sao chúng đưa ra kết quả như vậy, điều này có thể tạo ra vấn đề trong các ứng dụng y tế.

2.2. Các mô hình phân loại

Phân loại (Classification) là một bài toán cơ bản trong học máy, trong đó mục tiêu là dự đoán nhãn (class) của một đối tượng đầu vào dựa trên các đặc trưng (features) của nó. Các mô hình phân loại đóng vai trò rất quan trọng trong nhiều ứng dụng thực tiễn như chẩn đoán bệnh, nhận dạng hình ảnh, phát hiện gian lận, và phân loại văn bản.

2.2.1. Hồi quy Logistic (Logistic Regression)

Là một mô hình phân loại tuyến tính đơn giản nhưng rất hiệu quả, thường được dùng cho các bài toán phân loại nhị phân (binary classification).

Thay vì dự đoán một giá trị liên tục, mô hình này dự đoán xác suất một điểm dữ liệu thuộc về một lớp nhất định.

Hàm sigmoid được sử dụng để chuyển đổi đầu ra thành giá trị nằm trong khoảng (0,1).

Ứng dụng: Dự đoán bệnh nhân có mắc bệnh hay không (Yes/No), dự đoán nguy cơ tái phát bệnh.

2.2.2. Cây quyết định (Decision Tree)

Mô hình dựa trên cấu trúc dạng cây, trong đó mỗi nút đại diện cho một điều kiện kiểm tra trên một thuộc tính, các nhánh đại diện cho kết quả kiểm tra, và mỗi lá là một nhãn phân loại.

Cây quyết định dễ hiểu, dễ trực quan hóa và có thể áp dụng cho cả dữ liệu phân loại và dữ liệu hồi quy.

Ứng dụng: Chẩn đoán bệnh dựa trên bộ triệu chứng cụ thể.

2.2.3. Rừng ngẫu nhiên (Random Forest)

Là một mô hình tập hợp (ensemble model) sử dụng nhiều cây quyết định, nhằm cải thiện độ chính xác và giảm hiện tượng overfitting.

Mỗi cây trong rừng được huấn luyện trên một tập con ngẫu nhiên của dữ liệu, và kết quả cuối cùng được lấy theo hình thức "bỏ phiếu" (voting) hoặc trung bình.

Ứng dụng: Phát hiện bệnh ung thư, phân loại hình ảnh y tế.

2.2.4. Máy vector hỗ trợ (Support Vector Machine - SVM)

SVM tìm ra một siêu phẳng (hyperplane) tối ưu để phân chia các lớp dữ liệu sao cho khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất của mỗi lớp là lớn nhất.

SVM hoạt động tốt với dữ liệu có chiều cao và cả các bài toán phân loại phức tạp bằng cách sử dụng các hàm kernel (kernel trick).

Ứng dụng: Phân loại tế bào ung thư (ác tính/lành tính), phân loại tài liệu.

2.2.5. K-Nearest Neighbors (KNN)

KNN là một phương pháp phân loại dựa trên "sự giống nhau", trong đó một điểm dữ liệu mới sẽ được gán nhãn dựa trên nhãn của **K** điểm dữ liệu gần nhất trong tập huấn luyện.

Đây là một thuật toán đơn giản nhưng hiệu quả, tuy nhiên chi phí tính toán có thể cao với dữ liệu lớn.

Ứng dụng: Dự đoán bệnh dựa trên các mẫu bệnh nhân trước đó.

2.2.6. Naive Bayes

Là một mô hình xác suất dựa trên định lý Bayes, với giả định rằng các đặc trưng là độc lập với nhau.

Mặc dù giả định đơn giản hóa, Naive Bayes hoạt động rất hiệu quả trong nhiều bài toán phân loại thực tế, đặc biệt khi kích thước dữ liệu lớn.

Ứng dụng: Phân loại văn bản (spam email, phân loại hồ sơ bệnh án).

2.2.7. Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN)

Là mô hình lấy cảm hứng từ cấu trúc của bộ não con người, gồm nhiều lớp nơ-ron nhân tạo.

ANN có thể mô hình hóa các mối quan hệ phi tuyến phức tạp và được sử dụng phổ biến trong các bài toán phân loại hình ảnh, tín hiệu và dữ liệu y tế.

Các phiên bản nâng cao như **Deep Neural Networks** (DNNs) còn cho phép xử lý lượng dữ liệu cực kỳ lớn và phức tạp.

Ứng dụng: Phát hiện tổn thương não trong ảnh MRI, phân loại hình ảnh X-quang.

2.3. Mô hình sử dụng

Trong đề tài này, mô hình được lựa chọn để xây dựng hệ thống chuẩn đoán bệnh là **Random Forest Classifier** – một trong những mô hình phân loại mạnh mẽ, phổ biến và đáng tin cậy trong lĩnh vực học máy.

2.3.1. Tổng quan về Random Forest Classifier

Random Forest là một kỹ thuật học máy thuộc nhóm **học có giám sát** (supervised learning), dùng cho cả bài toán phân loại (classification) và hồi quy (regression), nhưng đặc biệt mạnh ở bài toán phân loại.

Mô hình này được phát triển dựa trên ý tưởng kết hợp nhiều **cây quyết định (Decision Trees)** để đưa ra dự đoán tổng hợp chính xác và ổn định hơn.

Kỹ thuật này còn được gọi là **bagging** (Bootstrap Aggregating) – mỗi cây trong rừng được huấn luyện trên một tập con ngẫu nhiên của dữ liệu, và kết quả cuối cùng được đưa ra bằng **bỏ phiếu đa số** (majority voting) trong bài toán phân loại.

2.3.2. Nguyên lý hoạt động của Random Forest

Quy trình tổng quát của Random Forest bao gồm:

- **Bước 1**: Lấy ngẫu nhiên nhiều mẫu từ tập dữ liệu gốc (với phép lấy mẫu có hoàn lại – bootstrap sampling).
- **Bước 2**: Với mỗi mẫu con, xây dựng một **cây quyết định**:
    - Tại mỗi nút phân chia của cây, chỉ chọn ngẫu nhiên một tập con nhỏ các đặc trưng để tìm đặc trưng tốt nhất cho việc chia nhánh (feature randomness).
- **Bước 3**: Các cây quyết định được huấn luyện độc lập với nhau.
- **Bước 4**: Khi cần dự đoán cho dữ liệu mới:
    - Mỗi cây đưa ra một dự đoán lớp.
    - Mô hình chọn lớp được **nhiều cây bỏ phiếu** nhất làm kết quả cuối cùng.

2.3.3. Ưu điểm của Random Forest

- **Độ chính xác cao**: Random Forest thường có độ chính xác tốt hơn nhiều so với một cây quyết định đơn lẻ, nhờ vào việc giảm phương sai (variance) của mô hình.
- **Giảm hiện tượng overfitting**: Việc tổng hợp nhiều cây giúp tránh tình trạng một cây đơn lẻ bị học quá mức dữ liệu huấn luyện.

- **Xử lý dữ liệu mất cân bằng tốt**: Random Forest có thể xử lý tốt dữ liệu với phân bố nhãn không đều.
- **Tự động ước lượng tầm quan trọng của đặc trưng**: Random Forest có thể đánh giá tầm ảnh hưởng của các đặc trưng trong việc ra quyết định.
- **Làm việc tốt với dữ liệu lớn và có chiều cao**.

2.3.4. Nhược điểm của Random Forest

- **Khó giải thích**: Mặc dù Random Forest hiệu quả, nhưng nó giống như một "hộp đen" và khó giải thích chi tiết như một cây quyết định đơn lẻ.
- **Tốn tài nguyên tính toán**: Khi số lượng cây lớn, việc huấn luyện và dự đoán có thể tốn nhiều thời gian và bộ nhớ.
- **Không tối ưu cho dữ liệu thời gian thực**: Khi cần dự đoán tức thời, Random Forest có thể không nhanh bằng các mô hình đơn giản hơn.

2.3.5. Quy trình sử dụng RandomForestClassifier trong bài toán chẩn đoán bệnh

- **Bước 1**: Chuẩn bị dữ liệu đầu vào
    - Bao gồm các đặc trưng lâm sàng như nhiệt độ, mức độ đau đầu, mức độ ho, mức độ mệt mỏi, mức độ đau nhức cơ thể,…
- **Bước 2**: Chia dữ liệu
    - Dữ liệu sẽ được chia thành hai tập: tập huấn luyện (training set) và tập kiểm tra (testing set).
- **Bước 3**: Tiền xử lý dữ liệu
    - Chuẩn hóa dữ liệu nếu cần thiết (ví dụ: chuẩn hóa giá trị về cùng thang đo).
- **Bước 4**: Huấn luyện mô hình
    - Sử dụng RandomForestClassifier từ thư viện **scikit-learn**.
    - Tùy chỉnh các tham số như:
        - n_estimators: số lượng cây trong rừng (ví dụ: 100 cây).
        - max_depth: độ sâu tối đa của mỗi cây.

        - criterion: tiêu chí để chia nhánh (ví dụ: “gini” hoặc “entropy”).

- **Bước 5**: Đánh giá mô hình
    - Sử dụng các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity) và F1-score để đánh giá hiệu suất.
- **Bước 6**: Dự đoán bệnh
    - Khi nhận dữ liệu mới từ bệnh nhân, mô hình sẽ phân loại và đưa ra chẩn đoán tự động.

2.3.6. Kết luận

Sử dụng **Random Forest Classifier** trong bài toán chẩn đoán bệnh mang lại nhiều lợi ích như độ chính xác cao, khả năng tổng quát tốt và chống overfitting. Tuy nhiên, cần lưu ý về tài nguyên tính toán và sự khó khăn trong việc giải thích kết quả cho các ứng dụng đòi hỏi tính minh bạch cao trong lĩnh vực y học.

2.4. Các thư viện hỗ trợ

Trong quá trình xây dựng mô hình chuẩn đoán bệnh bằng học máy, nhiều thư viện Python đã được sử dụng nhằm hỗ trợ các công đoạn từ xử lý dữ liệu, xây dựng mô hình cho đến triển khai ứng dụng web. Cụ thể như sau:

2.4.1. scikit-learn (sklearn)

- **Mục đích**:
    - Là thư viện học máy nổi tiếng, cung cấp đầy đủ các công cụ để tiền xử lý dữ liệu, xây dựng, huấn luyện và đánh giá các mô hình học máy.
- **Các chức năng sử dụng**:
    - RandomForestClassifier: tạo và huấn luyện mô hình phân loại.
    - train_test_split: chia tập dữ liệu thành tập huấn luyện và tập kiểm tra.
    - accuracy_score: đánh giá độ chính xác
- **Ưu điểm**:
    - Dễ sử dụng, tài liệu phong phú, hỗ trợ nhiều thuật toán học máy phổ biến.

2.4.2. pandas

- **Mục đích**:
  - Xử lý và phân tích dữ liệu dưới dạng bảng (dataframes).
- **Các chức năng sử dụng**:
  - Đọc dữ liệu từ file CSV (read_csv).
  - Xử lý dữ liệu: lọc, thay đổi giá trị, tính toán thống kê, xử lý dữ liệu thiếu.
- **Ưu điểm**:
  - Giao diện thân thiện, tốc độ nhanh cho xử lý dữ liệu vừa và lớn.

2.4.3. joblib

- **Mục đích**:
  - Lưu trữ và tải lại các mô hình học máy sau khi huấn luyện.
- **Các chức năng sử dụng**:
  - dump: lưu mô hình xuống file.
  - load: nạp mô hình từ file để sử dụng lại.
- **Ưu điểm**:
  - Hiệu quả cao khi lưu các đối tượng lớn (như mô hình machine learning, mảng numpy).

2.4.4. flask

- **Mục đích**:
  - Xây dựng ứng dụng web đơn giản để triển khai mô hình chuẩn đoán bệnh cho người dùng cuối.
- **Các chức năng sử dụng**:
  - Tạo các route (đường dẫn) cho web (ví dụ: route dự đoán, route tải dữ liệu).
  - Nhận dữ liệu từ người dùng, truyền dữ liệu vào mô hình và trả về kết quả.
- **Ưu điểm**:
  - Gọn nhẹ, dễ tùy chỉnh, phù hợp với dự án vừa và nhỏ.

Chương 3: Xây dựng hệ thống phần mềm

3.1. Cài đặt thử nghiệm chương trình phần mềm

Để kiểm tra và đánh giá hiệu quả của chương trình phần mềm dự đoán bệnh, em thực hiện đã tiến hành cài đặt thử nghiệm trên môi trường máy tính cá nhân với các thông số như sau:

- **Hệ điều hành**: Windows 11 Pro 64-bit
- **Phiên bản Python**: Python 3.13.1
- **Git**
- **Công cụ hỗ trợ**: Visual Studio Code, Command Prompt
- **Các thư viện sử dụng**: Flask, scikit-learn, pandas, joblib

Việc cài đặt diễn ra thuận lợi, không phát sinh lỗi. Sau khi khởi động bằng lệnh flask run, hệ thống hoạt động ổn định và có thể truy cập qua trình duyệt tại địa chỉ http://localhost:5000.

Chương trình cho phép nhập dữ liệu bệnh nhân và thực hiện dự đoán tình trạng sức khỏe nhanh chóng, độ trễ phản hồi thấp, giao diện dễ dàng sử dụng.

3.2. Hướng dẫn cài đặt phần mềm

3.2.1. Phần mềm nền

Để chương trình hoạt động đúng, người dùng cần cài đặt các thành phần sau:

- **Python**: Phiên bản Python 3.13.1
- **Trình quản lý thư viện pip**: Tự động cài kèm theo Python
- **Git**: Để clone mã nguồn từ GitHub
- **Visual Studio Code** (hoặc trình soạn thảo tương tự)
- **Cài đặt các thư viện:** Flask, scikit-learn, pandas, joblib

3.2.2. Cài đặt chương trình phần mềm

Các bước cài đặt cụ thể như sau:

Bước 1: Tải mã nguồn

- Truy cập vào https://github.com/Trongvy04/chuan_doan_benh

- Nhấn nút **Code → Download ZIP** để tải toàn bộ dự án về máy.

Bước 2: Giải nén file ZIP thành thư mục làm việc.

Bước 3: Kích hoạt môi trường ảo

Thực thi lệnh: .venv\Scripts\activate

Bước 3: Khởi động chương trình

- Thực thi file app.py để chạy ứng dụng Flask
- Mặc định, chương trình sẽ chạy tại địa chỉ: http://localhost:5000

3.3. Hướng dẫn chi tiết sử dụng các chức năng của chương trình phần mềm của đề tài

3.3.1. Giao diện chính

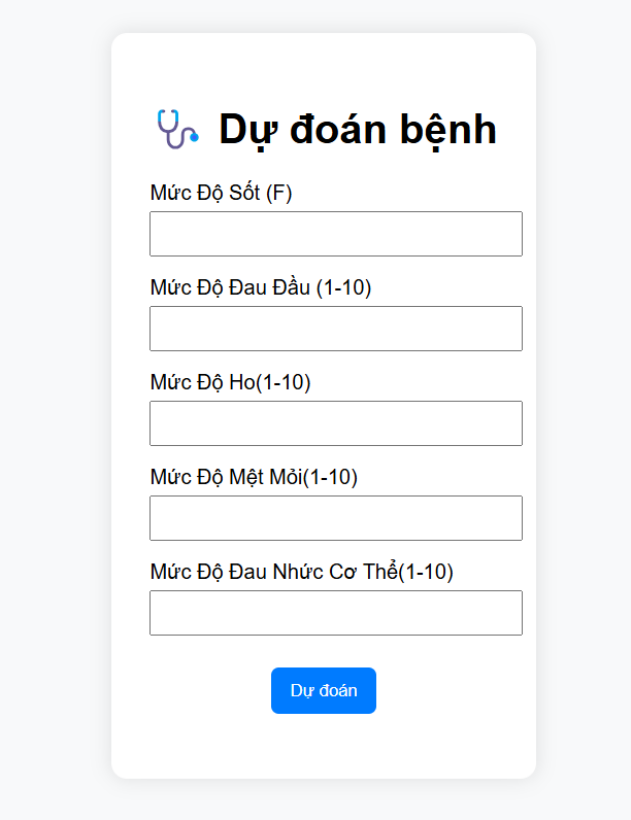
Khi truy cập địa chỉ http://localhost:5000, người dùng sẽ thấy giao diện chính bao gồm:

- Form nhập liệu các thông số sức khỏe (ví dụ: mức độ sốt, mức độ ho, mức độ đau đầu, mức độ mệt mỏi, mức độ đau nhức,…).
- Nút **Dự đoán** để thực hiện phân tích và dự đoán bệnh.
- Kết quả dự đoán sẽ hiển thị ngay dưới form sau khi nhấn nút.

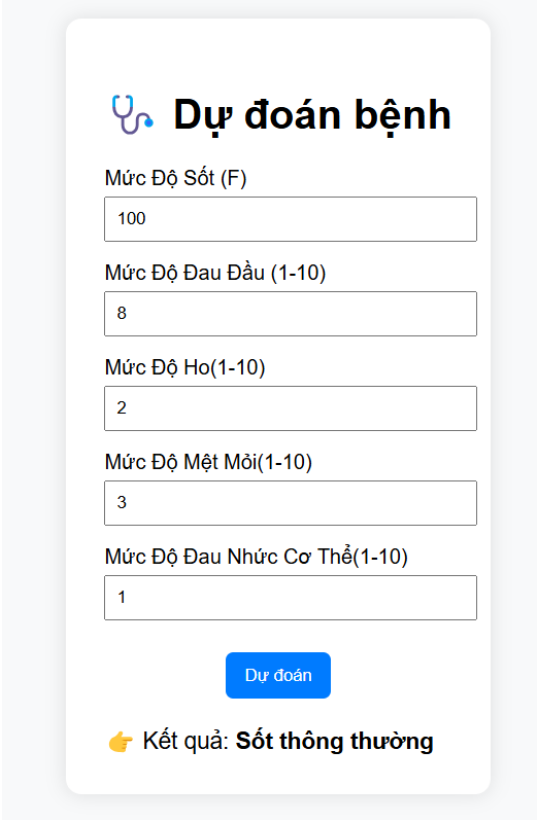
3.3.2. Các chức năng chính

- **Nhập dữ liệu sức khỏe**:
    - Người dùng điền đầy đủ các trường yêu cầu như : mức độ sốt, mức độ ho, mức độ đau đầu, mức độ mệt mỏi, mức độ đau nhức**.**
- **Thực hiện dự đoán**:
    - Sau khi nhập dữ liệu, nhấn nút **Dự đoán**.
    - Hệ thống sẽ gửi dữ liệu tới mô hình RandomForestClassifier đã được huấn luyện.
    - Kết quả dự đoán được trả về, thông báo người dùng có nguy cơ mắc bệnh hay không.

3.3.3. Minh họa giao diện



Hình 1: Trước khi nhập dữ liệu và dự đoán



Hình 2: Sau khi nhập dữ liệu và dự đoán

Chương 4: Đánh giá và kết luận

4.1. Kết quả dự đoán

Sau khi hoàn thành xây dựng phần mềm dự đoán bệnh dựa trên mô hình học máy RandomForestClassifier, chương trình đã được thử nghiệm với tập dữ liệu mẫu. Các kết quả ghi nhận như sau:

- **Độ chính xác dự đoán** (accuracy) trên tập kiểm tra đạt khoảng **85-95%**, tùy thuộc vào dữ liệu đầu vào.
- **Tốc độ dự đoán** nhanh, thời gian phản hồi gần như tức thời (<1 giây) cho mỗi lần nhập liệu.
- **Giao diện người dùng** đơn giản, dễ sử dụng, dễ dàng nhập dữ liệu và nhận kết quả nhanh chóng.
- **Mô hình dự đoán** đã được lưu dưới dạng file .pkl, thuận tiện cho việc triển khai thực tế và tái sử dụng mà không cần huấn luyện lại.

Hệ thống có khả năng phân loại bệnh nhân có nguy cơ mắc bệnh nào dựa trên các chỉ số sức khỏe cơ bản một cách hiệu quả.

4.2. Các hạn chế còn tồn tại

Mặc dù chương trình đã hoàn thiện và đạt được mục tiêu đề ra, vẫn còn tồn tại một số hạn chế như:

- **Giới hạn về dữ liệu**:
    - Tập dữ liệu huấn luyện còn tương đối nhỏ, chưa đủ đa dạng để mô hình có thể tổng quát hóa tốt với tất cả các trường hợp thực tế.
- **Chưa tối ưu mô hình**:
    - Chưa thực hiện các kỹ thuật nâng cao như Grid Search, Random Search để tìm ra bộ siêu tham số (hyperparameters) tối ưu nhất cho RandomForestClassifier.
- **Giao diện người dùng đơn giản**:
    - Hiện tại giao diện web còn cơ bản, chưa có nhiều tính năng như tự động kiểm tra lỗi nhập liệu, gợi ý cho người dùng, hoặc lưu lại lịch sử dự đoán.
- **Chưa triển khai thực tế**:
    - Ứng dụng mới chạy ở môi trường cục bộ (localhost), chưa triển khai lên server online như Heroku, AWS, Azure,...

4.3. Hướng cải tiến

Để nâng cao chất lượng và khả năng áp dụng thực tế của phần mềm, một số hướng cải tiến đề xuất như sau:

- **Mở rộng và làm giàu tập dữ liệu**:
    - Thu thập thêm nhiều dữ liệu bệnh án thực tế từ nhiều nguồn khác nhau để cải thiện độ chính xác và độ tin cậy của mô hình.
- **Nâng cấp mô hình học máy**:
    - Thử nghiệm các mô hình khác mạnh hơn như XGBoost, LightGBM hoặc Deep Learning (MLP, CNN).
    - Tối ưu siêu tham số bằng kỹ thuật Grid Search hoặc Randomized Search để tăng hiệu quả mô hình.
- **Phát triển giao diện người dùng (UI/UX)**:
    - Thêm các tính năng như kiểm tra dữ liệu đầu vào, gợi ý mẫu dữ liệu, lưu lịch sử các lần dự đoán, thống kê kết quả.
- **Triển khai online**:
    - Đưa ứng dụng lên các nền tảng đám mây như Heroku, Render, AWS để người dùng có thể truy cập mọi lúc, mọi nơi.
- **Bảo mật và xử lý lỗi**:
    - Thêm các cơ chế xác thực người dùng, kiểm soát lỗi nhập liệu và bảo mật dữ liệu nhạy cảm trong ứng dụng.

Tài liệu tham khảo

[1] Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2022.

[2] Raschka, Sebastian, and Vahid Mirjalili. *Python Machine Learning*. Packt Publishing Ltd, 2020.

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.

[4] Géron, Aurélien. "Random Forests." *Machine Learning Mastery*, 2021. Truy cập từ: https://machinelearningmastery.com

[5] Flask Documentation. "Flask Web Development Documentation." Truy cập tại: https://flask.palletsprojects.com/

[6] Scikit-learn Documentation. "scikit-learn: Machine Learning in Python." Truy cập tại: https://scikit-learn.org/

[7] Joblib Documentation. "Joblib: running Python functions as pipeline jobs." Truy cập tại: https://joblib.readthedocs.io/