**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****-----o0o----

**ĐỒ ÁN MÔN HỌC**

**KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: DỰ ĐOÁN BỆNH DỰA TRÊN BIỂU HIỆN BAN ĐẦU**

**GVHD: Trần Như Ý**

**Lớp: 13DHTH04**

***Thành viên nhóm:***

**1. Lại Phước Thịnh - 2001224956**

**2. Lê Trần Ngọc Yến - 2001226134**

**3. Phạm Hồ Thúy Vy - 2001225958**

***TP. Hồ Chí Minh, tháng 05 năm 2025***

**LỊCH LÀM VIỆC NHÓM HÀNG TUẦN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Thời gian** | **Nội dung công việc nhóm** | **Hình thức họp** |
| Tuần 1 | Thứ 3 (19h00 - 20h30) | Phân chia đề tài, thảo luận hướng tiếp cận ban đầu | Họp online (Google Meet) |
| Tuần 2 | Thứ 5 (19h00 - 21h00) | Thu thập và chuẩn hóa dữ liệu đầu vào | Họp online (Google Meet) |
| Tuần 3 | Chủ nhật (14h00 - 16h00) | Xây dựng mô hình dự đoán (Decision Tree, Random Forest) | Họp online (Google Meet) |
| Tuần 4 | Thứ 4 (20h00 - 21h30) | Thử nghiệm phân cụm KMeans và luật kết hợp Apriori/FP-Growth | Họp online Zalo |
| Tuần 5 | Thứ 6 (17h30 - 21h00) | Xây dựng giao diện Streamlit, kết nối mô hình với giao diện | Làm việc trực tiếp |
| Tuần 6 | Chủ nhật (9h00 - 11h00) | Kiểm thử hệ thống, đánh giá mô hình, chuẩn bị báo cáo | Online + Làm báo cáo |
| Tuần 7 |  | Hoàn thiện báo cáo, trình bày slide và chuẩn bị báo cáo |  |

**BẢNG ĐÁNH GIÁ THỰC HIỆN CÔNG VIỆC NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Đánh giá mức độ hoàn thành** |
| 1. | Lại Phước Thịnh | 2001224956 | 100% |
| 2. | Lê Trần Ngọc Yến | 2001226134 | 100% |
| 3. | Phạm Hồ Thúy Vy | 2001225958 | 100% |

**PHÂN CÔNG CÔNG VIỆC TRONG NHÓM**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ & tên** | **Nội dung công việc thực hiện** |
| 1. | Lại Phước Thịnh |  |
| 2. | Lê Trần Ngọc Yến |  |
| 3. | Phạm Hồ Thúy Vy |  |

# LỜI CẢM ƠN

Đề tài nhóm môn Khai phá dữ liệu “Dự đoán bệnh dựa trên biểu hiện ban đầu” được hoàn thành dưới sự hướng dẫn của giảng viên cô Trần Như Ý. Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến cô về sự hướng dẫn tận tâm, kiên nhẫn và luôn động viên chúng em trong suốt thời gian học tập và nghiên cứu thực hiện bài tiểu luận.

Đồng thời chúng em cũng muốn bày tỏ sự biết ơn đối với việc cô đánh giá bài tiểu luận của chúng em với sự chú ý và chi tiết, cho phép chúng em hiểu rõ hơn về những điểm mạnh và yếu của bài làm. Những góp ý của cô đã giúp nhóm em cải thiện nhiều và phát triển kỹ năng viết và nghiên cứu của mình.

Xin trân trọng cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 6](#_Toc28732)

[1.1 Phạm vi đề tài 6](#_Toc18333)

[1.2 Mục tiêu 6](#_Toc26127)

[1.3 Sự cần thiết và lý do chọn đề tài 6](#_Toc11238)

[CHƯƠNG 2: PHÂN TÍCH DỮ LIỆU 8](#_Toc2576)

[2.1 Thu thập/ xây dựng dữ liệu 8](#_Toc26358)

[2.2 Tăng cường dữ liệu/ phân tích dữ liệu 9](#_Toc5833)

[2.2.1 Tăng cường dữ liệu 9](#_Toc28077)

[2.2.2 Phân tích dữ liệu 10](#_Toc18453)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 12](#_Toc9492)

[3.1 Đề xuất sử dụng thuật toán 12](#_Toc27943)

[3.1.1 Thuật toán học máy 12](#_Toc19629)

[3.1.2 Kỹ thuật tăng cường dữ liệu 12](#_Toc18447)

[3.1.3 Phân cụm và luật kết hợp 12](#_Toc1146)

[3.2 Cách thức giải quyết bài toán 12](#_Toc3634)

[3.2.1 Tiền xử lý dữ liệu 12](#_Toc5141)

[3.2.2 Tăng cường và trích chọn đặc trưng 13](#_Toc14717)

[3.2.3 Xây dựng mô hình 13](#_Toc897)

[3.2.4 Điều chỉnh hiệu suất dự đoán 13](#_Toc6255)

[3.2.5 Phân tích cụm và luật kết hợp 14](#_Toc23464)

[3.3 Giao diện người dùng và ứng dụng web 14](#_Toc21405)

[CHƯƠNG 4: CÀI ĐẶT ỨNG DỤNG BÀI TOÁN 15](#_Toc3600)

[4.1 Môi trường và công cụ phát triển 15](#_Toc6429)

[4.2 Cài đặt giao diện ứng dụng 15](#_Toc14069)

[4.3 Luồng xử lý của ứng dụng 16](#_Toc5603)

[4.4 Một số giao diện chính 17](#_Toc12140)

[CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 23](#_Toc24774)

[5.1 Kết luận 23](#_Toc12294)

[5.2 Định hướng phát triển 23](#_Toc16584)

[CHƯƠNG 6: TÀI LIỆU THAM KHẢO 25](#_Toc22957)

# GIỚI THIỆU

## Phạm vi đề tài

Đồ án tập trung vào việc xây dựng một hệ thống dự đoán bệnh dựa trên các biểu hiện ban đầu của bệnh nhân. Phạm vi nghiên cứu bao gồm: thu thập dữ liệu về triệu chứng, phân tích dữ liệu, xây dựng mô hình dự đoán và triển khai ứng dụng hỗ trợ chẩn đoán sơ bộ. Hệ thống hướng đến việc hỗ trợ bác sĩ và bệnh nhân trong việc phát hiện sớm các bệnh lý phổ biến, tập trung vào các bệnh như cảm cúm, viêm phổi, tiểu đường, và các bệnh liên quan đến hệ hô hấp hoặc tiêu hóa.

## Mục tiêu

Xây dựng mô hình máy học có khả năng dự đoán với độ chính xác cao hơn dựa trên các biểu hiện.

Phát triển một ứng dụng đơn giản, thân thiện với người dùng, cho phép nhập các triệu chứng và nhận kết quả dự đoán tương ứng.

Cung cấp những gợi ý ban đầu hỗ trợ bác sĩ chẩn đoán, đồng thời nâng cao nhận thức của bệnh nhân về tình trạng sức khỏe của mình.

Thiết kế hệ thống có khả năng mở rộng để tích hợp thêm các loại bệnh và triệu chứng trong tương lai.

## Sự cần thiết và lý do chọn đề tài

Trong bối cảnh nhu cầu chăm sóc sức khỏe ngày càng gia tăng, việc phát hiện sớm các bệnh lý dựa trên triệu chứng ban đầu đóng vai trò quan trọng trong việc nâng cao hiệu quả điều trị và giảm chi phí y tế. Theo thống kê của Bộ Y tế, số lượng bệnh nhân mắc các bệnh đường hô hấp đang tăng đáng kể qua từng năm, dẫn đến tình trạng quá tải tại nhiều cơ sở y tế.

Các hệ thống chẩn đoán truyền thống thường yêu cầu bác sĩ trực tiếp thăm khám, điều này gây áp lực lớn lên nguồn nhân lực y tế. Việc ứng dụng các mô hình học máy để hỗ trợ dự đoán bệnh có thể giúp giảm tải cho các cơ sở y tế, đặc biệt tại những khu vực thiếu nhân lực, đồng thời giúp người dân chủ động theo dõi sức khỏe và sớm nhận biết các dấu hiệu bất thường.

Đề tài được lựa chọn vì tính ứng dụng thực tiễn cao, khả năng kết hợp giữa công nghệ và y học, đồng thời phù hợp với xu hướng phát triển trí tuệ nhân tạo trong lĩnh vực chăm sóc sức khỏe. Trong bối cảnh các vấn đề sức khỏe toàn cầu ngày càng phức tạp, việc ứng dụng công nghệ vào hỗ trợ chẩn đoán và điều trị bệnh là hết sức cần thiết.

# PHÂN TÍCH DỮ LIỆU

## Thu thập/ xây dựng dữ liệu

Dữ liệu đầu vào được xây dựng từ hai nguồn chính:

- Tập dữ liệu bệnh *(data\_benh.xlsx)*: Tập dữ liệu này chứa thông tin hồ sơ bệnh nhân bao gồm tuổi, giới tính, các triệu chứng đang gặp phải (dưới dạng nhị phân), tiền sử bệnh, và bệnh hiện tại được chẩn đoán như:

* Tuổi: Giá trị số nguyên từ 0 đến 80, đại diện cho độ tuổi của bệnh nhân.
* Giới tính: Được mã hóa bằng phương pháp LabelEncoder, với giá trị 0 tương ứng với "Nam" và 1 tương ứng với "Nữ".
* Triệu chứng: Các cột nhị phân (giá trị 0 hoặc 1), biểu thị sự xuất hiện của các triệu chứng điển hình như "Sốt", "Ho", "Khó thở", "Buồn nôn",... Tổng cộng có khoảng 25 triệu chứng phổ biến được theo dõi.
* Bệnh tiền sử: Các cột nhị phân (giá trị 0 hoặc 1) biểu diễn sự hiện diện của các bệnh nền như "Tiểu đường", "Cao huyết áp", "Hen suyễn",…
* Bệnh hiện tại: Là biến mục tiêu (target), thể hiện bệnh được chẩn đoán (ví dụ: "Covid-19", "Cúm", "Viêm phổi",...), được mã hóa nhãn bằng LabelEncoder để phục vụ mô hình học máy.
* Nguồn dữ liệu được tổng hợp từ các bộ dữ liệu y tế công khai (như Kaggle, WHO), khảo sát từ chuyên gia hoặc dữ liệu mô phỏng được xây dựng thủ công nhằm đảm bảo tính đa dạng và cân bằng giữa các lớp bệnh.

- Dữ liệu bệnh viện gợi ý *(benhviengoiy.xlsx)*: Đây là tập dữ liệu chứa danh sách các bệnh viện uy tín theo khu vực (Miền Bắc, Trung, Nam) và theo từng bệnh, được tổng hợp thủ công từ các nguồn đáng tin cậy như website chính thức của các bệnh viện lớn (Vinmec, Bạch Mai, Chợ Rẫy,…), cổng thông tin của Bộ y tế, và các tổ chức y tế có thẩm quyền.

*Tiền xử lý dữ liệu:*

Để chuẩn bị cho giai đoạn huấn luyện mô hình, dữ liệu được xử lý theo các bước sau:

* Chuẩn hóa tên cột: loại bỏ khoảng trắng, ký tự dư thừa, và các cột bị trùng lặp, thống nhất định dạng để đảm bảo tính nhất quán khi xử lý.
* Mã hóa nhãn: sử dụng LabelEncoder cho cột có dữ liệu phân loại như "Giới tính" và "Bệnh hiện tại".
* Kiểm tra và xử lý dữ liệu không hợp lệ: Loại bỏ hoặc thay thế các giá trị sai định dạng, ví dụ: giá trị giới tính không nằm trong tập hợp {0, 1, "Nam", "Nữ"}.
* Xử lý giá trị thiếu: Các ô trống hoặc dữ liệu thiếu được thay thế bằng giá trị mặc định là 0.
* Chuyển kiểu dữ liệu: Các cột đặc trưng (triệu chứng và bệnh nền) được ép kiểu về kiểu số nguyên (int) để tương thích với các thuật toán học máy như Decision Tree, Random Forest,...

## Tăng cường dữ liệu/ phân tích dữ liệu

### Tăng cường dữ liệu

Để đảm bảo tính cân bằng trong dữ liệu huấn luyện, đặc biệt khi có sự chênh lệch lớn giữa số lượng mẫu của các loại bệnh, đề tài áp dụng kỹ thuật SMOTE (Synthetic Minority Over-sampling Technique). Kỹ thuật này giúp tạo thêm các mẫu tổng hợp cho các lớp bệnh ít xuất hiện, nhằm giảm sự thiên lệch của mô hình và cải thiện độ chính xác khi dự đoán.

Dữ liệu đầu vào được kiểm tra để đảm bảo tính hợp lệ:

* Dữ liệu nhị phân (các triệu chứng và bệnh nền) được kiểm tra đảm bảo chỉ chứa các giá trị hợp lệ {0, 1}. Các giá trị không hợp lệ nếu có sẽ bị loại bỏ hoặc thay thế.
* Tần suất xuất hiện của các đặc trưng (triệu chứng, bệnh nền) được thống kê. Các đặc trưng có tần suất thấp (ví dụ dưới 5%) sẽ được đánh dấu để xử lý hoặc loại bỏ nhằm tránh gây nhiễu trong quá trình huấn luyện.

### Phân tích dữ liệu

**Khám phá phân tích (Exploratory Data Analysis - EDA):** Tiến hành thống kê số lượng bệnh nhân theo từng loại bệnh và trực quan hóa bằng biểu đồ cột sử dụng thư viện *plotly.express*. Phân tích này giúp nhận diện các bệnh phổ biến nhất trong tập dữ liệu và kiểm tra mức độ phân bố, mất cân bằng giữa các lớp bệnh.

**Phân tích tương quan:** Áp dụng hệ số tương quan Pearson để xác định mối quan hệ giữa các đặc trưng (triệu chứng, bệnh nền) với biến mục tiêu (loại bệnh). Chỉ giữ lại các đặc trưng có hệ số tương quan tuyệt đối lớn hơn 0.1 nhằm làm nổi bật các yếu tố ảnh hưởng mạnh đến việc chẩn đoán. Kết quả được lưu trữ tạm thời trong *st.session\_state.correlation\_results* để phục vụ việc hiển thị và điều chỉnh trong giao diện người dùng.

**Luật kết hợp (Association Rules):** Áp dụng thuật toán Apriori (hoặc FP-Growth) từ thư viện mlxtend để khai phá các tập phổ biến (frequent itemsets) và xây dựng các luật kết hợp (association rules). Các luật dạng:

{Triệu chứng A, Bệnh nền B} ⇒ Bệnh X

Các luật này được sử dụng nhằm hỗ trợ quá trình chẩn đoán bằng cách cung cấp giải thích rõ ràng cho kết quả mô hình. Các tham số như *min\_support*, *min\_confidence* và *min\_lift* được điều chỉnh linh hoạt thông qua giao diện người dùng để kiểm soát chất lượng luật rút ra.

**Huấn luyện mô hình dự đoán:**Áp dụng các mô hình học máy như:

* Decision Tree
* Random Forest

So sánh độ chính xác trên tập kiểm tra, đánh giá bằng các chỉ số:

* Độ chính xác (Accuracy)
* Độ bao phủ (Recall)
* Độ chính xác phân loại (Precision)
* Chỉ số F1-score

**Phân cụm và luật kết hợp:** Áp dụng KMeans để chia bệnh nhân thành các cụm tương đồng dựa trên triệu chứng và bệnh nền. Trong mỗi cụm, tiếp tục sử dụng thuật toán Apriori hoặc FP-Growth để khai phá các luật đặc trưng riêng biệt cho từng nhóm bệnh nhân,từ đó giúp phát hiện các mô hình bệnh lý tiềm ẩn theo từng nhóm.

Ví dụ: {Sốt, Đau họng, Khó thở} → Viêm phổi

**Ứng dụng trong tư vấn:** Dựa trên kết quả dự đoán của mô hình và cụm, hệ thống đưa ra các gợi ý cá nhân hóa cho người dùng bao gồm:

* Mức độ nghiêm trọng của bệnh.
* Hành động nên thực hiện (theo hướng dẫn của Bộ y tế).
* Danh sách bệnh viện phù hợp theo khu vực và loại bệnh.

# THIẾT KẾ HỆ THỐNG

## Đề xuất sử dụng thuật toán

Để giải quyết bài toán dự đoán bệnh từ triệu chứng ban đầu, hệ thống sử dụng kết hợp các thuật toán học máy và khai phá dữ liệu như sau:

### Thuật toán học máy

Decision Tree (Cây quyết định): Phù hợp với các bài toán phân loại dữ liệu nhị phân. Tham số class\_weight='balanced' được sử dụng để xử lý bài toán mất cân bằng dữ liệu giữa các lớp bệnh.

Random Forest (Rừng ngẫu nhiên): Là tổ hợp của nhiều cây quyết định (với n\_estimators=100), giúp cải thiện độ chính xác và khắc phục hiện tượng overfitting.

Logistic Regression (Hồi quy logistic): Được sử dụng trong bước phân cụm để ước lượng xác suất mắc bệnh trong từng cụm bệnh nhân.

### Kỹ thuật tăng cường dữ liệu

SMOTE (Synthetic Minority Over-sampling Technique): Tạo dữ liệu nhân tạo cho các lớp bệnh ít xuất hiện nhằm cân bằng dữ liệu đầu vào cho mô hình.

### Phân cụm và luật kết hợp

KMeans: Phân nhóm các bệnh nhân dựa vào triệu chứng và bệnh nền từ đó hỗ trợ mô hình xác định các nhóm bệnh có đặc điểm tương đồng.

Apriori / FP-Growth: Khai thác luật kết hợp giữa triệu chứng, bệnh nền và bệnh hiện tại, với các tham số min\_support , min\_trust , và min\_lift được sử dụng để lọc ra các luật có ý nghĩa thống kê.

Ví dụ: {Sốt, Mất vị giác} ⇒ Covid-19

## Cách thức giải quyết bài toán

### Tiền xử lý dữ liệu

Chuẩn hóa tên cột: xóa khoảng trắng thừa, cột trùng lặp, loại bỏ giá trị không hợp lệ.

Mã hóa nhãn (dùng LabelEncoder) cho các cột phân loại như "Giới tính", "Bệnh hiện tại".

Chuyển các cột triệu chứng và bệnh nền về dạng nhị phân (0/1).

Dữ liệu bị thiếu sẽ được thay thế bằng 0 để đảm bảo tính toàn vẹn.

### Tăng cường và trích chọn đặc trưng

Sử dụng kỹ thuật SMOTE để tạo thêm dữ liệu cho các bệnh có số mẫu ít gặp, từ đó giảm thiên lệch trong huấn luyện mô hình.

Tách các cột đặc trưng đầu vào gồm: Tuổi, Giới tính, Triệu chứng, Bệnh tiền sử.

Chọn lọc các triệu chứng có tần suất xuất hiện cao, loại bỏ những cột có giá trị quá thấp hoặc toàn 0 nhằm giảm nhiễu và tăng hiệu suất mô hình.

### Xây dựng mô hình

Học máy: Huấn luyện Decision Tree (Cây quyết định) và Random Forest (Rừng ngẫu nhiên) để dự đoán bệnh. Đánh giá mô hình bằng các chỉ số Accuracy, Precision, Recall, F1-score.

Phân cụm và khai phá luật: Sử dụng KMeans để phân chia dữ liệu thành các cụm bệnh nhân tương đồng. Áp dụng thuật toán Apriori / FP-Growth trong từng cụm để trích xuất các luật giữa triệu chứng, bệnh tiền sử và bệnh hiện tại. Mô hình Logistic Regression được huấn luyện trong từng cụm để hỗ trợ xác suất dự đoán bệnh từ thông tin đầu vào của người dùng.

### Điều chỉnh hiệu suất dự đoán

Kết hợp giữa hệ số tương quan Pearson và luật kết hợp để tăng độ chính xác của dự đoán, đặc biệt cho các bệnh nhẹ như cúm, viêm họng, sốt nhẹ, amidan,...

Tăng xác suất dự đoán cho các bệnh nhẹ nếu bệnh nhân chỉ có ≤ 4 triệu chứng và không xuất hiện các triệu chứng nghiêm trọng (như khó thở, đau ngực, mất vị giác,...).

Kết quả được trực quan hóa qua biểu đồ để so sánh hiệu suất các mô hình.

### Phân tích cụm và luật kết hợp

Áp dụng KMeans để phân loại bệnh nhân thành các nhóm tương đồng.

Trong mỗi cụm, khai thác luật kết hợp bằng Apriori hoặc FP-Growth để tìm các mẫu bệnh phổ biến.

***Đánh giá và tối ưu hóa***

Đánh giá mô hình bằng cách sử dụng các loại hiệu suất và báo cáo chỉ báo.

Có thể điều chỉnh các thông số như cụm số (k = 3 đến 15) và các tham số min\_support, min\_confidence, min\_lift trong Apriori qua giao diện Streamlit để tối ưu hóa kết quả.

***Lưu ý:*** Hệ thống yêu cầu tối thiểu 2 triệu chứng để mô hình học máy hoạt động và tối thiểu 3 triệu chứng để phân cụm có ý nghĩa.

## Giao diện người dùng và ứng dụng web

Hệ thống được xây dựng dưới dạng ứng dụng web sử dụng Streamlit với các chức năng:

* Tab "Phân tích dữ liệu": Hiển thị thống kê, biểu đồ tần suất triệu chứng, mối tương quan giữa các đặc trưng.
* Tab "Dự đoán bệnh": Cho phép người dùng nhập thông tin (tuổi, giới tính, triệu chứng, bệnh sử) và nhận kết quả dự đoán 3 bệnh có xác suất cao nhất, mức độ nghiêm trọng và hướng xử lý.
* Tab "Phân cụm": Hiển thị kết quả phân cụm, luật kết hợp, biểu đồ hỗ trợ chẩn đoán.

Tab "Chatbox": Người dùng chọn khu vực, nhập triệu chứng và bệnh nền. Hệ thống trả lời về bệnh có khả năng hiển thị cao nhất, mức độ nghiêm trọng, hướng xử lý và danh sách bệnh viện gợi ý từ benhviengoiy.xlsx .

# CÀI ĐẶT ỨNG DỤNG BÀI TOÁN

## Môi trường và công cụ phát triển

- Ngôn ngữ lập trình: Python 3.10  
 - Thư viện sử dụng:  
 - pandas, numpy: Xử lý dữ liệu  
 - scikit-learn: Huấn luyện mô hình học máy (Decision Tree, Random Forest, Logistic Regression)  
 - imblearn.SMOTE: Xử lý mất cân bằng dữ liệu  
 - mlxtend: Khai thác luật kết hợp bằng thuật toán Apriori / FP- Growth  
 - plotly: Trực quan hóa dữ liệu và kết quả mô hình  
 - streamlit: Tạo giao diện web tương tác  
 - IDE: Visual Studio Code  
 - Hệ điều hành: Windows 11

## Cài đặt giao diện ứng dụng

Ứng dụng được phát triển dưới dạng web sử dụng Streamlit, với các thành phần chính:

- Frontend: Giao diện người dùng được xây dựng bằng HTML, CSS (tùy chọn tinh chỉnh chiến lược hành động) và các thành phần Streamlit như st.selectbox, st.checkbox, st.form.

- Backend: Python với các thư viện scikit-learn, imblearn, mlxtend vàploly.express để xử lý dữ liệu, huấn luyện mô hình và trực quan hóa .

- Chức năng:

- Phân tích dữ liệu: hiển thị bảng và biểu đồ phân bố bệnh, tương quan các đặc trưng.

- Dự đoán bệnh: Người dùng nhập thông tin, hệ thống sử dụng Rừng ngẫu nhiên và Cây quyết định để dự đoán, điều chỉnh dựa trên cơ sở tương quan và luật hợp lý.

- Phân cụm: Sử dụng KMeans và Apriori để tìm các mẫu bệnh trong từng cụm, kết hợp Hồi quy logistic hợp lý để dự đoán bệnh.

- Chatbox: Gợi ý hành động, cung cấp mức độ nghiêm trọng, và danh sách bệnh viện dựa trên khu vực từ benhviengoiy.xlsx .

- Tính năng nổi bật: Giao diện thân thiện, hỗ trợ tải dữ liệu từ tệp Excel, trực quan hóa kết quả qua biểu đồ và khả năng điều chỉnh tham số (số cụm, min\_support , min\_trust , min\_lift ).

## Luồng xử lý của ứng dụng

1. Người dùng tải file dữ liệu mẫu (.xlsx)  
 2. Hệ thống tiền xử lý dữ liệu: chuẩn hóa, mã hóa, chọn đặc trưng  
 3. Huấn luyện mô hình học máy (Decision Tree, Random Forest)  
 4. Phân cụm dữ liệu bằng KMeans  
 5. Sinh luật kết hợp từ triệu chứng và bệnh sử  
 6. Nhập thông tin dự đoán: Tuổi, Giới tính, Triệu chứng, Tiền sử bệnh  
 7. Hệ thống trả về:  
 - Top 3 bệnh có khả năng cao nhất (theo xác suất)  
 - Luật kết hợp xác nhận bệnh  
 - Gợi ý hành động  
 - Danh sách bệnh viện phù hợp

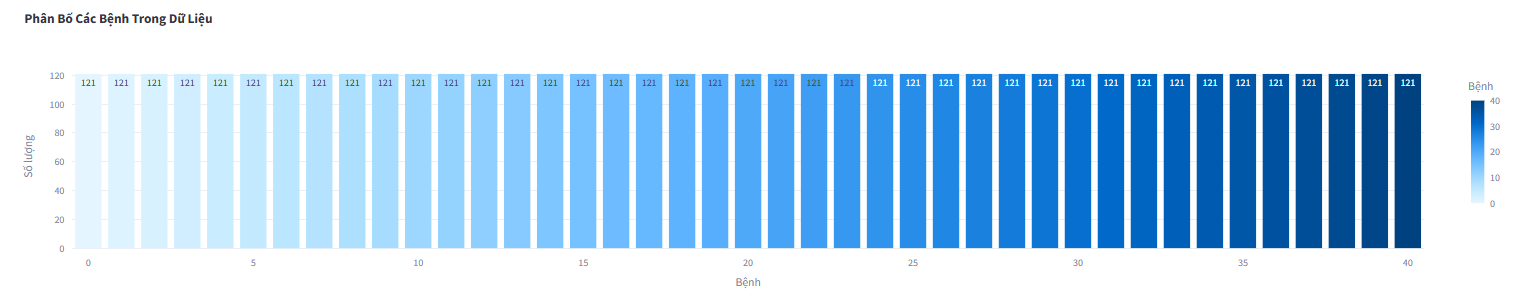
***Tệp chương trình chính***

- ‘main.py’: tập trung xử lý backend, huấn luyện mô hình, phân tích luật kết hợp và logic gợi ý bệnh viện  
 - ‘codetest.py’: là file chạy chính giao diện web Streamlit, tích hợp toàn bộ logic xử lý vào giao diện  
 - ‘benhviengoiy.xlsx’: chứa danh sách bệnh viện gợi ý theo bệnh và khu vực  
 - ‘data\_benh.xlsx’: chứa dữ liệu bệnh nhân dùng cho huấn luyện và kiểm thử

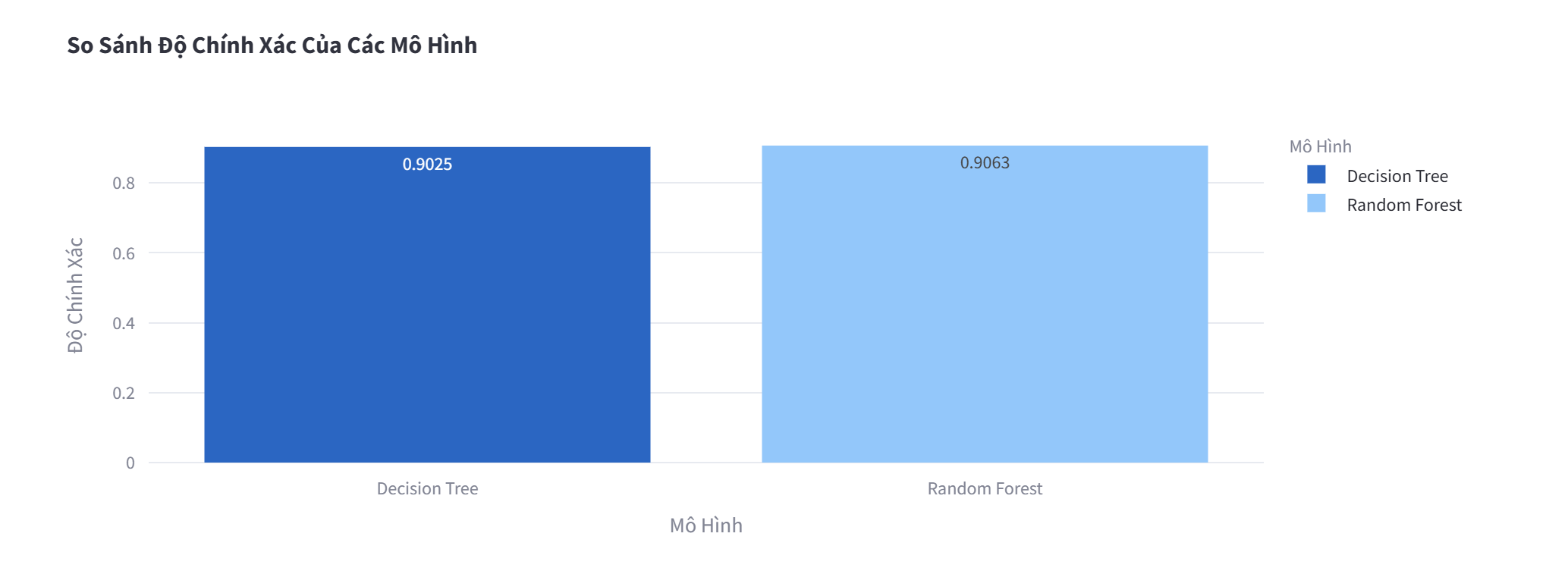
## Một số giao diện chính

- Giao diện bắt đầu:



Với tab “Phân tích dữ liệu” ta có thể xem biểu đồ phân bố các bệnh trong tập dữ liệu cũng như có thể xem các phân tích tương quan giữa triệu chứng bệnh và bệnh.

- Với tab “Dự đoán bệnh” chúng ta có thể xem biểu đồ so sánh độ chính xác của các mô hình



Biểu đồ trên thể hiện sự so sánh độ chính xác giữa hai mô hình học máy được áp dụng trong hệ thống dự đoán bệnh:

- Decision Tree (Cây quyết định) đạt độ chính xác 90.25%

- Random Forest (Rừng ngẫu nhiên) đạt độ chính xác 90.63%

Phân tích:

Mô hình Random Forest có độ chính xác cao hơn nhẹ so với Decision Tree. Điều này phù hợp với kỳ vọng vì Random Forest là tập hợp của nhiều cây quyết định, giúp giảm thiểu hiện tượng overfitting và cải thiện tổng thể độ chính xác.

Cả hai mô hình đều đạt độ chính xác trên 90%, chứng tỏ dữ liệu đầu vào và kỹ thuật tiền xử lý (bao gồm chuẩn hóa, mã hóa nhãn, và cân bằng dữ liệu với SMOTE) đã phát huy hiệu quả.

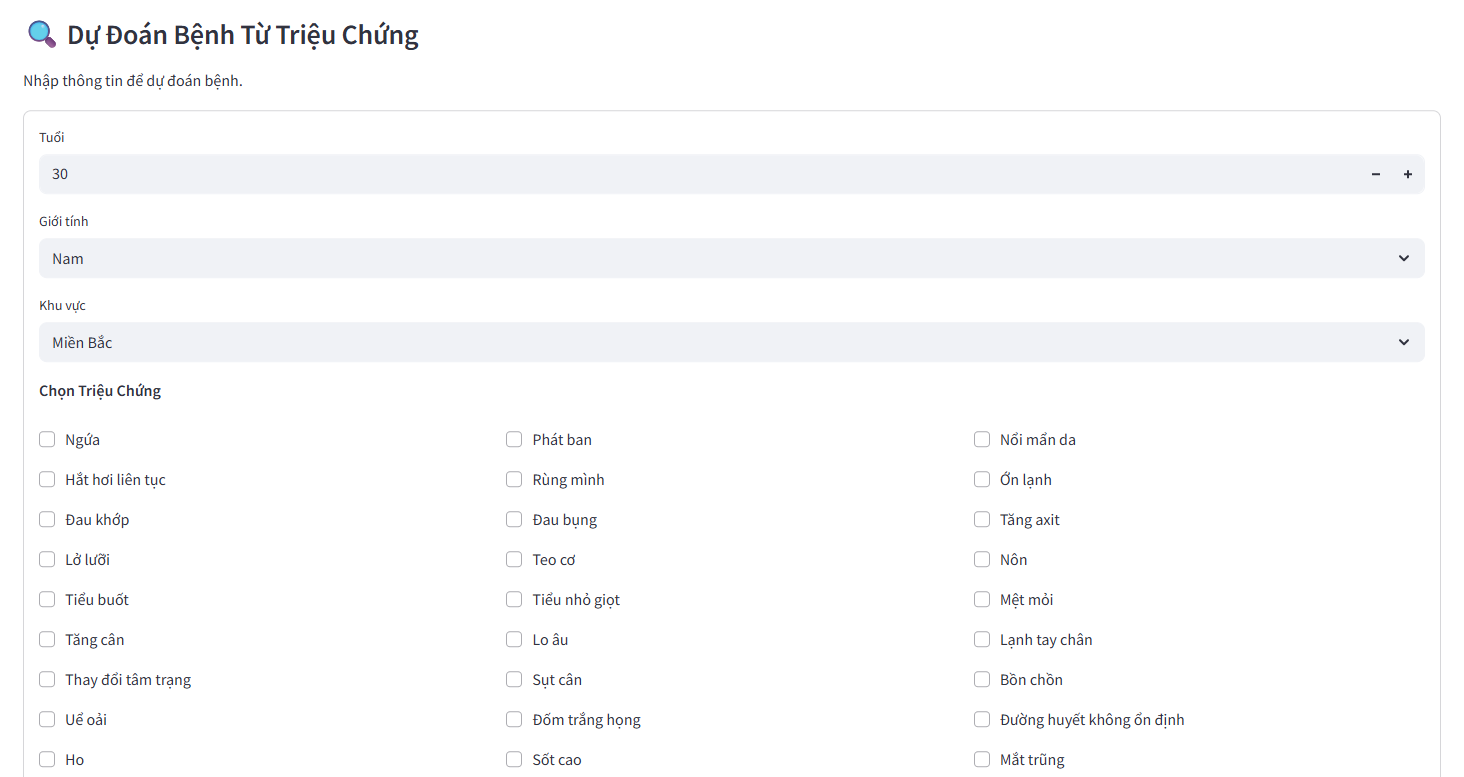
Sự khác biệt về độ chính xác tuy không lớn nhưng cho thấy Random Forest là lựa chọn ưu tiên cho việc triển khai dự đoán bệnh chính trong hệ thống.

*Kết luận:*

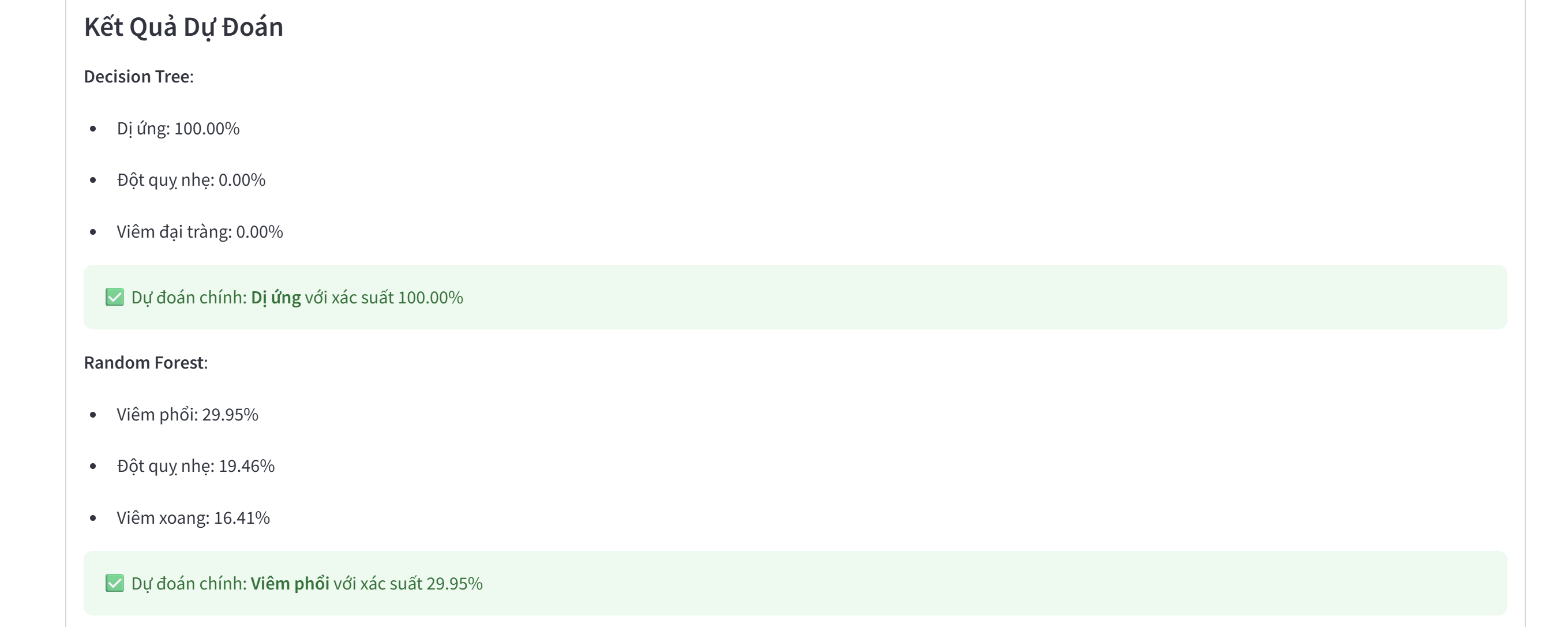
Mô hình Random Forest được lựa chọn làm mô hình chính cho chức năng dự đoán bệnh do có độ chính xác cao nhất, khả năng tổng quát tốt hơn, và hiệu quả trong việc xử lý dữ liệu đa dạng về triệu chứng và bệnh nền.

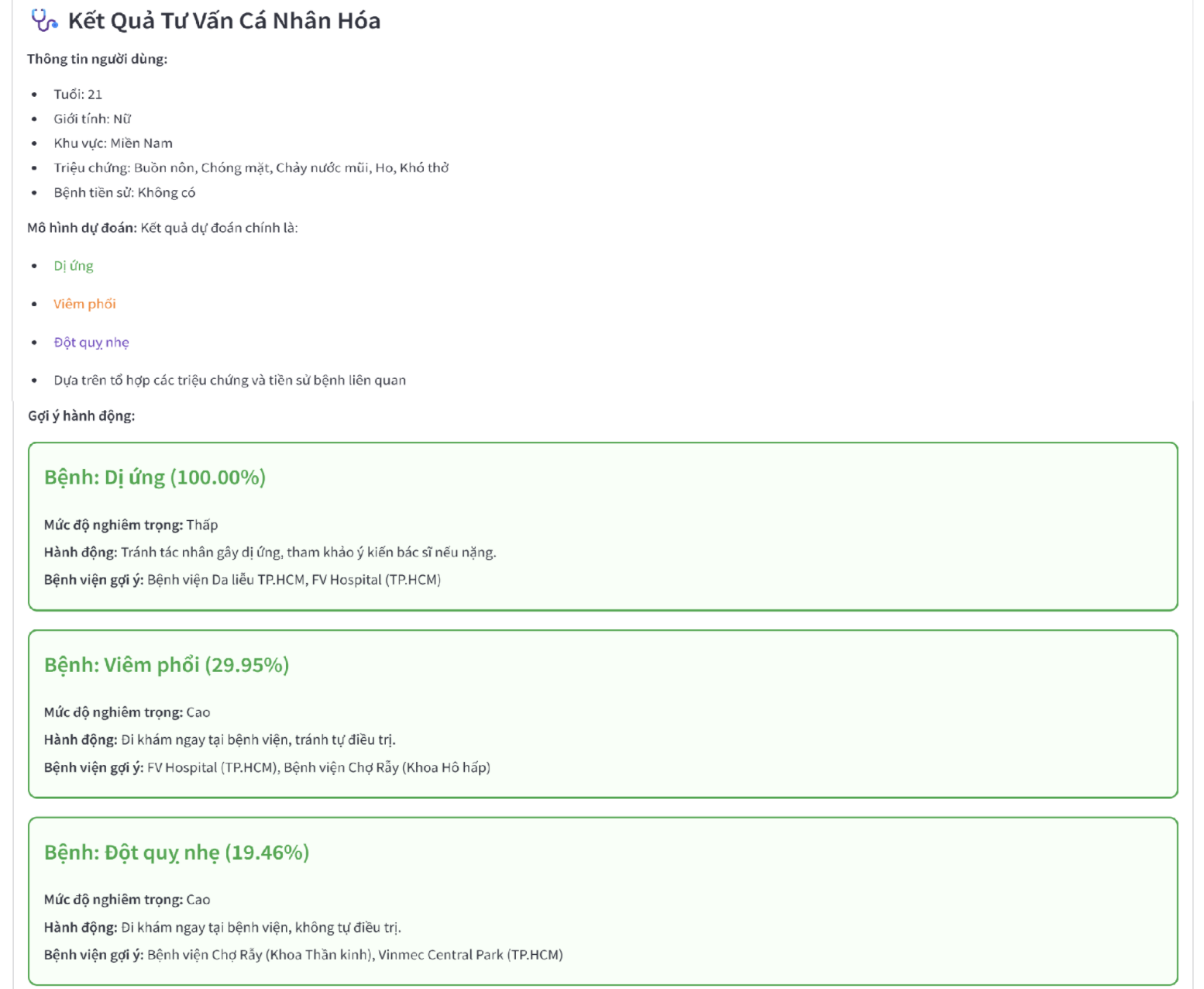
- Giao diện dự đoán bệnh từ triệu chứng

Sau khi nhập thông tin:



Sau đó bấm nút dự đoán và thu được kết quả như giao diện bên dưới:

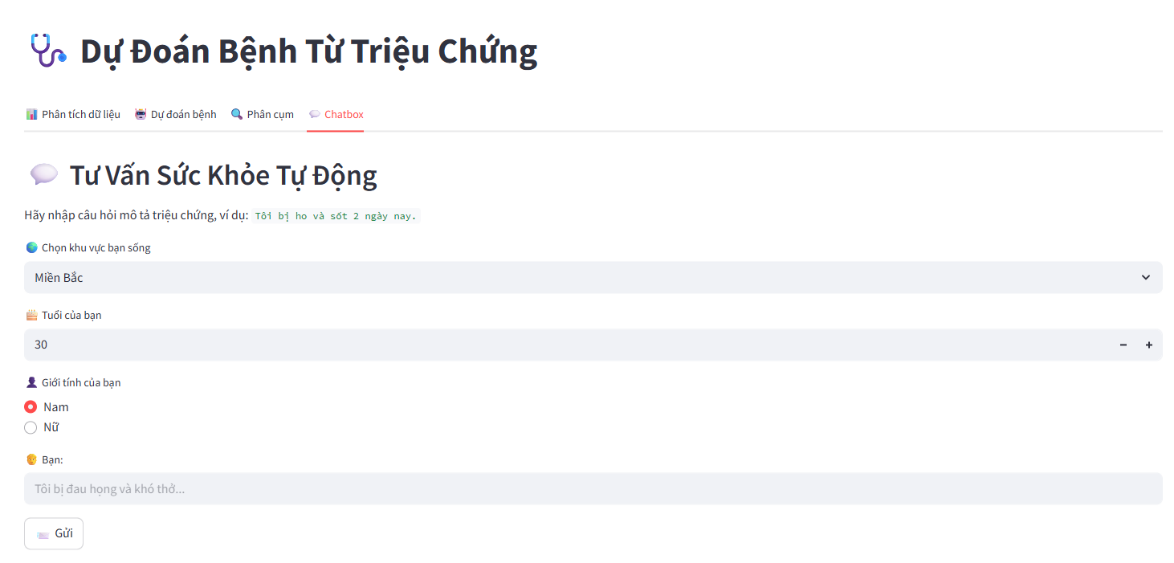




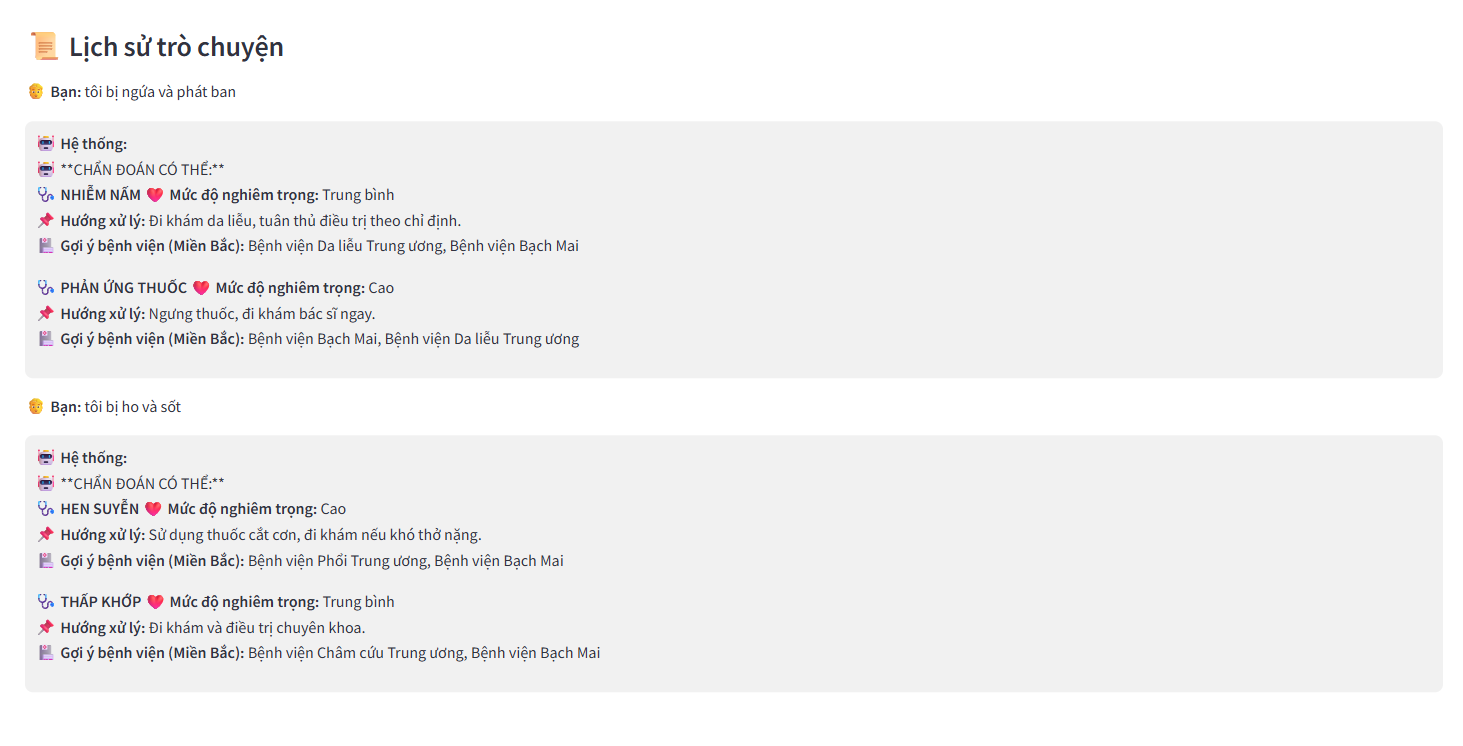
- Giao diện phân cụm





 - Giao diện chatbox:

Đây là kết quả sau khi hỏi về triệu chứng bệnh để chatbox tư vấn: Sau mỗi lần hỏi xong thì lịch sử trò chuyện của người dùng và hệ thống sẽ được lưu lại



# KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## Kết luận

Trong đề tài này, hệ thống dự đoán bệnh từ triệu chứng ban đầu đã được xây dựng thành công, dựa trên việc kết hợp các thuật toán học máy với khai phá luật kết hợp và phân tích dữ liệu. Hệ thống hỗ trợ người dùng trong việc chẩn đoán bệnh nhanh chóng, đồng thời đề xuất các bệnh viện phù hợp theo khu vực.

Thông qua việc áp dụng các kỹ thuật như Decision Tree, Random Forest, phân cụm KMeans, luật kết hợp Apriori/FP-Growth, và xử lý mất cân bằng dữ liệu bằng SMOTE, hệ thống đạt được khả năng dự đoán tương đối chính xác, minh bạch và dễ hiểu.

Bên cạnh chức năng dự đoán, hệ thống còn cung cấp thông tin bổ trợ như mức độ nghiêm trọng của bệnh, gợi ý hành động xử lý, và danh sách bệnh viện gần nhất (dựa trên dữ liệu thực tế từ benhviengoiy.xlsx), giúp nâng cao hiệu quả ứng dụng trong thực tiễn chăm sóc sức khỏe.

## Định hướng phát triển

- Mở rộng tập dữ liệu: Thu thập dữ liệu bệnh thực tế từ các nguồn uy tín như Bộ Y tế, WHO, Kaggle để tăng độ tin cậy cho mô hình.  
 - Phân tích ngôn ngữ tự nhiên: Bổ sung khả năng xử lý văn bản tự do từ người dùng (NLU - Natural Language Understanding), cho phép người dùng nhập mô tả triệu chứng thay vì chọn checkbox cố định.

- Tích hợp học sâu (deep learning): Kết hợp mô hình mạng nơ-ron sâu để nâng cao độ chính xác, đặc biệt với dữ liệu phi cấu trúc hoặc phức tạp.

- Tư vấn sức khỏe tự động: Tích hợp hệ thống tư vấn, chatbot sức khỏe, hoặc kết nối trực tiếp đến bác sĩ qua video/ứng dụng di động.

- Triển khai đa nền tảng: Xây dựng ứng dụng web và mobile để mở rộng đối tượng người dùng và nâng cao tính thực tiễn.

- Tối ưu hiệu năng: Cải thiện thời gian xử lý và khả năng mở rộng khi làm việc với tập dữ liệu lớn hơn hoặc truy cập đồng thời từ nhiều người dùng.

# TÀI LIỆU THAM KHẢO

1. Scikit-learn developers. (n.d.). Scikit-learn documentation. <https://scikit-learn.org/>
2. Streamlit. (n.d.). Streamlit documentation. <https://docs.streamlit.io/>
3. Kaggle. (2023). Medical symptoms dataset. https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning
4. Lê, V. Đ. (n.d.). Khai thác luật kết hợp với thuật toán Apriori trong khai phá dữ liệu. Lê Văn Đông Blog.
5. Viện Huyết học - Truyền máu Trung ương. (n.d.). Hướng dẫn chẩn đoán và điều trị một số bệnh lý huyết học.
6. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
7. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357.
8. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106.
9. Trường Đại học Công Thương TP. Hồ Chí Minh. (n.d.). Slide bài giảng môn Trí tuệ nhân tạo. Khoa Công nghệ Thông tin.
10. Trường Đại học Công Thương TP. Hồ Chí Minh. (n.d.). Slide bài giảng môn Khai phá dữ liệu. Khoa Công nghệ Thông tin.