**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------**🙡🕮🙣**----------**



**BÁO CÁO ĐỒ ÁN**

**Xây Dựng Mô Hình Dự Đoán Bệnh Tiểu Đường Dựa Trên**

**Phân Tích Dữ Liệu Y Tế**

|  |  |  |
| --- | --- | --- |
| **Thành viên nhóm:** | Bùi Nguyên Minh Thư | 3121410488 |
|  | Đặng Lê Tâm Như | 3121410372 |
| **Giảng viên hướng dẫn:** | Thầy Phan Thành Huấn |  |
|  |  |  |
| **Học phần:** | Phân tích dữ liệu |  |

**HỌC KỲ I**

**NĂM HỌC 2024 - 2025**

**Thành phố Hồ Chí Minh, ngày 10 tháng 12 năm 2024**

**MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU VÀ ĐỊNH HƯỚNG ĐỒ ÁN 8**](#_k9sqdtyk0lbh)

[**1.1. Lý do lựa chọn đề tài: 8**](#_fk9jcmvznpzl)

[**1.2. Mục tiêu đề tài: 8**](#_8txzvruik1wz)

[**1.3. Mô tả tập dữ liệu sử dụng: 9**](#_aqglhe396u23)

[**1.4. Thống kê mô tả dữ liệu: 10**](#_xxfbgmq9l5vl)

[**CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU 12**](#_fxn07mp86p4x)

[**2.1. Đọc và chuẩn bị dữ liệu: 12**](#_77sqv1t10s1i)

[**2.2. Làm sạch dữ liệu: 13**](#_9zzbhjn04c0g)

[2.2.1. Loại bỏ các giá trị không hợp lệ: 13](#_d2yjn08imxhj)

[2.2.1.1. Thống kê các dữ liệu không hợp lệ: 14](#_c69xvn5fuwul)

[2.2.1.2. Xóa các dữ liệu không hợp lệ: 14](#_skvsae6qpkkp)

[2.2.2. Chuẩn hóa dữ liệu: 15](#_k3wdrkuhgylf)

[2.2.3. Xử lý dữ liệu thiếu: 16](#_zkqg9j4tdw7)

[**2.3. Chuyển đổi và biến đổi dữ liệu: 19**](#_6zuhgbsjnayn)

[2.3.1. Lọc dữ liệu: 19](#_2o7n51s1rkei)

[2.3.2. Biến đổi dữ liệu: 21](#_p27gph9sgglb)

[**2.4. Tạo đặc trưng và dữ liệu mới: 21**](#_4v3ds5z66w4x)

[**CHƯƠNG 3: PHÂN TÍCH VÀ KHÁM PHÁ DỮ LIỆU 23**](#_8vepyaers20r)

[**3.1. Các phương pháp phân tích sử dụng: 23**](#_w8t7masf2ofv)

[**3.2. Phân tích mối quan hệ giữa các yếu tố ảnh hưởng đến bệnh tiểu đường: 24**](#_k6nn7ye0dmew)

[3.2.1.Hồi quy Logistic (Logistic Regression) 26](#_9p41nlp2xu5q)

[3.2.2. Cây quyết định (Decision Tree) 30](#_4dwkwjc5kcjl)

[3.2.3. Rừng ngẫu nhiên (Random Forest) 35](#_8zpauqajogkm)

[3.2.4. K láng giềng gần nhất (K-Nearest Neighbors) 39](#_yrcuxvn3jnt3)

[3.2.5. Máy vector hỗ trợ (Support Vector Regression) 43](#_xopo0uh0mbx2)

[**3.3. Kết luận về quá trình phân tích: 47**](#_sud915n44oyk)

[Kết luận 48](#_lfhnzp94ihf4)

[**CHƯƠNG 4: XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH DỰ ĐOÁN BỆNH TIỂU ĐƯỜNG 50**](#_yg1kh74puxwu)

[**4.1. Tổng quát về tập dữ liệu và thuật toán sử dụng: 50**](#_o73eetf99r4h)

[**4.2. Thuật toán sử dụng xây dựng mô hình dự đoán bệnh tiểu đường: 55**](#_xes6fd263hns)

[4.2.1. Thuật toán ID3: 56](#_o0aqo250ypdf)

[4.2.2. Thuật toán C4.5: 56](#_2gol126mi5rc)

[4.2.3. Thuật toán CART: 57](#_3ztepuex9ylm)

[**4.3. Quy trình triển khai và đánh giá mô hình: 58**](#_sactv2l3zl7k)

[4.3.1. Xử lý dữ liệu đầu vào: 58](#_lpfdif6y2fub)

[4.3.2. Áp dụng thuật toán ID3: 61](#_lftmboozy3u8)

[4.3.3. Áp dụng thuật toán C4.5: 71](#_tmfrswaavoqj)

[4.3.4. Áp dụng thuật toán CART: 83](#_fmikpy9llq2)

[**CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 93**](#_t17jam24c1t8)

[**5.1. Tổng quan kết quả đạt được: 93**](#_vkjh8mv55em6)

[**5.2. Các hạn chế và rủi ro gặp phải: 93**](#_ycibk6uqvm99)

[**5.3. Định hướng phát triển: 93**](#_idnjwhtvj0gs)

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.

[2] Wikipedia contributors. (n.d.). Diabetes. Wikipedia, The Free Encyclopedia. Retrieved December 2, 2024, from https://en.wikipedia.org/wiki/Diabetes

[3] Scikit-learn Developers. (n.d.). Learning curve. Scikit-learn. Retrieved December 2, 2024, from https://scikit-learn.org/dev/modules/generated/sklearn.model\_selection.learning\_curve.html

[4] Hugues, S. (2021). Basic ML: Best of 10 classifiers [Jupyter notebook]. Kaggle. Retrieved December 2, 2024, from https://www.kaggle.com/code/hugues/basic-ml-best-of-10-classifiers

**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn - thầy Phan Thành Huấn, là giảng viên bộ môn Phân tích dữ liệu, người đã tận tình hướng dẫn, chỉ bảo nhóm của chúng em trong suốt quá trình làm đồ án môn học.

Chúng em cũng xin gửi lời cảm ơn chân thành nhất đến các thầy cô trong trường Đại học Sài Gòn nói chung và các thầy cô khoa Công nghệ thông tin nói riêng đã truyền đạt cho chúng em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình chúng em tham gia học tập.

Với điều kiện về thời gian cũng như lượng kiến thức, kinh nghiệm còn hạn chế, đồ án môn học của nhóm chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để nhóm chúng em có điều kiện bổ sung, nâng cao ý thức của mình, hoàn thành đồ án môn học tốt hơn.

Nhóm chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày 10 tháng 12 năm 2024.

**NHẬN XÉT CỦA GIẢNG VIÊN**

**…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Thành viên - MSSV** | **Công việc** | **Phần trăm đóng góp** |
| Đặng Lê Tâm Như - 3121410372 | - Viết báo cáo Word  - Tiền xử lý dữ liệu  - Phân tích dữ liệu ( mô hình Logistic Regression, Decision Tree)  - Xử lý dữ liệu đầu vào  - Xây dựng mô hình Cây quyết định để phân tích bệnh tiểu đường (Phương pháp CART) | 50% |
| Bùi Nguyên Minh Thư - 3121410488 | - Làm PowerPoint  - Tìm dữ liệu  - Mô tả dữ liệu  - Phân tích dữ liệu ( mô hình Random Forest, K-Nearest Neighbors, SVC)  - Xây dựng mô hình Cây quyết định để phân tích bệnh tiểu đường (Phương pháp ID3, C4.5) | 50% |

# CHƯƠNG 1: GIỚI THIỆU VÀ ĐỊNH HƯỚNG ĐỒ ÁN

## 1.1. Lý do lựa chọn đề tài:

Bệnh tiểu đường là một trong những bệnh lý mãn tính phổ biến và có tỷ lệ mắc bệnh ngày càng gia tăng trên toàn thế giới. Việc phát hiện sớm và kiểm soát bệnh tiểu đường đóng vai trò quan trọng trong việc ngăn ngừa các biến chứng nguy hiểm như bệnh tim mạch, suy thận và tổn thương thần kinh. Bệnh tiểu đường loại 1 là kết quả của việc tuyến tụy không sản xuất đủ insulin. Bệnh tiểu đường loại 2 bắt đầu bằng tình trạng kháng insulin, một tình trạng mà các tế bào không phản ứng đúng với insulin. Tính đến năm 2015, ước tính có 415 triệu người mắc bệnh tiểu đường trên toàn thế giới, trong đó bệnh tiểu đường loại 2 chiếm khoảng 90% các trường hợp. Con số này chiếm 8,3% dân số trưởng thành.

Tính đến năm 2021, ước tính có 537 triệu người mắc bệnh tiểu đường trên toàn thế giới, chiếm 10,5% dân số người lớn, trong đó loại 2 chiếm khoảng 90% tổng số ca. Tổ chức Y tế Thế giới đã báo cáo rằng tiểu đường là "một trong 10 nguyên nhân hàng đầu gây tử vong vào năm 2021, với tỷ lệ tăng đáng kể 95% kể từ năm 2000." Dự đoán rằng đến năm 2045, khoảng 783 triệu người lớn, tức 1 trong 8 người, sẽ sống chung với bệnh tiểu đường, tương ứng với mức tăng 46% so với các số liệu hiện tại. Tỷ lệ mắc bệnh tiếp tục gia tăng, đặc biệt là ở các quốc gia có thu nhập thấp và trung bình. Tỷ lệ này tương đương ở cả phụ nữ và nam giới, với tiểu đường là nguyên nhân thứ bảy gây tử vong toàn cầu. Chi phí toàn cầu cho chăm sóc sức khỏe liên quan đến tiểu đường ước tính lên tới 760 tỷ USD mỗi năm.

Dựa trên sự phát triển mạnh mẽ của khoa học dữ liệu và các thuật toán học máy, việc xây dựng mô hình dự đoán nguy cơ mắc bệnh tiểu đường không chỉ hỗ trợ các bác sĩ trong việc chẩn đoán mà còn nâng cao khả năng phòng ngừa bệnh cho cộng đồng. Do đó, nhóm chúng em sẽ áp dụng kiến thức phân tích dữ liệu và học máy để xây dựng một mô hình dự đoán hiệu quả, góp phần vào việc cải thiện chất lượng chăm sóc y tế và nâng cao nhận thức về tầm quan trọng của việc phòng chống bệnh tiểu đường.

## 1.2. Mục tiêu đề tài:

Mục tiêu của đề tài này là xây dựng mô hình dự đoán bệnh tiểu đường dựa trên phân tích dữ liệu y tế, nhằm xác định các yếu tố nguy cơ có thể dẫn đến căn bệnh này. Cụ thể, đề tài sẽ:

* **Phân tích các số liệu y khoa chính**: Tìm hiểu và xác định các số liệu sức khỏe như nồng độ glucose, huyết áp, chỉ số BMI, độ dày da và các yếu tố khác có ảnh hưởng đến nguy cơ mắc bệnh tiểu đường.
* **Phát triển mô hình dự đoán:** Sử dụng các thuật toán học máy như hồi quy, cây quyết định,... để phát triển một mô hình dự đoán có khả năng phân loại bệnh nhân thành hai nhóm: mắc bệnh tiểu đường và không mắc bệnh tiểu đường dựa trên các thuộc tính y tế đã được phân tích.
* **Đánh giá hiệu suất mô hình:** Kiểm tra độ chính xác và hiệu suất của mô hình bằng cách sử dụng các phương pháp đánh giá khác nhau như ma trận nhầm lẫn, độ chính xác, độ nhạy và độ đặc hiệu.
* **Đề xuất giải pháp phòng ngừa:** Trên cơ sở kết quả phân tích và dự đoán, đề xuất các giải pháp phòng ngừa bệnh tiểu đường phù hợp cho các nhóm đối tượng có nguy cơ cao, góp phần nâng cao nhận thức về sức khỏe cộng đồng.

Mục tiêu chính của đề tài là nâng cao nhận thức về bệnh tiểu đường và tầm quan trọng của việc phát hiện sớm thông qua việc phát triển một công cụ dự đoán hiệu quả, giúp bệnh nhân và các chuyên gia y tế có được thông tin cần thiết để đưa ra quyết định phù hợp về điều trị và chăm sóc sức khỏe.

## 1.3. Mô tả tập dữ liệu sử dụng:

Tập dữ liệu Diabetes.csv được lấy từ trang web <https://kaggle.com/>, chứa thông tin sức khỏe cơ bản của bệnh nhân liên quan đến bệnh tiểu đường. Mục tiêu là phân loại bệnh nhân thành hai nhóm: mắc bệnh tiểu đường và không mắc bệnh tiểu đường

Tập dữ liệu này bao gồm các thuộc tính sau: 768 dòng x 9 cột

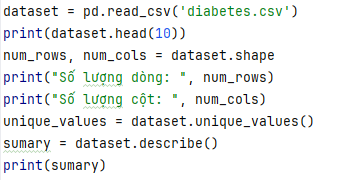
* **Pregnancies:** Số lần mang thai của bệnh nhân (giá trị số nguyên). Yếu tố này đặc biệt liên quan đến nữ giới và có thể ảnh hưởng đến nguy cơ mắc tiểu đường, vì sự thay đổi hormone trong quá trình mang thai có thể tác động đến khả năng xử lý glucose của cơ thể.
* **Glucose**: Lượng glucose trong máu, một chỉ số quan trọng để đánh giá nguy cơ mắc bệnh tiểu đường.
* **BloodPressure**: Huyết áp, có thể liên quan đến các vấn đề về sức khỏe và nguy cơ bệnh tiểu đường.
* **SkinThickness**: Độ dày da, có thể dùng làm dấu hiệu cho sự tích tụ mỡ dưới da.
* **Insulin**: Mức insulin, một hormone quan trọng trong điều chỉnh glucose trong máu.
* **BMI**: Chỉ số khối cơ thể (Body Mass Index), giúp đánh giá tình trạng cân nặng của bệnh nhân.
* **DiabetesPedigreeFunction**: Thước đo có tính chất gia đình đối với bệnh tiểu đường, dùng để xác định khả năng di truyền.
* **Age**: Tuổi của bệnh nhân, có liên quan mật thiết đến nguy cơ mắc bệnh.
* **Outcome**: Kết quả về tình trạng tiểu đường, là biến mục tiêu để xác định liệu bệnh nhân có bị bệnh tiểu đường hay không.

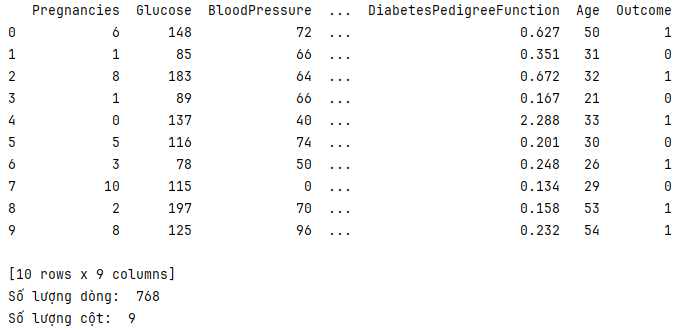
**Bảng mô tả dữ liệu:**

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Pregnancies | int | Số lần mang thai của bệnh nhân. |
| Glucose | float | Nồng độ glucose trong máu (mg/dL). |
| BloodPressure | float | Huyết áp tâm trương (mmHg). |
| SkinThickness | float | Độ dày da (mm) đo bằng cách sử dụng caliper. |
| Insulin | float | Nồng độ insulin trong máu (μU/mL). |
| BMI | float | Chỉ số khối cơ thể, được tính toán dựa trên chiều cao và cân nặng của bệnh nhân. |
| DiabetesPedigreeFunction | float | Chỉ số di truyền cho thấy mức độ ảnh hưởng của yếu tố di truyền đến nguy cơ mắc bệnh tiểu đường. |
| Age | int | Độ tuổi của bệnh nhân (năm). |
| Outcome | int | Biến mục tiêu cho biết tình trạng của bệnh nhân là mắc bệnh tiểu đường (1) hay không mắc (0). |

## 1.4. Thống kê mô tả dữ liệu:

Khi kiểm tra mười hàng dữ liệu đầu tiên, một điều dễ nhận thấy ngay là có các phép đo bằng không đối với cả mức insulin và độ dày da.

****

****

Sẽ là một vấn đề y khoa nghiêm trọng nếu bệnh nhân có mức insulin và độ dày da bằng 0. Như vậy, chúng ta có thể kết luận rằng tập dữ liệu này sử dụng số 0 để biểu diễn dữ liệu bị thiếu hoặc dữ liệu rỗng. Ở đây, chúng ta có thể thấy rằng có tới một nửa số hàng chứa các cột có dữ liệu bị thiếu.

# 

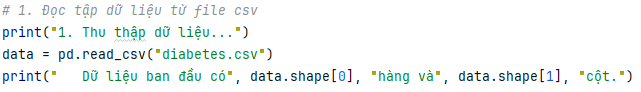
# CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU

## 2.1. Đọc và chuẩn bị dữ liệu:

Trong bước này, dữ liệu được thu thập từ tệp diabetes.csv, chứa thông tin sức khỏe của 768 bệnh nhân nhằm mục tiêu dự đoán nguy cơ mắc bệnh tiểu đường. Tệp dữ liệu này gồm 9 thuộc tính, trong đó thuộc tính Outcome đóng vai trò là biến mục tiêu (1: có tiểu đường, 0: không có tiểu đường).

Phân tích ban đầu: Trước khi xử lý sâu, dữ liệu cần được kiểm tra để hiểu rõ cấu trúc và các thông tin có trong tập dữ liệu. Cụ thể, ta sẽ sử dụng các công cụ như .shape() và countplot để kiểm tra số lượng mẫu và phân bố của các giá trị trong biến mục tiêu.

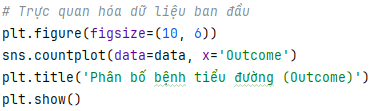
Dữ liệu sẽ được tải vào sử dụng thư viện Pandas và kiểm tra kích thước của nó, nhằm xác định số lượng hàng và cột của dữ liệu.



Kết quả:



Tiếp theo, sử dụng biểu đồ countplot để trực quan hóa phân bố biến mục tiêu Outcome giúp chúng ta đánh giá sự cân bằng giữa các lớp (bị tiểu đường và không bị tiểu đường).





*Biểu đồ countplot phân bố tình trạng mắc bệnh tiểu đường (Outcome)*

Biểu đồ cột cho thấy sự phân bố của biến Outcome, giúp đánh giá độ cân bằng giữa hai nhóm: người bị tiểu đường và không bị tiểu đường.

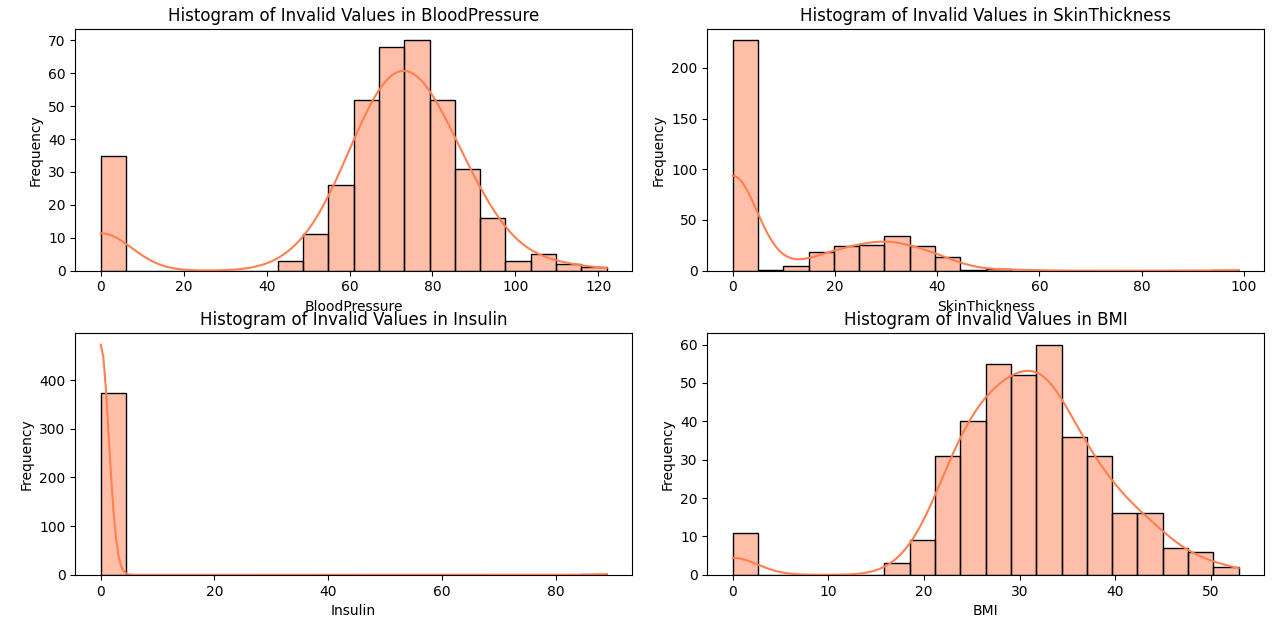
## 2.2. Làm sạch dữ liệu:

### 2.2.1. Loại bỏ các giá trị không hợp lệ:

* **Dữ liệu thiếu:** Là những giá trị bị bỏ trống hoặc không có trong một số ô của tập dữ liệu. Nguyên nhân có thể do lỗi nhập liệu, thu thập dữ liệu không đầy đủ, hoặc lỗi khi tải dữ liệu.
* **Dữ liệu bị trùng lặp:** Là những dòng dữ liệu được lặp lại nhiều lần trong tập dữ liệu. Dữ liệu trùng lặp có thể làm tăng độ nhiễu và gây sai lệch trong mô hình.
* **Dữ liệu phân phối không đồng đều:** Nếu có sự chênh lệch trong phân phối của một hoặc nhiều thuộc tính, có thể ảnh hướng tới hiệu suất của mô hình
* **Giá trị ngoại lai (Outliers):** Là những giá trị nằm cách xa đáng kể so với các giá trị khác trong tập dữ liệu. Ngoại lệ có thể gây nhiễu và làm ảnh hưởng đến độ chính xác của mô hình.
* **Dữ liệu lỗi thời (Outdated Data):** Là những giá trị đã lỗi thời hoặc không còn phản ánh đúng tình trạng thực tế hiện tại. Dữ liệu lỗi thời có thể gây ra những kết luận sai lầm nếu không được cập nhật.

#### 2.2.1.1. Thống kê các dữ liệu không hợp lệ:

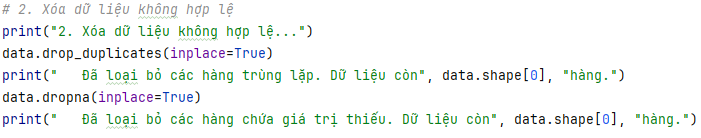
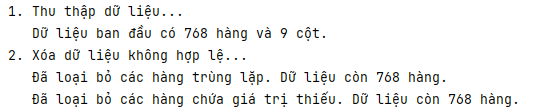
Vẽ các biểu đồ Histogram thống kê các dữ liệu không hợp lệ



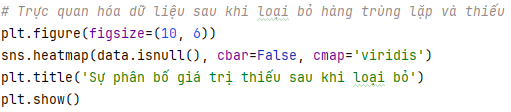
*Biểu đồ Histogram thống kê các dữ liệu không hợp lệ*

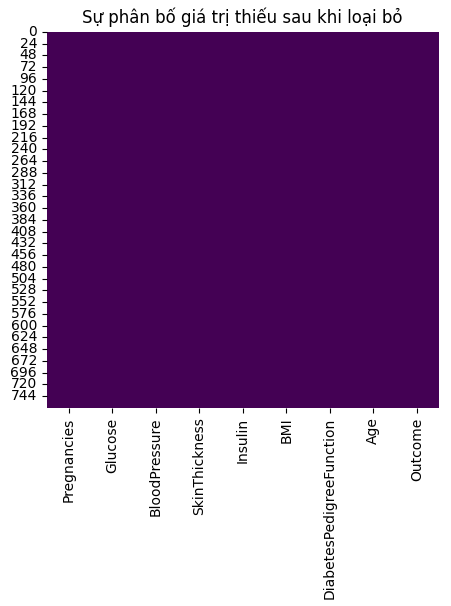
#### 2.2.1.2. Xóa các dữ liệu không hợp lệ:

Chúng em xóa dữ liệu không hợp lệ bằng hàm drop\_duplicates() như sau :



Để kiểm tra dữ liệu thiếu, ta sử dụng một biểu đồ heatmap để minh họa sự phân bố của các giá trị thiếu.





*Biểu đồ heatmap về sự phân bố giá trị thiếu sau khi loại bỏ*

Biểu đồ heatmap cho thấy không còn giá trị thiếu nào trong dữ liệu.

### 2.2.2. Chuẩn hóa dữ liệu:

Cần đảm bảo ràng dữ liệu trong tập dữ liệu được chuẩn hóa và đồng nhất chuyển đổi các giá trị về cùng một đơn vị đo lường, đơn vị tiền tệ hoặc định dạng. Mục tiêu của chuẩn hóa dữ liệu là tạo ra một tập dữ liệu nhất quán, giúp việc phân tích và xử lý trở nên dễ dàng và chính xác hơn.

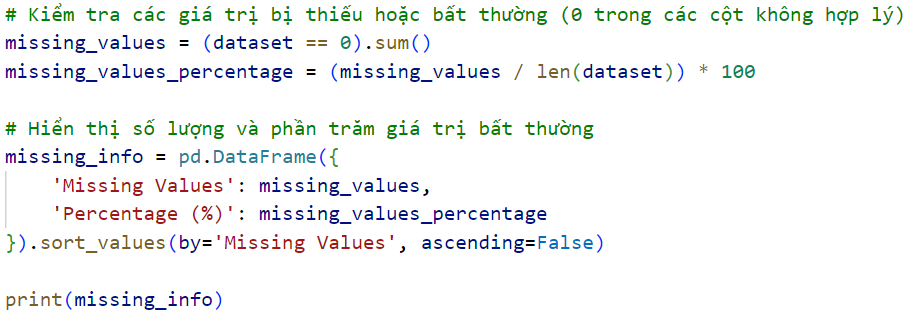
Trong bài toán này, dữ liệu không được chuẩn hóa vì các đặc trưng trong bộ dữ liệu có phạm vi giá trị gần nhau và không có sự chênh lệch lớn về đơn vị đo lường. Cụ thể, các đặc trưng như Glucose, BMI, và Age đều có các giá trị có thể so sánh trực tiếp với nhau mà không gây ảnh hưởng quá lớn đến hiệu quả của mô hình.

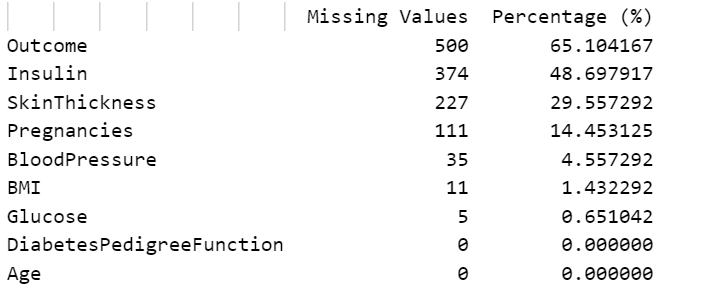
Ngoài ra, một số mô hình sử dụng trong nghiên cứu này, như Random Forest và Decision Tree, không nhạy cảm với việc chuẩn hóa dữ liệu, vì chúng không phụ thuộc vào phạm vi giá trị của các đặc trưng trong quá trình học mô hình. Do đó, không cần thiết phải chuẩn hóa dữ liệu đối với các mô hình này.

### 2.2.3. Xử lý dữ liệu thiếu:

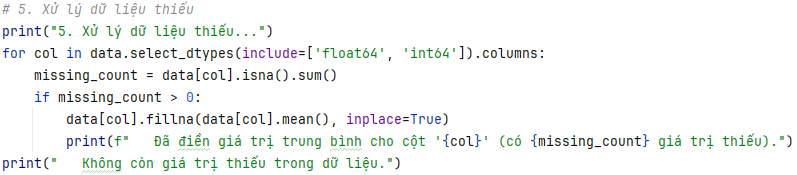
Đảm bảo rằng tập dữ liệu không có lỗ hổng, tránh gây sai lệch cho các phân tích và mô hình sau này. Nếu phát hiện giá trị thiếu trong các cột số học, các giá trị này sẽ được thay thế bằng giá trị trung bình của cột tương ứng để tránh mất mát dữ liệu quan trọng.

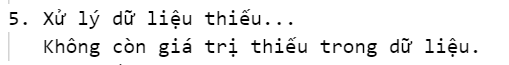
Trong bộ dữ liệu này, các cột như Glucose, BloodPressure, SkinThickness, Insulin, và BMI có thể chứa giá trị 0, điều này không hợp lý về mặt y tế (vì huyết áp hoặc nồng độ glucose không thể bằng 0). Chúng ta cần thay các giá trị 0 trong các cột này thành giá trị NaN để sau đó có thể điền các giá trị thay thế thích hợp.

**Bước 1:** Xác định các giá trị bị thiếu ****

****

**Bước 2:** Điền vào giá trịtrung bình cho các ô trống

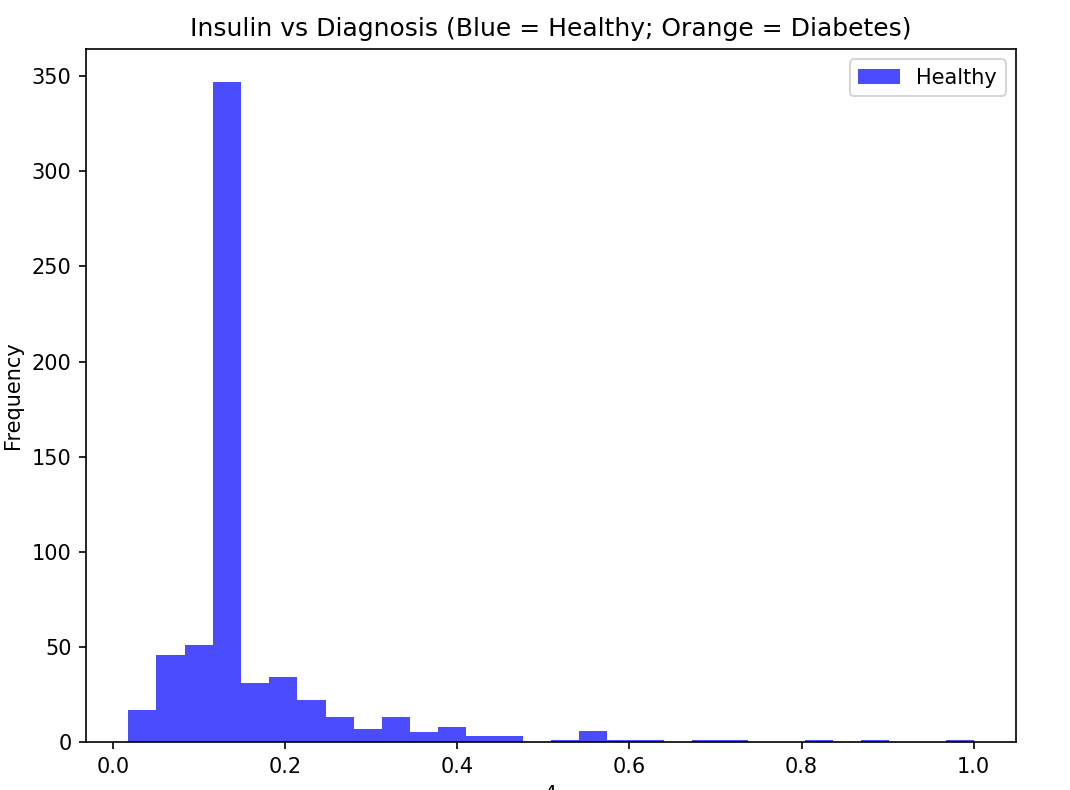




Kiểm tra từng cột dữ liệu dạng số (int64 hoặc float64). Nếu phát hiện cột có giá trị thiếu, các ô trống (NaN) sẽ được điền bằng giá trị trung bình của cột đó. Việc này giúp lấp đầy các giá trị còn thiếu một cách hợp lý mà không cần xóa bỏ dữ liệu.

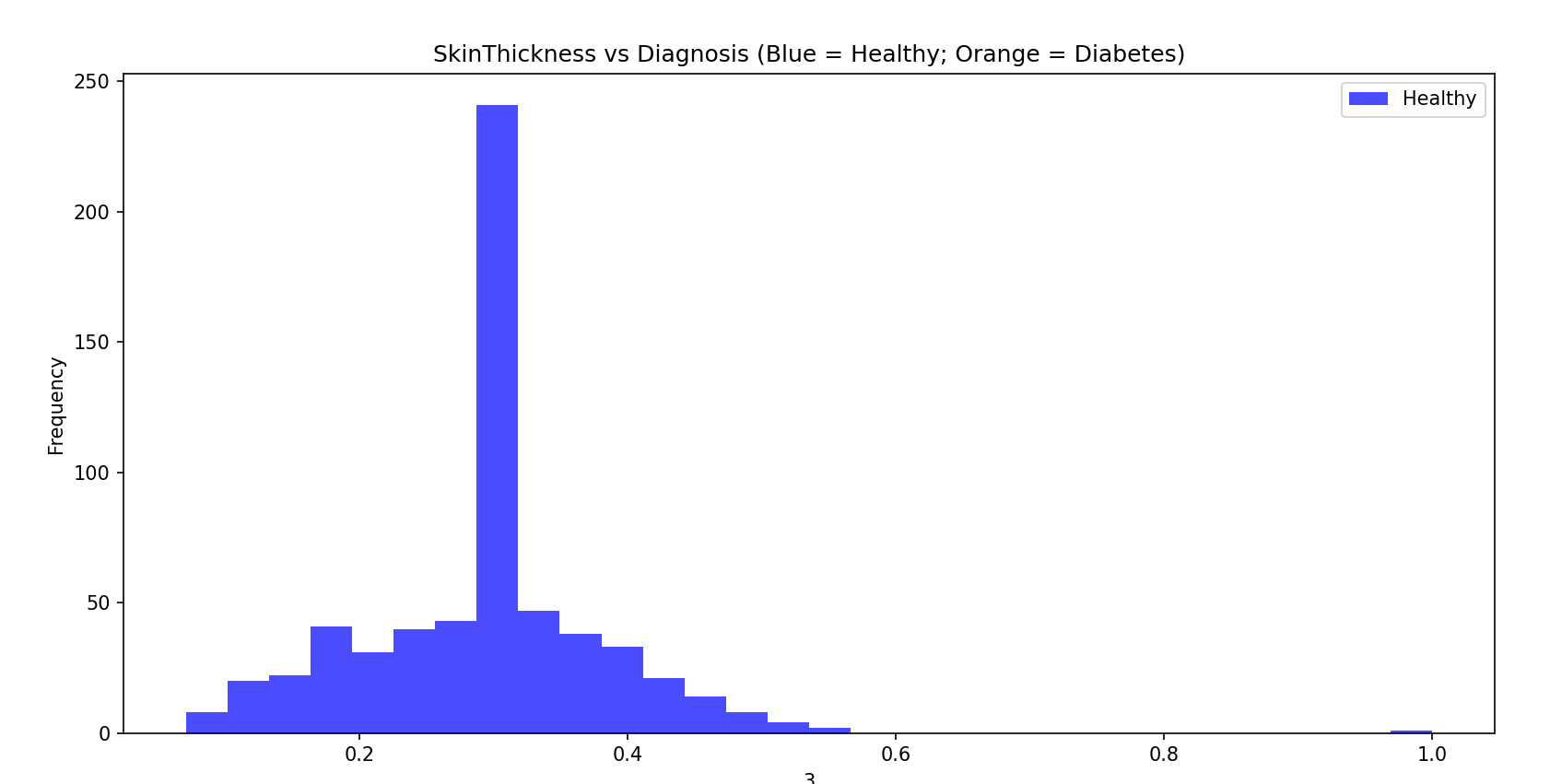
Các phép đo Insulin và Độ dày da không có mối tương quan cao với bất kỳ kết quả cụ thể nào và do đó, chúng ta có thể loại trừ mối lo ngại về rò rỉ dữ liệu. Tuy nhiên, giá trị 0 trong các danh mục này vẫn có sai sót và do đó không nên đưa vào mô hình của chúng tôi. Vì vậy, thay thế các giá trị này bằng một số phân bố giá trị gần với số đo trung vị là tốt nhất. Và lưu ý rằng tốt nhất nên áp dụng các giá trị này sau hàm train\_test\_split để ngăn chặn một dạng rò rỉ dữ liệu khác (tức là không nên sử dụng dữ liệu thử nghiệm khi tính giá trị trung bình để sử dụng trong quá trình quy định).

Biểu đồ sau đây minh họa rằng các giá trị null thực sự đã được thay thế bằng các giá trị trung bình.



*Biểu đồ phân phối mức Insulin theo tình trạng bệnh tiểu đường*

**Biểu đồ Insulin**: Ở biểu đồ này có thể thấy phân phối của mức Insulin trong dữ liệu, cho biết mức độ phổ biến của các giá trị Insulin khác nhau trong tập huấn luyện. Nếu có sự phân biệt rõ giữa các bệnh nhân mắc bệnh tiểu đường và người khỏe mạnh, biểu đồ này có thể giúp bạn quan sát được xu hướng phân phối của giá trị Insulin.

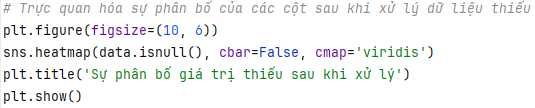


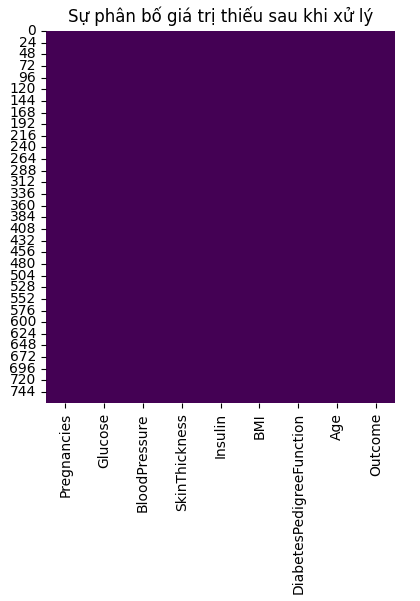
*Biểu đồ phân phối độ dày da (SkinThickness) sau khi xử lý giá trị thiếu*

**Biểu đồ SkinThickness**: Tương tự, biểu đồ này giúp bạn quan sát sự phân bố của độ dày da, một yếu tố có thể liên quan đến bệnh tiểu đường.

Sau bước này, dữ liệu không còn chứa giá trị thiếu. Tập dữ liệu đầy đủ và có tính toàn vẹn hơn, giúp đảm bảo rằng phân tích sau này sẽ chính xác và mô hình hóa không bị gián đoạn bởi các ô trống.

Một lần nữa sử dụng heatmap để kiểm tra dữ liệu thiếu sau khi xử lý.





*Biểu đồ heatmap về sự phân bố giá trị thiếu sau khi loại bỏ*

Biểu đồ heatmap xác nhận không còn giá trị thiếu.

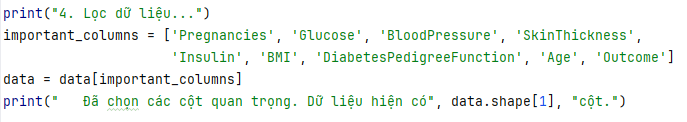
## 2.3. Chuyển đổi và biến đổi dữ liệu:

Trong giai đoạn này, chúng ta thực hiện các phép biến đổi dữ liệu cần thiết để làm cho dữ liệu sẵn sàng cho các phân tích tiếp theo hoặc sử dụng trong mô hình học máy. Chuyển đổi và biến đổi dữ liệu là một phần quan trọng trong quá trình tiền xử lý để cải thiện chất lượng dữ liệu và đảm bảo dữ liệu ở dạng thích hợp cho các thuật toán phân tích.

### 2.3.1. Lọc dữ liệu:

Lọc dữ liệu là quá trình chọn ra những cột và bản ghi quan trọng, phù hợp với mục tiêu phân tích hoặc mô hình hóa. Trong trường hợp này, chúng ta chọn lọc các cột quan trọng liên quan đến bệnh tiểu đường từ bộ dữ liệu "diabetes.csv".

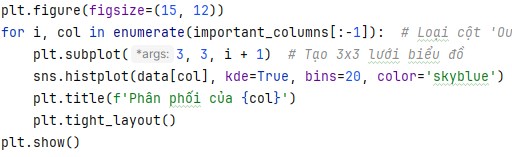
Cách lọc được thực hiện qua đoạn mã sau:

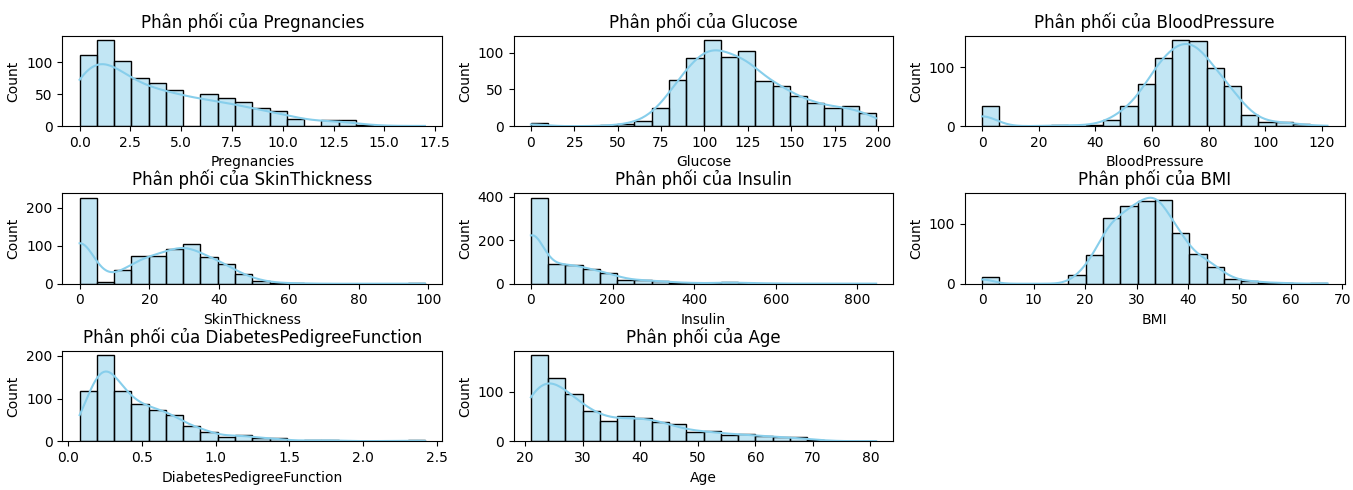


Sau khi thực hiện, dữ liệu chỉ còn lại các cột quan trọng này, giúp giảm thiểu sự phức tạp và tập trung vào các yếu tố có ảnh hưởng trực tiếp đến việc chẩn đoán bệnh tiểu đường.



Để xác nhận rằng các cột quan trọng được giữ lại, một biểu đồ heatmap hiển thị ma trận tương quan giữa các biến số được sử dụng. Điều này giúp xác định các mối quan hệ tiềm năng giữa các thuộc tính và biến mục tiêu.





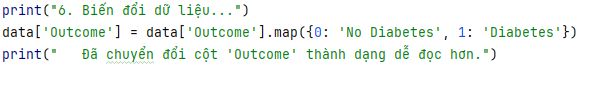
*Biểu đồ heatmap ma trận tương quan giữa các biến số*

### 2.3.2. Biến đổi dữ liệu:

Biến đổi dữ liệu là quá trình chuyển đổi các giá trị trong dữ liệu thành dạng dễ hiểu hoặc dễ sử dụng cho các phân tích tiếp theo. Trong quá trình này, một cột cụ thể đã được biến đổi:

* **Outcome:** Cột này chứa giá trị nhị phân (0 hoặc 1), với 0 đại diện cho không có bệnh tiểu đường và 1 đại diện cho có bệnh tiểu đường. Chúng ta đã chuyển đổi các giá trị này thành dạng văn bản dễ đọc hơn:
  + 0 → 'No Diabetes'
  + 1 → 'Diabetes'

Đoạn mã thực hiện biến đổi dữ liệu:



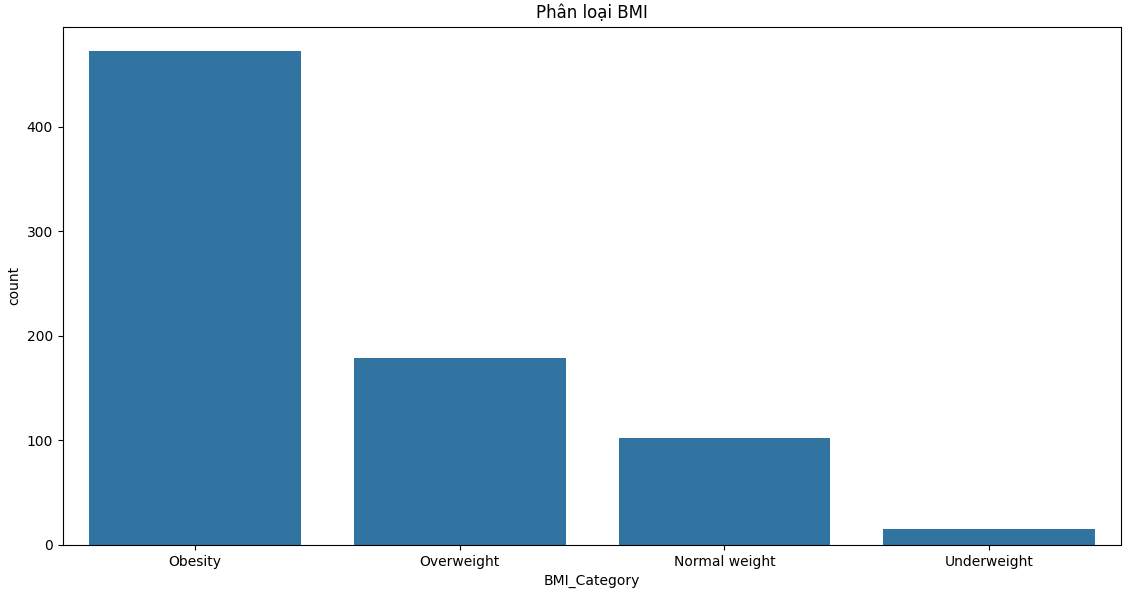
Sau khi thực hiện, cột Outcome sẽ dễ đọc và dễ hiểu hơn đối với người sử dụng hoặc các mô hình phân tích sau này.



## 2.4. Tạo đặc trưng và dữ liệu mới:

Sau khi thực hiện các bước làm sạch và biến đổi dữ liệu, giai đoạn tiếp theo là tạo ra các đặc trưng (features) mới, giúp làm phong phú bộ dữ liệu và hỗ trợ cho các mô hình phân tích hoặc học máy.

Dựa trên giá trị BMI của bệnh nhân, một cột mới BMI\_Category được tạo ra để phân loại bệnh nhân vào các nhóm như "Underweight" (Thiếu cân), "Normal weight" (Cân nặng bình thường), "Overweight" (Thừa cân), và "Obesity" (Béo phì). Cột này cung cấp thông tin chi tiết hơn về trạng thái sức khỏe của bệnh nhân và có thể hữu ích trong phân tích.



*Biểu đồ phân bố nhóm BMI của bệnh nhân*

Biểu đồ sẽ cho thấy số lượng bệnh nhân rơi vào từng nhóm BMI. Thông qua biểu đồ này, chúng ta có thể nhanh chóng nhận ra nhóm nào chiếm ưu thế trong dữ liệu, và nếu có sự mất cân đối trong các nhóm BMI, ví dụ, nếu nhóm "Underweight" có số lượng quá ít hoặc "Obesity" quá nhiều.

# 

# CHƯƠNG 3: PHÂN TÍCH VÀ KHÁM PHÁ DỮ LIỆU

## 3.1. Các phương pháp phân tích sử dụng:

Trong quá trình phân tích dữ liệu, việc lựa chọn phương pháp phân tích phù hợp là rất quan trọng để đạt được kết quả chính xác và có giá trị. Đối với bài toán dự đoán bệnh tiểu đường, chúng tôi sử dụng một số phương pháp phân tích dữ liệu phổ biến, bao gồm các kỹ thuật học máy (machine learning) và thống kê, để khám phá các yếu tố ảnh hưởng đến nguy cơ mắc bệnh và xây dựng mô hình dự đoán.

* **Phân tích thống kê mô tả.**

Đây là phương pháp cơ bản giúp mô tả các đặc trưng của dữ liệu, chẳng hạn như độ lệch chuẩn, trung bình, và các phân phối của các biến số quan trọng như mức độ glucose, huyết áp, và chỉ số BMI. Mục đích là hiểu rõ các đặc điểm của dữ liệu và xác định các giá trị bất thường hoặc thiếu sót.

* **Các thuật toán học máy.**

Áp dụng các mô hình học máy như cây quyết định, rừng ngẫu nhiên, K-nearest neighbors (KNN) để xây dựng các mô hình phân loại nhằm dự đoán nguy cơ mắc bệnh tiểu đường. Các mô hình này có khả năng tự học từ dữ liệu và tạo ra những dự đoán chính xác về tình trạng bệnh của người dùng.

* **Phân tích mối quan hệ giữa các yếu tố.**

Phân tích này giúp chúng ta khám phá các mối liên hệ giữa các yếu tố khác nhau như chỉ số glucose, huyết áp, BMI và nguy cơ mắc bệnh tiểu đường. Các phương pháp phân tích tương quan và hồi quy sẽ được sử dụng để đánh giá mức độ ảnh hưởng của các yếu tố này.

* **Kiểm định mô hình và đánh giá hiệu quả.**

Các mô hình học máy sau khi được huấn luyện sẽ được kiểm tra và đánh giá với các chỉ số như độ chính xác (accuracy), độ nhớ (recall), độ chính xác của lớp dương tính (precision) và điểm F1 (F1-score) để đánh giá hiệu suất của mô hình.

Để đánh giá hiệu suất của mô hình ML, độ đo cho bài toán chúng em sử

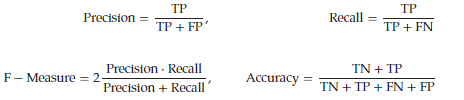
dụng confusion matrix là một cách bố trí bảng cụ thể cho phép trực quan hóa

hiệu suất của thuật toán học có giám sát. Mỗi hàng của ma trận biểu thị các thể

hiện trong lớp ‘thực tế’ trong khi mỗi cột biểu thị các thể hiện trong lớp ‘dự

đoán’.Confusion matrix có các chỉ số đánh giá sau :precision, recall, accuracy, FMeasure và AUC.Cụ thể:

* Accuracy tóm tắt hiệu suất của nhiệm vụ phân loại và đo lường số lượng trường hợp được dự đoán chính xác trong số tất cả các trường hợp dữ liệu.
* Recall ghi lại tỷ lệ các trường hợp mắc bệnh tiểu đường và được phân loại chính xác là có mắc bệnh tiểu dường hay không.
* Precision cho biết có baonhiêu người được chẩn đoán mắc bệnh CKD thuộc nhóm này.
* F-Measure là giá trị trung bình hài hòa của precision và recall .Các số liệu được xác định như sau



Trong đó:

* True Positive (TP): Số lượng dự đoán đúng là lớp dương (1).
* True Negative (TN): Số lượng dự đoán đúng là lớp âm (0).
* False Positive (FP): Số lượng dự đoán sai là lớp dương (mô hình dự đoán 1 nhưng thực tế là 0).
* False Negative (FN): Số lượng dự đoán sai là lớp âm (mô hình dự đoán 0 nhưng thực tế là 1).

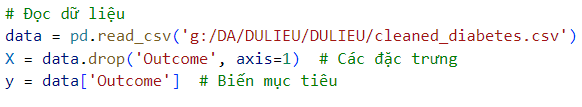
## 3.2. Phân tích mối quan hệ giữa các yếu tố ảnh hưởng đến bệnh tiểu đường:

Nhóm chúng em đã thay thế tất cả các giá trị sai, bị thiếu và không có giá trị bằng các giá trị trung bình nên hiện tại tập dữ liệu đã sẵn sàng đào tạo và đánh giá các mô hình của mình để dự đoán bệnh tiểu đường.

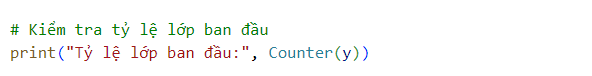
Trong bài toán phân loại bệnh tiểu đường, việc đánh giá hiệu quả của các mô hình phân loại là rất quan trọng để xác định mô hình nào hoạt động tốt nhất với tập dữ liệu. Để làm điều này, em đã xây dựng một hàm chung evaluate\_model để đánh giá hiệu quả của mô hình dựa trên các chỉ số như độ chính xác (accuracy), precision, recall và f1-score.

**Các bước chuẩn bị dữ liệu:**

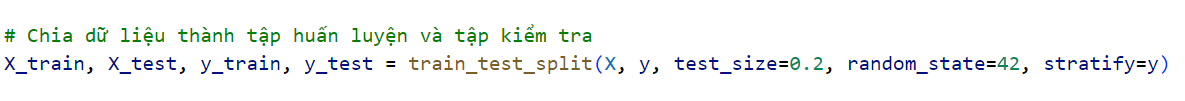
* Chuẩn bị dữ liệu: Tập dữ liệu được chia thành hai phần: đặc trưng (X) và biến mục tiêu (y). Biến mục tiêu (y) chứa thông tin về kết quả của bệnh tiểu đường (0: không bị bệnh, 1: bị bệnh), trong khi các đặc trưng (X) là các yếu tố đầu vào như glucose, BMI, độ tuổi, v.v.



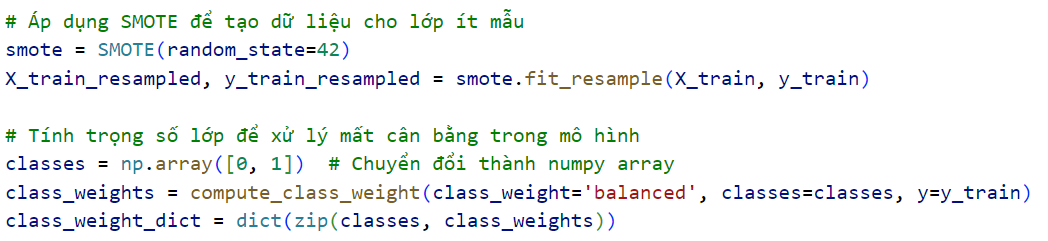
* Kiểm tra tỉ lệ lớp: Trước khi chia tập dữ liệu, ta kiểm tra tỷ lệ giữa các lớp trong biến mục tiêu để đảm bảo tính cân đối của dữ liệu.



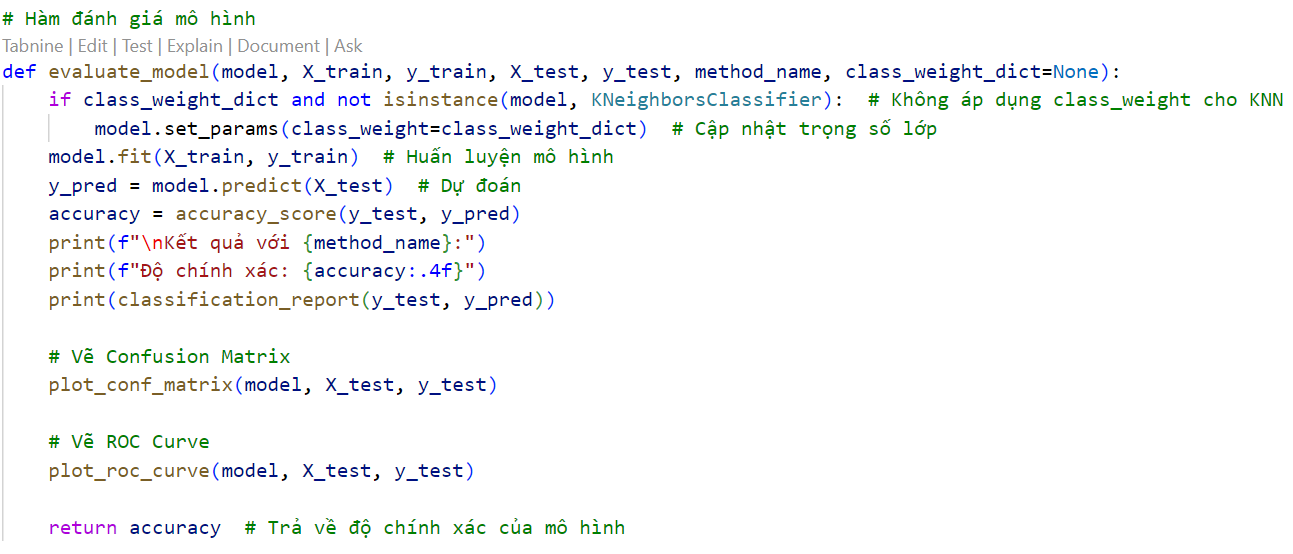
* Chia dữ liệu: Dữ liệu được chia thành tập huấn luyện và tập kiểm tra với tỷ lệ 80% huấn luyện và 20% kiểm tra, sử dụng train\_test\_split. Tùy chọn stratify=y giúp giữ tỷ lệ giữa các lớp trong cả hai tập.



Trong tập dữ liệu ban đầu, tỷ lệ giữa các lớp không đồng đều (lớp "Không mắc bệnh" thường chiếm ưu thế so với lớp "Mắc bệnh"). Điều này dẫn đến mô hình thiên vị, thường dự đoán chính xác hơn cho lớp chiếm ưu thế nhưng bỏ qua lớp thiểu số quan trọng. Vì vậy để giảm thiểu sự chênh lệch dữ liệu giữa hai lớp 0 và 1, chúng em đã sử dụng SMOTE để tạo ra các mẫu tổng hợp mới cho lớp thiểu số bằng cách nội suy giữa các điểm dữ liệu hiện có. Điều này giúp cân bằng tỷ lệ giữa các lớp, cải thiện khả năng học của mô hình với lớp thiểu số (lớp bệnh nhân mắc bệnh). Và kết quả là SMOTE cải thiện độ “Recall” và “F1\_score cho lớp thiểu số, đảm bảo rằng mô hình không bỏ sót các trường hợp quan trọng, đặc biệt trong bài toán y tế, nơi việc nhận diện đúng bệnh nhân là rất quan trọng. Bên cạnh đó, sử dụng thêm kỹ thuật Trọng số lớp (Class Weight) để bổ sung cho SMOTE nhằm cân bằng dữ liệu và trọng số lớp để tối ưu hóa mô hình sẽ đảm bảo mô hình đạt độ chính xác cao cho cả hai lớp, đặc biệt là lớp bệnh nhân mắc bệnh.



**Hàm Đánh Giá Mô Hình**

Hàm evaluate\_model được sử dụng để huấn luyện và đánh giá các mô hình phân loại. Hàm này nhận vào mô hình, dữ liệu huấn luyện và kiểm tra, sau đó tính toán độ chính xác và các chỉ số đánh giá khác như precision, recall, và f1-score.

Trong quá trình phân tích và đánh giá hiệu quả của các mô hình học máy trong việc dự đoán bệnh tiểu đường, chúng em sẽ sử dụng một số loại biểu đồ để giúp trực quan hóa và hiểu rõ hơn về hiệu suất của các mô hình.

* **Confusion Matrix (Ma trận nhầm lẫn)**: Là công cụ quan trọng giúp đánh giá khả năng phân loại của mô hình. Nó cho thấy số lượng dự đoán đúng và sai của các lớp, từ đó giúp đánh giá mức độ chính xác trong việc phân loại bệnh nhân mắc và không mắc bệnh. Các giá trị chính trong confusion matrix bao gồm số lượng đúng/ sai đối với lớp "0" (Không mắc bệnh) và lớp "1" (Mắc bệnh), giúp hiểu rõ hơn về khả năng phân loại của mô hình đối với các lớp này.
* **ROC Curve (Đường cong ROC)**: Giúp đánh giá hiệu quả của mô hình phân loại qua các tỷ lệ dương tính giả (False Positive Rate - FPR) và tỷ lệ dương tính thật (True Positive Rate - TPR). Đường cong này cho phép đánh giá khả năng phân biệt giữa các lớp của mô hình ở các ngưỡng phân loại khác nhau. AUC (Area Under the Curve) là một chỉ số quan trọng trong ROC Curve, cho biết mức độ chính xác của mô hình, với giá trị càng cao (gần 1) chứng tỏ mô hình càng tốt.
* **Scatter plot (biểu đồ phân tán)**: Được sử dụng để biểu diễn mối quan hệ giữa hai hoặc nhiều biến, giúp xác định các xu hướng, tương quan (thuận, nghịch, hoặc không tương quan) và phát hiện các giá trị bất thường (outliers) trong dữ liệu. Ngoài ra, biểu đồ này hỗ trợ trực quan hóa sự phân bố của dữ liệu và đánh giá hiệu quả của các mô hình phân loại hoặc dự đoán.

Các biểu đồ này không chỉ giúp dễ dàng nhận diện mô hình nào hoạt động tốt hơn mà còn cung cấp những thông tin chi tiết về cách mỗi mô hình xử lý các lớp khác nhau trong bài toán phân loại, từ đó đưa ra quyết định về mô hình phù hợp nhất cho bài toán dự đoán bệnh tiểu đường.

### 3.2.1.Hồi quy Logistic (Logistic Regression)

Logistic Regression là một mô hình phân loại trong học máy (machine learning), thường được sử dụng để giải quyết các bài toán phân loại nhị phân (binary classification). Dù tên gọi là "hồi quy" (regression), nhưng Logistic Regression thực chất lại là một thuật toán phân loại, không phải dự báo số liệu liên tục như hồi quy tuyến tính.

Logistic Regression (hay còn gọi là Logit Regression) được sử dụng phổ biến để ước lượng xác suất 1 điểm dữ liệu có thể thuộc về 1 lớp nào đó (ví dụ tính xác suất để 1 email là spam). Nếu xác suất > 50% thì mô hình dự đoán có thể khẳng định được điểm dữ liệu đó thuộc về lớp 1 (nhãn là 1) hoặc ngược lại (nhãn là 0). Việc này được gọi là phân loại nhị phân (chỉ có 0 với 1)

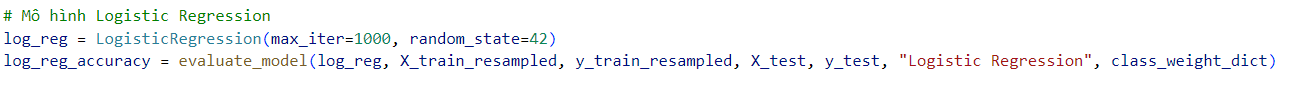
Hồi quy Logistic dùng để mô hình hóa mối quan hệ giữa biến phụ thuộc (y) và các biến độc lập (X). Mục tiêu là tính toán xác suất mà một đối tượng thuộc vào một lớp nhất định. Công thức của hàm sigmoid (logistic function) được sử dụng trong hồi quy logistic như sau:

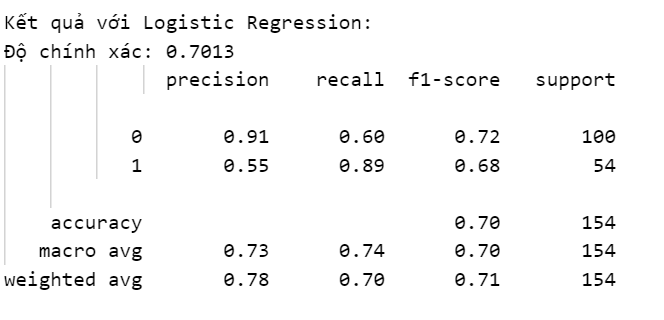


Trong đó:

* *P(y=1∣X)* là xác suất mà đối tượng thuộc lớp 1.
* *z=β0+β1X1+β2X2+⋯+βnXn*​ là hàm tuyến tính, với *β* là các hệ số (parameters) cần được học trong quá trình huấn luyện.

**Bước 1:**Thực hiện tạo và huấn luyện mô hình Logistic Regression trên dữ liệu huấn luyện, dự đoán giá trị trên tập kiểm tra:

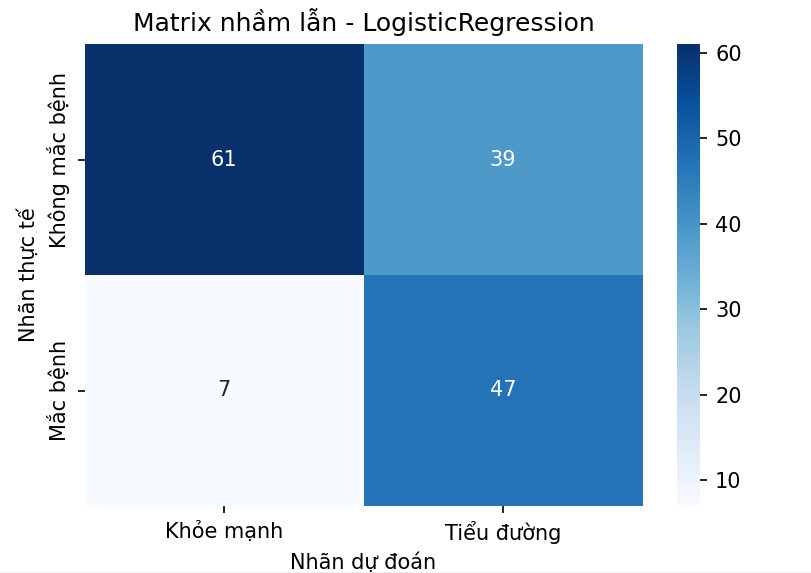




**Phân tích** **Kết quả :**

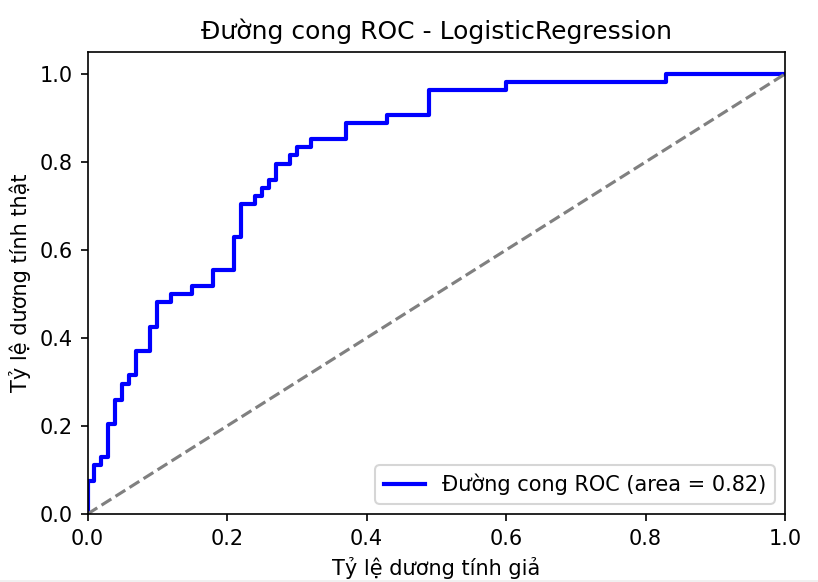
* Mô hình Logistic Regression đạt độ chính xác 0.7013
* Precision cho lớp 0 khá cao (0.91), cho thấy mô hình có khả năng phân loại tốt cho người không mắc bệnh.
* Tuy nhiên, recall cho lớp 0 chỉ đạt 0.60, cho thấy mô hình còn bỏ sót khá nhiều trường hợp không mắc bệnh. Mặc dù recall cho lớp 1 (mắc bệnh) cao (0.89), nhưng precision cho lớp 1 chỉ đạt 0.55, điều này có nghĩa là mô hình đôi khi nhận diện sai những trường hợp không mắc bệnh là mắc bệnh.

**Bước 2:** Vẽ sơ đồ Ma trận nhầm lẫn và đường cong ROC cho mô hình Logistic Regression



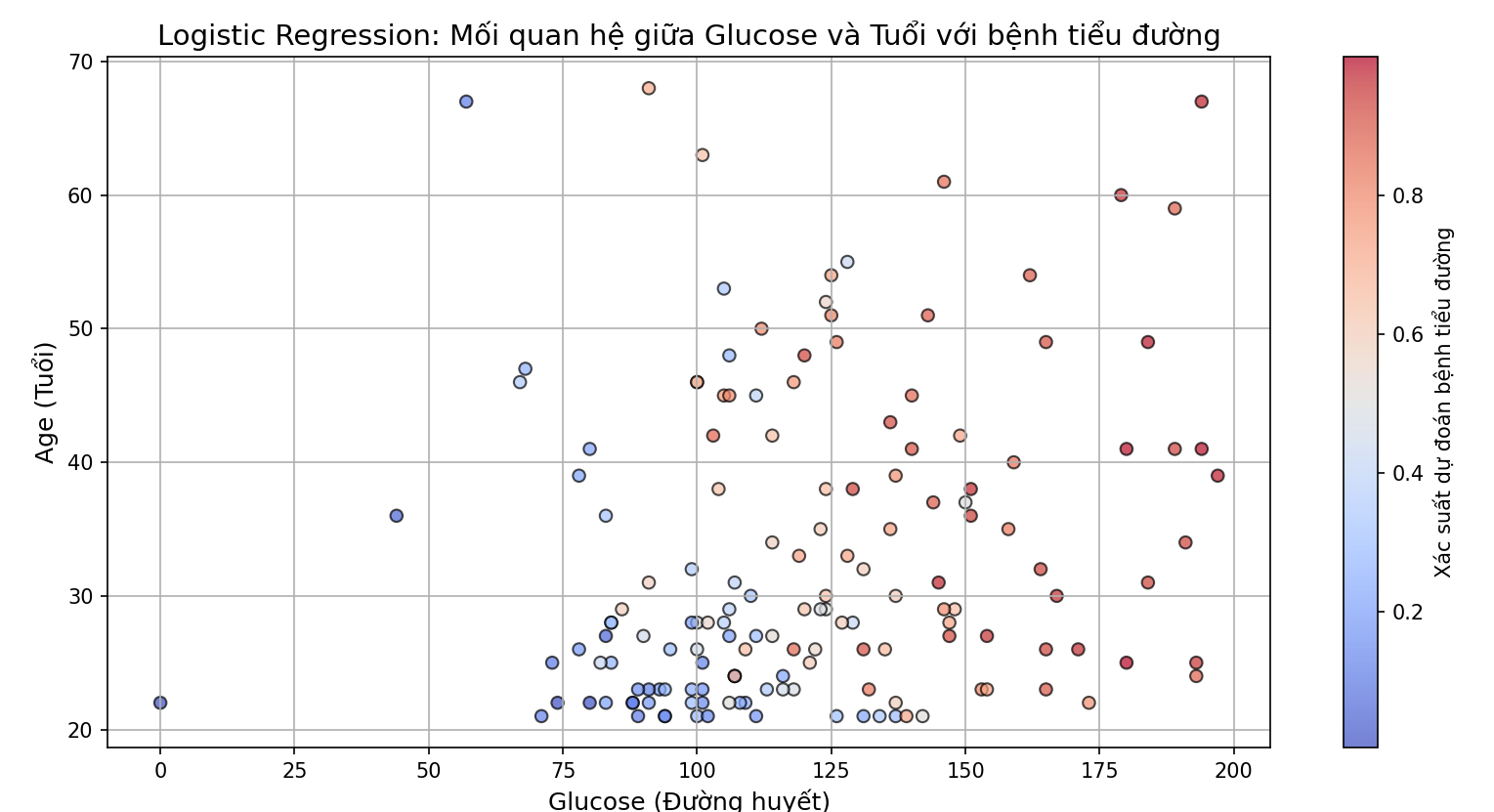
*Biểu đồ ma trận nhầm lẫn Logistic Regression*

Logistic Regression cho thấy sự cân bằng tốt giữa việc dự đoán đúng các lớp 0 (Negative) và lớp 1 (Positive), với số lượng True Positive (TP) và True Negative (TN) cao. Điều này cho thấy mô hình này hoạt động khá hiệu quả trong việc đạt được sự cân bằng giữa precision và recall.

**

*Biểu đồ Đường cong ROC Logistic Regression*

**Bước 3:** Vẽ biểu đồ Scatter plot cho mô hình *Logistic Regression*

**

*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình Logistic Regression*

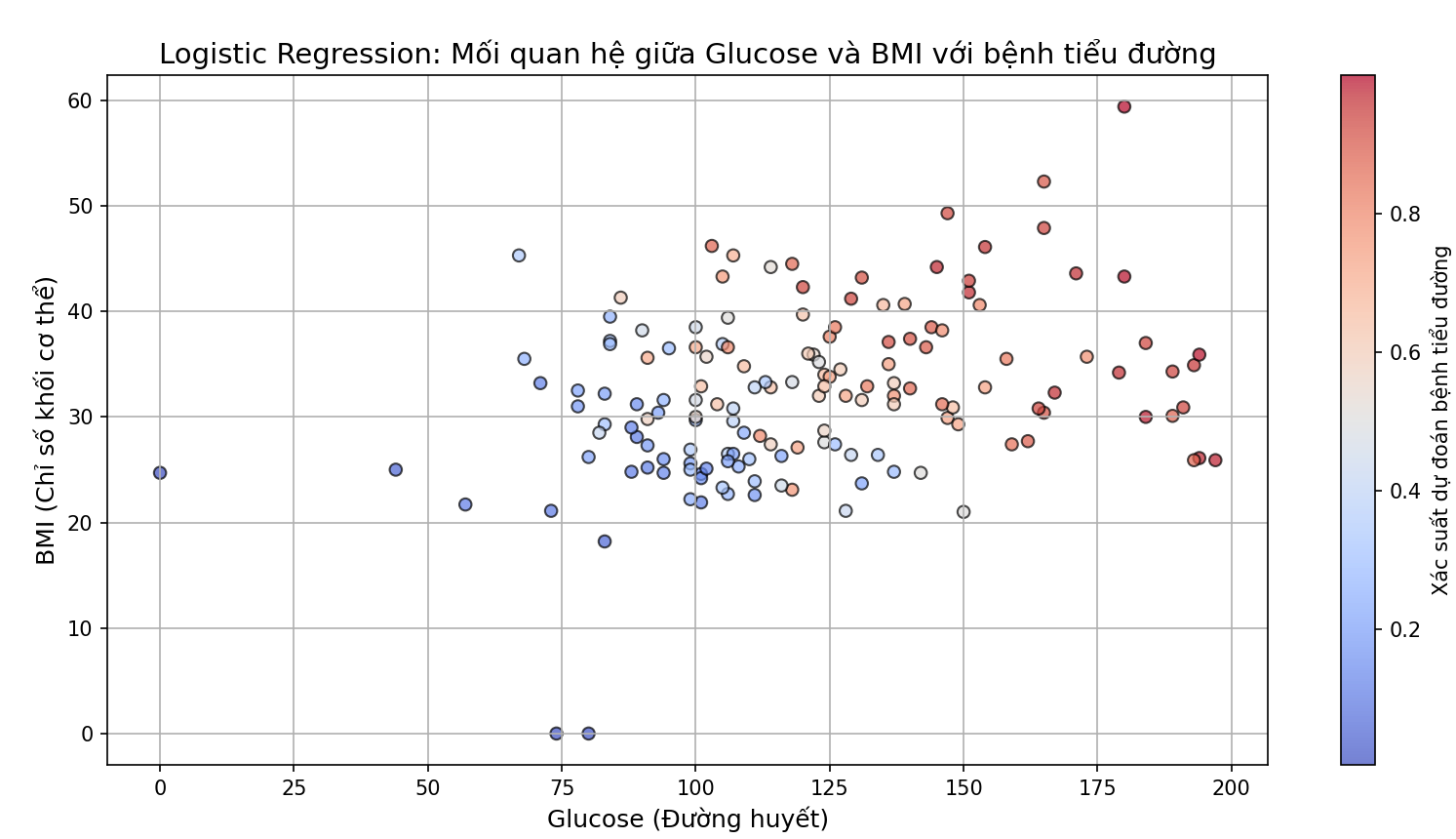
**Trục x (Glucose - Đường huyết):**Giá trị tăng từ 0 đến 1, biểu diễn mức đường huyết của các cá nhân. Mức đường huyết cao hơn thường liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn.

**Trục y (Age - Tuổi):**Giá trị tăng từ 0 đến 0.8 (đã được chuẩn hóa), biểu thị độ tuổi của cá nhân.

* Các điểm màu đỏ thể hiện xác suất dự đoán cao của bệnh tiểu đường.
* Các điểm màu xanh biểu thị xác suất thấp hơn.

Từ biểu đồ Scatter plot mô hình Logistic Regression chúng em nhận thấy:

* Mức đường huyết (Glucose) có mối tương quan rõ ràng với nguy cơ mắc bệnh tiểu đường: khi mức đường huyết cao hơn, xác suất mắc bệnh tiểu đường tăng lên.
* Tuổi (Age) cũng có ảnh hưởng nhất định: người lớn tuổi có xu hướng dễ mắc tiểu đường hơn, đặc biệt khi mức đường huyết cũng cao.



*Biểu đồ Scatter Mối quan hệ giữa Glucose và BMI với bệnh tiểu đường - mô hình Logistic Regression*

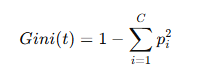
### 3.2.2. Cây quyết định (Decision Tree)

Cây quyết định là một kỹ thuật học máy giám sát, được sử dụng để phân loại bệnh tiểu đường dựa trên các đặc trưng của bệnh nhân (như Glucose, BloodPressure, BMI). Phương pháp này sẽ chia dữ liệu thành các nhóm con dựa trên các đặc trưng, qua đó giúp xác định các yếu tố quan trọng nhất trong việc dự đoán bệnh tiểu đường.

Cây quyết định là cây mà mỗi nút biểu diễn một đặc trưng(tính chất), mỗi nhánh(branch) biểu diễn một quy luật(rule) và mỗi lá biểu diễn một kết quả (giá trị cụ thể hay một nhánh tiếp tục)

Cây quyết định phân chia dữ liệu dựa trên các giá trị của các đặc trưng để phân loại các đối tượng. Việc phân chia được thực hiện dựa trên các tiêu chí như Gini Index hoặc Entropy.

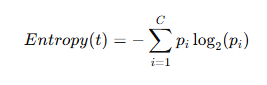
**Gini Index:**

****

Trong đó:

* *C* là số lớp trong bài toán.
* *pi*​ là tỷ lệ các đối tượng thuộc lớp i trong tập con *t.*

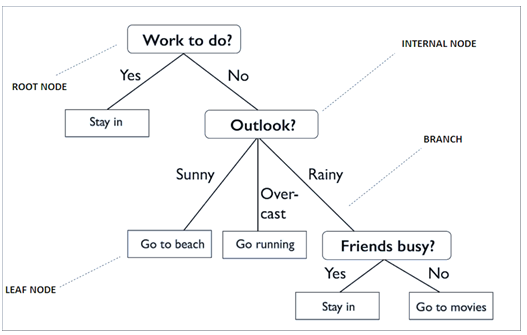
**Entropy:**



Trong đó:

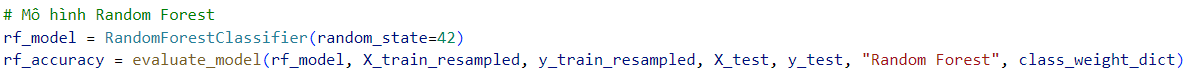
* *pi*​ là xác suất của các đối tượng thuộc lớp *i* trong tập con *t*.
* Entropy nằm trong khoảng (0,1)

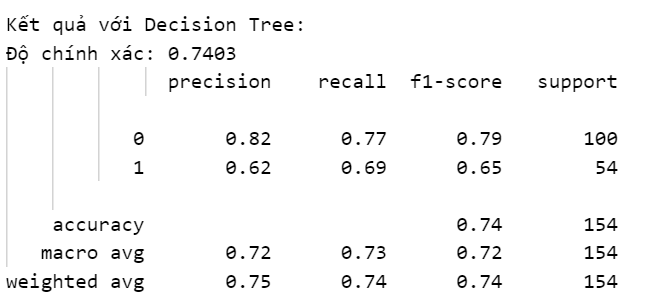
Cây quyết định sẽ tìm kiếm các giá trị đặc trưng để phân chia sao cho giá trị **Gini** hoặc **Entropy** là nhỏ nhất, giúp cải thiện tính đồng nhất trong mỗi nhóm con.



*Hình ảnh trực quan hóa cây quyết định*

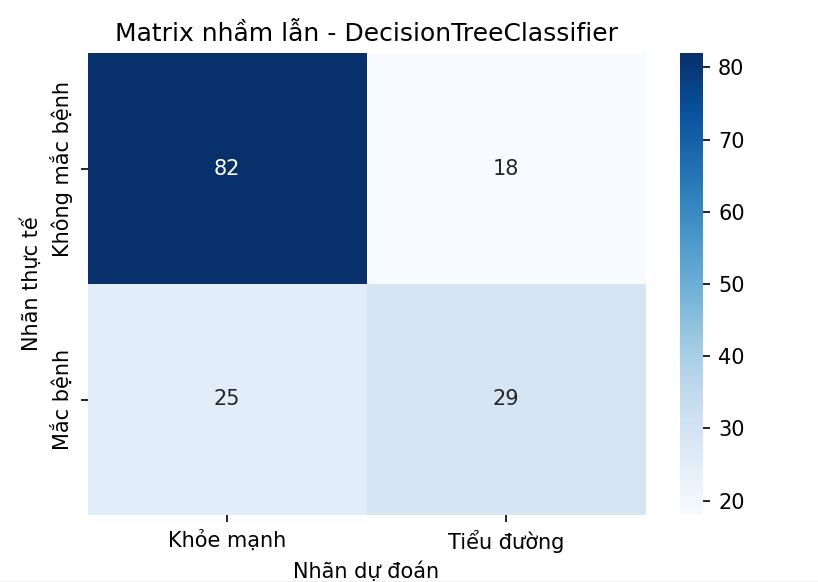
**Bước 1:**Thực hiện tạo và huấn luyện mô hình Cây quyết định trên dữ liệu huấn luyện, dự đoán giá trị trên tập kiểm tra:





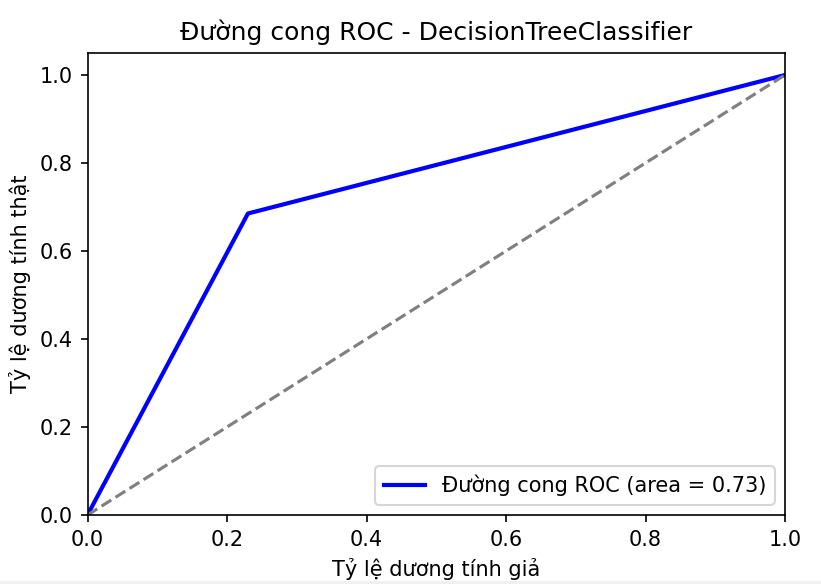
*Kết quả về độ chính xác và các chỉ số của mô hình Cây quyết định*

* Mô hình Decision Tree đạt độ chính xác 0.7403
* Precision cho lớp 0 là 0.82 và recall 0.77, cho thấy mô hình này khá tốt trong việc phân loại người không mắc bệnh.
* Precision cho lớp 1 là 0.62 và recall là 0.69, chứng tỏ mô hình này có hiệu quả trong việc nhận diện những người mắc bệnh tiểu đường. Tuy nhiên, so với Logistic Regression, Decision Tree có sự cân bằng tốt hơn giữa precision và recall cho cả hai lớp.

**Bước 2:** Vẽ sơ đồ Ma trận nhầm lẫn và đường cong ROC cho mô hình Decision Tree

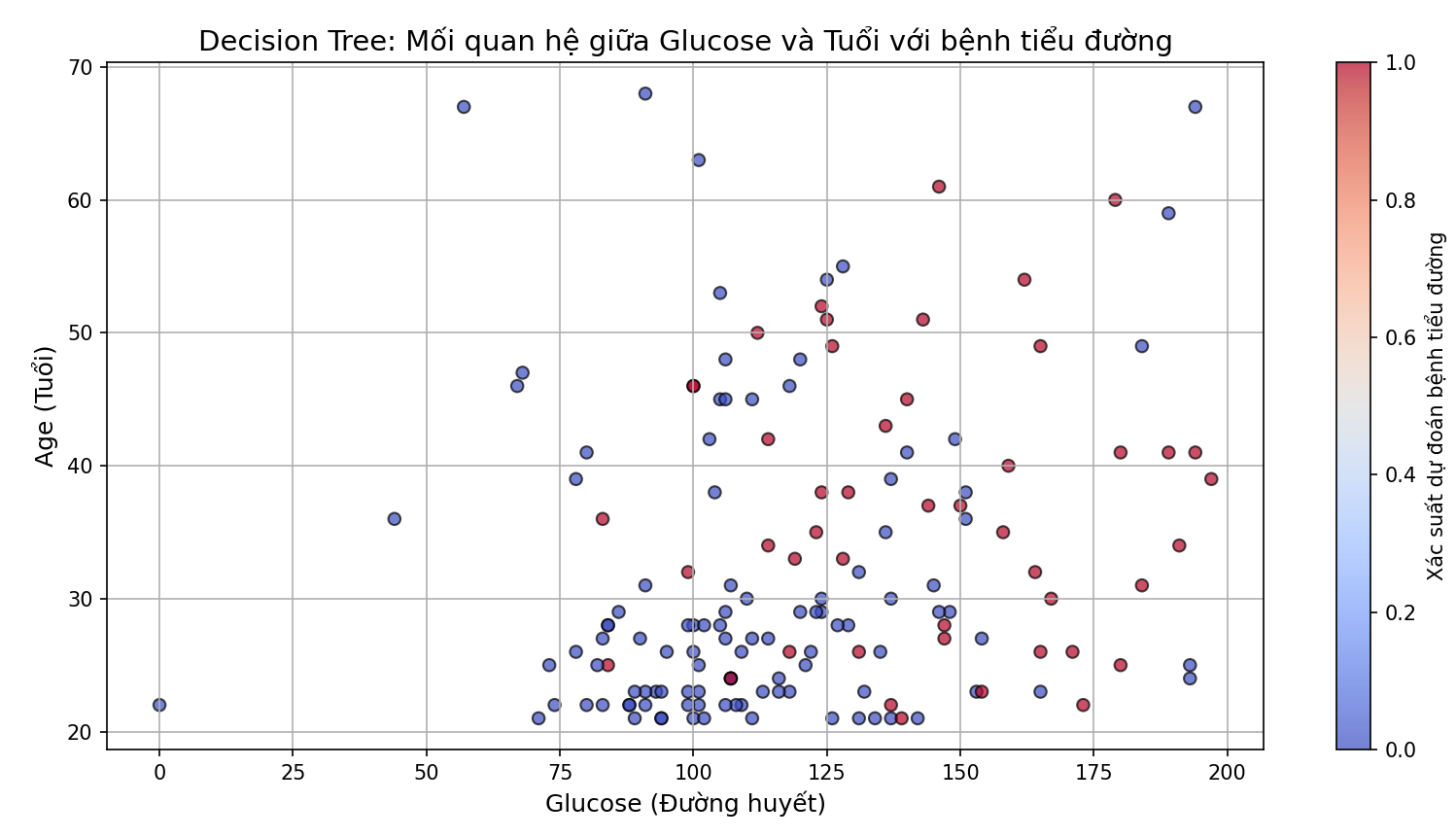
*Biểu đồ Ma trận nhầm lẫn Decision Tree*

Decision Tree nổi bật với khả năng nhận diện chính xác các trường hợp thuộc lớp 1 (cao nhất về TP), nhưng lại gặp vấn đề với số lượng False Positive (FP) cao, dẫn đến độ chính xác của lớp 0 bị ảnh hưởng. Mặc dù có khả năng nhận diện bệnh tiểu đường tốt (lớp 1), nhưng mô hình này lại dễ dàng bị đánh lừa khi phân loại các trường hợp không mắc bệnh (lớp 0).



*Biểu đồ Đường cong ROC Decision Tree*

**Bước 3:** Vẽ biểu đồ Scatter plot cho mô hình *Decision Tree*

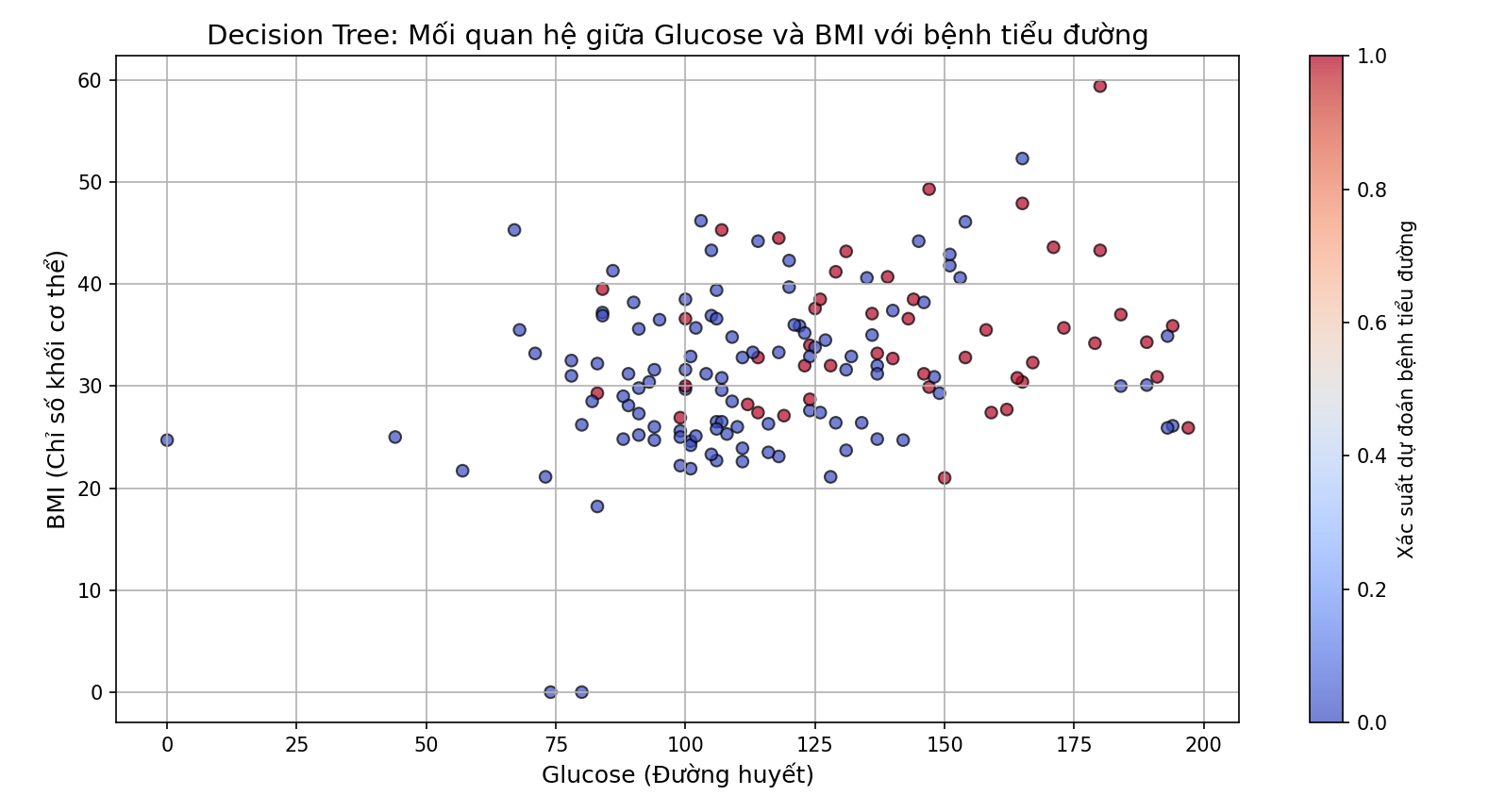
**

*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình Decision Tree*

**Trục x (Glucose - Đường huyết):**Giá trị tăng từ 0 đến 1, biểu diễn mức đường huyết của các cá nhân. Mức đường huyết cao hơn thường liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn.

**Trục y (Age - Tuổi):**Giá trị tăng từ 0 đến 0.8 (đã được chuẩn hóa), biểu thị độ tuổi của cá nhân.

* Các điểm màu đỏ thể hiện xác suất dự đoán cao của bệnh tiểu đường.
* Các điểm màu xanh biểu thị xác suất thấp hơn.



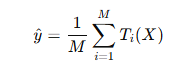
*Biểu đồ Scatter Mối quan hệ giữa Glucose và BMI với bệnh tiểu đường - mô hình Decision Tree*

### 3.2.3. Rừng ngẫu nhiên (Random Forest)

Rừng ngẫu nhiên là một thuật toán học máy mạnh mẽ, hoạt động bằng cách tạo ra nhiều cây quyết định và kết hợp kết quả của chúng. Phương pháp này giúp cải thiện độ chính xác và giảm thiểu hiện tượng overfitting (quá khớp) so với cây quyết định đơn lẻ. Với dữ liệu bệnh tiểu đường, rừng ngẫu nhiên có thể giúp phát hiện các yếu tố quan trọng hơn trong việc dự đoán bệnh.

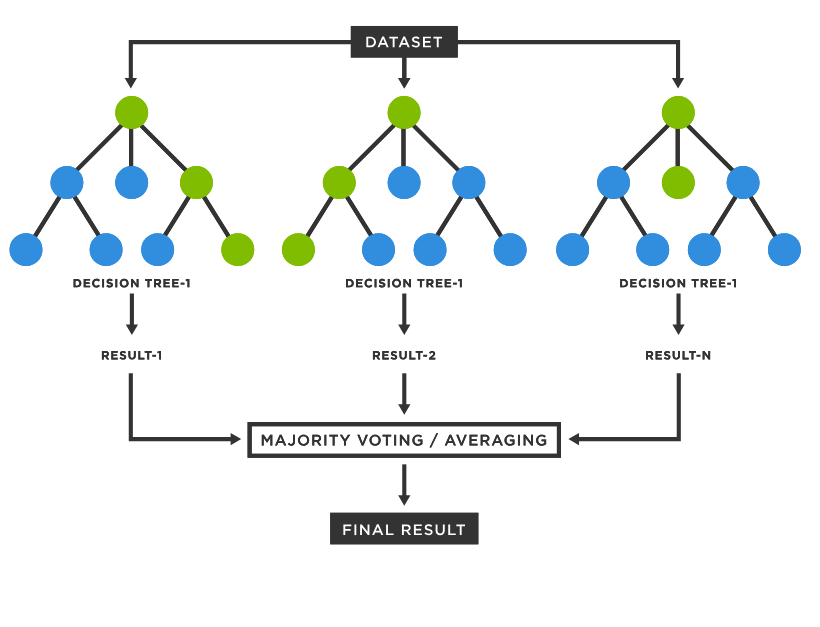
Mỗi cây trong rừng được huấn luyện trên một mẫu ngẫu nhiên với việc chọn các đặc trưng ngẫu nhiên. Để dự đoán, các cây trong rừng sẽ đưa ra quyết định và kết quả cuối cùng là sự kết hợp của các quyết định đó.

Công thức tính tổng quát cho một rừng ngẫu nhiên:



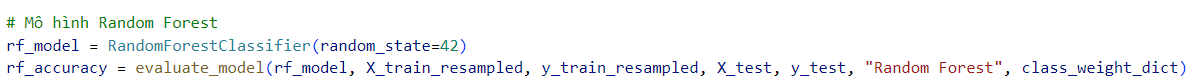
Trong đó:

* *M* là số cây trong rừng.
* *Ti(X)* là dự đoán của cây *i* cho đối tượng *X*.

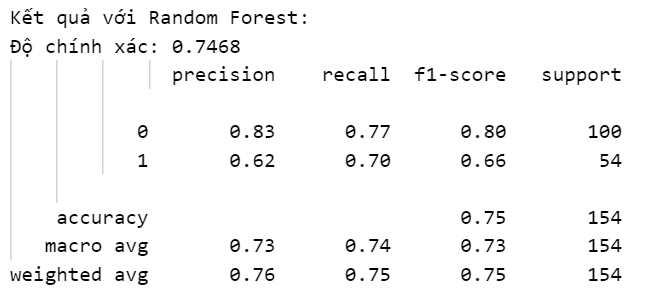


*Hình ảnh trực quan hóa rừng ngẫu nhiên*

**Bước 1:**Thực hiện tạo và huấn luyện mô hình Random Forest trên dữ liệu huấn luyện, dự đoán giá trị trên tập kiểm tra:



Kết quả:



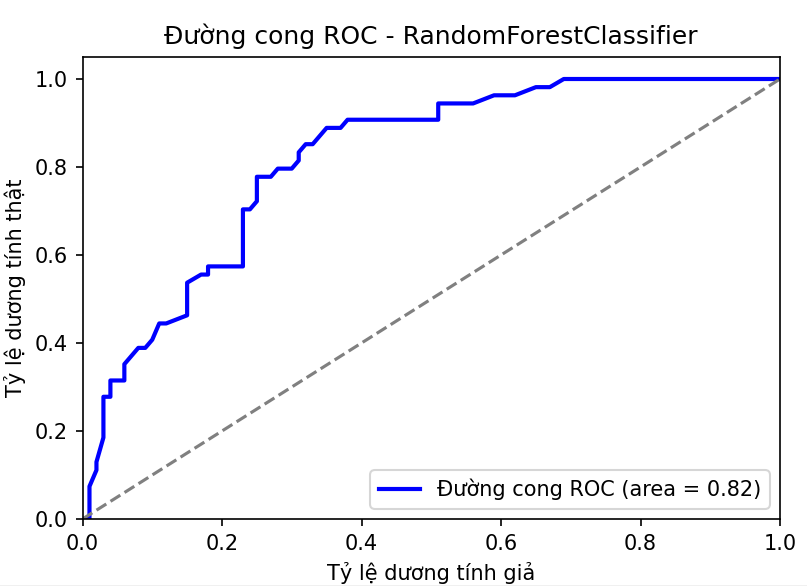
* Random Forest đạt độ chính xác cao nhất trong số các mô hình thử nghiệm với 0.7468.
* Precision cho lớp 0 là 0.83 và recall là 0.77, cho thấy mô hình này hoạt động rất tốt trong việc phân loại người không mắc bệnh. Dù precision cho lớp 1 không cao (0.62)
* Recall của lớp 1 đạt 0.70, cho thấy Random Forest có thể nhận diện tốt hơn những người mắc bệnh tiểu đường so với các mô hình trước. Mô hình này có sự cân bằng khá tốt giữa precision và recall cho cả hai lớp.

**Bước 2:** Vẽ sơ đồ Ma trận nhầm lẫn và đường cong ROC cho mô hình Rừng ngẫu nhiên



*Biểu đồ Ma trận nhầm lẫn Random Forest*

Random Forest cho thấy sự cân bằng tốt giữa việc dự đoán đúng các lớp 0 (Negative) và lớp 1 (Positive), với số lượng True Positive (TP) và True Negative (TN) cao. Điều này cho thấy mô hình này hoạt động khá hiệu quả trong việc đạt được sự cân bằng giữa precision và recall.



*Biểu đồ Đường cong ROC Random Forest*

**Bước 3:** Vẽ biểu đồ Scatter plot cho mô hình Rừng ngẫu nhiên



*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình Random Forest*

**Trục x (Glucose - Đường huyết):**Giá trị tăng từ 0 đến 1, biểu diễn mức đường huyết của các cá nhân. Mức đường huyết cao hơn thường liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn.

**Trục y (Age - Tuổi):**Giá trị tăng từ 0 đến 0.8 (đã được chuẩn hóa), biểu thị độ tuổi của cá nhân.

* Các điểm màu đỏ thể hiện xác suất dự đoán cao của bệnh tiểu đường.
* Các điểm màu xanh biểu thị xác suất thấp hơn.



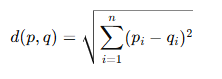
*Biểu đồ Scatter Mối quan hệ giữa Glucose và BMI với bệnh tiểu đường - mô hình Random Forest*

### 3.2.4. K láng giềng gần nhất (K-Nearest Neighbors)

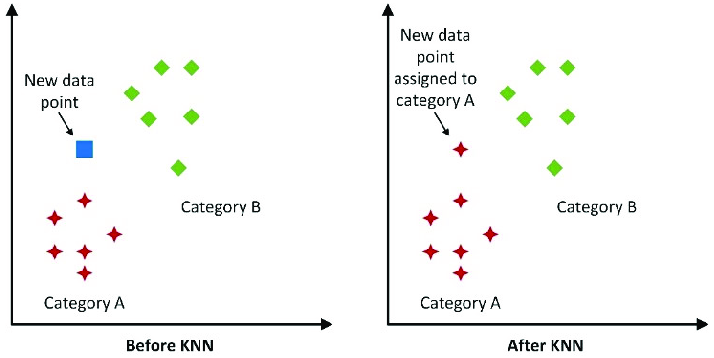
Thuật toán KNN sử dụng thông tin từ các đối tượng tương tự để dự đoán kết quả. Trong bài toán dự đoán bệnh tiểu đường, chúng ta có thể áp dụng KNN để phân loại bệnh nhân là có hay không mắc bệnh tiểu đường dựa trên các đặc trưng của họ (như tuổi tác, BMI, mức độ glucose trong máu). Phương pháp này dễ dàng điều chỉnh và có thể đạt được kết quả tốt với các tập dữ liệu nhỏ và không phức tạp.

Dựa trên nguyên lý rằng các điểm dữ liệu gần nhau có xu hướng thuộc cùng một lớp. Để phân loại một điểm, thuật toán tính khoảng cách giữa điểm đó và các điểm trong tập huấn luyện, sau đó chọn *k* điểm gần nhất để đưa ra dự đoán.

Công thức tính khoảng cách Euclidean giữa hai điểm *p=(p1,p2,...,pn)* và *q=(q1,q2,...,qn)* là:



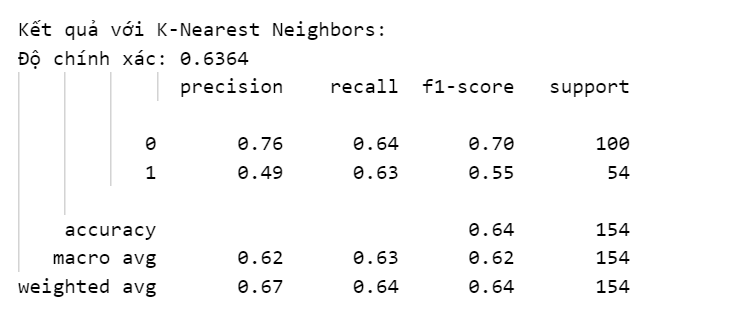
Sau khi tính được khoảng cách, KNN sẽ chọn k điểm gần nhất và đưa ra quyết định dựa trên sự đa số của các lớp của những điểm này.



*Hình ảnh trực quan hóa K láng giềng gần*

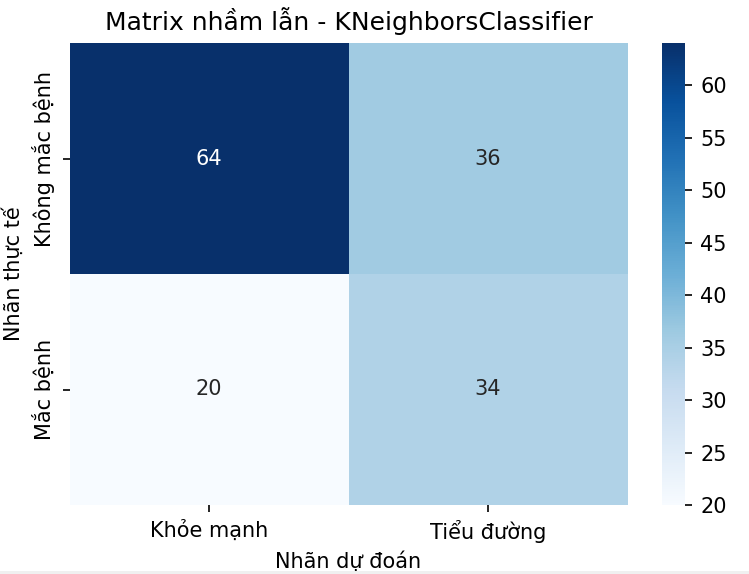
**Bước 1:** Thực hiện tạo và huấn luyện mô hình K-Nearest Neighbors trên dữ liệu huấn luyện, dự đoán giá trị trên tập kiểm tra:





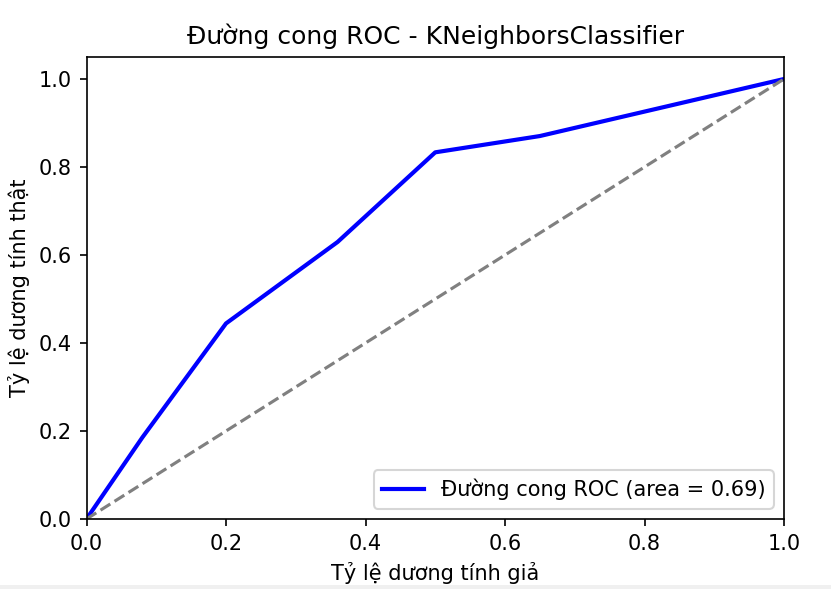
* **KNN** có độ chính xác thấp trong các mô hình với 0.6364.
* Mặc dù precision cho lớp 0 đạt 0.76, recall cho lớp 0 chỉ đạt 0.64, cho thấy mô hình gặp khó khăn trong việc phân loại chính xác các trường hợp không mắc bệnh.
* Precision và recall cho lớp 1 cũng không tốt, lần lượt chỉ đạt 0.49 và 0.63, cho thấy KNN không thể nhận diện hiệu quả nhóm người mắc bệnh tiểu đường.

**Bước 2:** Vẽ sơ đồ Ma trận nhầm lẫn và đường cong ROC cho mô hình KNN



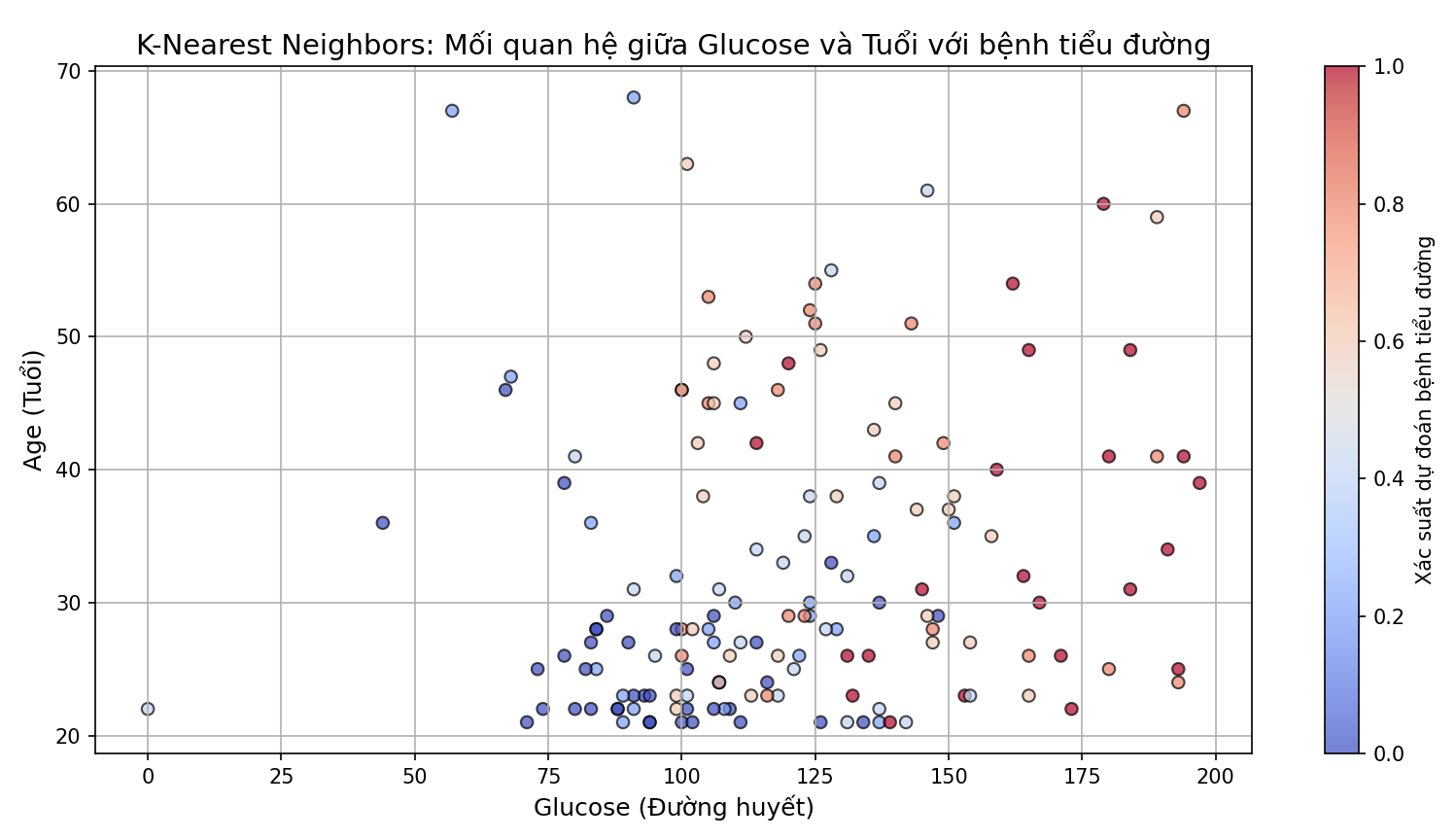
*Biểu đồ Ma trận nhầm lẫn K-Nearest Neighbors*

* K-Nearest Neighbors (KNN) có hiệu suất thấp nhất trong các mô hình thử nghiệm, với số lượng False Negative (FN) và False Positive (FP) cao. Điều này cho thấy KNN không hiệu quả trong việc phân loại các lớp đúng, đặc biệt là với các trường hợp mắc bệnh tiểu đường.



*Biểu đồ Đường cong ROC K-Nearest Neighbors*

**Bước 3:** Vẽ biểu đồ Scatter plot cho mô hình *K-Nearest Neighbors*

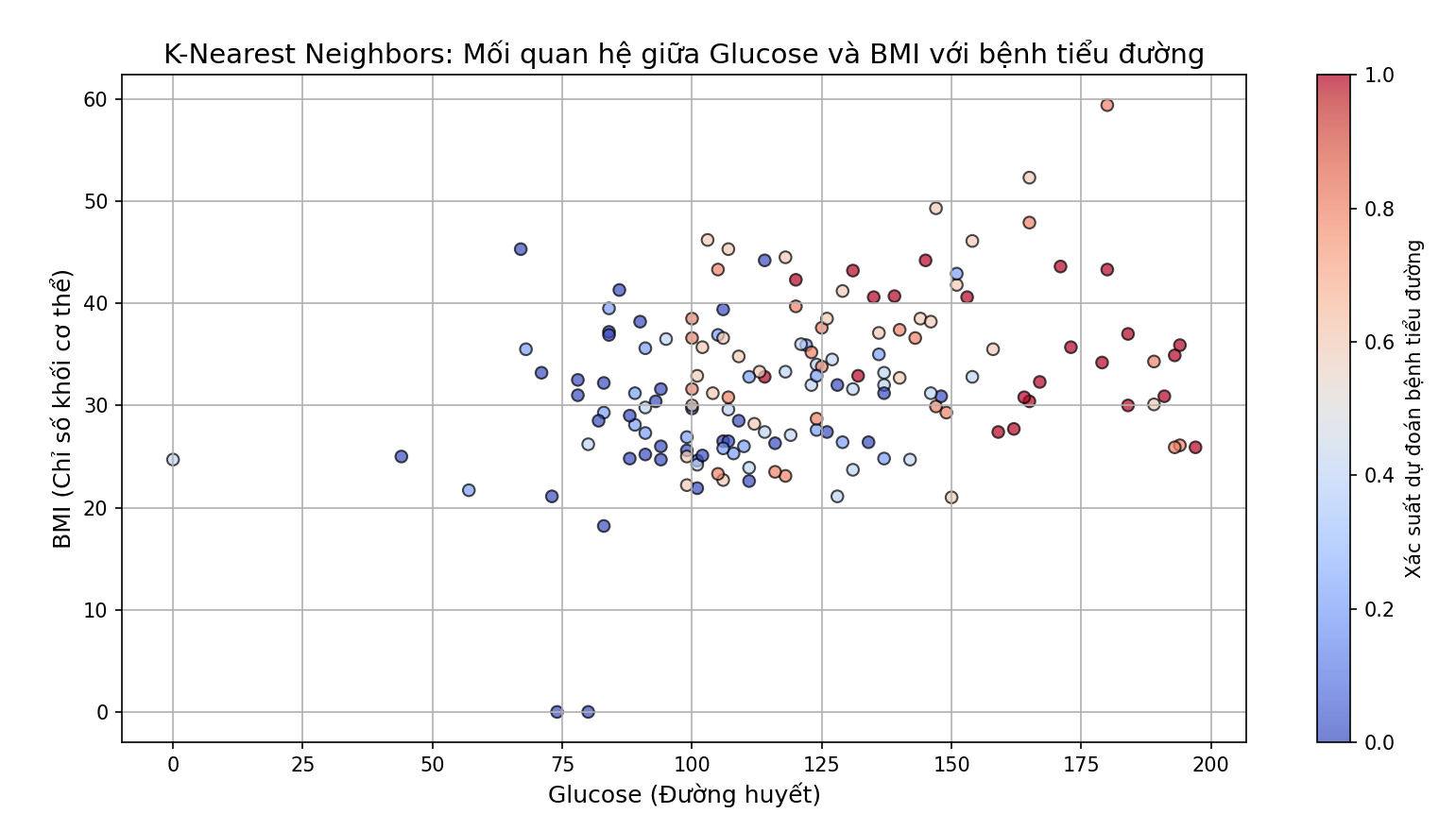
**

*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình K-Nearest Neighbors*

**Trục x (Glucose - Đường huyết):**Giá trị tăng từ 0 đến 1, biểu diễn mức đường huyết của các cá nhân. Mức đường huyết cao hơn thường liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn.

**Trục y (Age - Tuổi):**Giá trị tăng từ 0 đến 0.8 (đã được chuẩn hóa), biểu thị độ tuổi của cá nhân.

* Các điểm màu đỏ thể hiện xác suất dự đoán cao của bệnh tiểu đường.
* Các điểm màu xanh biểu thị xác suất thấp hơn.



*Biểu đồ Scatter Mối quan hệ giữa Glucose và BMI với bệnh tiểu đường - mô hình K-Nearest Neighbors*

### 3.2.5. Máy vector hỗ trợ (Support Vector Regression)

Máy vector hỗ trợ (SVM) là một kỹ thuật mạnh mẽ trong việc phân loại. Với bệnh tiểu đường, SVM có thể tìm ra đường biên tối ưu để phân tách nhóm người mắc bệnh tiểu đường và không mắc bệnh dựa trên các đặc trưng quan sát được (như nồng độ glucose, huyết áp, và BMI). SVM có khả năng làm việc tốt với các dữ liệu có chiều cao và phân bố không đồng đều.

Là thuật toán phân loại tìm một siêu phẳng tối ưu để phân tách các lớp trong không gian đặc trưng sao cho khoảng cách từ các điểm gần nhất của mỗi lớp đến siêu phẳng (được gọi là margin) là lớn nhất.

Công thức của SVM với các điểm dữ liệu *xi* có nhãn *yi∈{−1,1}* là:



Mục tiêu của SVM là tối thiểu hóa hàm mục tiêu sau:

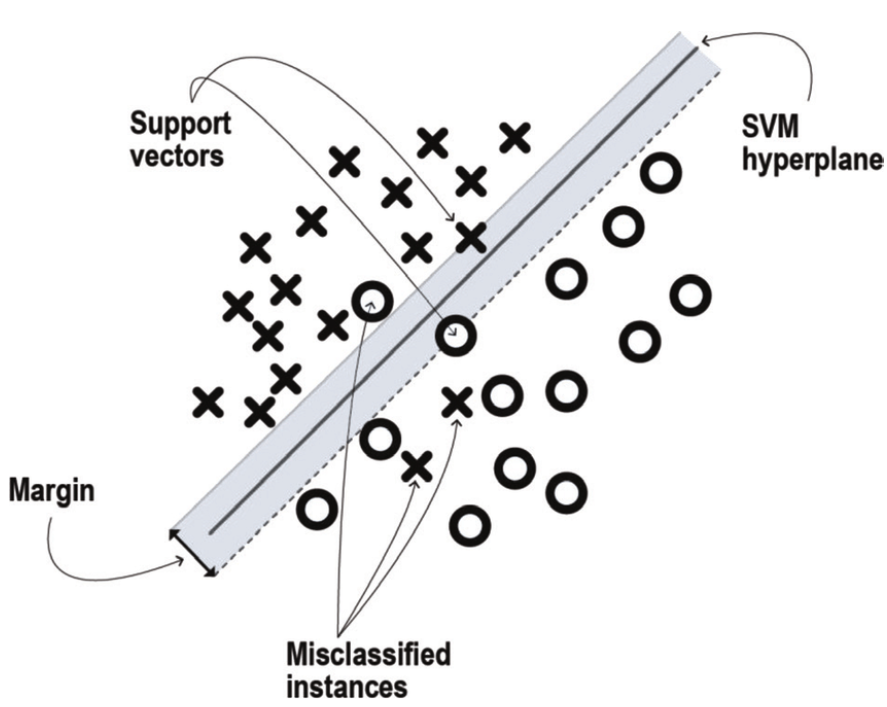


với ràng buộc:



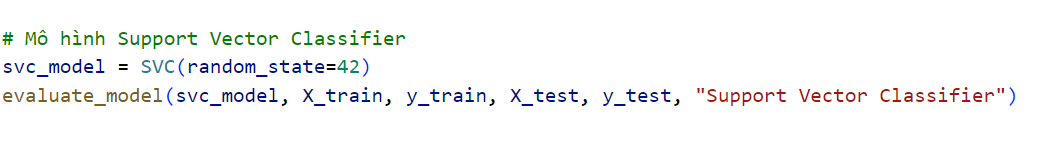
Trong đó:

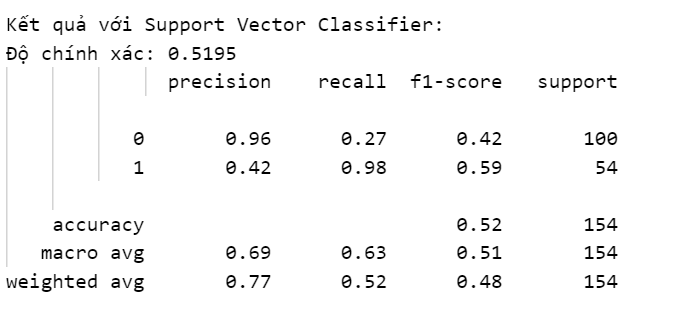
* **w** là trọng số (weights) của siêu phẳng.
* *b* là độ chệch (bias).
* *xi​* là đặc trưng của điểm dữ liệu thứ *i*.



*Hình ảnh trực quan hóa Máy vectơ hỗ trợ*

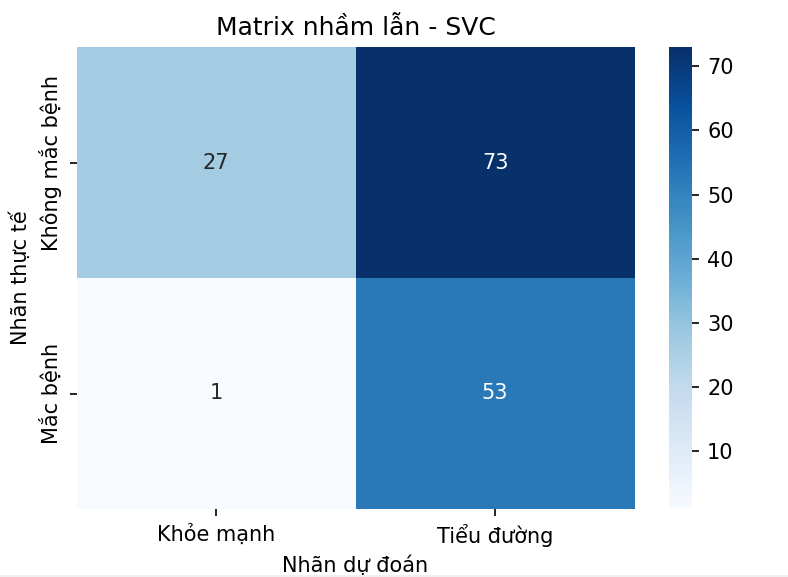
**Bước 1:**Thực hiện tạo và huấn luyện mô hình SVC trên dữ liệu huấn luyện, dự đoán giá trị trên tập kiểm tra:





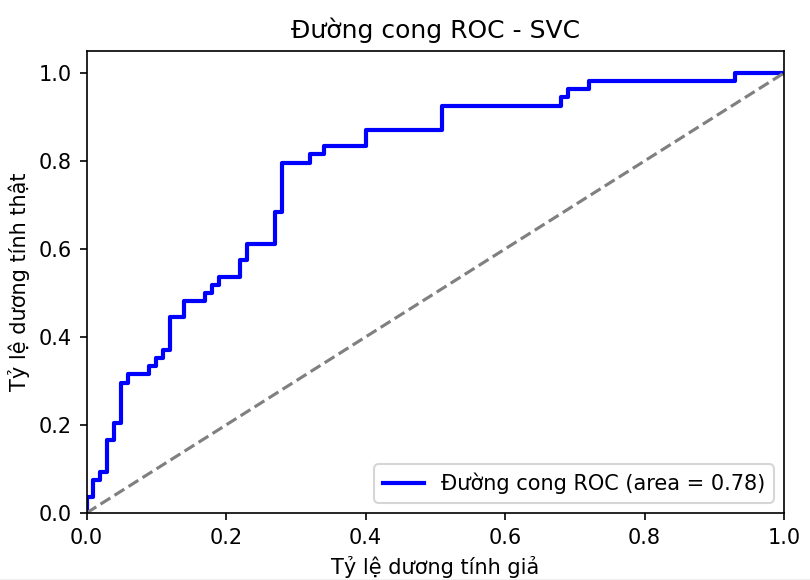
* SVC cho thấy độ chính xác rất thấp chỉ với 0.5195.
* Precision cho lớp 0 rất cao (0.96), nhưng recall chỉ đạt 0.27, cho thấy mô hình này gặp vấn đề nghiêm trọng trong việc nhận diện lớp không mắc bệnh. Đối với lớp 1, recall đạt rất cao (0.98), nhưng precision lại rất thấp (0.42), cho thấy mô hình dễ nhận diện quá mức những trường hợp mắc bệnh nhưng lại rất dễ nhầm lẫn giữa các trường hợp.

**Bước 2:** Vẽ sơ đồ Ma trận nhầm lẫn và đường cong ROC cho mô hình SVC



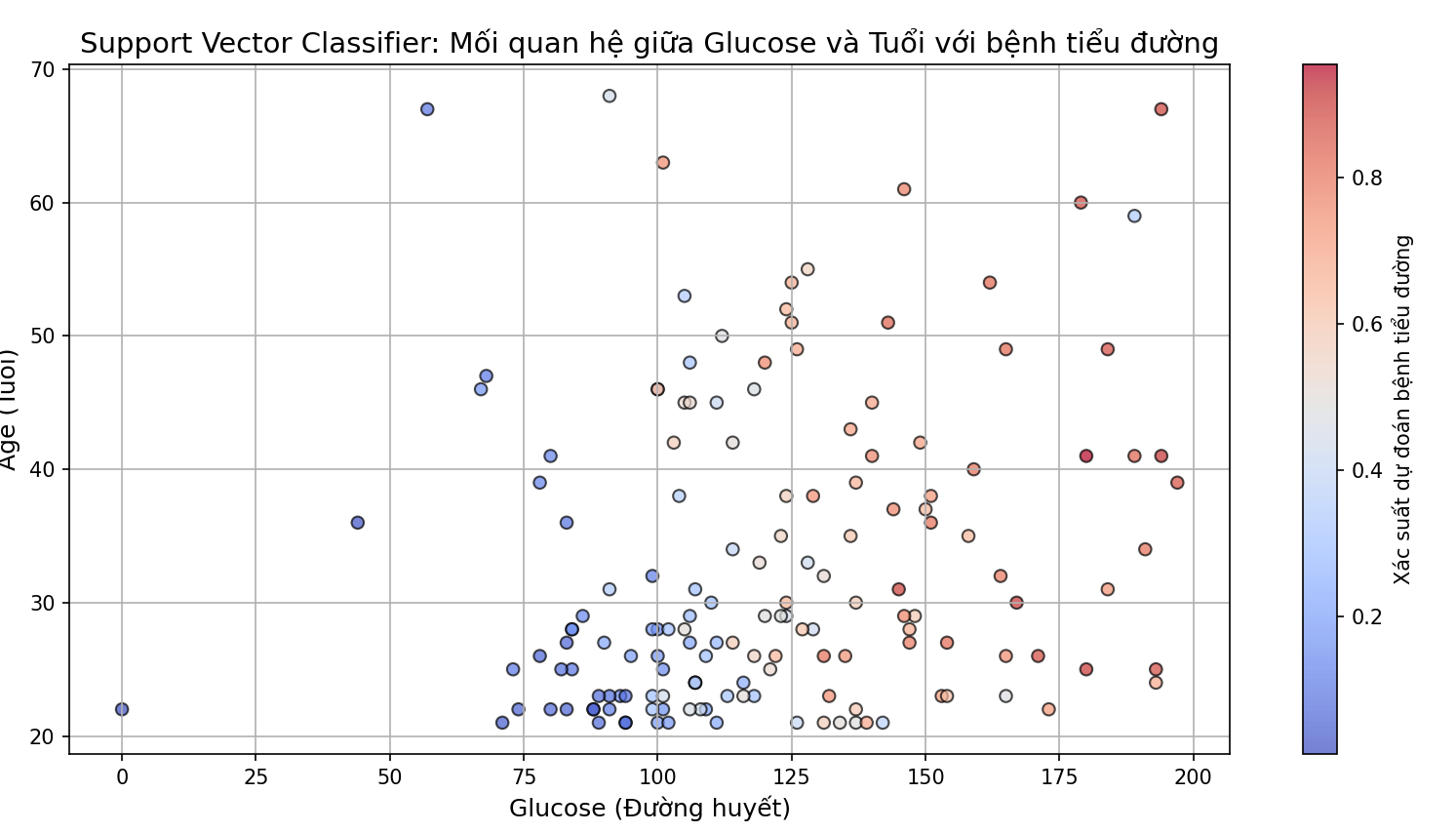
*Biểu đồ Ma trận nhầm lẫn SVC*

Support Vector Machines (SVM) hoạt động rất tốt trong việc phân loại lớp 0 (có TN cao nhất), nhưng lại gặp khó khăn trong việc phát hiện lớp 1, dẫn đến recall thấp cho lớp này. Mô hình này có xu hướng thiên về dự đoán lớp không mắc bệnh và gặp khó khăn khi phân loại các trường hợp mắc bệnh.



*Biểu đồ Đường cong ROC SVC*

**Bước 3:** Vẽ biểu đồ Scatter plot cho mô hình *SVC*

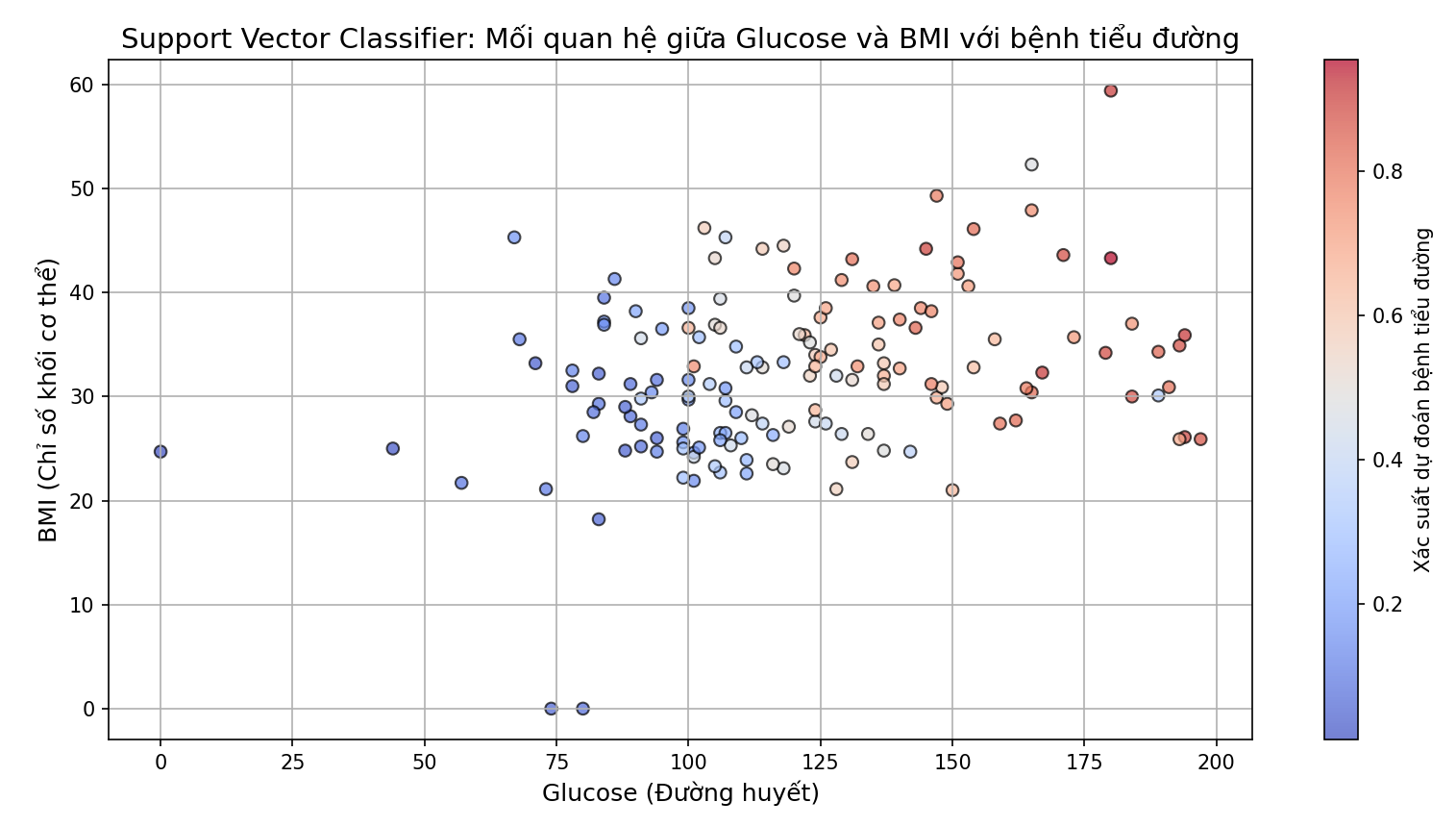
**

*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình SVC*

**Trục x (Glucose - Đường huyết):**Giá trị tăng từ 0 đến 1, biểu diễn mức đường huyết của các cá nhân. Mức đường huyết cao hơn thường liên quan đến nguy cơ mắc bệnh tiểu đường cao hơn.

**Trục y (Age - Tuổi):**Giá trị tăng từ 0 đến 0.8 (đã được chuẩn hóa), biểu thị độ tuổi của cá nhân.

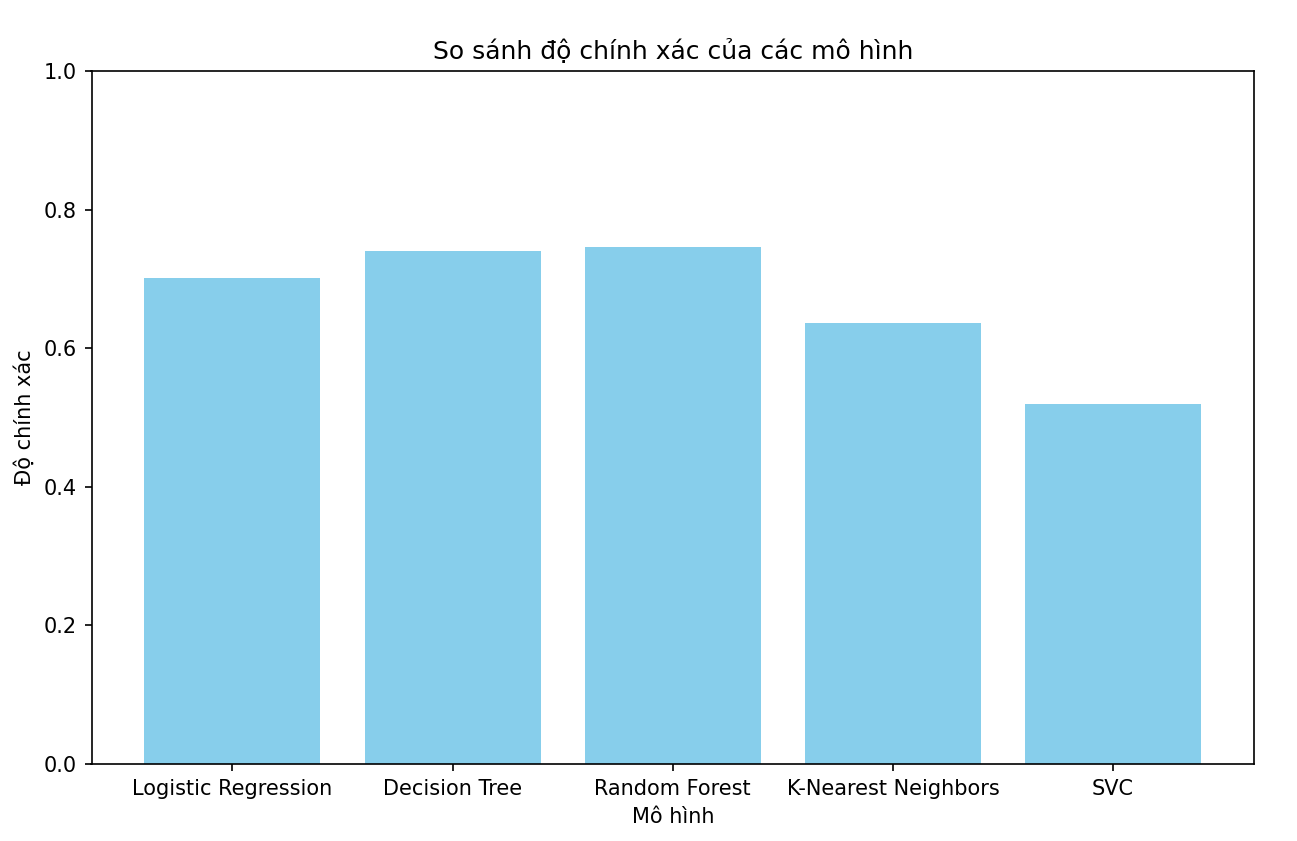
* Các điểm màu đỏ thể hiện xác suất dự đoán cao của bệnh tiểu đường.
* Các điểm màu xanh biểu thị xác suất thấp hơn.



*Biểu đồ Scatter Mối quan hệ giữa Glucose và Tuổi với bệnh tiểu đường - mô hình SVC*

## 3.3. Kết luận về quá trình phân tích:

Trong quá trình phân tích và đánh giá các mô hình học máy để dự đoán bệnh tiểu đường, chúng tôi đã thử nghiệm một số thuật toán phổ biến như Hồi quy Logistic, Cây quyết định, Rừng ngẫu nhiên, K-Nearest Neighbors (KNN), và Máy vector hỗ trợ (SVM). Mỗi mô hình có những đặc điểm riêng và việc lựa chọn mô hình tốt nhất phụ thuộc vào các chỉ số như độ chính xác (Accuracy), precision, recall, và F1-Score.



*Biểu đồ so sánh độ chính xác của các mô hình*

Biểu đồ cột so sánh độ chính xác của các mô hình với kết quả:

* Logistic Regression: 0.7013
* Decision Tree: 0.7403
* Random Forest: 0.7468
* KNN: 0.6364
* SVC: 0.5195

### Kết luận

* **Logistic Regression** cho kết quả tương đối ổn với độ chính xác 0.7013, nhưng recall cho lớp 0 thấp, cho thấy mô hình còn bỏ sót nhiều trường hợp không mắc bệnh.
* **Decision Tree** có kết quả khá tốt với độ chính xác 0.7403, khả năng nhận diện lớp 1 (mắc bệnh) khá tốt.
* **Random Forest** là mô hình đạt kết quả tốt nhất với độ chính xác 0.7468, với sự cân bằng tốt giữa precision và recall cho cả lớp 0 và lớp 1.
* **KNN** có độ chính xác thấp nhất (0.6364) và chưa thực sự hiệu quả trong việc phân loại, đặc biệt là đối với lớp mắc bệnh.
* **SVC** có độ chính xác rất thấp (0.5195), và mặc dù có khả năng nhận diện tốt lớp mắc bệnh, nhưng lại thiếu hiệu quả trong việc phân loại lớp không mắc bệnh.

Sau khi đánh giá các mô hình, Random Forest đạt độ chính xác cao nhất (0.7468) và có sự cân bằng tốt giữa precision và recall cho cả lớp 0 và lớp 1. Decision Tree cũng đạt kết quả khá tốt (độ chính xác 0.7403) và có khả năng phân loại ổn định hơn các mô hình khác. Các mô hình Logistic Regression, KNN, và SVC có độ chính xác thấp hơn và gặp khó khăn trong việc phân loại chính xác, đặc biệt là lớp mắc bệnh. Vì vậy, Cây Quyết Định (Decision Tree) là lựa chọn ưu tiên cho bài toán này.

# 

# CHƯƠNG 4: XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH DỰ ĐOÁN BỆNH TIỂU ĐƯỜNG

Dựa trên kết quả phân tích các mô hình học máy ở chương 3, nhóm em đã quyết định chọn Cây quyết định (Decision Tree) làm thuật toán chính để xây dựng mô hình dự đoán bệnh tiểu đường. Quyết định này được đưa ra dựa trên các chỉ số quan trọng như Precision, Recall, F1-Score, và Độ chính xác (Accuracy).

* **Độ chính xác**: Cây quyết định đạt độ chính xác gần bằng Random Forest (khoảng 74%), cho thấy mô hình có khả năng phân loại tổng thể tốt.
* **Precision và Recall**: Decision Tree thể hiện khả năng phân loại cân bằng với precision và recall khá tốt cho cả hai lớp, trong đó lớp 0 đạt precision 0.82 và recall 0.77, còn lớp 1 đạt precision 0.62 và recall 0.69. Mô hình này cũng dễ hiểu và giải thích, giúp người dùng có thể nắm bắt được cách thức phân loại. Bên cạnh đó, Decision Tree có tốc độ huấn luyện nhanh và ít tốn tài nguyên tính toán.
* **F1-Score**: Cây quyết định đạt F1-Score khá ấn tượng (0.79), thể hiện sự cân bằng giữa Precision và Recall đối với lớp mắc bệnh. Điều này cho thấy mô hình này có khả năng tìm ra các bệnh nhân mắc bệnh tiểu đường với tỷ lệ chấp nhận được mà không bỏ sót quá nhiều trường hợp.
* So sánh với các mô hình khác: Mặc dù Random Forest có độ chính xác tổng thể tốt hơn (74,68%) và Logistic Regression có độ chính xác đạt 70%, nhưng Cây quyết định vượt trội hơn về khả năng phát hiện bệnh tiểu đường. Các mô hình như KNN và SVM gặp khó khăn trong việc phát hiện lớp mắc bệnh tiểu đường, với Recall và Precision thấp. Cây quyết định, với khả năng nhận diện cao lớp mắc bệnh (TP cao) và ít False Negative (FN), cho thấy ưu thế rõ rệt trong việc phát hiện bệnh tiểu đường.

Với những lý do trên, nhóm em quyết định chọn Cây quyết định là mô hình chính cho dự đoán bệnh tiểu đường, nhờ vào sự cân bằng tốt giữa các chỉ số và khả năng giải thích kết quả một cách dễ dàng. Cây quyết định không chỉ mang lại độ chính xác ổn định mà còn giúp hiểu rõ hơn về các yếu tố ảnh hưởng đến bệnh tiểu đường, như glucose, huyết áp, và BMI.

## 4.1. Tổng quát về tập dữ liệu và thuật toán sử dụng:

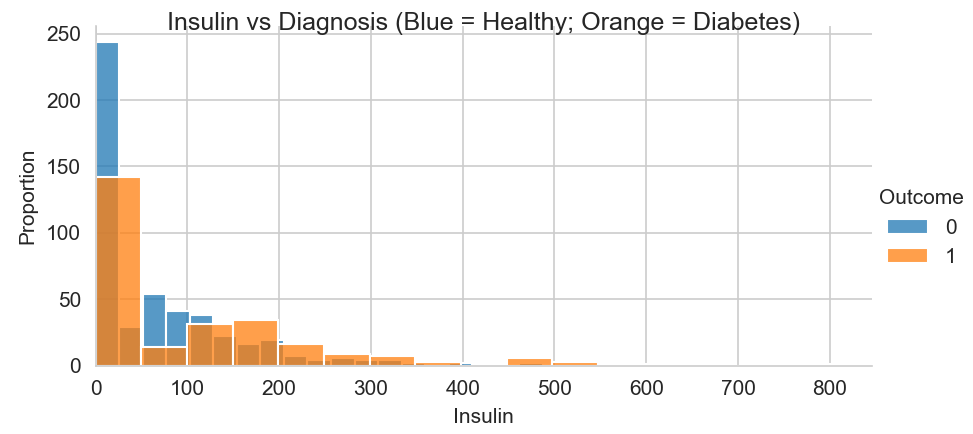
Để xây dựng mô hình dự đoán bệnh tiểu đường, chúng em sử dụng một tập dữ liệu y tế chứa các đặc trưng sức khỏe của bệnh nhân, như là mức độ glucose trong máu, huyết áp, độ dày da, chỉ số BMI, và các yếu tố di truyền. Tập dữ liệu này chứa thông tin của các bệnh nhân được chia thành hai nhóm: những người mắc bệnh tiểu đường và những người không mắc bệnh tiểu đường, dựa trên biến mục tiêu là Outcome (biến này có giá trị 1 nếu bệnh nhân bị tiểu đường và 0 nếu không).

Tập dữ liệu Diabetes.csv gồm 9 thuộc tính, trong đó cột ‘Outcome’ làm thuộc tính quyết định. Cột “Outcome” trong dữ liệu cho thấy bệnh nhân có mắc bệnh tiểu đường hay không dựa trên các chỉ số y tế.

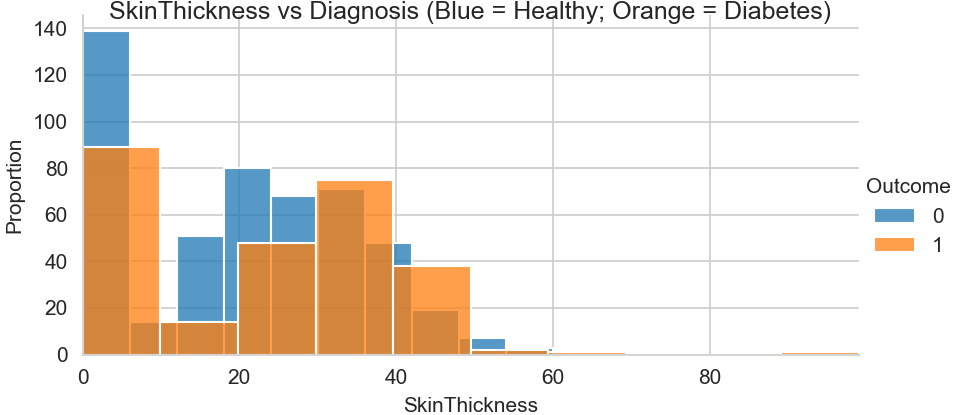
Dữ liệu được biểu diễn dưới dạng số thực

* **Pregnancies**: Số lần mang thai của bệnh nhân (giá trị số nguyên). Yếu tố này đặc biệt liên quan đến nữ giới và có thể ảnh hưởng đến nguy cơ mắc tiểu đường, vì sự thay đổi hormone trong quá trình mang thai có thể tác động đến khả năng xử lý glucose của cơ thể.
* **Glucose**: Lượng glucose trong máu, một chỉ số quan trọng để đánh giá nguy cơ mắc bệnh tiểu đường.
* **BloodPressure**: Huyết áp, có thể liên quan đến các vấn đề về sức khỏe và nguy cơ bệnh tiểu đường.
* **SkinThickness**: Độ dày da, có thể dùng làm dấu hiệu cho sự tích tụ mỡ dưới da.
* **Insulin**: Mức insulin, một hormone quan trọng trong điều chỉnh glucose trong máu.
* **BMI**: Chỉ số khối cơ thể (Body Mass Index), giúp đánh giá tình trạng cân nặng của bệnh nhân.
* **DiabetesPedigreeFunction**: Thước đo có tính chất gia đình đối với bệnh tiểu đường, dùng để xác định khả năng di truyền.
* **Age**: Tuổi của bệnh nhân, có liên quan mật thiết đến nguy cơ mắc bệnh.
* **Outcome**: Kết quả về tình trạng tiểu đường, là biến mục tiêu để xác định liệu bệnh nhân có bị bệnh tiểu đường hay không.

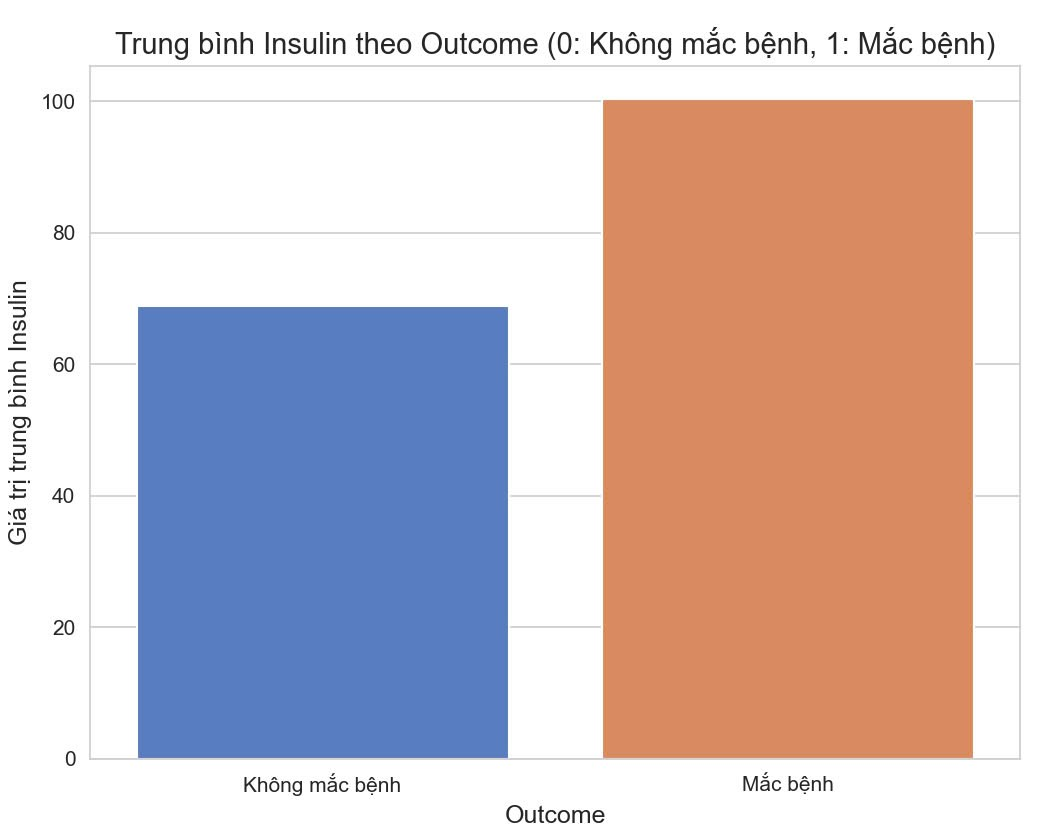
Sử dụng biểu đồ histogram để minh họa sự phân bố của các đặc trưng Insulin và SkinThickness trong hai nhóm đối tượng: người khỏe mạnh và người mắc bệnh tiểu đường. Cụ thể, biểu đồ "Insulin vs Diagnosis" và "SkinThickness vs Diagnosis" cho thấy sự khác biệt về phân phối mức độ Insulin và SkinThickness giữa hai nhóm này, với nhóm người khỏe mạnh được biểu diễn bằng màu xanh và nhóm người mắc bệnh tiểu đường được biểu diễn bằng màu cam. Các biểu đồ này giúp chúng ta nhận diện sự phân biệt rõ ràng giữa các nhóm dựa trên các đặc trưng này, từ đó đưa ra nhận xét về mối liên hệ giữa chúng và bệnh tiểu đường. Việc có sự phân bố khác biệt giữa các nhóm là dấu hiệu cho thấy các đặc trưng này có thể đóng vai trò quan trọng trong việc dự đoán bệnh tiểu đường. Thông qua đó, các biểu đồ này hỗ trợ phân tích và lựa chọn các đặc trưng phù hợp cho mô hình học máy, giúp nâng cao hiệu quả dự đoán bệnh tiểu đường.



*Biểu đồ histogram để minh họa sự phân bố của các đặc trưng Insulin vs Diagnosis*

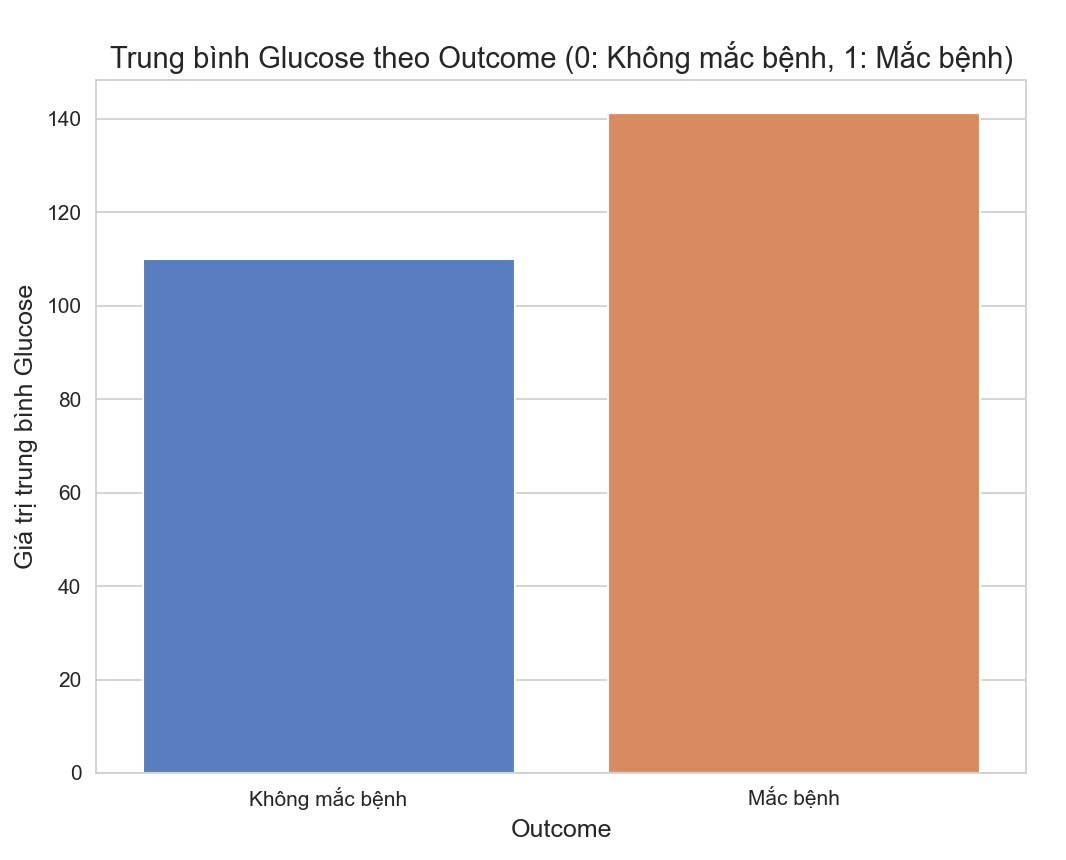


*Biểu đồ histogram để minh họa sự phân bố của các đặc trưng Skin Thickness vs Diagnosis*



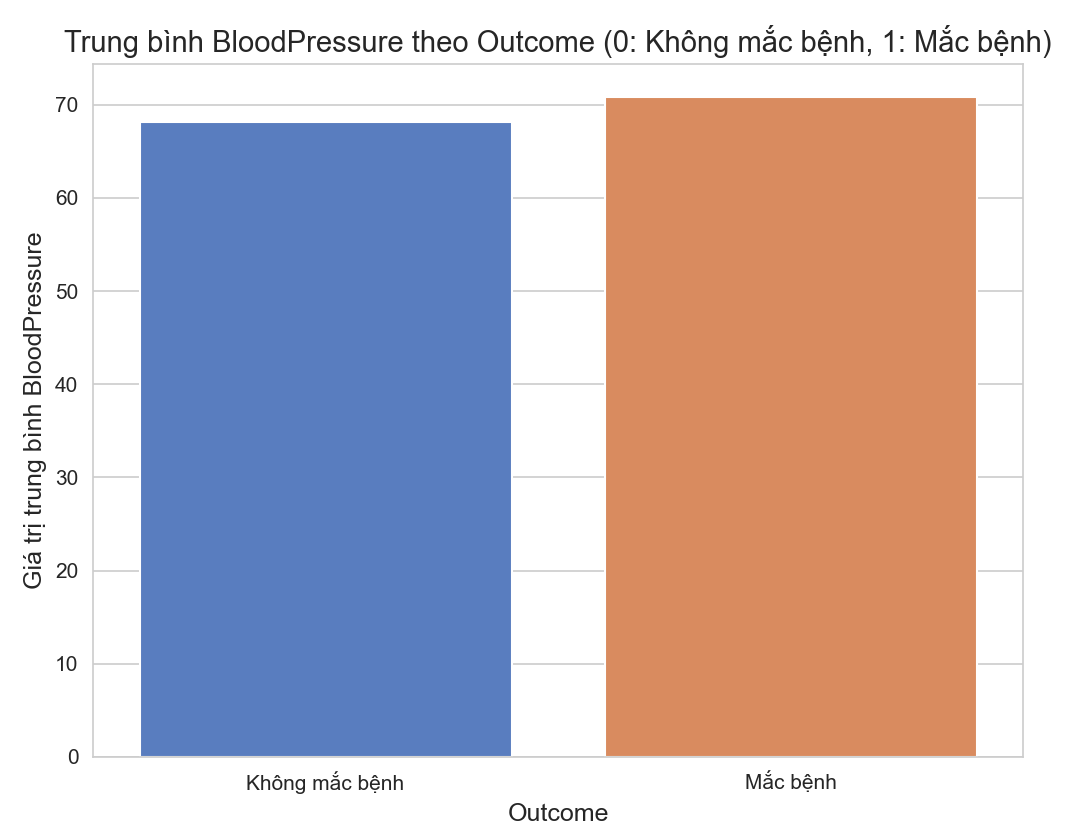
*Biểu đồ trung bình Insulin*

Dựa trên biểu đồ trung bình Insulin, ta thấy có phân phối khác biệt rõ rệt giữa hai nhóm Outcome (0: không mắc bệnh, 1: mắc bệnh). Điều này cho thấy Insulin là một trong những yếu tố có mối tương quan mạnh mẽ với nguy cơ mắc bệnh tiểu đường.Mặc dù sự phân bổ của Insulin cũng khác biệt giữa hai nhóm, dữ liệu có thể chỉ ra rằng mức Insulin có xu hướng cao hơn hoặc thấp hơn bất thường ở nhóm mắc bệnh (Outcome = 1). Điều này phản ánh khả năng rối loạn chức năng Insulin là một trong những nguyên nhân chính của bệnh tiểu đường.



*Biểu đồ trung bình Glucose*

Dựa trên biểu đồ trung bình glucose, ta thấy có phân phối khác biệt rõ rệt giữa hai nhóm Outcome (0: không mắc bệnh, 1: mắc bệnh). Điều này cho thấy glucose là một trong những yếu tố có mối tương quan mạnh mẽ với nguy cơ mắc bệnh tiểu đường.Giá trị Glucose cao thường tập trung ở nhóm người mắc bệnh (Outcome = 1), cho thấy rằng mức đường huyết đóng vai trò quan trọng trong việc phân biệt giữa hai nhóm. Điều này phù hợp với các nghiên cứu y học cho rằng chỉ số đường huyết cao là một dấu hiệu chính của bệnh tiểu đường.



*Biểu đồ trung bình BMI*

Dựa trên biểu đồ trung bình BMI, ta thấy sự phân bố khá cân bằng giữa hai nhóm Outcome (0: không mắc bệnh, 1: mắc bệnh). Điều này cho thấy BMI có ảnh hưởng nhất định đến nguy cơ mắc bệnh tiểu đường, nhưng mức độ tác động không quá mạnh như các yếu tố khác. Tuy nhiên, chỉ số này vẫn đóng vai trò hỗ trợ trong việc phân loại và dự đoán bệnh, đặc biệt khi kết hợp với các đặc trưng quan trọng hơn như Glucose hay Insulin.

## 4.2. Thuật toán sử dụng xây dựng mô hình dự đoán bệnh tiểu đường:

Cây quyết định (Decision Tree) là thuật toán học máy dùng để phân loại, xây dựng mô hình dựa trên việc chia nhỏ dữ liệu thành các nhánh theo đặc trưng.Thuật toán này rất phù hợp trong việc chẩn đoán bệnh, vì nó có thể xác định các yếu tố quan trọng ảnh hưởng đến tình trạng bệnh, đồng thời giúp giải thích các quyết định mô hình một cách dễ dàng và áp dụng trực tiếp vào thực tế, như trong việc dự đoán bệnh tiểu đường. Có nhiều thuật toán trong xây dựng cây quyết định, nhóm chúng em quyết định sử dụng 3 thuật toán đó là ID3 (Iterative Dichotomiser 3), C4.5 và CART (Classification and Regression Trees). Một thuật toán này đều có đặc điểm và ưu nhược điểm riêng, phản ánh sự đa dạng trong cách tiếp cận và xử lý dữ liệu.

### 4.2.1. Thuật toán ID3:

Thuật toán ID3 (Iterative Dichotomiser 3) là một thuật toán xây dựng cây quyết định nổi tiếng, được phát triển bởi Ross Quinlan. ID3 sử dụng khái niệm "entropy" để đo độ không chắc chắn trong dữ liệu và "information gain" (lượng thông tin tăng thêm) để lựa chọn thuộc tính phân nhánh tốt nhất tại mỗi nút trong cây. Quá trình phân nhánh của ID3 diễn ra tuần tự qua từng thuộc tính, chọn thuộc tính có lượng thông tin tăng cao nhất để chia nhỏ tập dữ liệu. Công thức tính **Information Gain (IG)** cho một thuộc tính A là:



Trong đó:

* *Entropy(D)*: độ hỗn loạn của tập dữ liệu *D*.
* *Values(A)*: các giá trị có thể có của thuộc tính *A*.
* *Dv*: con tập con của D với thuộc tính A có giá trị *v*.

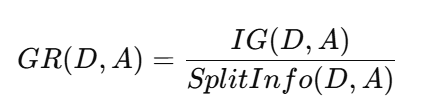
Thuật toán tiếp tục quá trình này cho đến khi tất cả các thuộc tính được phân nhánh hoặc khi không còn dữ liệu cần phân loại. ID3 thường được sử dụng trong các bài toán phân loại, giúp đưa ra những quyết định hiệu quả dựa trên dữ liệu huấn luyện.

Ưu điểm của thuật toán ID3:

* Sử dụng Entropy: ID3 đo lường sự hỗn loạn của dữ liệu qua Entropy. Tiêu chí này đảm bảo các thuộc tính được chọn sẽ giảm thiểu sự hỗn loạn, giúp phân loại dữ liệu chính xác hơn.
* Tối ưu hóa qua Information Gain: Các thuộc tính được lựa chọn dựa trên mức tăng thông tin lớn nhất, nghĩa là các đặc trưng có khả năng phân loại dữ liệu tốt nhất sẽ được ưu tiên.
* Dễ hiểu và trực quan: Cây quyết định do ID3 tạo ra giúp người dùng dễ dàng diễn giải và hiểu rõ mối quan hệ giữa các đặc trưng và kết quả phân loại.

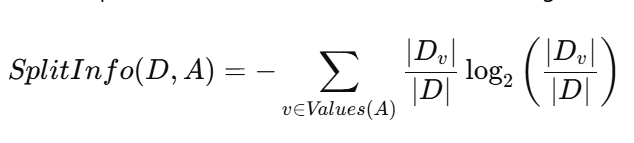
### 4.2.2. Thuật toán C4.5:

C4.5 là phiên bản cải tiến của ID3, cũng do Ross Quinlan phát triển, nhằm khắc phục một số hạn chế của ID3. Thuật toán này cũng dựa trên khái niệm "information gain," nhưng thay vì sử dụng entropy, C4.5 dùng "gain ratio" (tỷ lệ thông tin tăng thêm) để đo mức độ phân nhánh tốt hơn và tránh thiên vị cho các thuộc tính có nhiều giá trị. Công thức tính Gain Ratio cho một thuộc tính A là:



Trong đó:

* *SplitInfo(D, A)* là mức độ phân chia của thuộc tính A, tính theo công thức:



C4.5 hỗ trợ dữ liệu rời rạc và liên tục, đồng thời có khả năng xử lý dữ liệu bị thiếu giá trị, điều mà ID3 không làm được. Ngoài ra, C4.5 cũng cho phép "pruning" (cắt tỉa) cây để tránh overfitting (quá khớp) dữ liệu, giúp cây quyết định có tính khái quát cao hơn khi áp dụng vào dữ liệu mới.

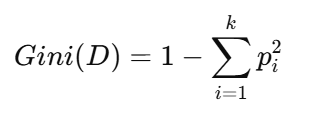
Ưu điểm của thuật toán C4.5:

* Xử lý dữ liệu liên tục: C4.5 tự động tạo ra các điểm cắt (threshold) để phân chia các thuộc tính liên tục (như Glucose hoặc BMI), giúp tăng độ chính xác khi dữ liệu không đồng nhất.
* Giảm thiểu thiên lệch thông qua Gain Ratio: Thay vì chỉ sử dụng Information Gain, C4.5 sử dụng Gain Ratio để cân bằng giữa các thuộc tính có số lượng giá trị khác nhau.
* Hỗ trợ pruning (cắt tỉa cây): Giảm kích thước của cây để tránh overfitting, cải thiện khả năng tổng quát hóa của mô hình.
* Tối ưu hóa quá trình phân loại giữa hai lớp Diabetes và No Diabetes bằng cách tận dụng các thuộc tính liên tục và giảm sự thiên lệch trong lựa chọn thuộc tính.
* Xây dựng mô hình cây quyết định mạnh mẽ, dễ hiểu, nhưng vẫn đảm bảo khả năng tổng quát hóa tốt trên dữ liệu mới.
* Trực quan hóa cây quyết định để hiểu rõ cách mô hình đưa ra các quyết định dựa trên các đặc trưng đầu vào.

### 4.2.3. Thuật toán CART:

CART (Classification and Regression Tree) là thuật toán xây dựng cây quyết định dùng cho cả hai bài toán phân loại (classification) và hồi quy (regression). Trong bối cảnh bài toán phân loại tiểu đường, CART được áp dụng để xác định mối quan hệ giữa các yếu tố sức khỏe (như glucose, BMI, huyết áp, v.v.) và khả năng mắc bệnh tiểu đường.

Trong phân loại, CART chia nhỏ tập dữ liệu dựa trên chỉ số "Gini impurity" (độ nhiễm bẩn Gini) thay vì entropy, và tiếp tục chia các nút trong cây dựa trên độ giảm nhiễm bẩn Gini lớn nhất.



Trong đó:

* *pi​*: tỷ lệ của lớp *i* trong tập dữ liệu *D*.

Đối với bài toán hồi quy, CART sử dụng phương pháp tối thiểu hóa sai số bình phương trung bình để chọn thuộc tính phân nhánh. CART tạo ra các cây nhị phân (binary tree), tức là mỗi nút chỉ có hai nhánh con. Giống như C4.5, thuật toán này cũng sử dụng kỹ thuật "pruning" để giảm thiểu hiện tượng overfitting, giúp cây quyết định được tinh gọn và có khả năng dự đoán tốt hơn trên dữ liệu mới.

## 4.3. Quy trình triển khai và đánh giá mô hình:

### 4.3.1. Xử lý dữ liệu đầu vào:

Trong quy trình triển khai và đánh giá mô hình, việc xử lý dữ liệu đầu vào là một bước quan trọng giúp đảm bảo chất lượng dữ liệu đầu vào, từ đó tạo nền tảng vững chắc cho mô hình học máy. Mục tiêu của phần xử lý dữ liệu đầu vào là làm sạch, chuẩn hóa và phân tách dữ liệu sao cho phù hợp với các thuật toán học máy.

Quy trình xử lý dữ liệu đầu vào:

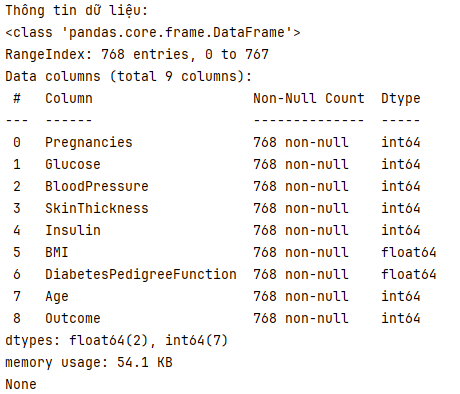
**Bước 1**: Đọc Dữ Liệu và Kiểm Tra Thông Tin Dữ Liệu

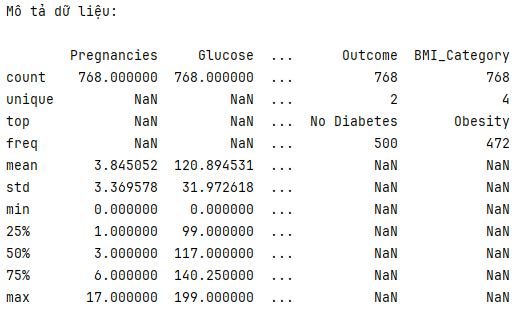
Bước đầu tiên trong quy trình là tải dữ liệu từ tệp CSV. Dữ liệu chứa các thông tin về các đặc trưng sức khỏe của bệnh nhân (như mức đường huyết, huyết áp, chỉ số BMI) và kết quả của bệnh tiểu đường (Outcome). Thư viện Pandas được sử dụng để đọc tệp CSV và kiểm tra thông tin về cấu trúc dữ liệu.



Sau khi dữ liệu được tải, chúng ta cần kiểm tra các thông tin cơ bản về dữ liệu như kiểu dữ liệu của các cột, số lượng giá trị hợp lệ, giá trị thiếu (nếu có) và các thống kê mô tả (trung bình, độ lệch chuẩn, giá trị min/max) của từng cột. Các hàm info() và describe() của Pandas được sử dụng để thực hiện điều này:







Kết quả này cung cấp cái nhìn tổng quan về bộ dữ liệu, giúp chúng ta hiểu rõ về các cột và các thuộc tính trong bộ dữ liệu, bao gồm các giá trị cơ bản như trung bình, độ lệch chuẩn, và các giá trị min/max.

**Bước 2**: Tách Dữ Liệu Thành Đặc Trưng và Mục Tiêu

Sau khi tải và kiểm tra dữ liệu, bước tiếp theo là phân tách dữ liệu thành các đặc trưng (features) và mục tiêu (target). Trong bài toán này, cột "Outcome" là cột mục tiêu, còn lại các cột khác sẽ là đặc trưng đầu vào.



Trong đó:

* X chứa tất cả các đặc trưng (features) của bệnh nhân như Glucose, BloodPressure, BMI, v.v.
* y chứa kết quả bệnh tiểu đường, với các giá trị là 0 hoặc 1, ứng với "No Diabetes" và "Diabetes".

**Bước 3**: Mã hóa Nhãn Kết Quả (Outcome)

Cột "Outcome" trong bộ dữ liệu có thể chứa các giá trị dạng chuỗi như "No Diabetes" và "Diabetes". Tuy nhiên, các thuật toán học máy yêu cầu dữ liệu phải ở dạng số. Do đó, chúng ta cần mã hóa các nhãn này thành các giá trị số (0 và 1) để phù hợp với yêu cầu của các mô hình học máy.



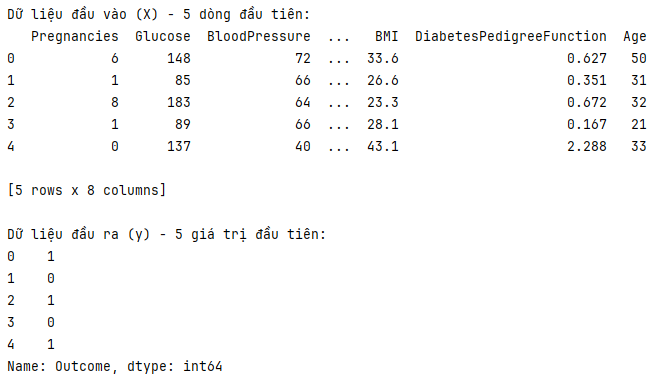
Sau bước này, y sẽ chứa các giá trị 0 và 1, phù hợp với yêu cầu của các thuật toán học máy.

**Bước 4**: Hiển Thị Dữ Liệu Đầu Vào và Đầu Ra

Sau khi tách dữ liệu thành các đặc trưng và mục tiêu, chúng ta có thể in ra một số dòng đầu tiên của dữ liệu đầu vào (X) và dữ liệu đầu ra (y) để kiểm tra tính chính xác của quá trình xử lý. Đoạn mã sau giúp kiểm tra 5 dòng đầu tiên của X và y:



Kết quả in ra sẽ cung cấp cái nhìn tổng quan về các đặc trưng đầu vào và các giá trị mục tiêu, giúp kiểm tra tính chính xác của quá trình phân tách và mã hóa dữ liệu.



Sau khi hoàn thành các bước xử lý, chúng ta đã có bộ dữ liệu sẵn sàng cho việc huấn luyện và đánh giá mô hình học máy. Bộ dữ liệu đã được phân tách và mã hóa hợp lý, giúp các thuật toán học máy hoạt động hiệu quả hơn trong việc dự đoán bệnh tiểu đường.

### 4.3.2. Áp dụng thuật toán ID3:

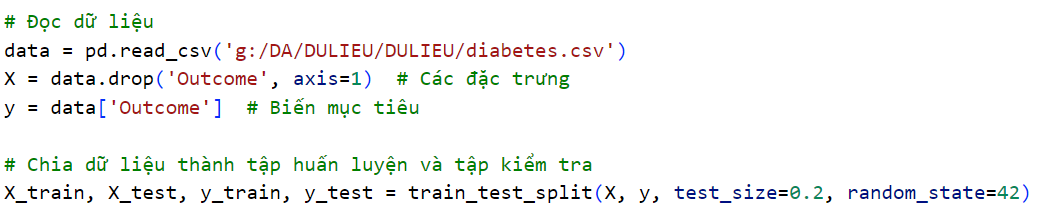
Thuật toán ID3 (Iterative Dichotomiser 3) là một trong những thuật toán cơ bản để xây dựng cây quyết định. ID3 sử dụng Entropy và Information Gain để lựa chọn các thuộc tính tốt nhất tại mỗi bước phân chia dữ liệu. Trong bài toán phân loại bệnh tiểu đường, ID3 giúp xác định các đặc trưng quan trọng nhất và phân loại dữ liệu thành các nhánh:

* Diabetes (Mắc bệnh tiểu đường)
* No Diabetes (Không mắc bệnh tiểu đường)

**Bước 1:** Chuẩn bị Dữ Liệu

Trước khi triển khai thuật toán ID3, chúng ta cần đảm bảo rằng bộ dữ liệu đã được xử lý và chuẩn bị kỹ càng. Quy trình chuẩn bị dữ liệu đã được thực hiện ở các bước trước như:

* Đọc dữ liệu từ tệp CSV.
* Kiểm tra và làm sạch dữ liệu (bao gồm việc xử lý các giá trị thiếu, kiểm tra kiểu dữ liệu của các cột, và thống kê mô tả các thuộc tính).
* Mã hóa các nhãn (Outcome và BMI\_Category) thành các giá trị số để phù hợp với các thuật toán học máy.
* Tách bộ dữ liệu thành các đặc trưng (features) và nhãn (labels).
* Chia bộ dữ liệu thành hai phần: 80% cho huấn luyện (training) và 20% cho kiểm tra (testing) với việc sử dụng hàm train\_test\_split từ thư viện sklearn.



**Bước 2:** Xây Dựng Mô Hình ID3

Thuật toán ID3 sử dụng *Entropy và Information Gain* làm tiêu chí chính để lựa chọn thuộc tính tốt nhất tại mỗi bước phân chia. Thuật toán sẽ tiếp tục phân chia dữ liệu đến khi đạt được điều kiện dừng, chẳng hạn như cây đạt đến độ sâu tối đa hoặc số mẫu trong một nhánh nhỏ hơn ngưỡng quy định.

Để xây dựng mô hình ID3, chúng ta sử dụng lớp DecisionTreeClassifier từ thư viện sklearn với các tham số đặc thù:

* criterion='entropy': Sử dụng Entropy làm tiêu chí để đo lường độ không thuần nhất và lựa chọn thuộc tính tốt nhất.
* max\_depth: Giới hạn độ sâu của cây quyết định nhằm tránh overfitting. Chúng ta cần chọn giá trị hợp lý, ví dụ: 5.
* random\_state: Đảm bảo tính tái lặp của mô hình bằng cách cố định các giá trị ngẫu nhiên.

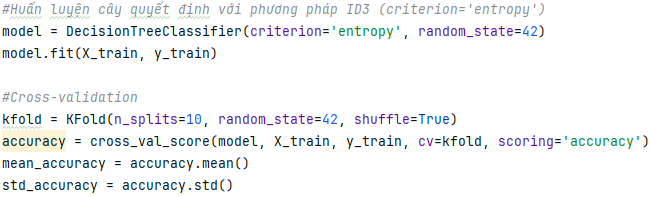
**Bước 3:** Đánh Giá Mô Hình

Sau khi xây dựng và huấn luyện mô hình ID3, chúng ta tiến hành đánh giá hiệu suất dự đoán trên tập kiểm tra. Phương pháp đánh giá phổ biến bao gồm:

* Độ chính xác (Accuracy): Tính toán tỷ lệ phần trăm dự đoán đúng trên tổng số mẫu trong tập kiểm tra.
* Ma trận nhầm lẫn (Confusion Matrix): Cung cấp cái nhìn chi tiết về số lượng dự đoán đúng và sai đối với mỗi lớp.
* ROC Learning curve: Hiển thị sự thay đổi độ chính xác của mô hình theo kích thước của tập huấn luyện.

**Đánh Giá Độ Chính Xác**

Sau khi huấn luyện xong, chúng ta sẽ sử dụng accuracy\_score để đánh giá độ chính xác của mô hình trên tập kiểm tra.



Sau khi huấn luyện và dự đoán với mô hình ID3 trên tập kiểm tra, chúng ta sẽ in ra độ chính xác của mô hình. Độ chính xác là tỷ lệ dự đoán đúng trên tổng số mẫu trong tập kiểm tra.

Kết quả:

* Accuracy là giá trị từ 0 đến 1, càng gần 1 thì mô hình càng chính xác.
* Ở đây là 0.72 tương ứng với 72%. Điều này có nghĩa là mô hình ID3 đã dự đoán đúng 72% số trường hợp trong tập huấn luyện.

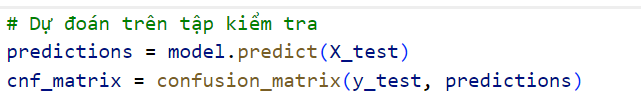


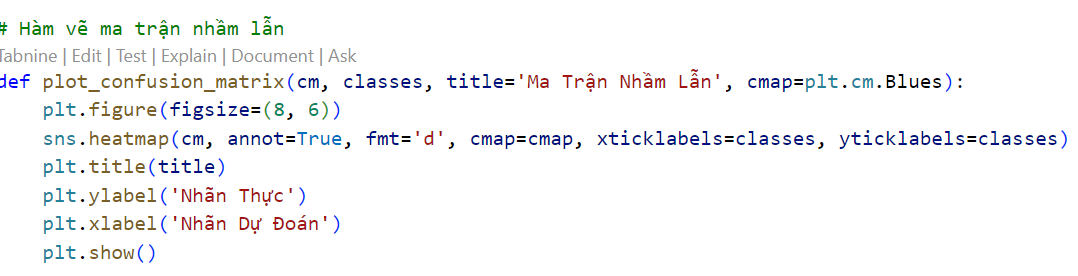
Kết quả này giúp người dùng hiểu mức độ chính xác mà mô hình ID3 đạt được khi dự đoán kết quả bệnh tiểu đường từ tập kiểm tra.

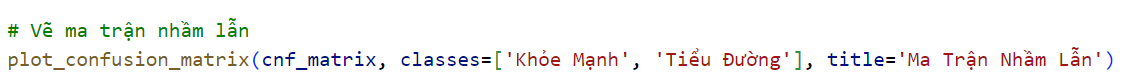
Kết quả càng cao, mô hình càng hiệu quả trong việc phân loại các trường hợp mắc bệnh tiểu đường và không mắc bệnh.

**Ma Trận Nhầm Lẫn (Confusion Matrix)**

Ma trận nhầm lẫn giúp chúng ta hiểu rõ hơn về hiệu quả phân loại của mô hình bằng cách chỉ ra số lượng các trường hợp dự đoán đúng và sai. Chúng ta có thể vẽ ma trận nhầm lẫn bằng cách sử dụng ConfusionMatrixDisplay từ thư viện sklearn.

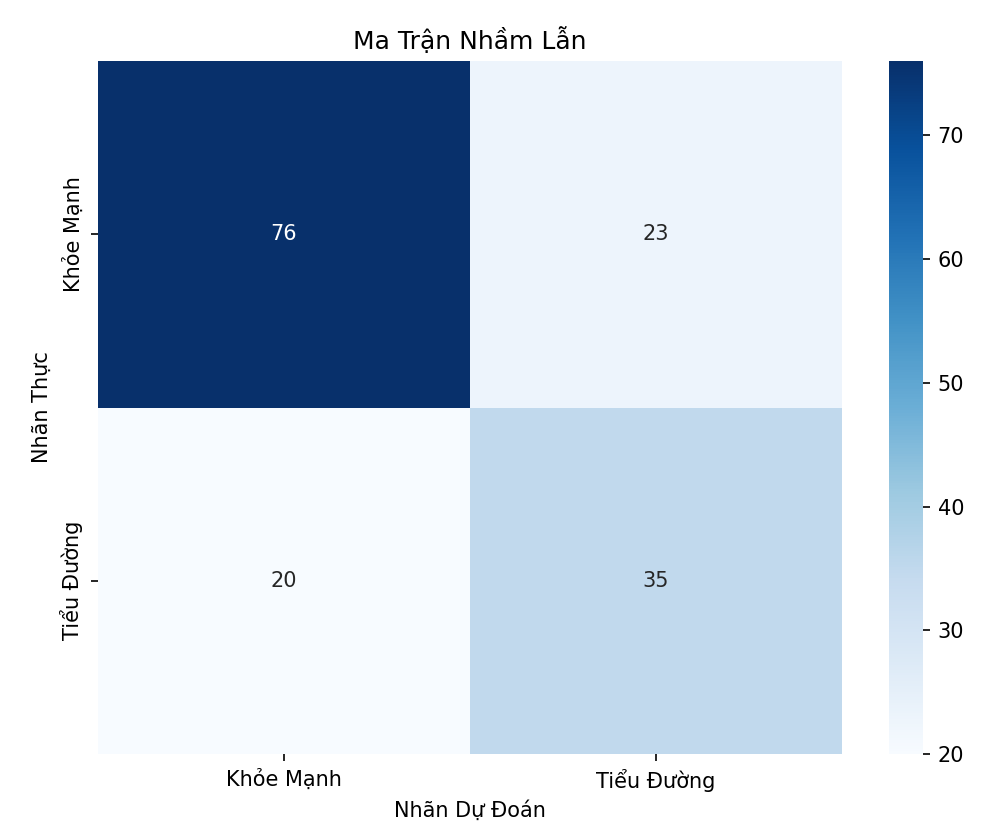






Ma trận nhầm lẫn (confusion matrix) sẽ thể hiện số lượng dự đoán đúng và sai của mô hình đối với mỗi lớp (Diabetes vs No Diabetes). Nó bao gồm 4 giá trị:

* True Positive (TP): Số trường hợp thực sự mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Positive (FP): Số trường hợp không mắc bệnh tiểu đường nhưng mô hình lại dự đoán là mắc bệnh.
* True Negative (TN): Số trường hợp không mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Negative (FN): Số trường hợp mắc bệnh tiểu đường nhưng mô hình lại dự đoán không mắc bệnh.

****

*Biểu đồ ma trận nhầm lẫn của phương pháp ID3*

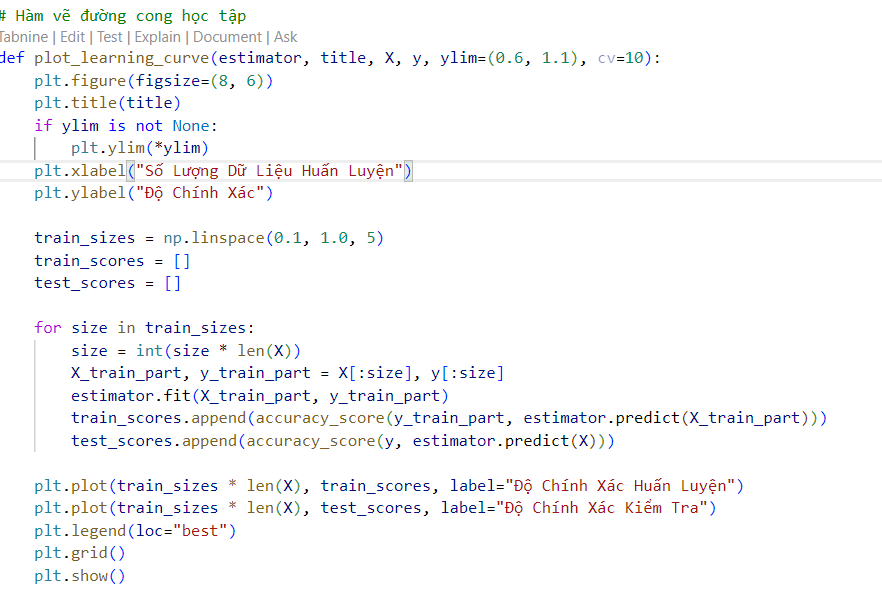
* **76** là số trường hợp không mắc bệnh tiểu đường và mô hình dự đoán đúng là "Khỏe Mạnh" (True Negative).
* **23** là số trường hợp không mắc bệnh tiểu đường nhưng mô hình lại dự đoán sai là "Tiểu Đường" (False Positive).
* **20** là số trường hợp mắc bệnh tiểu đường nhưng mô hình lại dự đoán sai là "Khỏe Mạnh" (False Negative).
* **35** là số trường hợp mắc bệnh tiểu đường và mô hình dự đoán đúng là "Tiểu Đường" (True Positive).

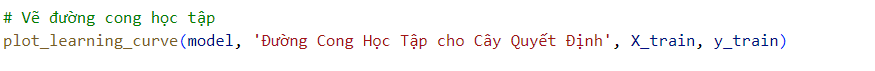
Ma trận nhầm lẫn cho phép chúng ta hiểu rõ hơn về việc mô hình xử lý đúng và sai như thế nào đối với từng lớp.

Càng nhiều giá trị True Positive và True Negative, mô hình càng chính xác trong việc phân loại đúng các trường hợp mắc bệnh tiểu đường.

**Đường Cong Học (Learning Curve)**

Để đánh giá sự thay đổi độ chính xác khi tăng kích thước của tập huấn luyện, chúng ta có thể vẽ đường cong học. Điều này giúp nhận diện liệu mô hình có bị overfitting hay underfitting khi sử dụng tập huấn luyện lớn hơn.



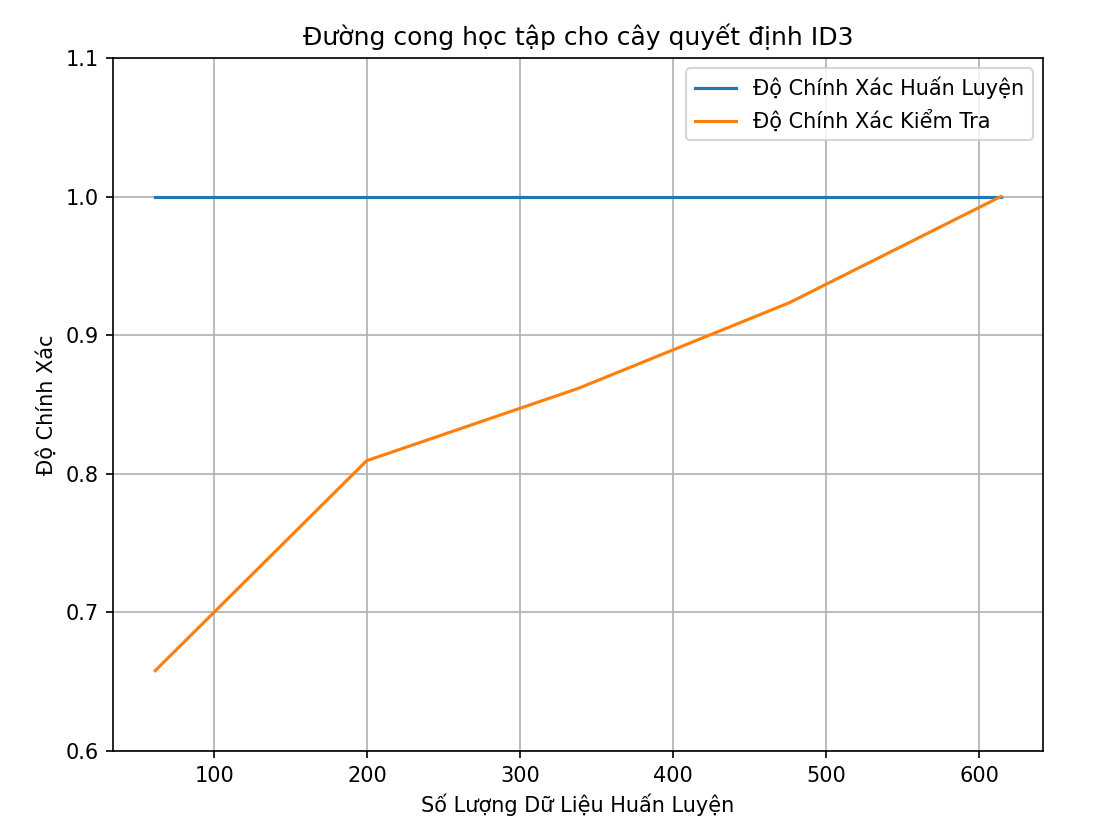


**Kết quả:**

* Độ chính xác huấn luyện: Đo lường độ chính xác của mô hình trên tập huấn luyện, có thể sẽ tăng lên khi số lượng mẫu huấn luyện tăng.
* Độ chính xác kiểm tra: Đo lường độ chính xác trên tập kiểm tra, có thể sẽ bắt đầu ổn định hoặc giảm nếu mô hình bị overfitting.

**Ví dụ về đường cong:**

* Nếu độ chính xác huấn luyện tăng và độ chính xác kiểm tra ổn định hoặc giảm sau một điểm nhất định, điều này chỉ ra rằng mô hình có thể bị overfitting ( khi sử dụng quá nhiều dữ liệu huấn luyện.
* Nếu cả hai đều thấp và có xu hướng ổn định, mô hình có thể bị underfitting do không đủ khả năng học từ dữ liệu huấn luyện.

****

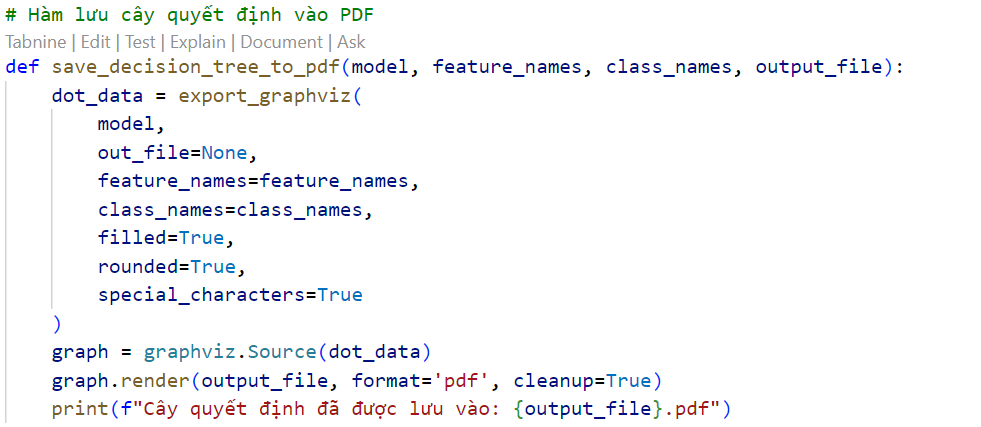
*Biểu đồ đường cong học tập của phương pháp ID3*

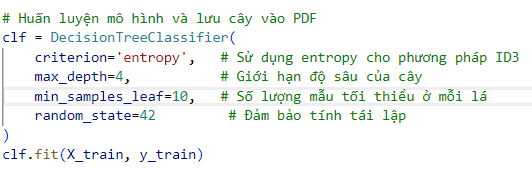
Đường cong học giúp xác định xem mô hình có học tốt từ dữ liệu huấn luyện và có thể tổng quát tốt cho dữ liệu kiểm tra hay không.

Đánh giá này quan trọng để đảm bảo mô hình không bị overfitting hoặc underfitting, điều này giúp mô hình có khả năng dự đoán tốt hơn trong thực tế.

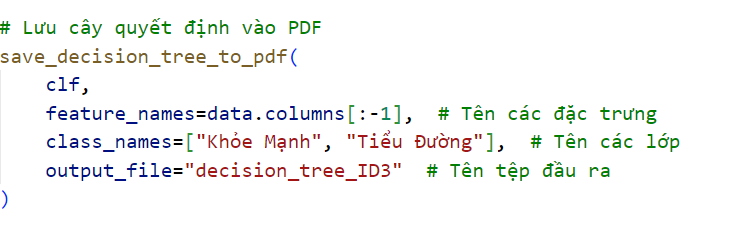
**Bước 4:** Trực Quan Hóa Cây Quyết Định

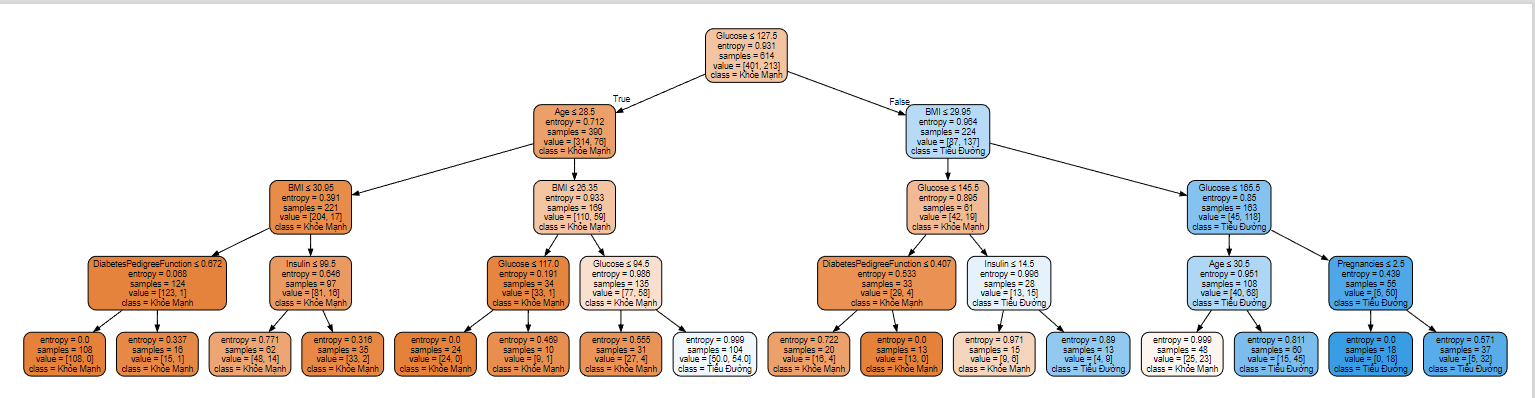
Một trong những điểm mạnh của thuật toán ID3 là khả năng trực quan hóa cây quyết định. Bằng cách sử dụng export\_graphviz và graphviz, chúng ta có thể vẽ cây quyết định để hiểu cách mô hình phân chia dữ liệu và ra quyết định.





Sau khi huấn luyện mô hình ID3, chúng ta có thể trực quan hóa cây quyết định để hiểu rõ hơn về cách mô hình phân chia các nhánh và đưa ra quyết định dựa trên các đặc trưng của dữ liệu (như mức đường huyết, huyết áp, BMI). Cây quyết định cho thấy các quyết định được đưa ra tại mỗi nút phân chia và các thuộc tính có vai trò quan trọng trong việc phân loại bệnh tiểu đường.



****

*Cây quyết định theo ID3*

Cây quyết định giúp người dùng hiểu cách mô hình phân chia dữ liệu và đưa ra quyết định. Mỗi nhánh trong cây thể hiện một điều kiện phân chia và các nhánh con thể hiện kết quả phân loại.

Trực quan hóa cây giúp kiểm tra tính logic và tính minh bạch trong quá trình ra quyết định của mô hình. Cây quyết định đơn giản, dễ hiểu và có thể áp dụng trực tiếp vào thực tế.

**Kết quả:**

Cây quyết định ID3 này sử dụng tiêu chí entropy để phân chia dữ liệu, dựa trên các thuộc tính như Glucose, BMI, Age, và DiabetesPedigreeFunction.

* Lớp gốc: Cây bắt đầu với điều kiện Glucose <= 127.5, phân loại dữ liệu thành "Khỏe Mạnh" (nếu đúng) và "Tiểu Đường" (nếu sai).
* Nhánh trái: Khi Glucose <= 127.5, cây tiếp tục phân chia theo Age <= 28.5. Nếu đúng, kết quả là "Khỏe Mạnh".
* Nhánh phải: Khi Glucose > 127.5, phân chia theo BMI <= 29.95. Nếu đúng, kết quả là "Tiểu Đường", nếu sai, tiếp tục phân chia theo các thuộc tính khác như SkinThickness.

Cây quyết định giúp phân loại người dùng thành "Khỏe Mạnh" hoặc "Tiểu Đường" dựa trên các đặc trưng quan trọng như Glucose và BMI.

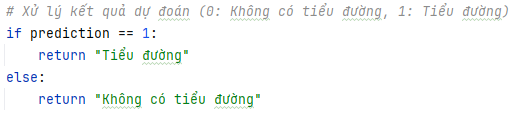
**Bước 5:** Xây dựng hàm dự đoán kết quả bệnh tiểu đường dựa trên cây quyết định đã huấn luyện theo ID3

Hàm predict\_diabetes nhận đầu vào là mô hình huấn luyện (model) và một danh sách các đặc trưng (input\_data) của bệnh nhân.

Sử dụng phương thức model.predict() để dự đoán giá trị lớp (0 hoặc 1) từ mô hình. Hàm predict() yêu cầu đầu vào dưới dạng một danh sách (hoặc mảng 2 chiều), nên input\_data phải được bọc trong một danh sách.

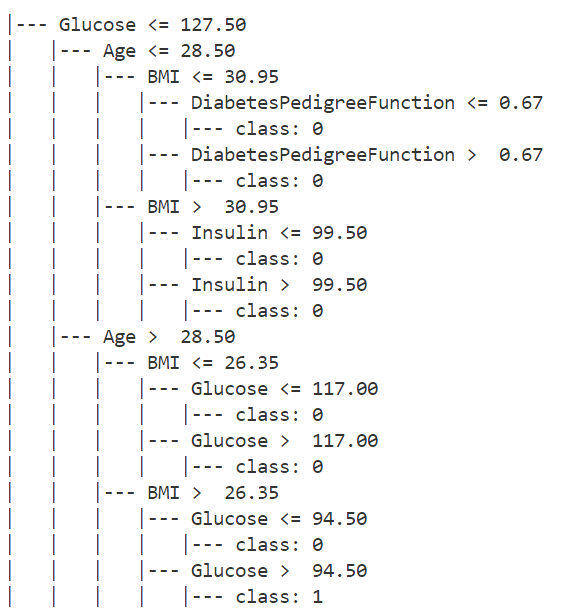


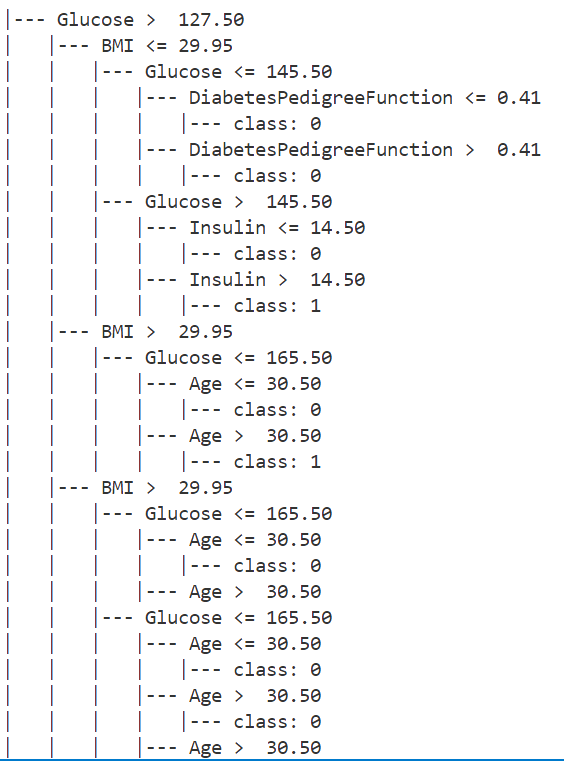
Nếu giá trị trả về là 1 (tương ứng với "Tiểu đường"), hàm sẽ trả về kết quả "Tiểu đường". Ngược lại, nếu là 0 (tương ứng với "Không có tiểu đường"), hàm sẽ trả về "Không có tiểu đường".

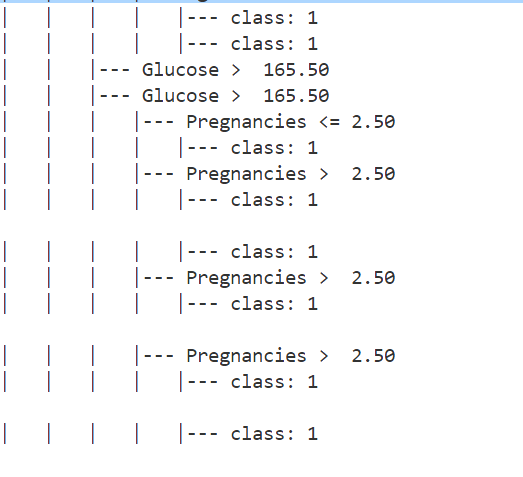


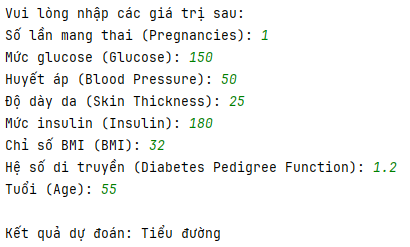
Sau đó, dùng hàm predict\_diabetes\_from\_input yêu cầu người dùng nhập các giá trị của các đặc trưng (đặc điểm y tế) của bệnh nhân, sau đó sử dụng mô hình đã huấn luyện để dự đoán bệnh tiểu đường. Kết quả dự đoán sẽ được in ra màn hình cho người dùng biết nguy cơ mắc bệnh tiểu đường của mình.

Với cây quyết định dưới dạng văn bản như sau:









Quy trình phân loại:

* Glucose <= 127.50 (False, vì 150 > 127.50)

-> Chuyển đến nhánh "Glucose > 127.50"

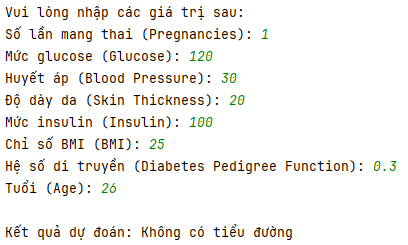
* BMI <= 29.95 (False, vì 32 > 29.95)

-> Chuyển đến nhánh "BMI > 29.95"

* Glucose <= 165.50 (True, vì 150 <= 165.50)

-> Age <= 30.50 (False, vì 55 > 30.50)

Kết quả: Tiểu đường (class = 1).



Quy trình phân loại:

* Glucose <= 127.50 (True, vì 120 <= 127.50)
* Age <= 28.50 (True, vì 26 <= 28.50)
* BMI <= 30.95 (True, vì 25 <= 30.95)
* DiabetesPedigreeFunction <= 0.67 (True, vì 0.3 <= 0.67)

Kết quả: Không tiểu đường (class = 0).

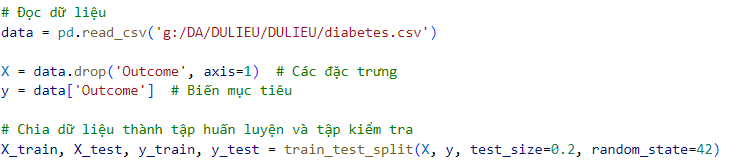
### 4.3.3. Áp dụng thuật toán C4.5:

C4.5 là một phiên bản cải tiến của ID3, khắc phục được nhiều hạn chế của ID3 và trở thành một trong những thuật toán cây quyết định phổ biến nhất. Thay vì chỉ sử dụng Entropy và Information Gain, C4.5 bổ sung các tính năng mạnh mẽ như xử lý dữ liệu liên tục và xử lý giá trị thiếu. C4.5 là sự lựa chọn lý tưởng khi làm việc với dữ liệu hỗn hợp, phức tạp, hoặc khi muốn xây dựng một mô hình cây quyết định có tính tổng quát cao hơn so với ID3.

**Bước 1:** Chuẩn bị Dữ Liệu

Trước khi triển khai thuật toán C4.5, chúng ta cần đảm bảo rằng bộ dữ liệu đã được xử lý và chuẩn bị kỹ càng. Quy trình chuẩn bị dữ liệu đã được thực hiện ở các bước trước như:

* Đọc dữ liệu từ tệp CSV.
* Kiểm tra và làm sạch dữ liệu (bao gồm việc xử lý các giá trị thiếu, kiểm tra kiểu dữ liệu của các cột, và thống kê mô tả các thuộc tính).
* Mã hóa các nhãn (Outcome và BMI\_Category) thành các giá trị số để phù hợp với các thuật toán học máy.
* Tách bộ dữ liệu thành các đặc trưng (features) và nhãn (labels).
* Chia bộ dữ liệu thành hai phần: 80% cho huấn luyện (training) và 20% cho kiểm tra (testing) với việc sử dụng hàm train\_test\_split từ thư viện sklearn.



**Bước 2:** Xây Dựng Mô Hình C4.5

C4.5 là phiên bản mở rộng của ID3, sử dụng tiêu chí Gain Ratio thay vì chỉ dựa vào Information Gain. Ngoài ra, thuật toán C4.5 hỗ trợ phân chia dữ liệu liên tục và xử lý các giá trị thiếu.

* Trong Python, chúng ta có thể giả lập mô hình C4.5 bằng cách điều chỉnh các tham số trong lớp DecisionTreeClassifier để bắt chước các tính năng của thuật toán:
* Dữ liệu liên tục: Phân chia các thuộc tính liên tục thành các điểm ngưỡng tự động.
* criterion='entropy': Dựa vào Entropy để đo lường độ hỗn loạn và kết hợp Gain Ratio để tối ưu hóa việc chọn thuộc tính.
* pruning (cắt tỉa cây): Tránh overfitting bằng cách kiểm soát các tham số như min\_samples\_split và min\_samples\_leaf.

**Bước 3:** Đánh Giá Mô Hình

Sau khi huấn luyện mô hình, bước tiếp theo là đánh giá mô hình này trên tập kiểm tra X\_test và y\_test. Một số cách đánh giá phổ biến bao gồm:

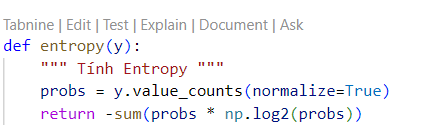
* Độ chính xác (Accuracy): Tính toán tỷ lệ phần trăm dự đoán đúng trên tổng số mẫu trong tập kiểm tra.
* Ma trận nhầm lẫn (Confusion Matrix): Cung cấp cái nhìn chi tiết về số lượng dự đoán đúng và sai đối với mỗi lớp.
* ROC Learning curve: Hiển thị sự thay đổi độ chính xác của mô hình theo kích thước của tập huấn luyện.

Huấn luyện mô hình:

* Tính Gain Ratio và Huấn Luyện Mô Hình

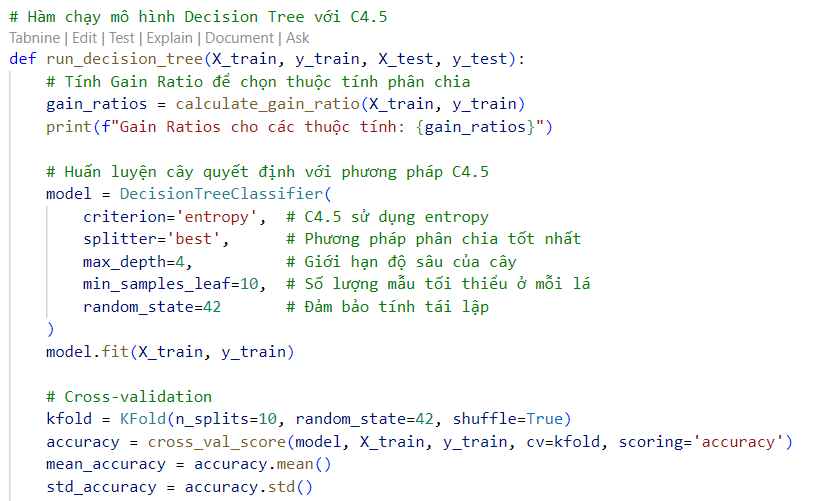
Tính Gain Ratio: Ta sử dụng hàm calculate\_gain\_ratio() để tính toán Gain Ratio cho từng thuộc tính trong tập huấn luyện. Việc này giúp lựa chọn thuộc tính tốt nhất để phân chia tại mỗi nút trong cây quyết định, theo phương pháp C4.5.



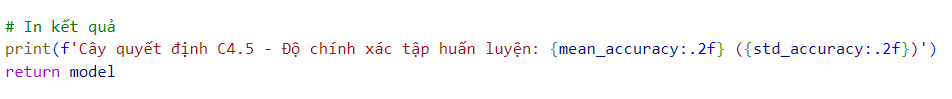


**Đánh Giá Độ Chính Xác**

Sau khi huấn luyện xong, chúng ta sẽ sử dụng accuracy\_score để đánh giá độ chính xác của mô hình trên tập kiểm tra.



Sau khi huấn luyện và dự đoán với mô hình C4.5 trên tập kiểm tra, chúng ta sẽ in ra độ chính xác của mô hình. Độ chính xác là tỷ lệ dự đoán đúng trên tổng số mẫu trong tập kiểm tra.



**Kết quả:**

* Accuracy là giá trị từ 0 đến 1, càng gần 1 thì mô hình càng chính xác.
* Nếu kết quả là 0.75, có nghĩa là mô hình dự đoán chính xác 75% số trường hợp trong tập kiểm tra.



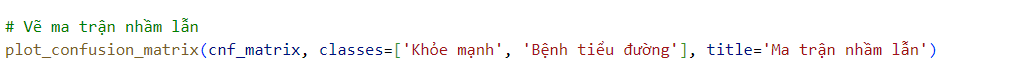
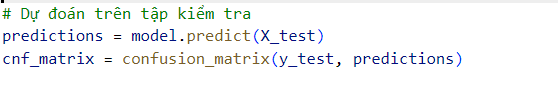
Kết quả này giúp người dùng hiểu mức độ chính xác mà mô hình CART đạt được khi dự đoán kết quả bệnh tiểu đường từ tập kiểm tra.

Kết quả càng cao, mô hình càng hiệu quả trong việc phân loại các trường hợp mắc bệnh tiểu đường và không mắc bệnh.

**Ma Trận Nhầm Lẫn (Confusion Matrix)**

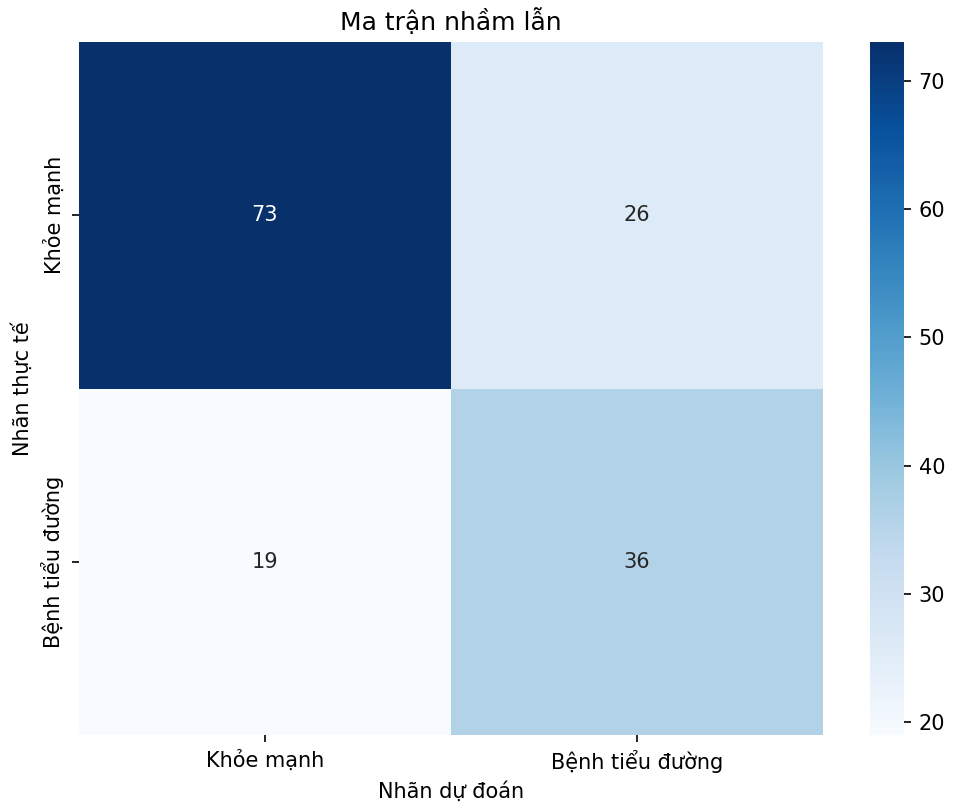
Ma trận nhầm lẫn giúp chúng ta hiểu rõ hơn về hiệu quả phân loại của mô hình bằng cách chỉ ra số lượng các trường hợp dự đoán đúng và sai. Chúng ta có thể vẽ ma trận nhầm lẫn bằng cách sử dụng ConfusionMatrixDisplay từ thư viện sklearn.





Ma trận nhầm lẫn (confusion matrix) sẽ thể hiện số lượng dự đoán đúng và sai của mô hình đối với mỗi lớp (Diabetes vs No Diabetes). Nó bao gồm 4 giá trị:

* True Positive (TP): Số trường hợp thực sự mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Positive (FP): Số trường hợp không mắc bệnh tiểu đường nhưng mô hình lại dự đoán là mắc bệnh.
* True Negative (TN): Số trường hợp không mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Negative (FN): Số trường hợp mắc bệnh tiểu đường nhưng mô hình lại dự đoán không mắc bệnh.

****

*Biểu đồ ma trận nhầm lẫn của phương pháp C4.5*

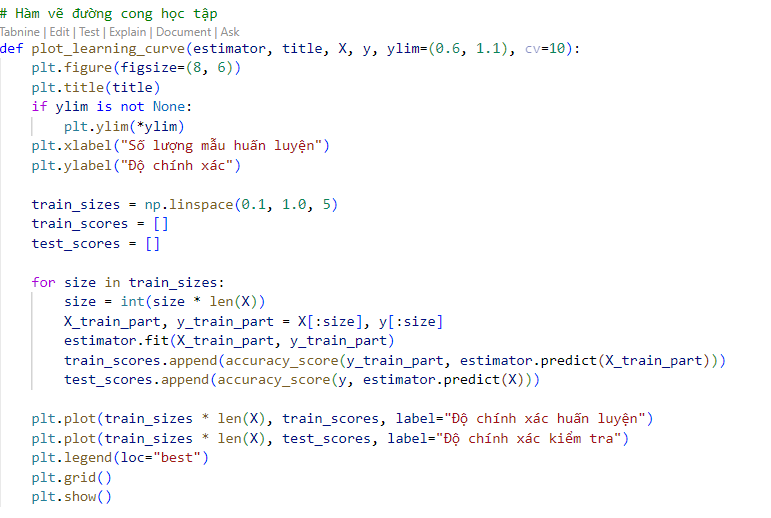
* **73** là số trường hợp không mắc bệnh tiểu đường mà mô hình dự đoán đúng.
* **26** là số trường hợp không mắc bệnh nhưng mô hình lại dự đoán là mắc bệnh.
* **19** là số trường hợp mắc bệnh nhưng mô hình lại dự đoán là không mắc bệnh.
* **36** là số trường hợp mắc bệnh mà mô hình dự đoán đúng.

Ma trận nhầm lẫn cho phép chúng ta hiểu rõ hơn về việc mô hình xử lý đúng và sai như thế nào đối với từng lớp.

Càng nhiều giá trị True Positive và True Negative, mô hình càng chính xác trong việc phân loại đúng các trường hợp mắc bệnh tiểu đường.

**Đường Cong Học (Learning Curve)**

Để đánh giá sự thay đổi độ chính xác khi tăng kích thước của tập huấn luyện, chúng ta có thể vẽ đường cong học. Điều này giúp nhận diện liệu mô hình có bị overfitting hay underfitting khi sử dụng tập huấn luyện lớn hơn.

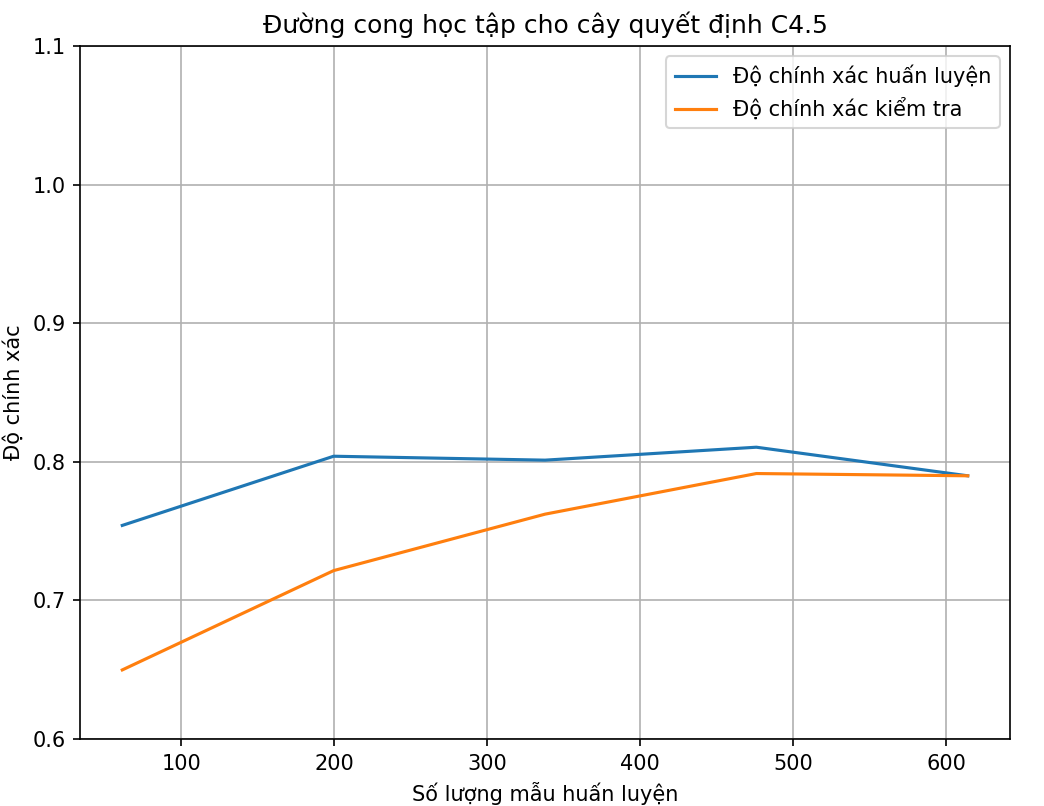


**Kết quả:**

* Training score: Đo lường độ chính xác của mô hình trên tập huấn luyện, có thể sẽ tăng lên khi số lượng mẫu huấn luyện tăng.
* Validation score: Đo lường độ chính xác trên tập kiểm tra, có thể sẽ bắt đầu ổn định hoặc giảm nếu mô hình bị overfitting.

**Ví dụ về đường cong:**

* Nếu training score tăng và validation score ổn định hoặc giảm sau một điểm nhất định, điều này chỉ ra rằng mô hình có thể bị overfitting khi sử dụng quá nhiều dữ liệu huấn luyện.
* Nếu both scores đều thấp và có xu hướng ổn định, mô hình có thể bị underfitting do không đủ khả năng học từ dữ liệu huấn luyện.

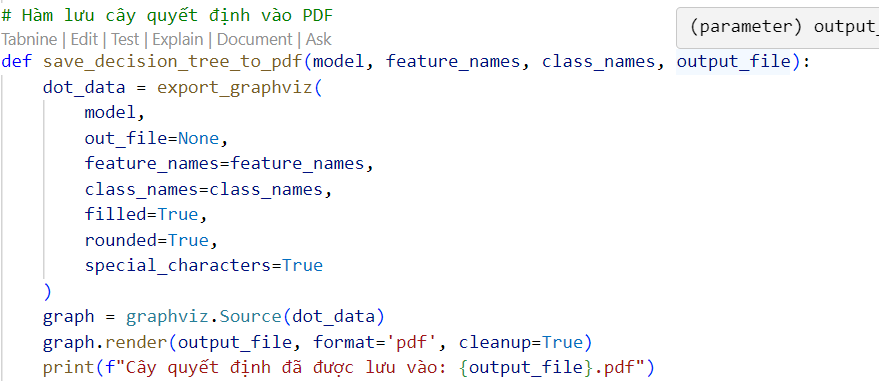
****

*Biểu đồ đường cong học tập của phương pháp C4.5*

* Đường cong học giúp xác định xem mô hình có học tốt từ dữ liệu huấn luyện và có thể tổng quát tốt cho dữ liệu kiểm tra hay không.
* Đánh giá này quan trọng để đảm bảo mô hình không bị overfitting hoặc underfitting, điều này giúp mô hình có khả năng dự đoán tốt hơn trong thực tế.

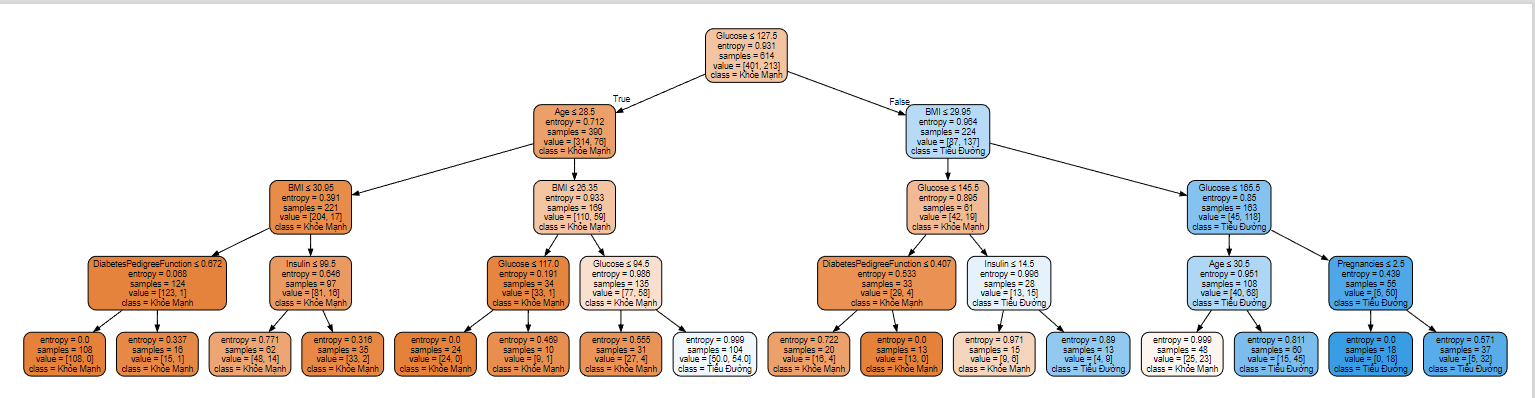
**Bước 4:** Trực Quan Hóa Cây Quyết Định

Một trong những điểm mạnh của thuật toán CART là khả năng trực quan hóa cây quyết định. Bằng cách sử dụng export\_graphviz và graphviz, chúng ta có thể vẽ cây quyết định để hiểu cách mô hình phân chia dữ liệu và ra quyết định.



Sau khi huấn luyện mô hình C4.5, chúng ta có thể trực quan hóa cây quyết định để hiểu rõ hơn về cách mô hình phân chia các nhánh và đưa ra quyết định dựa trên các đặc trưng của dữ liệu (như mức đường huyết, huyết áp, BMI). Cây quyết định cho thấy các quyết định được đưa ra tại mỗi nút phân chia và các thuộc tính có vai trò quan trọng trong việc phân loại bệnh tiểu đường.



****

*Cây quyết định theo C4.5*

Cây quyết định giúp người dùng hiểu cách mô hình phân chia dữ liệu và đưa ra quyết định. Mỗi nhánh trong cây thể hiện một điều kiện phân chia và các nhánh con thể hiện kết quả phân loại.

Trực quan hóa cây giúp kiểm tra tính logic và tính minh bạch trong quá trình ra quyết định của mô hình. Cây quyết định đơn giản, dễ hiểu và có thể áp dụng trực tiếp vào thực tế.

**Kết quả:**

Cây quyết định C4.5 sử dụng tiêu chí gain ratio để phân chia dữ liệu, tương tự như cách ID3 sử dụng entropy, nhưng C4.5 cải tiến bằng cách tính toán tỷ lệ lợi ích của việc phân chia các thuộc tính và chọn thuộc tính có gain ratio cao nhất.

* Lớp gốc: Cây bắt đầu với điều kiện Glucose <= 127.5, phân loại dữ liệu thành "Khỏe Mạnh" (nếu đúng) và "Tiểu Đường" (nếu sai).
* Nhánh trái: Khi Glucose <= 127.5, cây tiếp tục phân chia theo Age <= 28.5. Nếu đúng, kết quả là "Khỏe Mạnh".
* Nhánh phải: Khi Glucose > 127.5, phân chia theo BMI <= 30.95. Nếu đúng, kết quả là "Tiểu Đường", nếu sai, tiếp tục phân chia theo các thuộc tính khác như SkinThickness.

Cây quyết định giúp phân loại người dùng thành "Khỏe Mạnh" hoặc "Tiểu Đường" dựa trên các đặc trưng quan trọng như Glucose và BMI. C4.5 cải tiến ID3 bằng cách sử dụng gain ratio, giúp tránh sự thiên lệch đối với các thuộc tính có nhiều giá trị.

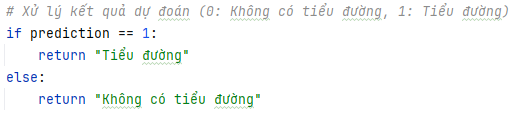
**Bước 5:** Xây dựng hàm dự đoán kết quả bệnh tiểu đường dựa trên cây quyết định đã huấn luyện theo C4.5

Hàm predict\_diabetes nhận đầu vào là mô hình huấn luyện (model) và một danh sách các đặc trưng (input\_data) của bệnh nhân.

Sử dụng phương thức model.predict() để dự đoán giá trị lớp (0 hoặc 1) từ mô hình. Hàm predict() yêu cầu đầu vào dưới dạng một danh sách (hoặc mảng 2 chiều), nên input\_data phải được bọc trong một danh sách.

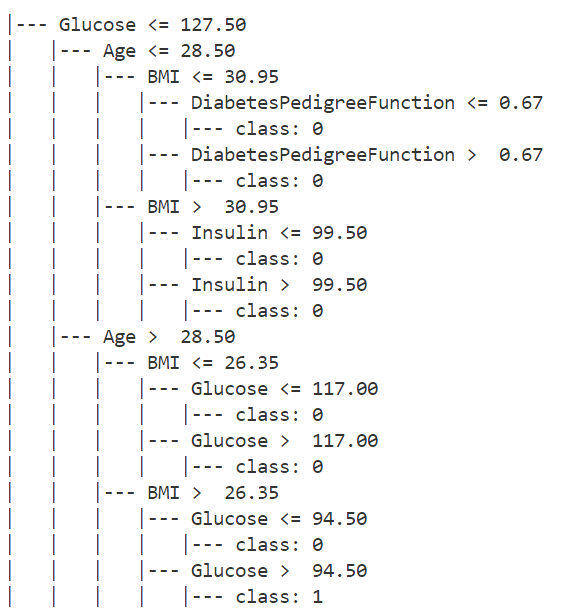


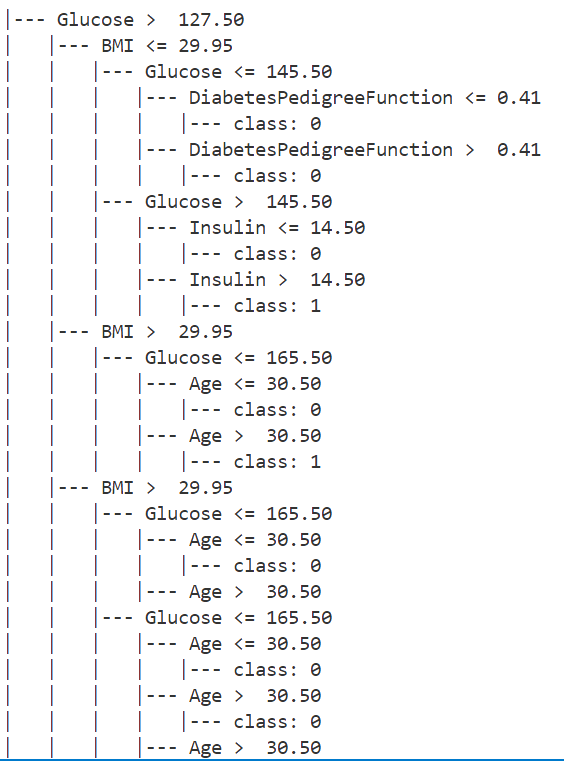
Nếu giá trị trả về là 1 (tương ứng với "Tiểu đường"), hàm sẽ trả về kết quả "Tiểu đường". Ngược lại, nếu là 0 (tương ứng với "Không có tiểu đường"), hàm sẽ trả về "Không có tiểu đường".

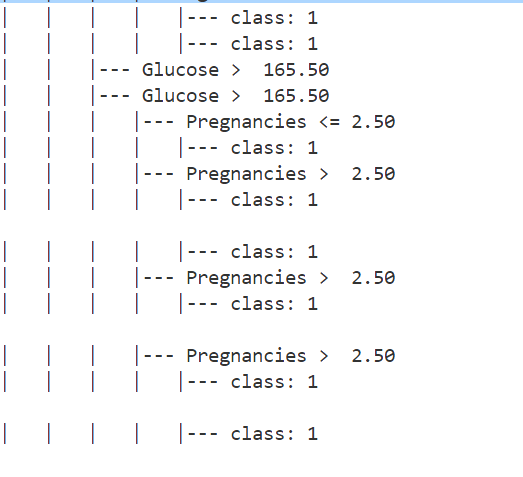


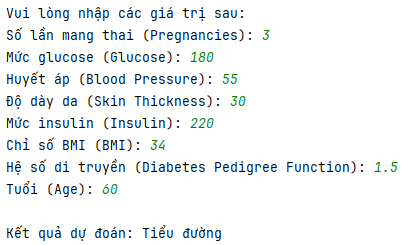
Sau đó, dùng hàm predict\_diabetes\_from\_input yêu cầu người dùng nhập các giá trị của các đặc trưng (đặc điểm y tế) của bệnh nhân, sau đó sử dụng mô hình đã huấn luyện để dự đoán bệnh tiểu đường. Kết quả dự đoán sẽ được in ra màn hình cho người dùng biết nguy cơ mắc bệnh tiểu đường của mình.

Với cây quyết định dưới dạng văn bản như sau:









Quy trình phân loại sẽ như sau:

* Glucose <= 127.50 (False, vì 180 > 127.50)

-> Chuyển đến nhánh "Glucose > 127.50"

* BMI <= 29.95 (False, vì 34 > 29.95)

-> Chuyển đến nhánh "BMI > 29.95"

* Glucose <= 165.50 (False, vì 180 > 165.50)

Kết quả: Tiểu đường (class = 1).



Glucose <= 127.50 (True, vì 120 <= 127.50)

* Age <= 28.50 (True, vì 24 <= 28.50)
* BMI <= 30.95 (True, vì 26 <= 30.95)
* DiabetesPedigreeFunction <= 0.67 (True, vì 0.4 <= 0.67)

Kết quả: Không tiểu đường (class = 0).

### 4.3.4. Áp dụng thuật toán CART:

Trong phần này, chúng ta sẽ đi vào chi tiết cách áp dụng thuật toán CART (Classification and Regression Trees) để xây dựng mô hình dự đoán bệnh tiểu đường. CART là một thuật toán cây quyết định mạnh mẽ, có thể được sử dụng cho cả bài toán phân loại và hồi quy. Đặc biệt trong bài toán phân loại bệnh tiểu đường, CART sẽ chia dữ liệu thành các nhánh theo cách giúp tối ưu hóa việc phân chia giữa các lớp "Diabetes" và "No Diabetes".

**Bước 1:** Chuẩn bị Dữ Liệu

Trước khi triển khai thuật toán CART, chúng ta cần đảm bảo rằng bộ dữ liệu đã được xử lý và chuẩn bị kỹ càng. Quy trình chuẩn bị dữ liệu đã được thực hiện ở các bước trước như:

* Đọc dữ liệu từ tệp CSV.
* Kiểm tra và làm sạch dữ liệu (bao gồm việc xử lý các giá trị thiếu, kiểm tra kiểu dữ liệu của các cột, và thống kê mô tả các thuộc tính).
* Mã hóa các nhãn (Outcome và BMI) thành các giá trị số để phù hợp với các thuật toán học máy.
* Tách bộ dữ liệu thành các đặc trưng (features) và nhãn (labels).
* Chia bộ dữ liệu thành hai phần: 80% cho huấn luyện (training) và 20% cho kiểm tra (testing) với việc sử dụng hàm train\_test\_split từ thư viện sklearn.





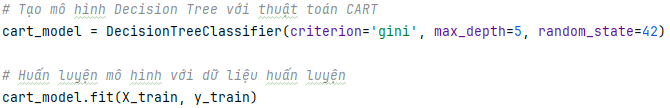
**Bước 2:** Xây Dựng Mô Hình CART

Thuật toán CART sử dụng tiêu chí "Gini Impurity" để phân chia dữ liệu tại mỗi nút trong cây. Gini Impurity đo lường mức độ "không thuần nhất" của các lớp trong một nhánh. Cây quyết định sẽ tiếp tục phân chia cho đến khi đạt được độ sâu nhất định (hoặc đến khi số lượng mẫu tại một nhánh nhỏ hơn ngưỡng quy định).

Để xây dựng mô hình CART trong Python, chúng ta sử dụng lớp DecisionTreeClassifier từ thư viện sklearn. Một số tham số quan trọng cần thiết là:

* criterion='gini': Sử dụng Gini Impurity làm tiêu chí để phân chia các nhánh.
* max\_depth: Giới hạn độ sâu của cây quyết định, nhằm tránh overfitting. Ta có thể chọn một giá trị hợp lý cho độ sâu cây (ví dụ: 5).
* random\_state: Để đảm bảo tính tái lặp của mô hình, chúng ta sử dụng tham số này để cố định các giá trị ngẫu nhiên.



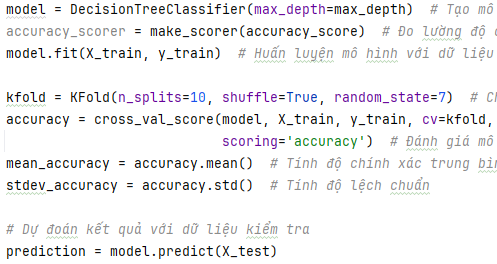


**Bước 3:** Đánh Giá Mô Hình

Sau khi huấn luyện mô hình, bước tiếp theo là đánh giá mô hình này trên tập kiểm tra X\_test và y\_test. Một số cách đánh giá phổ biến bao gồm:

* Độ chính xác (Accuracy): Tính toán tỷ lệ phần trăm dự đoán đúng trên tổng số mẫu trong tập kiểm tra.
* Ma trận nhầm lẫn (Confusion Matrix): Cung cấp cái nhìn chi tiết về số lượng dự đoán đúng và sai đối với mỗi lớp.
* Learning curve: Hiển thị sự thay đổi độ chính xác của mô hình theo kích thước của tập huấn luyện.
* **Đánh Giá Độ Chính Xác**

Sau khi huấn luyện xong, chúng ta sẽ sử dụng accuracy\_score để đánh giá độ chính xác của mô hình trên tập kiểm tra.



Sau khi huấn luyện và dự đoán với mô hình CART trên tập kiểm tra, chúng ta sẽ in ra độ chính xác của mô hình. Độ chính xác là tỷ lệ dự đoán đúng trên tổng số mẫu trong tập kiểm tra.



Kết quả:

* Accuracy là giá trị từ 0 đến 1, càng gần 1 thì mô hình càng chính xác.
* Nếu kết quả là 0.85, có nghĩa là mô hình dự đoán chính xác 85% số trường hợp trong tập kiểm tra.



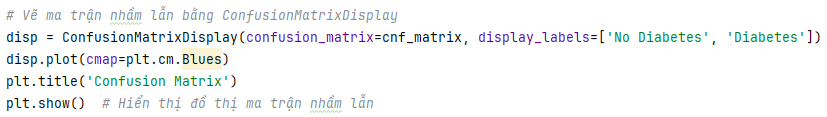
Kết quả này giúp người dùng hiểu mức độ chính xác mà mô hình CART đạt được khi dự đoán kết quả bệnh tiểu đường từ tập kiểm tra.

Kết quả càng cao, mô hình càng hiệu quả trong việc phân loại các trường hợp mắc bệnh tiểu đường và không mắc bệnh.

* **Ma Trận Nhầm Lẫn (Confusion Matrix)**

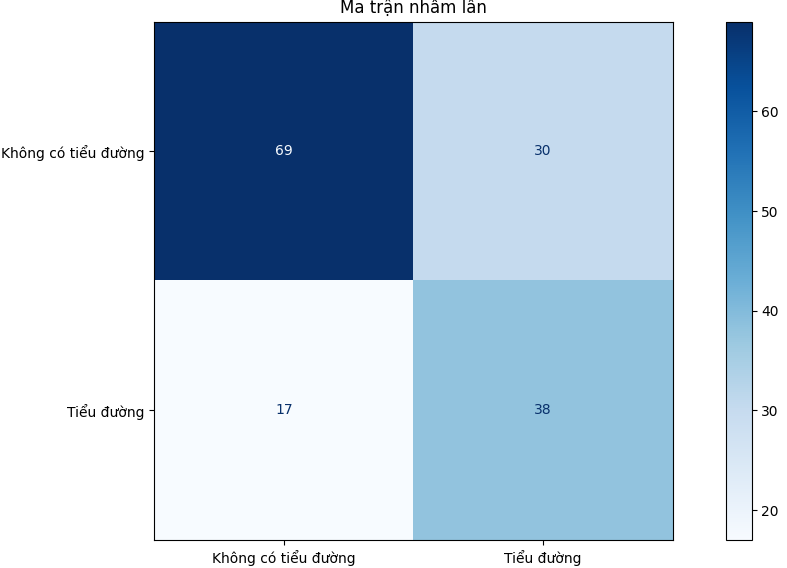
Ma trận nhầm lẫn giúp chúng ta hiểu rõ hơn về hiệu quả phân loại của mô hình bằng cách chỉ ra số lượng các trường hợp dự đoán đúng và sai. Chúng ta có thể vẽ ma trận nhầm lẫn bằng cách sử dụng ConfusionMatrixDisplay từ thư viện sklearn.





Ma trận nhầm lẫn (confusion matrix) sẽ thể hiện số lượng dự đoán đúng và sai của mô hình đối với mỗi lớp (Diabetes vs No Diabetes). Nó bao gồm 4 giá trị:

* True Positive (TP): Số trường hợp thực sự mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Positive (FP): Số trường hợp không mắc bệnh tiểu đường nhưng mô hình lại dự đoán là mắc bệnh.
* True Negative (TN): Số trường hợp không mắc bệnh tiểu đường và mô hình dự đoán đúng.
* False Negative (FN): Số trường hợp mắc bệnh tiểu đường nhưng mô hình lại dự đoán không mắc bệnh.



*Biểu đồ ma trận nhầm lẫn của phương pháp CART*

* **69** là số trường hợp không mắc bệnh tiểu đường mà mô hình dự đoán đúng.
* **30** là số trường hợp không mắc bệnh nhưng mô hình lại dự đoán là mắc bệnh.
* **17** là số trường hợp mắc bệnh nhưng mô hình lại dự đoán là không mắc bệnh.
* **38** là số trường hợp mắc bệnh mà mô hình dự đoán đúng.

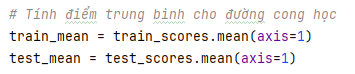
Ma trận nhầm lẫn cho phép chúng ta hiểu rõ hơn về việc mô hình xử lý đúng và sai như thế nào đối với từng lớp.

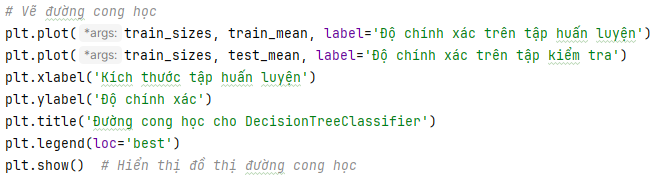
Càng nhiều giá trị True Positive và True Negative, mô hình càng chính xác trong việc phân loại đúng các trường hợp mắc bệnh tiểu đường.

* **Đường Cong Học (Learning Curve)**

Để đánh giá sự thay đổi độ chính xác khi tăng kích thước của tập huấn luyện, chúng ta có thể vẽ đường cong học. Điều này giúp nhận diện liệu mô hình có bị overfitting hay underfitting khi sử dụng tập huấn luyện lớn hơn.







**Kết quả:**

* Độ chính xác trên tập huấn luyện: Đo lường độ chính xác của mô hình trên tập huấn luyện, có thể sẽ tăng lên khi số lượng mẫu huấn luyện tăng.
* Độ chính xác trên tập kiểm tra: Đo lường độ chính xác trên tập kiểm tra, có thể sẽ bắt đầu ổn định hoặc giảm nếu mô hình bị overfitting.

**Ví dụ về đường cong:**

* Nếu độ chính xác trên tập huấn luyện tăng và độ chính xác trên tập kiểm tra ổn định hoặc giảm sau một điểm nhất định, điều này chỉ ra rằng mô hình có thể bị overfitting (hiện tượng quá khớp) khi sử dụng quá nhiều dữ liệu huấn luyện.
* Nếu cả hai đều thấp và có xu hướng ổn định, mô hình có thể bị underfitting do không đủ khả năng học từ dữ liệu huấn luyện.

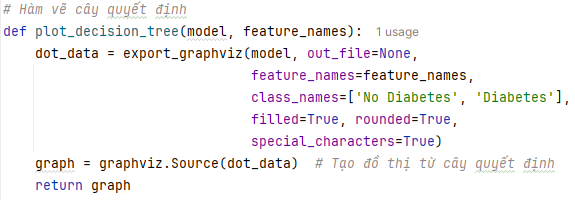


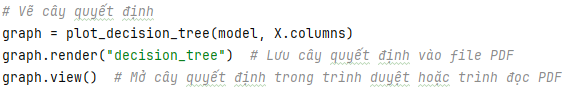
*Biểu đồ đường cong học tập của phương pháp CART*

* Đường cong học giúp xác định xem mô hình có học tốt từ dữ liệu huấn luyện và có thể tổng quát tốt cho dữ liệu kiểm tra hay không.
* Đánh giá này quan trọng để đảm bảo mô hình không bị overfitting hoặc underfitting, điều này giúp mô hình có khả năng dự đoán tốt hơn trong thực tế.

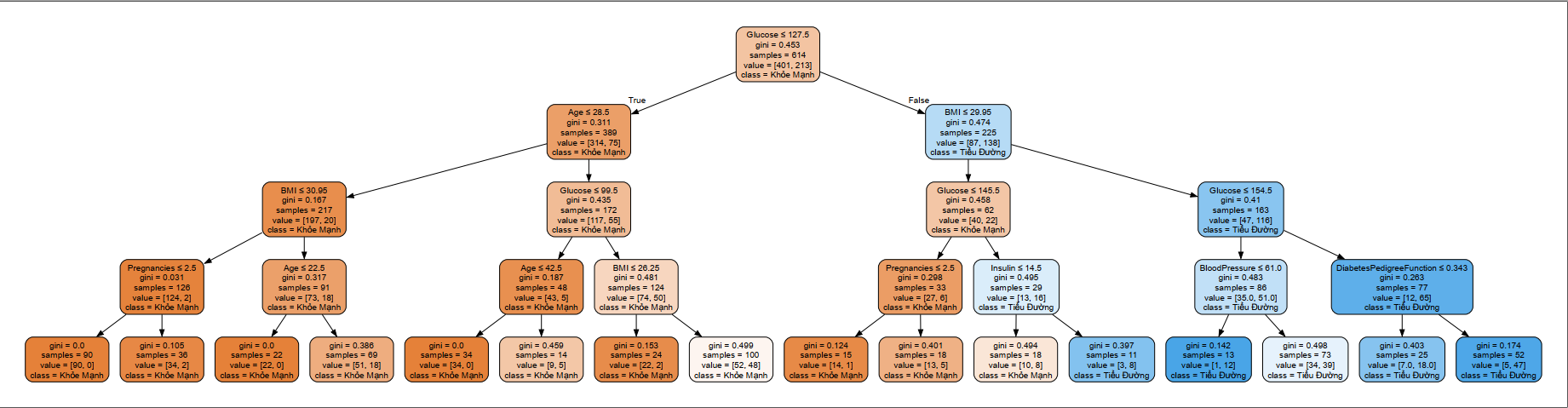
**Bước 4:** Trực Quan Hóa Cây Quyết Định

Một trong những điểm mạnh của thuật toán CART là khả năng trực quan hóa cây quyết định. Bằng cách sử dụng export\_graphviz và graphviz, chúng ta có thể vẽ cây quyết định để hiểu cách mô hình phân chia dữ liệu và ra quyết định.





Sau khi huấn luyện mô hình CART, chúng ta có thể trực quan hóa cây quyết định để hiểu rõ hơn về cách mô hình phân chia các nhánh và đưa ra quyết định dựa trên các đặc trưng của dữ liệu (như mức đường huyết, huyết áp, BMI). Cây quyết định cho thấy các quyết định được đưa ra tại mỗi nút phân chia và các thuộc tính có vai trò quan trọng trong việc phân loại bệnh tiểu đường.



Cây quyết định giúp người dùng hiểu cách mô hình phân chia dữ liệu và đưa ra quyết định. Mỗi nhánh trong cây thể hiện một điều kiện phân chia và các nhánh con thể hiện kết quả phân loại.

Trực quan hóa cây giúp kiểm tra tính logic và tính minh bạch trong quá trình ra quyết định của mô hình. Cây quyết định đơn giản, dễ hiểu và có thể áp dụng trực tiếp vào thực tế.

**Kết quả:**

Cây quyết định có thể sẽ hiển thị một số điều kiện phân chia:

* Nếu Glucose <= 127,5, mô hình phân loại là "Diabetes".
* Nếu BMI <= 29,95 và BloodPressure <=55, mô hình phân loại là "No Diabetes".

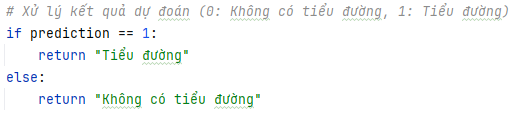
**Bước 5:** Xây dựng hàm dự đoán kết quả bệnh tiểu đường dựa trên cây quyết định đã huấn luyện theo CART

Hàm predict\_diabetes nhận đầu vào là mô hình huấn luyện (model) và một danh sách các đặc trưng (input\_data) của bệnh nhân.

Sử dụng phương thức model.predict() để dự đoán giá trị lớp (0 hoặc 1) từ mô hình. Hàm predict() yêu cầu đầu vào dưới dạng một danh sách (hoặc mảng 2 chiều), nên input\_data phải được bọc trong một danh sách.



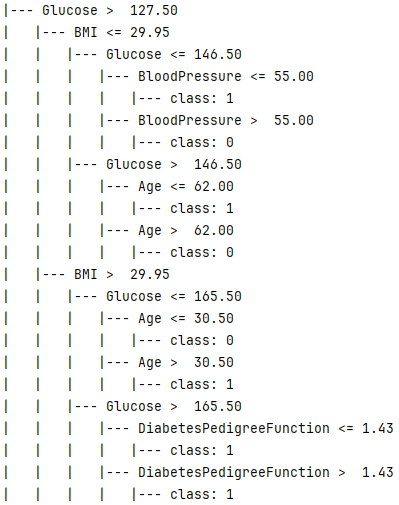
Nếu giá trị trả về là 1 (tương ứng với "Tiểu đường"), hàm sẽ trả về kết quả "Tiểu đường". Ngược lại, nếu là 0 (tương ứng với "Không có tiểu đường"), hàm sẽ trả về "Không có tiểu đường".

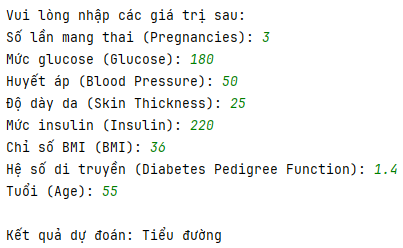


Sau đó, dùng hàm predict\_diabetes\_from\_input yêu cầu người dùng nhập các giá trị của các đặc trưng (đặc điểm y tế) của bệnh nhân, sau đó sử dụng mô hình đã huấn luyện để dự đoán bệnh tiểu đường. Kết quả dự đoán sẽ được in ra màn hình cho người dùng biết nguy cơ mắc bệnh tiểu đường của mình.

Với cây quyết định dưới dạng văn bản như sau:







Quy trình phân loại sẽ như sau:

* Glucose <= 127.50 (False, vì 180 > 127.50)

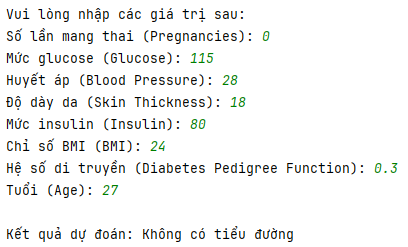
-> Chuyển đến nhánh "Glucose > 127.50"

* BMI <= 29.95 (False, vì 36 > 29.95)

-> Chuyển đến nhánh "BMI > 29.95"

* Glucose <= 165.50 (False, vì 180 > 165.50)

Kết quả: Tiểu đường (class = 1).



Quy trình phân loại sẽ như sau:

* Glucose <= 127.50 (True, vì 115 <= 127.50)
* Age <= 28.50 (True, vì 27 <= 28.50)
* BMI <= 30.95 (True, vì 24 <= 30.95)
* DiabetesPedigreeFunction <= 0.67 (True, vì 0.3 <= 0.67)

Kết quả: Không tiểu đường (class = 0).

# 

# CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## 5.1. Tổng quan kết quả đạt được:

Trong quá trình nghiên cứu và xây dựng mô hình dự đoán bệnh tiểu đường, nhóm chúng em đã đạt được một số kết quả quan trọng. Đầu tiên, quá trình tiền xử lý dữ liệu được thực hiện đầy đủ, giúp loại bỏ các giá trị thiếu và chuẩn hóa các đặc tính dữ liệu, đảm bảo tính đồng nhất. Mô hình cây quyết định đã được chọn làm mô hình chính để phân tích mối quan hệ giữa các yếu tố y tế như glucose, huyết áp, BMI và nguy cơ mắc bệnh tiểu đường.

Kết quả cho thấy mô hình cây quyết định có khả năng phân loại bệnh tiểu đường với độ chính xác tương đối cao. Cây quyết định sử dụng các thuộc tính như glucose, BMI và các yếu tố y tế khác để phân loại người dùng thành nhóm "Khỏe Mạnh" hoặc "Tiểu Đường". Mô hình này giúp xác định các yếu tố quan trọng liên quan đến nguy cơ mắc bệnh, đồng thời cung cấp một giải pháp hiệu quả cho việc dự đoán nguy cơ mắc bệnh tiểu đường trong các hệ thống y tế.

Từ đó, chúng em đã có thể xây dựng một hệ thống dự đoán nguy cơ mắc bệnh tiểu đường hiệu quả, có thể ứng dụng trong các hệ thống y tế và hỗ trợ các bác sĩ trong việc đưa ra quyết định điều trị kịp thời.

## 5.2. Các hạn chế và rủi ro gặp phải:

Mặc dù đạt được kết quả khả quan, nhóm em cũng gặp phải một số hạn chế và rủi ro trong quá trình nghiên cứu. Dữ liệu thu thập được có thể không đại diện đầy đủ cho toàn bộ dân số, dẫn đến mô hình có thể không phản ánh chính xác sự đa dạng của các yếu tố nguy cơ mắc bệnh tiểu đường. Các yếu tố như độ tuổi, di truyền, thói quen ăn uống và mức độ hoạt động thể chất chưa được đưa vào mô hình, gây thiếu sót trong dự đoán.

Bên cạnh đó, việc lựa chọn mô hình cây quyết định mặc dù đơn giản và dễ hiểu, nhưng vẫn có thể gặp phải các vấn đề như dễ bị overfitting nếu dữ liệu quá phức tạp hoặc không đủ đa dạng. Cây quyết định cũng có thể yêu cầu nhiều tài nguyên tính toán nếu cần xây dựng cây sâu hoặc phức tạp.

Cuối cùng, một vấn đề quan trọng là tính khả thi trong việc triển khai mô hình vào các hệ thống y tế hiện tại. Việc tích hợp mô hình vào các cơ sở hạ tầng y tế có thể gặp phải các khó khăn về vấn đề dữ liệu, bảo mật và quyền riêng tư của bệnh nhân.

## 5.3. Định hướng phát triển:

Trong tương lai, một số hướng phát triển sẽ được nghiên cứu và triển khai để cải thiện chất lượng và hiệu quả của mô hình dự đoán bệnh tiểu đường. Một trong những hướng đi quan trọng là thu thập và sử dụng thêm dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như thông tin di truyền và thói quen sống, để tăng tính chính xác và đầy đủ của mô hình.

Bên cạnh đó, để tăng tính hiệu quả của mô hình này, việc áp dụng các phương pháp học sâu (Deep Learning) có thể giúp cải thiện khả năng nhận diện các mẫu phức tạp trong dữ liệu y tế. Các mô hình học sâu như mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) hoặc mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) có thể được thử nghiệm để đạt được hiệu suất cao hơn trong việc dự đoán.

Ngoài ra, việc triển khai hệ thống dự đoán vào môi trường thực tế là một mục tiêu quan trọng. Các nghiên cứu tiếp theo có thể tập trung vào việc xây dựng một hệ thống dễ sử dụng và tương thích với các phần mềm y tế hiện có, đồng thời đảm bảo bảo mật thông tin người bệnh và tuân thủ các quy định về quyền riêng tư.

Cuối cùng, chúng em cũng sẽ tiếp tục đánh giá và cải thiện các mô hình thông qua việc thu thập thêm dữ liệu từ các nhóm đối tượng khác nhau và áp dụng các kỹ thuật tối ưu hóa mô hình để đảm bảo hiệu quả dự đoán trong thực tế.