## BỘ GIÁO DỤC VÀ ĐÀO TẠO



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

BÀI TẬP LỚN

**HỌC MÁY CƠ BẢN**

**APPLYING MACHINE LEARING TO PREDICT STROKE**

NGÀNH: CÔNG NGHỆ PHẦN MỀM

SINH VIÊN: **1. NGUYỄN VĂN LINH**

**2. ĐÀO ANH QUÂN**

LỚP: **12522W.2**

NGƯỜI HƯỚNG DẪN: **TS. HOÀNG QUỐC VIỆT**

**HƯNG YÊN – 2025**

# NHẬN XÉT

## Nhận xét của giáo viên hướng dẫn

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

## GIÁO VIÊN HƯỚNG DẪN

**Hoàng Quốc Việt**

**MỤC LỤC**

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 6](#_bookmark0)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 9](#_bookmark1)

* 1. [Bài toán 9](#_bookmark2)
  2. [Trình bày dữ liệu bài toán 9](#_bookmark3)
  3. [Tiền xử lý dữ liệu 11](#_bookmark7)
  4. [Làm sạch dữ liệu 12](#_bookmark8)
  5. [Trực quan hoá dữ liệu 12](#_bookmark9)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13](#_bookmark10)

* 1. [RandomForestClassifier 13](#_bookmark11)
     1. [Giới thiệu thuật toán RandomForestClassifier 13](#_bookmark12)
     2. [Cấu trúc của mô hình RandomForest 13](#_bookmark13)
     3. [Quá trình xây dựng mô hình RandomForest 13](#_bookmark14)
     4. [Ưu điểm của RandomForest 13](#_bookmark15)
     5. [Nhược điểm của RandomForest 14](#_bookmark16)
     6. [Đánh giá hiệu suất 14](#_bookmark17)
  2. [KNeighborsClassifier 14](#_bookmark18)
     1. [Giới thiệu thuật toán KNeighborsClassifier 14](#_bookmark19)
     2. [Cấu trúc của mô hình KNN 14](#_bookmark20)
     3. [Quá trình xây dựng mô hình KNN 15](#_bookmark21)
     4. [Ưu điểm của KNN 15](#_bookmark22)
     5. [Nhược điểm của KNN 15](#_bookmark23)
     6. [Đánh giá hiệu suất 15](#_bookmark24)
  3. [Support Vector Classification (SVC) 15](#_bookmark25)
     1. [Giới thiệu thuật toán SVC 15](#_bookmark26)
     2. [Cấu trúc của mô hình SVC 16](#_bookmark27)
     3. [Quá trình xây dựng mô hình SVC 16](#_bookmark28)
     4. [Ưu điểm của SVC 16](#_bookmark29)
     5. [Nhược điểm của SVC 16](#_bookmark30)
     6. [Đánh giá hiệu suất 16](#_bookmark31)

[CHƯƠNG 3: GIẢI PHÁP, CÀI ĐẶT THỰC HIỆN 17](#_bookmark32)

* 1. [Mã nguồn tiền xử lý dữ liệu 17](#_bookmark33)
  2. [Mã nguồn chức năng làm sạch dữ liệu 18](#_bookmark38)
     1. [Xử lý giá trị thiếu 18](#_bookmark39)
     2. [Xoá outlier 20](#_bookmark46)
     3. [Ma trận tương quan 21](#_bookmark49)
  3. [Mã nguồn chức năng Trực quan hoá dữ liệu 23](#_bookmark52)
     1. [Biểu đồ thể hiện phân phối đặc trưng các phân loại 23](#_bookmark53)
     2. [Biểu đồ thể hiện đếm số lần đột quỵ 24](#_bookmark57)
     3. [Biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark60)
     4. [Biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại 26](#_bookmark63)
     5. [Biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age,](#_bookmark66) [avg\_glucose\_level) 27](#_bookmark66)
     6. [Biểu đồ kiểm tra giá trị ngoại lai 27](#_bookmark69)
     7. [Biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark72)
  4. [Chuyển đổi dữ liệu 29](#_bookmark75)
     1. [Xử lý dữ liệu chuỗi 29](#_bookmark77)
     2. [Xử lý dữ liệu số 30](#_bookmark81)
     3. [Xử lý dữ liệu mất cân bằng bằng SMOTE 31](#_bookmark85)
  5. [Huấn luyện mô hình SKLEARN 31](#_bookmark87)
     1. [Tìm kiếm mô hình 31](#_bookmark88)
     2. [Điều chỉnh tham số 32](#_bookmark91)
     3. [Mô hình RandomForestClassifier 35](#_bookmark95)
     4. [Mô hình KNeighborsClassifier 36](#_bookmark98)
     5. [Mô hình Support Vector Classification 37](#_bookmark101)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG 38](#_bookmark104)

[KẾT LUẬN 39](#_bookmark106)

[TÀI LIỆU THAM KHẢO 41](#_bookmark107)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1. 1 Hiển thị 5 hàng đầu và 5 hàng cuối của dữ liệu 10](#_bookmark4)

[Hình 1. 2 Thông tin tổng quan về dữ liệu 10](#_bookmark5)

[Hình 1. 3 Thông tin mô tả về DataFrame 11](#_bookmark6)

[Hình 3. 1 Import thư viện cần thiết, load, hiển thị 5 hàng đầu của DataFrame 17](#_bookmark34)

[Hình 3. 2 Kết quả hiển thị 5 hàng đầu của DataFrame 17](#_bookmark35)

[Hình 3. 3 Mã nguồn xoá cột “id” 17](#_bookmark36)

[Hình 3. 4 Thông tin về DataFrame 18](#_bookmark37)

[Hình 3. 5 Mã nguồn xử lý giá trị thiếu 18](#_bookmark40)

[Hình 3. 6 Hiển thị phần trăm giá trị thiếu của từng cột 19](#_bookmark41)

[Hình 3. 7 Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu](#_bookmark42) [(NaN) và hiển thị 5 hàng cuối của DataFrame 19](#_bookmark42)

[Hình 3. 8 Kết quả hiển thị 5 hàng cuối của DataFrame 19](#_bookmark43)

[Hình 3. 9 Mã nguồn đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame](#_bookmark44) [data. 20](#_bookmark44)

[Hình 3. 10 Kết quả liệt kê số giá trị thiếu theo cột. 20](#_bookmark45)

[Hình 3. 11 Mã nguồn xử lý dữ liệu ngoại lai 20](#_bookmark47)

[Hình 3. 12 Boxplot minh họa dữ liệu sạch. 21](#_bookmark48)

[Hình 3. 13 Mã nguồn ma trận tương quan 21](#_bookmark50)

[Hình 3. 14 Kết quả Heatmap minh họa mức độ tương quan giữa các yếu tố 22](#_bookmark51)

[Hình 3. 15 Mã nguồn biểu đồ thể hiện phân phối đặc trưng các phân loại 23](#_bookmark54)

[Hình 3. 16 Kết quả biểu đồ bar thể hiện phân phối đặc trưng các phân loại 23](#_bookmark55)

[Hình 3. 17 Kết quả biểu đồ pie thể hiện phân phối đặc trưng các phân loại 24](#_bookmark56)

[Hình 3. 18 Mã nguồn biểu đồ thể hiện đếm số lần đột quỵ 24](#_bookmark58)

[Hình 3. 19 Kết quả biểu đồ thể hiện đếm số lần đột quỵ 25](#_bookmark59)

[Hình 3. 20 Mã nguồn biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark61)

[Hình 3. 21 Kết quả biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark62)

[Hình 3. 22 Mã nguồn biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn](#_bookmark64) [lại 26](#_bookmark64)

[Hình 3. 23 Kết quả biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại](#_bookmark65)

[. 26](#_bookmark65)

[Hình 3. 24 Mã nguồn biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho](#_bookmark67) [(age, avg\_glucose\_level) 27](#_bookmark67)

[Hình 3. 25 Kết quả biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age,](#_bookmark68) [avg\_glucose\_level) 27](#_bookmark68)

[Hình 3. 26 Mã nguồn biểu đồ kiểm tra giá trị ngoại lai 27](#_bookmark70)

[Hình 3. 27 Kết quả biểu đồ kiểm tra giá trị ngoại lai 28](#_bookmark71)

[Hình 3. 28 Mã nguồn biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark73)

[Hình 3. 29 Kết quả biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark74)

[Hình 3. 30 Import 1 số thư viện cần thiết 29](#_bookmark76)

[Hình 3. 31 Mã nguồn xử lý dữ liệu chuỗi 29](#_bookmark78)

[Hình 3. 32 Kết quả xử lý dữ liệu chuỗi 29](#_bookmark79)

[Hình 3. 33 Mã nguồn xoá cột phân loại gốc trong X\_train và X\_test 30](#_bookmark80)

[Hình 3. 34 Mã nguồn xử lý dữ liệu số 30](#_bookmark82)

[Hình 3. 35 Kết quả xử lý dữ liệu số 30](#_bookmark83)

[Hình 3. 36 Hiển thị số hàng và số cột có X\_train và y\_train 30](#_bookmark84)

[Hình 3. 37 Xử lý dữ liệu mất cân bằng bằng SMOTE 31](#_bookmark86)

[Hình 3. 38 Mã nguồn tìm kiếm mô hình dùng Lazypredict 31](#_bookmark89)

[Hình 3. 39 Kết quả tìm kiếm mô hình dùng Lazypredict 32](#_bookmark90)

[Hình 3. 40 Import các thư viện cần thiết 32](#_bookmark92)

[Hình 3. 41 Mã nguồn hàm tìm kiếm, điều chỉnh tham số tốt nhất 33](#_bookmark93)

[Hình 3. 42 Mã nguồn trực quan hoá, hiển thị kết quả sau khi chạy mô hình 34](#_bookmark94)

[Hình 3. 43 Mã nguồn mô hình RandomForestClassifier 35](#_bookmark96)

[Hình 3. 44 Kết quả mô hình RandomForestClassifier 35](#_bookmark97)

[Hình 3. 45 Mã nguồn mô hình KneighborsClassifier 36](#_bookmark99)

[Hình 3. 46 Kết quả mô hình KNeighborsClassifier 36](#_bookmark100)

[Hình 3. 47 Mã nguồn mô hình Support Vector Classification 37](#_bookmark102)

[Hình 3. 48 Kết quả mô hình Support Vector Classification 37](#_bookmark103)

[HÌnh 4. 1 Sử dụng Streamlit để đưa bài toán lên giao diện web 38](#_bookmark105)

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## Bài toán

Tổng sản phẩm quốc nội(GDP) là một trong những chỉ số quan trọng phản ánh mức độ phát triển của một quốc gia. Dự đoán chính xác GDP thông qua việc phân tích dữ liệu và áp dụng các thuật toán học máy có thể hỗ trợ các chính phủ, tổ chức và doanh nghiệp xây dựng chính sách kinh tế, kế hoạch đầu tư và các chiến lược phát triển bển vững.

Bài toán "Áp dụng các thuật toán học máy vào dự đoán GDP dựa trên dữ liệu cơ cấu lao động và tỷ lệ thất nghiệp" tập trung vào việc khai thác dữ liệu kinh tế - xã hội và áp dụng các thuật toán học máy để xây dựng mô hình dự báo GDP. Từ đó, chúng ta sẽ xây dựng các mô hình dự báo sử dụng các kỹ thuật học máy như hồi quy logistic, rừng ngẫu nhiên, và máy vector hỗ trợ (SVM) và các bộ dữ liệu được sử dụng bao gồm các thông tin cơ bản như: tên quốc gia, năm, tỷ lệ lao động trong nông nghiệp, tỷ lệ lao động trong công nghiệp, tỷ lệ lao động trong dịch vụ, tỷ lệ thất nghiệp .

Dựa vào tích dữ liệu GDP từng năm của các quốc gia, các mô hình học máy có thể phát hiện các mối quan hệ tiềm ẩn giữa cơ cấu lao động và tốc độ tang trưởng GDP. Từ đó, nghiên cứu hướng tới việc:

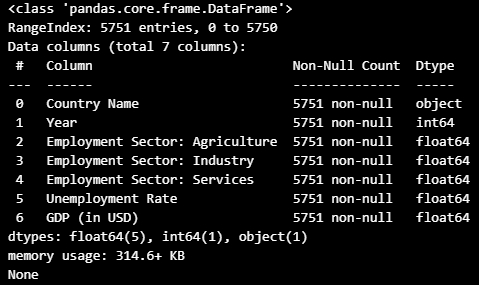
* + - Xác định mức độ ảnh hưởng của từng khu vực lao động đến tăng trưởng GDP.
    - Đánh giá vai trò của tỷ lệ thất nghiệp trong việc phản ánh tiềm năng phát triển của quốc gia.
    - Xây dựng so sách các mô hình học máy có độ chính xác cao trong dự đoán GDP hỗ trợ các chính sách và chiến lược phát triển.

Đề tài không chỉ thể hiện giá trị thực tiễn của học máy trong lĩnh vực kinh tế mà còn mang ý nghĩa khoa học sâu sắc trong việc sử dụng trí tuệ nhân tạo vào phân tích, dự báo và hỗ trợ ra quyết định. Kết quả nghiên cứu có thể góp phần nâng cao hiệu quả điều hành kinh tế, giúp các quốc gia ra chính sách, giải pháp hợp lý nhằm ổn định và phát triển kinh tế một cách bền vững.

* 1. **Trình bày dữ liệu bài toán** Dữ liệu được lấy từ trang web này: [Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)



*Hình 1. 1 Hiển thị 5 hàng đầu và 5 hàng cuối của dữ liệu*

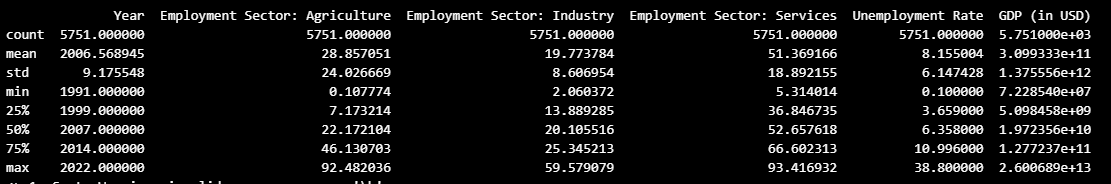
**

*Hình 1. 2 Thông tin tổng quan về cấu trúc Dataframe*

* Bộ dữ liệu được sử dụng trong bài toán này bao gồm các đặc trưng liên quan đến sức khỏe và nhân khẩu học của bệnh nhân, cụ thể như sau:
  1. Country Name: tên quốc gia .
  2. Year: năm quan sát.
  3. Employment Sector: Agriculture: Tỷ lệ lao động ngành Nông nghiệp.
  4. Employment Sector: Industry: Tỷ lệ lao động ngành Công nghiêp.
  5. Employment Sector: Services: Tỷ lệ lao động ngành dịch vụ
  6. Unemployment Rate: Tỷ lệ thất nghiệp
  7. GDP (in USD): Tổng sản phẩm quốc nội tính theo USD
* Dữ liệu bài toán là 1 file csv gồm 5751 rows × 7 columns

+ Có 7 feature và mỗi feature có 5751 dữ liệu đầu vào

-Sau khi mô tả dữ liệu ta có:



*Hình 1. 3 Thông tin mô tả về dữ liệu số*

+ Năm thống kê lớn nhất là **2022** và nhỏ nhất là **1991**.

+ Tỷ lệ lao động trong lĩnh vực **nông nghiệp (Agriculture)** lớn nhất là **92.48%** và nhỏ nhất là **0.11%**.

+ Tỷ lệ lao động trong lĩnh vực **công nghiệp (Industry)** lớn nhất là **59.58%** và nhỏ nhất là **2.06%**.

+ Tỷ lệ lao động trong lĩnh vực **dịch vụ (Services)** lớn nhất là **93.42%** và nhỏ nhất là **5.31%**.

+ **Tỷ lệ thất nghiệp (Unemployment Rate)** lớn nhất là **38.80%** và nhỏ nhất là **0.10%.**

+ **GDP (tính theo USD)** lớn nhất là **2.60×10¹³ USD** và nhỏ nhất là **7.23×10⁷ USD.**

## Tiền xử lý dữ liệu

1. Hiển thị 5 hàng đầu của dữ liệu
2. Hiển thị số hàng, cột của dữ liệu
3. Hiển thị thông tin của dữ liệu
4. Mô tả dữ liệu
5. Hiển thị giá trị duy nhất trong mỗi cột

## Làm sạch dữ liệu

1. Xử lý giá trị thiếu
2. Xoá outliner
3. Ma trận tương quan

## Trực quan hoá dữ liệu

1. Biểu đồ thể hiện phân phối đặc trưng các phân loại
2. Biểu đồ phân bố GDP
3. đồ ma trận tương quan
4. Biểu đồ phân tán GDP theo tỷ lệ lao động nông nghiệp
5. Biểu đồ phân tán GDP theo tỷ lệ lao động công nghiệp
6. **Biểu đồ phân tán GDP theo tỷ lệ lao động dịch vụ**
7. Biểu đồ hộp về tỷ lệ thất nghiệp

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## RandomForestClassifier

## Giới thiệu thuật toán RandomForestClassifier

Thuật toán RandomForestClassifier là một phương pháp học máy phổ biến được sử dụng trong cả bài toán phân loại và hồi quy. Random Forest là một tập hợp các cây quyết định, trong đó mỗi cây được xây dựng dựa trên một tập dữ liệu mẫu ngẫu nhiên (bootstrapping) và một tập con các đặc trưng (feature subset) được chọn ngẫu nhiên tại mỗi nút phân chia. Kết quả cuối cùng được lấy bằng cách thực hiện trung bình (hồi quy) hoặc lấy đa số phiếu (phân loại) từ các cây thành phần.

## Cấu trúc của mô hình RandomForest

Mô hình Random Forest bao gồm:

* + - * **Cây quyết định (Decision Trees):** Các cây quyết định hoạt động như các mô hình thành phần.
      * **Tập dữ liệu mẫu (Bootstrap Samples):** Mỗi cây được huấn luyện trên một tập con dữ liệu được lấy mẫu ngẫu nhiên có thay thế từ tập dữ liệu gốc.
      * **Tập con đặc trưng (Feature Subset):** Tại mỗi nút của cây, một tập hợp ngẫu nhiên các đặc trưng được chọn để tìm phân chia tốt nhất.

## Quá trình xây dựng mô hình RandomForest

* + - * **Khởi tạo các cây quyết định:** Xác định số lượng cây trong rừng (hyperparameter: n\_estimators).
      * **Lấy mẫu dữ liệu:** Tạo tập dữ liệu mẫu ngẫu nhiên có thay thế cho từng cây.
      * **Chọn tập con đặc trưng:** Tại mỗi nút, chọn ngẫu nhiên một tập con các đặc trưng.
      * **Xây dựng cây:** Phân chia dữ liệu tại các nút theo tiêu chí như Gini Index hoặc Entropy.
      * **Dự đoán:** Đối với phân loại, mỗi cây bỏ phiếu cho một nhãn, và nhãn cuối cùng được chọn dựa trên đa số phiếu.

## Ưu điểm của RandomForest

* + - * **Khả năng tổng quát hóa tốt:** Random Forest giảm thiểu vấn đề overfitting nhờ tích hợp nhiều cây.
      * **Hiệu suất cao:** Hoạt động hiệu quả trên nhiều loại dữ liệu khác nhau, bao gồm cả dữ liệu không tuyến tính.
      * **Đánh giá quan trọng đặc trưng:** Cung cấp thông tin về mức độ quan trọng của từng đặc trưng trong việc đưa ra dự đoán.
      * **Độ chính xác cao:** Thường đạt độ chính xác cao hơn các mô hình đơn lẻ như Decision Tree.
      * **Xử lý dữ liệu mất mát:** Random Forest có thể xử lý dữ liệu bị mất mát bằng cách ước tính giá trị bị thiếu.

## Nhược điểm của RandomForest

* + - * **Khó giải thích:** Mặc dù các cây quyết định riêng lẻ dễ hiểu, tập hợp Random Forest lại phức tạp hơn để diễn giải.
      * **Chi phí tính toán cao:** Xây dựng nhiều cây quyết định đòi hỏi tài nguyên tính toán lớn.
      * **Dễ bị bias:** Trong một số trường hợp, nếu dữ liệu không được chuẩn hóa tốt, mô hình có thể bị bias.

## Đánh giá hiệu suất

Hiệu suất của RandomForestClassifier thường được đánh giá thông qua các chỉ số như Accuracy, Precision, Recall, và F1 Score. Đối với bài toán dự đoán bệnh đột quỵ, mô hình được kiểm tra dựa trên khả năng dự đoán đúng các trường hợp bệnh lý.

## KNeighborsClassifier

## Giới thiệu thuật toán KNeighborsClassifier

Thuật toán K-Nearest Neighbors (KNN) là một phương pháp học máy không tham số, được sử dụng phổ biến cho bài toán phân loại và hồi quy. KNN hoạt động dựa trên khoảng cách giữa một điểm dữ liệu mới và các điểm dữ liệu đã biết trong tập huấn luyện. Dữ liệu mới được phân loại dựa trên nhãn của k điểm lân cận gần nhất.

## Cấu trúc của mô hình KNN

* + - * **Tập huấn luyện:** Bao gồm các điểm dữ liệu đã được gán nhãn.
      * **Khoảng cách:** Các khoảng cách như Euclidean, Manhattan, hoặc Minkowski được sử dụng để xác định mức độ gần gũi giữa các điểm.
      * **Số lượng lân cận (k):** Đây là tham số chính quyết định số lượng điểm lân cận được sử dụng để phân loại dữ liệu mới.

## Quá trình xây dựng mô hình KNN

* + - * **Xác định giá trị k:** Thông thường k được chọn là số lẻ để tránh trường hợp đồng phiếu.
      * **Tính khoảng cách:** Đo khoảng cách từ điểm cần dự đoán đến tất cả các điểm trong tập huấn luyện.
      * **Chọn k lân cận gần nhất:** Lấy k điểm có khoảng cách nhỏ nhất.
      * **Bỏ phiếu:** Xác định nhãn của điểm mới dựa trên nhãn của k điểm lân cận.

## Ưu điểm của KNN

* + - * **Đơn giản:** Thuật toán dễ hiểu và triển khai.
      * **Linh hoạt:** Có thể sử dụng cho cả bài toán phân loại và hồi quy.
      * **Không cần huấn luyện:** KNN không yêu cầu giai đoạn huấn luyện, chỉ cần lưu trữ dữ liệu.
      * **Hiệu quả với dữ liệu nhỏ:** Hoạt động tốt với các tập dữ liệu có kích thước nhỏ.

## Nhược điểm của KNN

* + - * **Chi phí tính toán cao:** Khi kích thước dữ liệu lớn, việc tính khoảng cách đến tất cả các điểm trở nên rất tốn kém.
      * **Nhạy cảm với dữ liệu nhiễu:** Các điểm nhiễu có thể ảnh hưởng lớn đến kết quả.
      * **Phụ thuộc vào k:** Việc chọn giá trị k phù hợp là một thách thức.
      * **Cần chuẩn hóa dữ liệu:** Khoảng cách giữa các điểm có thể bị ảnh hưởng bởi thang đo của các đặc trưng.

## Đánh giá hiệu suất

Hiệu suất của KNN được đánh giá thông qua các độ đo như Accuracy, Precision, và Recall. Đối với bài toán dự đoán bệnh đột quỵ, KNN cần xử lý tốt sự phân bố không đồng đều giữa các lớp dữ liệu.

## Support Vector Classification (SVC)

## Giới thiệu thuật toán SVC

Thuật toán Support Vector Classification (SVC) là một phương pháp học máy mạnh mẽ, đặc biệt hiệu quả với các bài toán phân loại. SVC hoạt động bằng cách tìm một siêu phẳng tối ưu để phân chia các lớp dữ liệu, đảm bảo khoảng cách giữa siêu phẳng và các điểm gần nhất của mỗi lớp (margin) là lớn nhất.

## Cấu trúc của mô hình SVC

* + - * **Siêu phẳng (Hyperplane):** Là đường hoặc mặt phẳng phân chia không gian đặc trưng.
      * **Margin:** Khoảng cách giữa siêu phẳng và điểm dữ liệu gần nhất từ mỗi lớp.
      * **Hạt nhân (Kernel):** Chức năng ánh xạ dữ liệu vào không gian đặc trưng cao hơn để xử lý dữ liệu không tuyến tính.

## Quá trình xây dựng mô hình SVC

* + - * **Lựa chọn hạt nhân:** Xác định hàm kernel như Linear, Polynomial, RBF (Gaussian), hoặc Sigmoid.
      * **Tìm siêu phẳng tối ưu:** Sử dụng thuật toán tối ưu hóa (như SMO) để xác định siêu phẳng phân chia dữ liệu.
      * **Đánh giá:** Kiểm tra hiệu suất dựa trên dữ liệu kiểm thử.

## Ưu điểm của SVC

* + - * **Hiệu quả cao:** Là một trong những thuật toán tốt nhất cho các bài toán phân loại nhị phân.
      * **Xử lý tốt dữ liệu không tuyến tính:** Nhờ vào các hàm kernel.
      * **Tổng quát hóa tốt:** Hạn chế overfitting trong nhiều trường hợp.
      * **Hiệu quả với dữ liệu nhỏ:** Hoạt động tốt ngay cả khi kích thước dữ liệu nhỏ.

## Nhược điểm của SVC

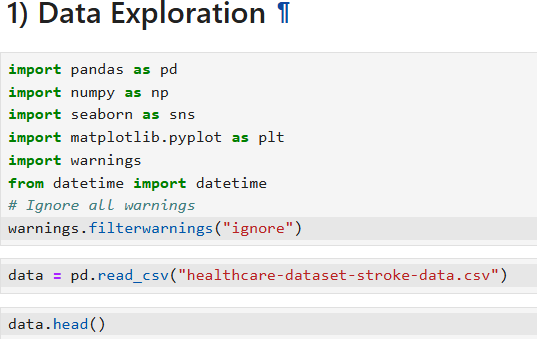
* + - * **Khó mở rộng:** Hiệu suất giảm khi kích thước dữ liệu tăng.
      * **Chi phí tính toán cao:** Việc tìm siêu phẳng tối ưu yêu cầu tài nguyên lớn.
      * **Nhạy cảm với tham số:** Cần lựa chọn cẩn thận tham số như C, gamma, và kernel.
      * **Không phù hợp với dữ liệu mất cân bằng:** SVC có thể bị bias khi dữ liệu không cân bằng.

## Đánh giá hiệu suất

Hiệu suất của SVC thường được đánh giá bằng các chỉ số như Accuracy, Precision, Recall, và F1 Score. Với bài toán dự đoán bệnh đột quỵ, SVC có thể tối ưu hóa phân loại nhờ vào khả năng xử lý dữ liệu không tuyến tính.

# CHƯƠNG 3: GIẢI PHÁP, CÀI ĐẶT THỰC HIỆN

## Mã nguồn tiền xử lý dữ liệu

****

*Hình 3. 1 Import thư viện cần thiết, load, hiển thị 5 hàng đầu của DataFrame*

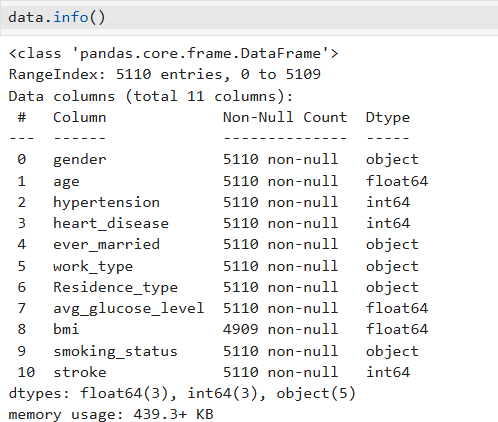
**

*Hình 3. 2 Kết quả hiển thị 5 hàng đầu của DataFrame*

**

*Hình 3. 3 Mã nguồn xoá cột “id”*

Đầu tiên em sử dụng hàm df.info() để kiểm tra các feature nào bị thiếu và có kiểu dữ liệu thế nào.Em hiển thị lại hàm df và dùng lại hàm info để kiểm tra xem dữ liệu. Và hình ảnh dưới là kết quả sau khi đã điền các dữ liệu:



*Hình 3. 4 Thông tin về DataFrame*

## Mã nguồn chức năng làm sạch dữ liệu

## Xử lý giá trị thiếu

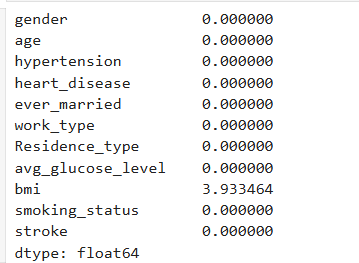
****

*Hình 3. 5 Mã nguồn xử lý giá trị thiếu*

Câu lệnh này tính toán tỷ lệ phần trăm giá trị thiếu trong từng cột của DataFrame data.

* + - * **data.isna()**: Xác định các giá trị thiếu (NaN) trong data và trả về DataFrame với giá trị True/False.
      * **.sum()**: Tính tổng số True (giá trị thiếu) cho mỗi cột.
      * **len(data)**: Số lượng hàng trong data.
      * **\*100**: Chuyển đổi tỷ lệ sang phần trăm.

Kết quả: Hiển thị phần trăm giá trị thiếu của từng cột, hỗ trợ xác định cần xử lý những cột nào.



*Hình 3. 6 Hiển thị phần trăm giá trị thiếu của từng cột*

**

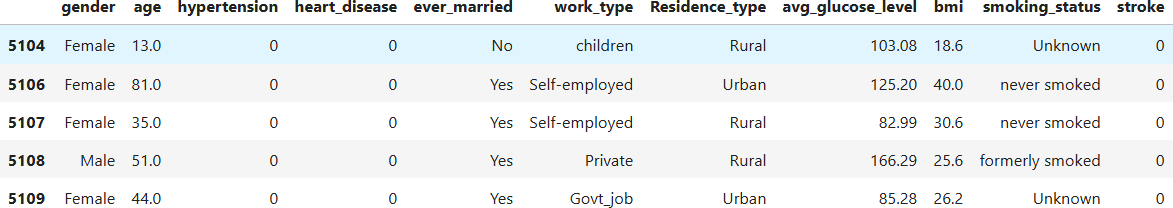
*Hình 3. 7 Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu (NaN) và hiển thị 5 hàng cuối của DataFrame*

Câu lệnh này được sử dụng để xử lý giá trị thiếu và kiểm tra dữ liệu sau khi loại bỏ các hàng chứa giá trị thiếu:

**data.dropna(inplace=True)**: Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu (NaN). Tham số inplace=True đảm bảo thay đổi được thực hiện trực tiếp trên DataFrame data mà không tạo một bản sao.

**data.tail()**: Hiển thị 5 hàng cuối cùng của DataFrame data sau khi loại bỏ các giá trị thiếu.

Mục đích: Loại bỏ dữ liệu không đầy đủ và xem nhanh các dòng cuối để đảm bảo dữ liệu đã được xử lý đúng cách.

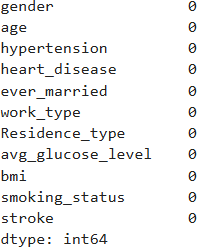


*Hình 3. 8 Kết quả hiển thị 5 hàng cuối của DataFrame*



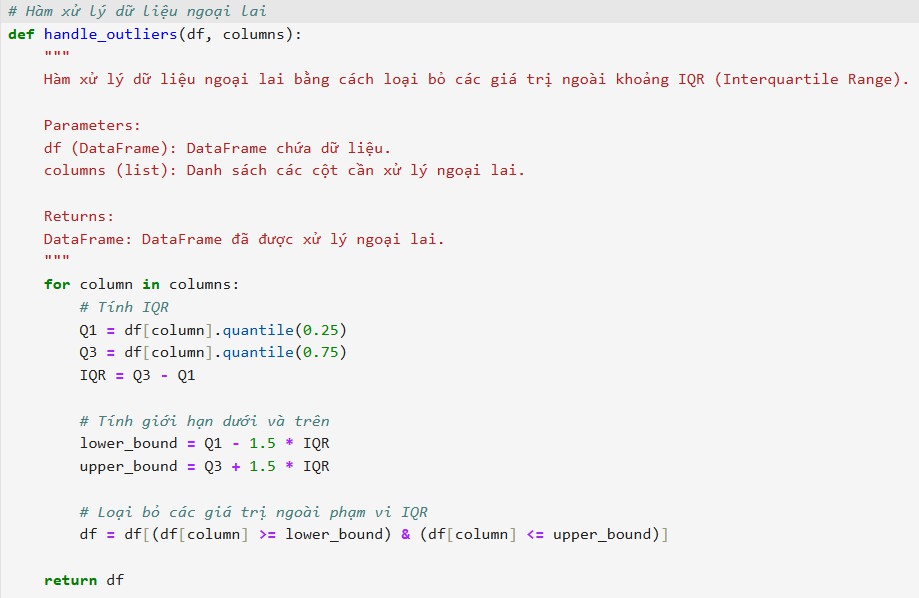
*Hình 3. 9 Mã nguồn đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame data.*

* + - * Đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame data.
      * Kết quả: Một bảng liệt kê số giá trị thiếu theo cột.



*Hình 3. 10 Kết quả liệt kê số giá trị thiếu theo cột.*

## Xoá outlier

****

*Hình 3. 11 Mã nguồn xử lý dữ liệu ngoại lai*

Hàm **handle\_outliers** loại bỏ giá trị ngoại lai trong các cột bằng cách sử dụng **IQR (Interquartile Range)**:

## Tính IQR:

* + Q1 (25%) và Q3 (75%) là các phân vị.
  + IQR = Q3 - Q1.

## Giới hạn:

* + **Lower bound** = Q1 - 1.5 \* IQR.
  + **Upper bound** = Q3 + 1.5 \* IQR.

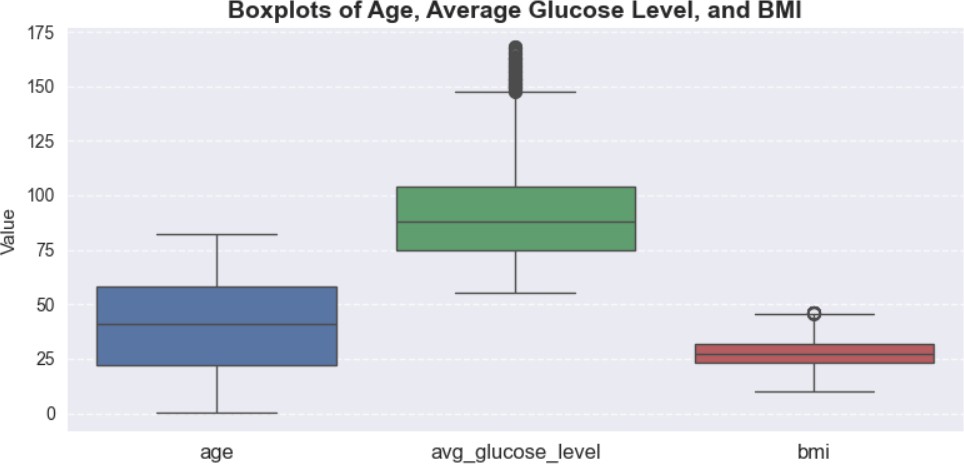
## Lọc dữ liệu:

* + Loại bỏ giá trị ngoài khoảng [lower\_bound, upper\_bound].

## boxplots(data\_no\_outliers):

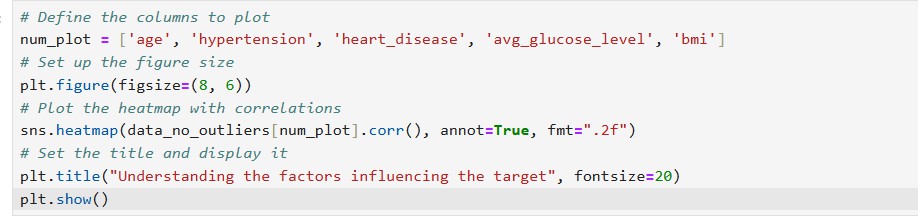
* + Hiển thị boxplot của dữ liệu sau khi xử lý ngoại lai.

Kết quả: DataFrame data\_no\_outliers không còn ngoại lai, và boxplot minh họa dữ liệu sạch.



*Hình 3. 12 Boxplot minh họa dữ liệu sạch.*

## Ma trận tương quan

****

*Hình 3. 13 Mã nguồn ma trận tương quan*

**Biểu đồ Heatmap** để thể hiện mối quan hệ tương quan giữa các cột số học trong dữ liệu đã xử lý ngoại lai:

## num\_plot:

* + Danh sách các cột cần tính tương quan: ['age', 'hypertension', 'heart\_disease', 'avg\_glucose\_level', 'bmi'].

## plt.figure(figsize=(8, 6)):

* + Thiết lập kích thước biểu đồ là 8x6 inch.

## sns.heatmap():

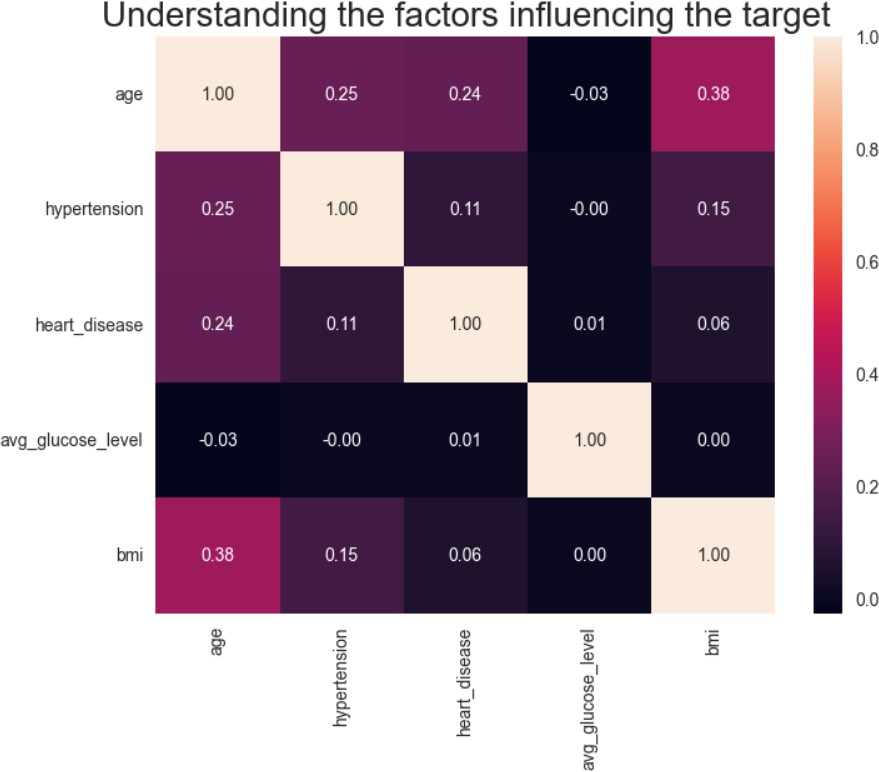
* + Tính toán ma trận tương quan giữa các cột trong data\_no\_outliers.
  + annot=True: Hiển thị giá trị tương quan trên heatmap.
  + fmt=".2f": Định dạng giá trị tương quan đến 2 chữ số thập phân.

## plt.title():

* + Đặt tiêu đề: "Understanding the factors influencing the target".

1. **plt.show()**: Hiển thị biểu đồ.

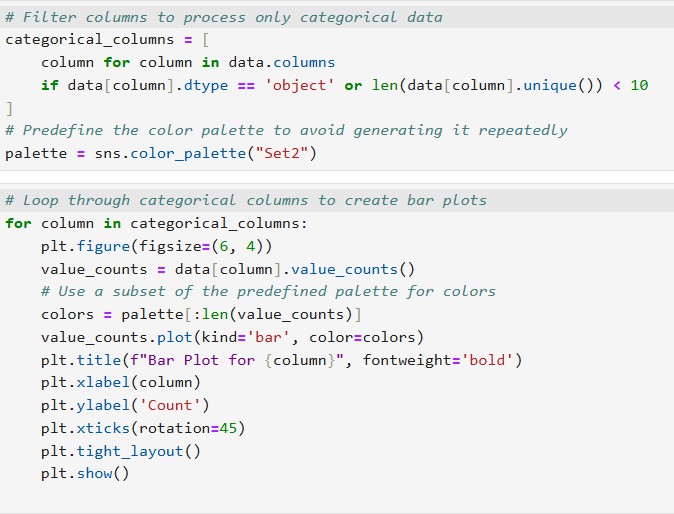
**Kết quả**: Heatmap minh họa mức độ tương quan giữa các yếu tố, giúp xác định yếu tố nào có tác động lớn hơn đến mục tiêu.



*Hình 3. 14 Kết quả Heatmap minh họa mức độ tương quan giữa các yếu tố*

## Mã nguồn chức năng Trực quan hoá dữ liệu

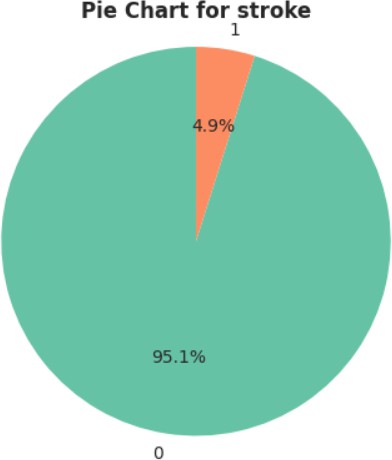
## Biểu đồ thể hiện phân phối đặc trưng các phân loại

****

*Hình 3. 15 Mã nguồn biểu đồ thể hiện phân phối đặc trưng các phân loại*

**

*Hình 3. 16 Kết quả biểu đồ bar thể hiện phân phối đặc trưng các phân loại*

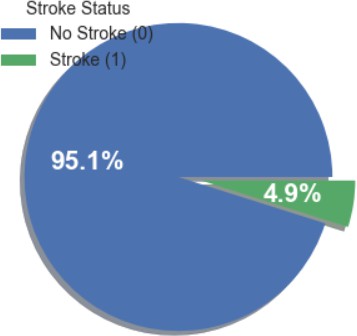


*Hình 3. 17 Kết quả biểu đồ pie thể hiện phân phối đặc trưng các phân loại*

## Biểu đồ thể hiện đếm số lần đột quỵ

****

*Hình 3. 18 Mã nguồn biểu đồ thể hiện đếm số lần đột quỵ*

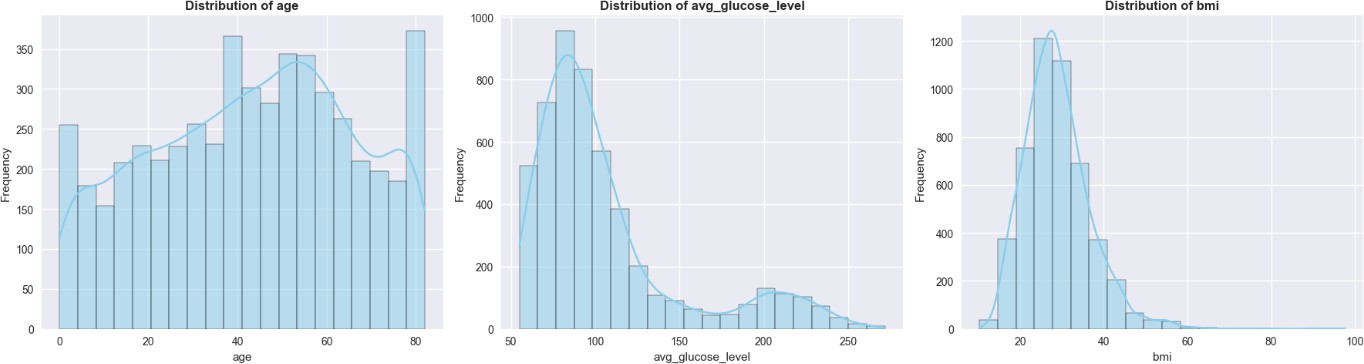


*Hình 3. 19 Kết quả biểu đồ thể hiện đếm số lần đột quỵ*

## Biểu đồ thể hiện các đặc trưng liên tục

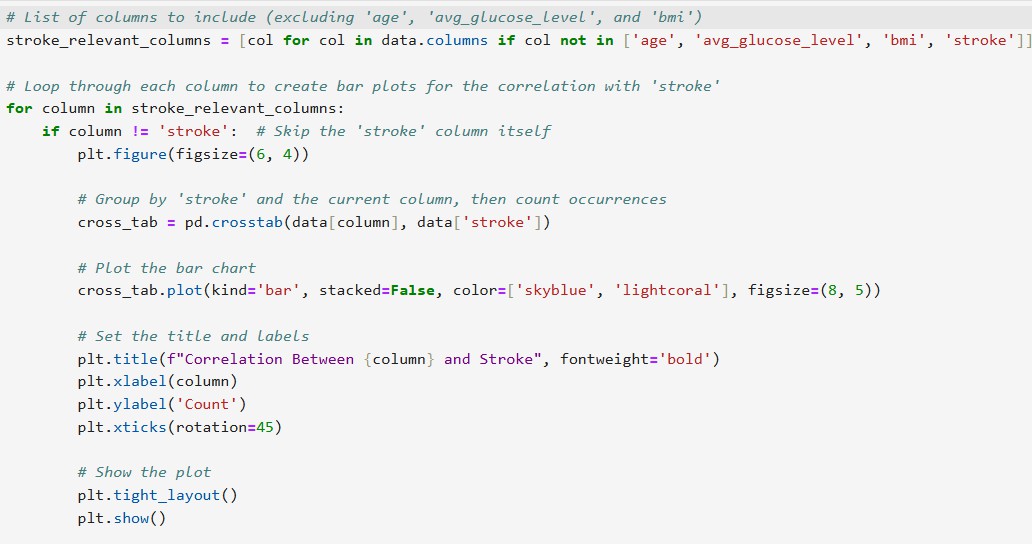
****

*Hình 3. 20 Mã nguồn biểu đồ thể hiện các đặc trưng liên tục*

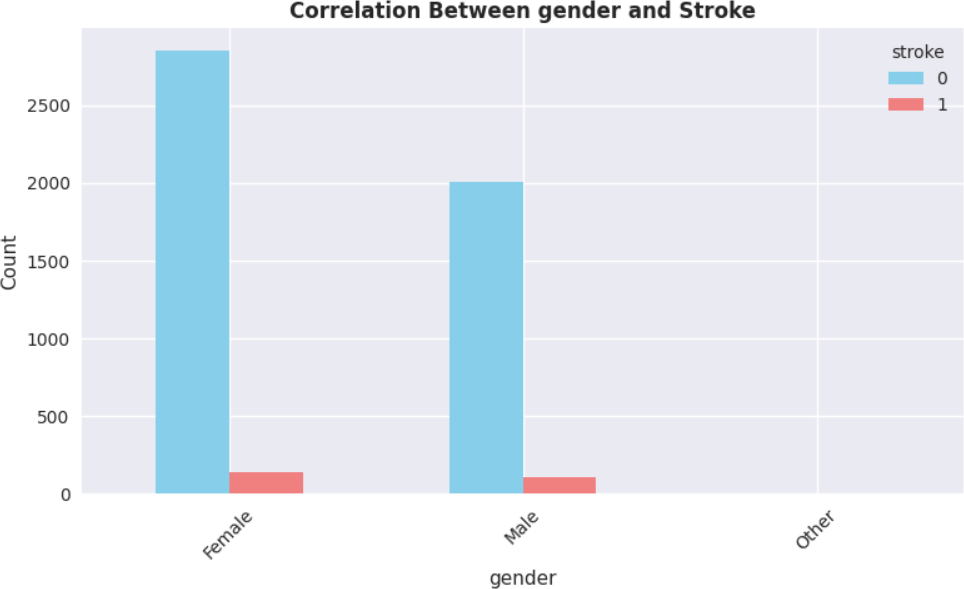
**

*Hình 3. 21 Kết quả biểu đồ thể hiện các đặc trưng liên tục*

## Biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại

****

*Hình 3. 22 Mã nguồn biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại*

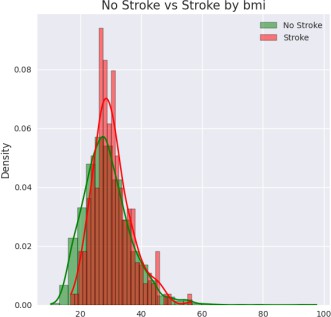
**

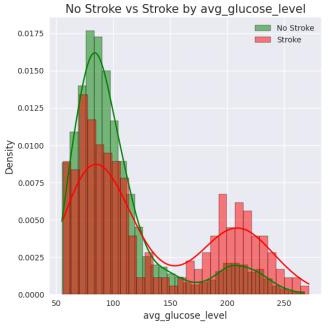
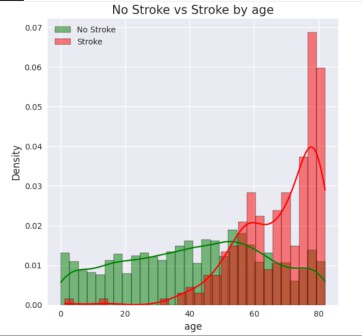
*Hình 3. 23 Kết quả biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại*

## Biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age, avg\_glucose\_level)

****

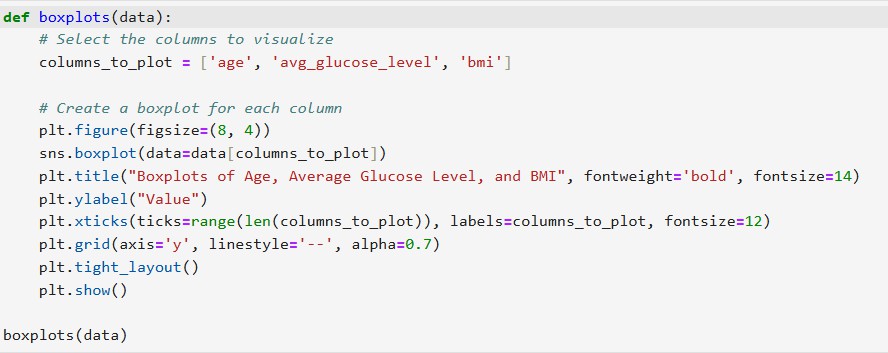
*Hình 3. 24 Mã nguồn biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age, avg\_glucose\_level)*

**

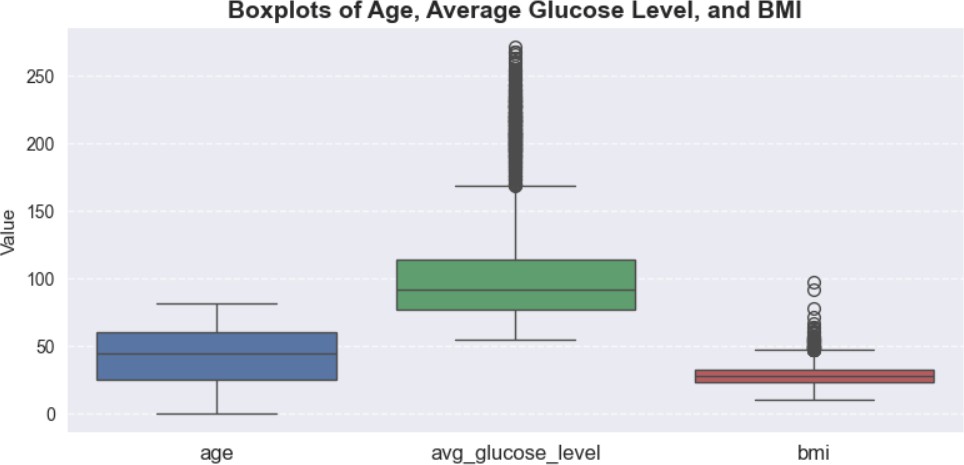


*Hình 3. 25 Kết quả biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age, avg\_glucose\_level)*

## Biểu đồ kiểm tra giá trị ngoại lai

****

*Hình 3. 26 Mã nguồn biểu đồ kiểm tra giá trị ngoại lai*

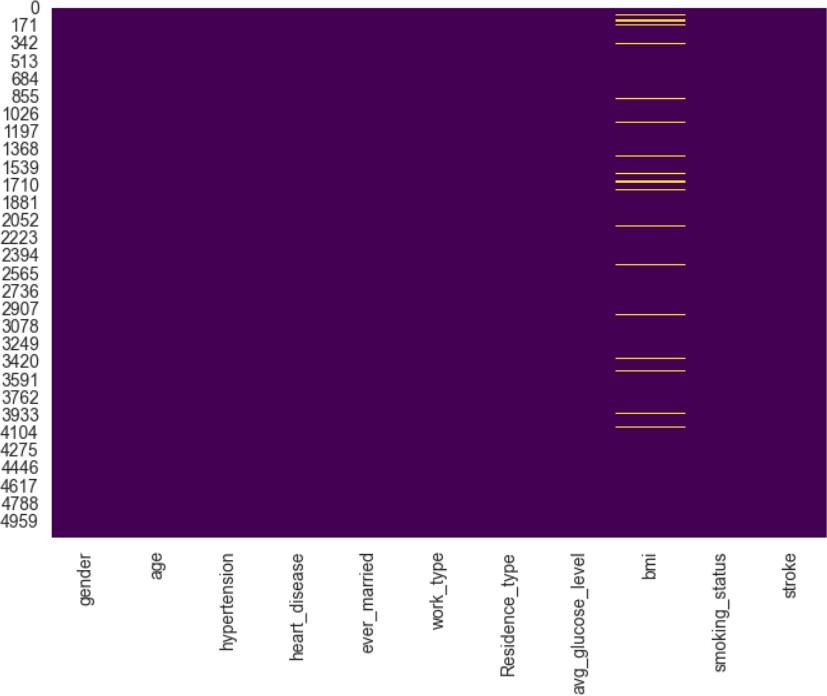


*Hình 3. 27 Kết quả biểu đồ kiểm tra giá trị ngoại lai*

## Biểu đồ thể hiện sự trực quan hoá giá trị thiếu

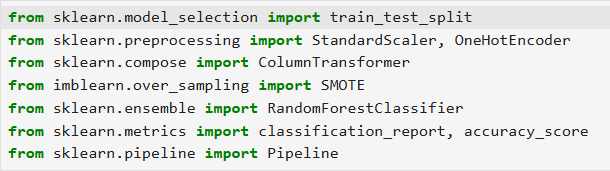
****

*Hình 3. 28 Mã nguồn biểu đồ thể hiện sự trực quan hoá giá trị thiếu*

**

*Hình 3. 29 Kết quả biểu đồ thể hiện sự trực quan hoá giá trị thiếu*

## Chuyển đổi dữ liệu

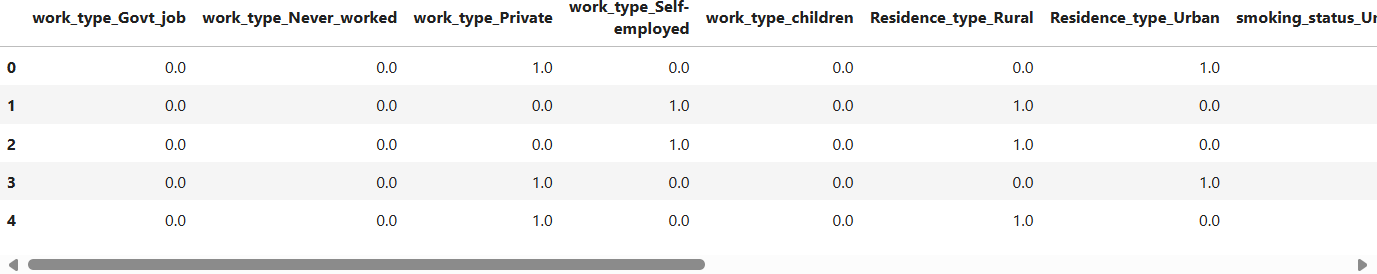
****

*Hình 3. 30 Import 1 số thư viện cần thiết*

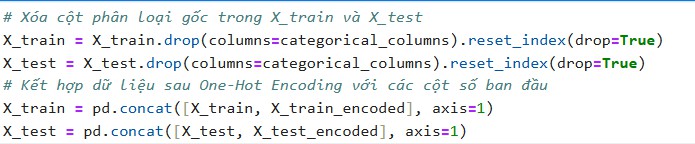
## Xử lý dữ liệu chuỗi

****

*Hình 3. 31 Mã nguồn xử lý dữ liệu chuỗi*

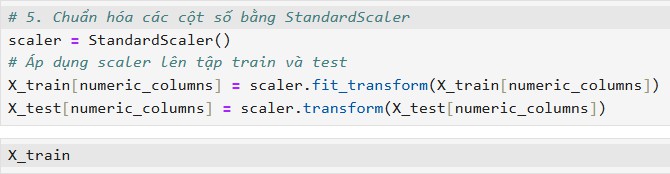
**

*Hình 3. 32 Kết quả xử lý dữ liệu chuỗi*

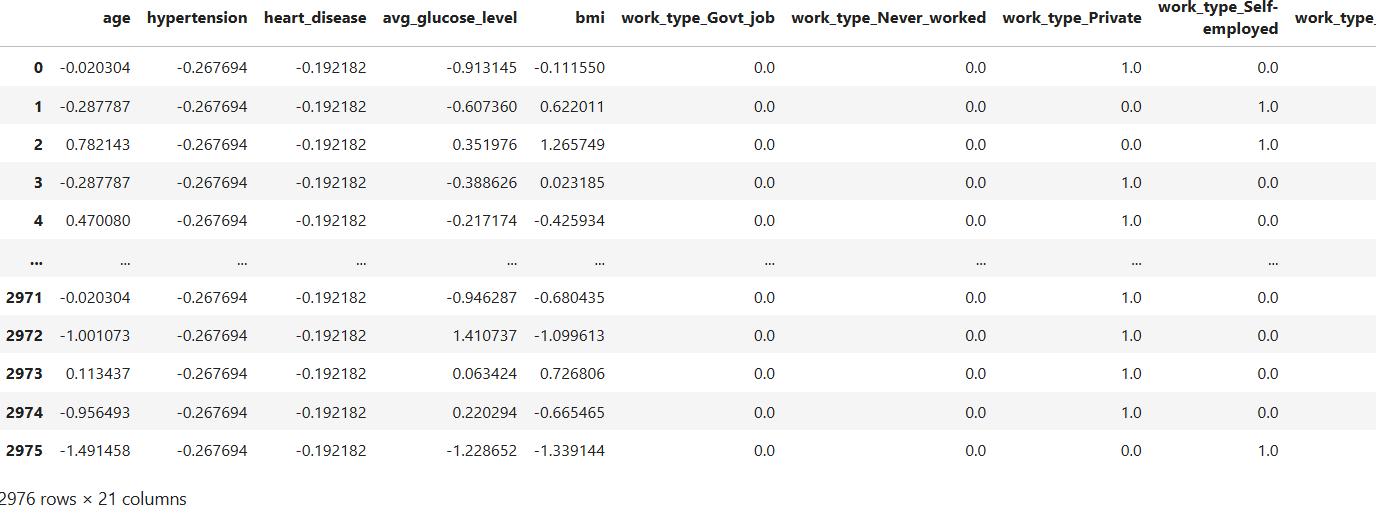


*Hình 3. 33 Mã nguồn xoá cột phân loại gốc trong X\_train và X\_test*

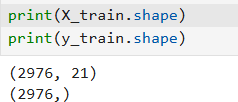
## Xử lý dữ liệu số

****

*Hình 3. 34 Mã nguồn xử lý dữ liệu số*

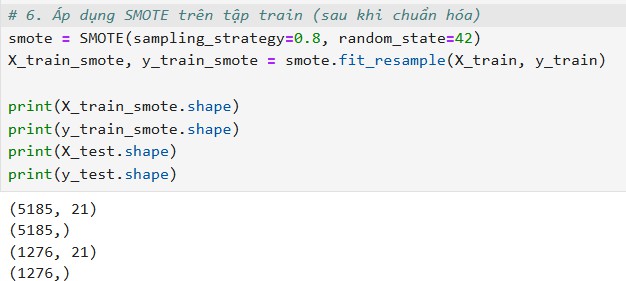
**

*Hình 3. 35 Kết quả xử lý dữ liệu số*

**

*Hình 3. 36 Hiển thị số hàng và số cột có X\_train và y\_train*

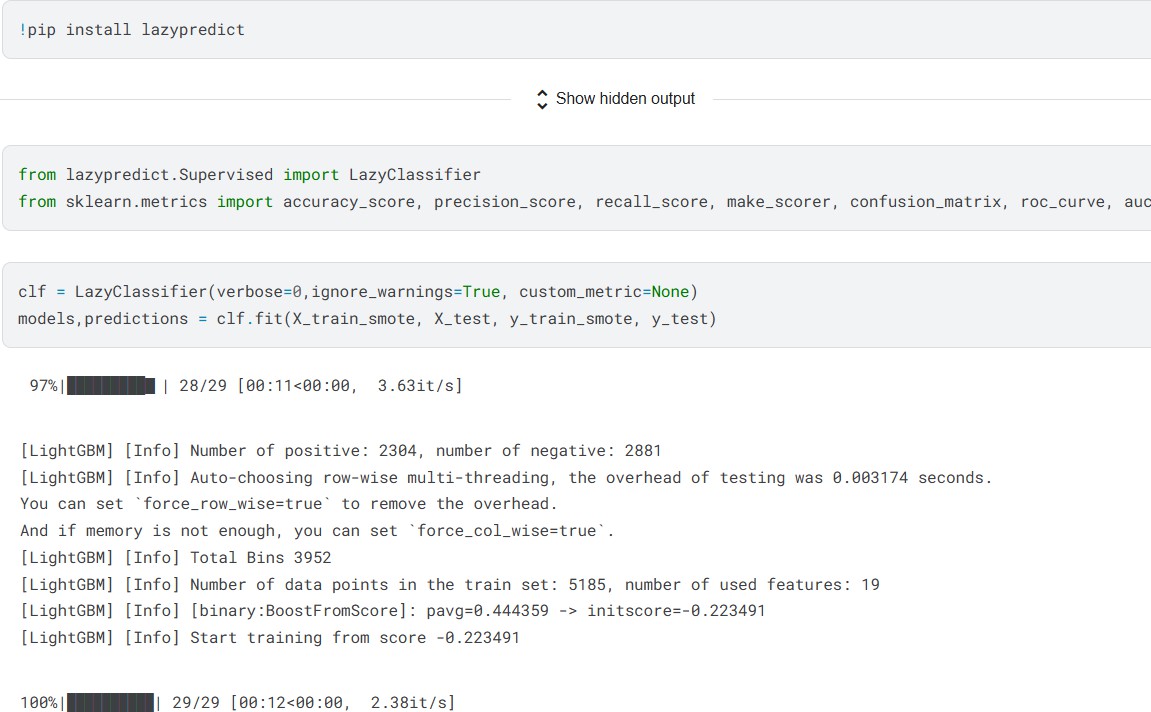
## Xử lý dữ liệu mất cân bằng bằng SMOTE

****

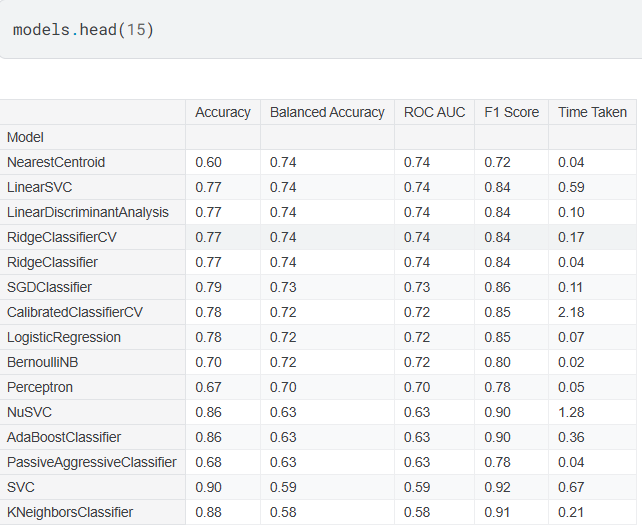
*Hình 3. 37 Xử lý dữ liệu mất cân bằng bằng SMOTE*

## Huấn luyện mô hình SKLEARN

## Tìm kiếm mô hình

****

*Hình 3. 38 Mã nguồn tìm kiếm mô hình dùng Lazypredict*

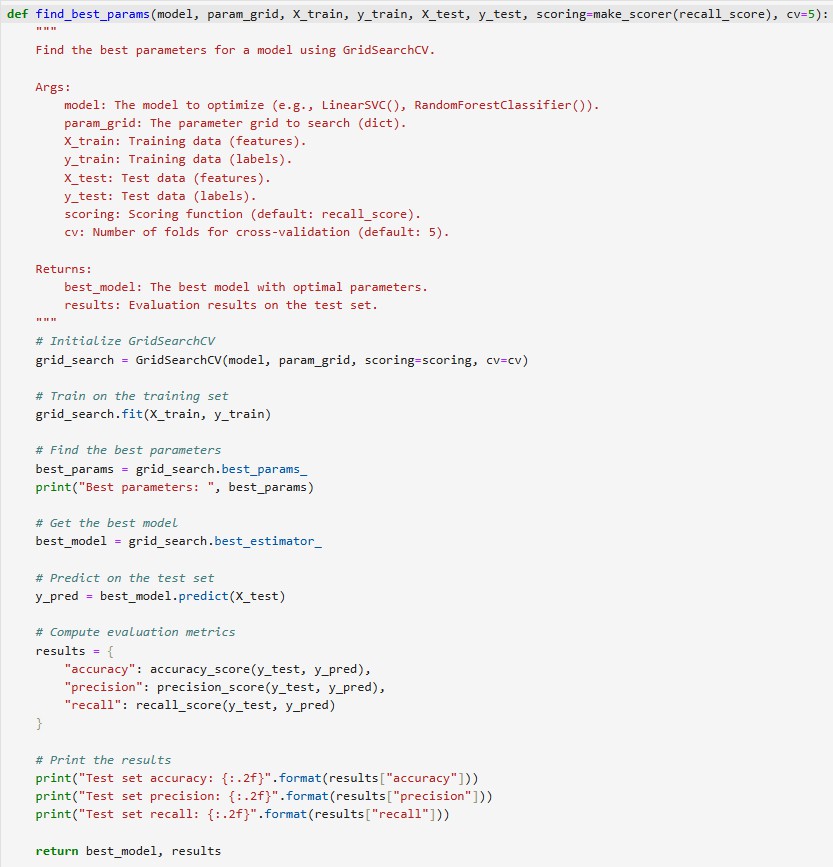


*Hình 3. 39 Kết quả tìm kiếm mô hình dùng Lazypredict*

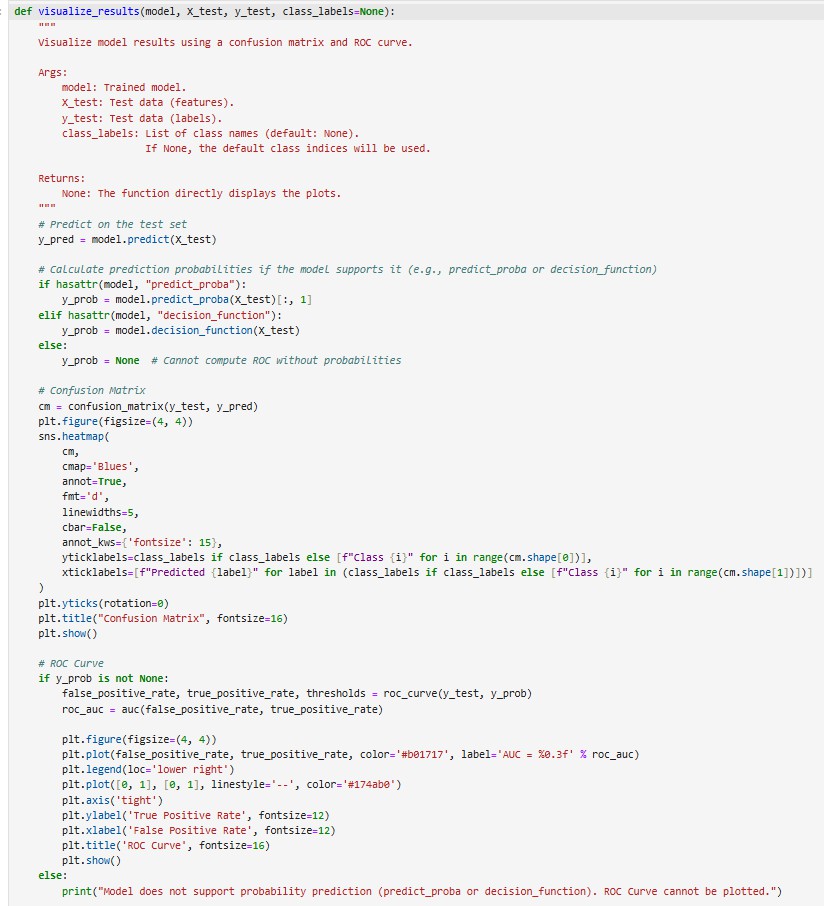
## Điều chỉnh tham số

****

*Hình 3. 40 Import các thư viện cần thiết*

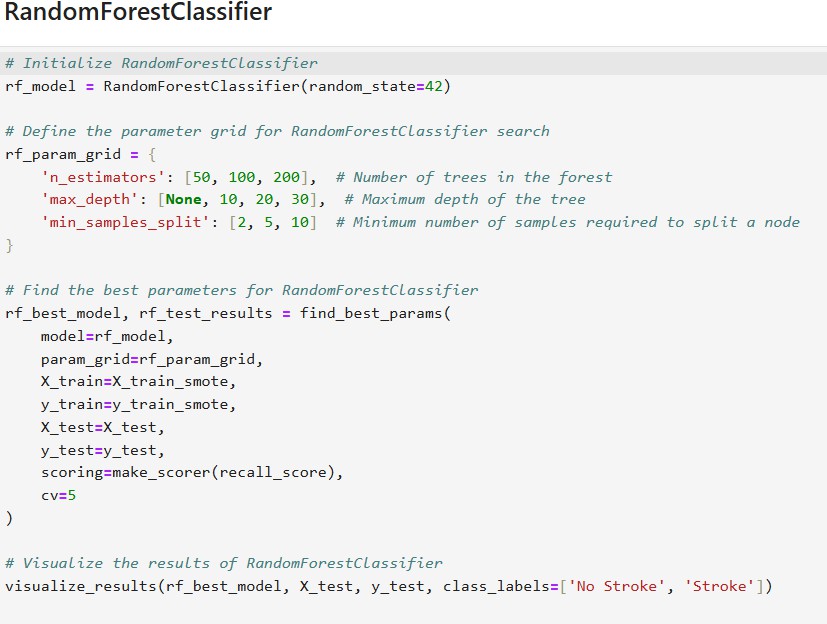


*Hình 3. 41 Mã nguồn hàm tìm kiếm, điều chỉnh tham số tốt nhất*

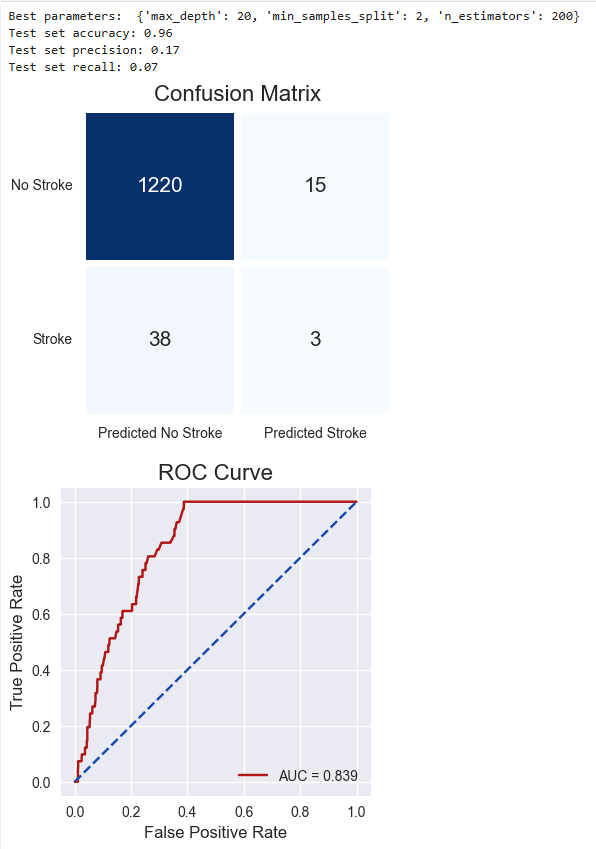


*Hình 3. 42 Mã nguồn trực quan hoá, hiển thị kết quả sau khi chạy mô hình*

## Mô hình RandomForestClassifier

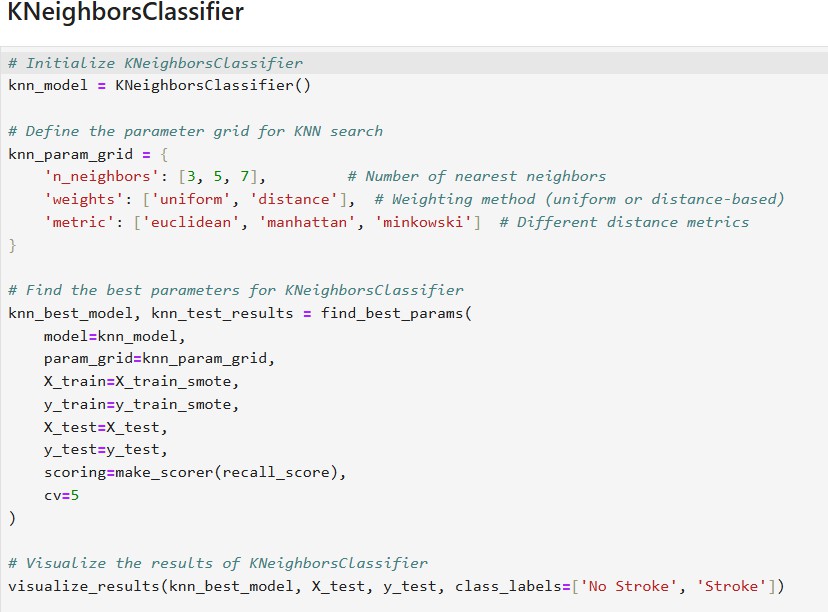
****

*Hình 3. 43 Mã nguồn mô hình RandomForestClassifier*

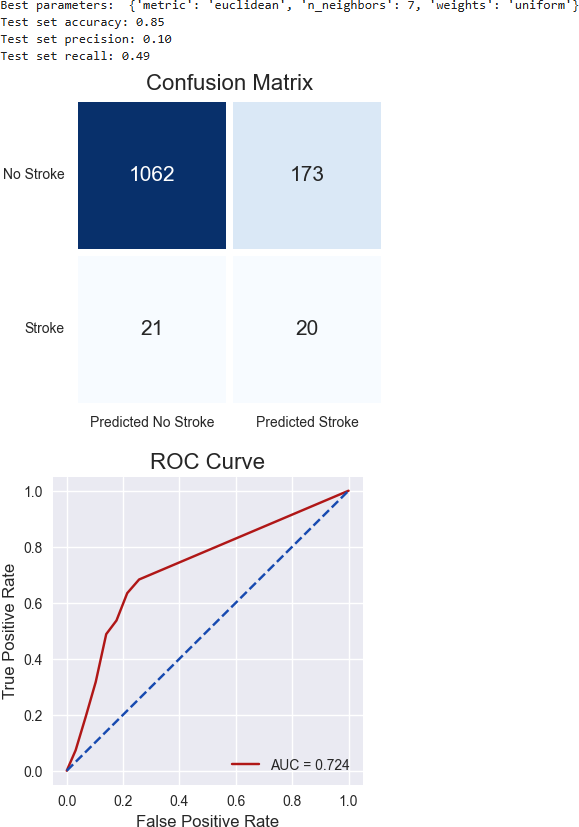
**

*Hình 3. 44 Kết quả mô hình RandomForestClassifier*

## Mô hình KNeighborsClassifier

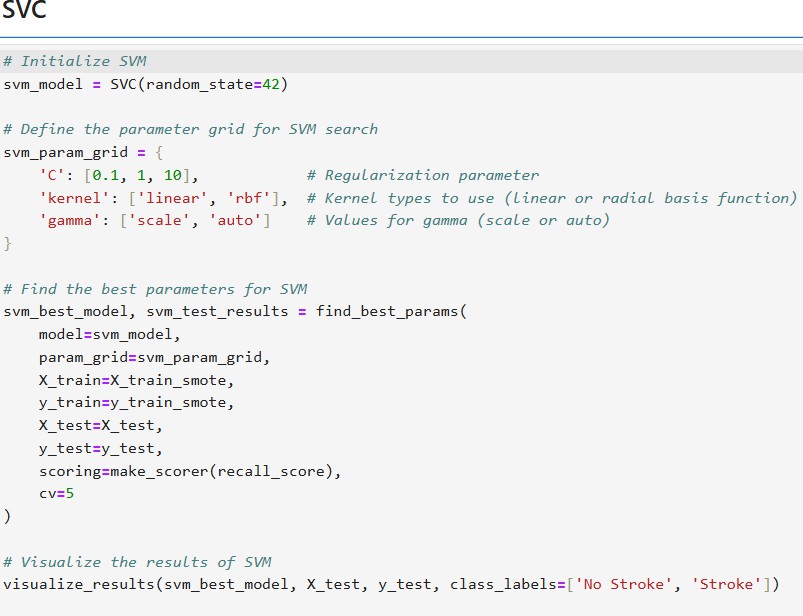
****

*Hình 3. 45 Mã nguồn mô hình KneighborsClassifier*

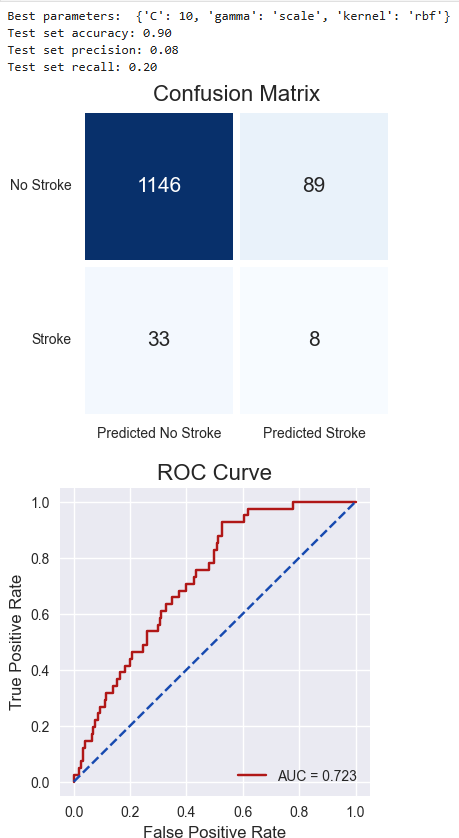
**

*Hình 3. 46 Kết quả mô hình KNeighborsClassifier*

## Mô hình Support Vector Classification

****

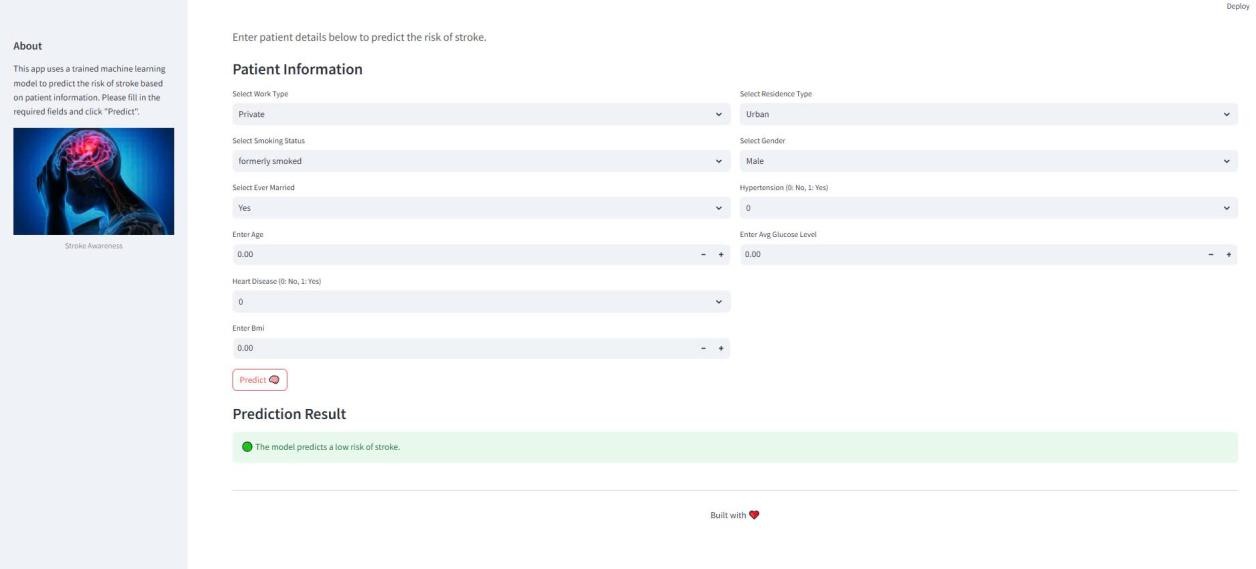
*Hình 3. 47 Mã nguồn mô hình Support Vector Classification*

**

*Hình 3. 48 Kết quả mô hình Support Vector Classification*

# CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG

Bằng cách sử dụng Streamlit của Python em đã đưa bài toán dự đoán bệnh đột quỵ lên giao diện web.



*HÌnh 4. 1 Sử dụng Streamlit để đưa bài toán lên giao diện web*

# KẾT LUẬN

## Kết quả đạt được trong kiến thức:

* + **Hiểu cách hoạt động và tầm quan trọng của các mô hình RandomForestClassifier, KNeighborsClassifier, và Support Vector Classification:**
    - Hiểu cách RandomForestClassifier tổ hợp nhiều cây quyết định để tăng độ chính xác và khả năng tổng quát hóa.
    - Nắm rõ nguyên tắc hoạt động của KNeighborsClassifier dựa trên người láng giềng gần nhất.
    - Hiểu biết về Support Vector Classification và cách nó tìm siêu phân cách tốt nhất trong không gian cao chiều.

## Áp dụng các mô hình này vào bài toán dự đoán bệnh đột quỵ:

* + - Triển khai RandomForestClassifier, KNeighborsClassifier, và Support Vector Classification để phân loại nguy cơ bị đột quỵ dựa trên các đặc trưng như huyết áp, mỡ máu, tiểu sử bệnh tật.
    - Lựa chọn và tinh chỉnh các tham số quan trọng như số cây, k trong KNN, hay tham số kernel trong SVC để tối ưu hóa hiệu suất mô hình.

## Xử lý, phân tích và trực quan hóa dữ liệu phù hợp với các mô hình:

* + - Thực hiện làm sạch dữ liệu, xử lý các giá trị thiếu, chuẩn hóa dữ liệu để đảm bảo tính nhất quán.
    - Trực quan hóa phân phối nguyên nhân để giúp nhân viên y tế hiểu tầm quan trọng của từng yếu tố.

## Mô hình dự đoán bệnh đột quỵ có thể áp dụng trong hệ thống y tế:

* + - Các mô hình này có thể tích hợp vào hệ thống cảnh báo sớm trong bệnh viện hoặc các trung tâm y tế.
    - Cung cấp khả năng phân tích nhanh theo thời gian thực, tăng cơ hộ chẩn đoán đột quỵ kịp thời.

## Hạn chế của đề tài:

* + **Biến động dữ liệu và yếu tố ngoài:**
    - Biến động nhanh chóng trong yếu tố nguy cơ như huyết áp hay cấp độ oxy có thể ảnh hưởng tới độ chính xác.

## Cân bằng dữ liệu:

* + - Sự mất cân đối trong tập dữ liệu (các trường hợp bị đột quỵ thường rất thấp so với không bị) làm cho mô hình thiên về phân lớp chiếm đa số.

## Giới hạn về tính tối ưu:

* + - Các mô hình như RandomForestClassifier và SVC phụ thuộc nhiều vào tham số tôi ưu hóa, trong khi KNN yêu cầu sự cân nhắc cấn thận với k.

## Hướng phát triển của đề tài:

* + **Tích hợp vào hệ thống y tế:**
    - Phát triển các ứng dụng trên nền tảng web và di động giúp bác sĩ và bệnh nhân giắm thiểu thời gian chẩn đoán.

## Nâng cao tính giải thích:

* + - Sử dụng các kỹ thuật trực quan hóa và giải thích như LIME hoặc SHAP để tăng sự minh bạch và đáng tin cậy của mô hình.

## Mở rộng tập dữ liệu:

* + - Thu thập thêm dữ liệu từ nhiều khu vực và nhóm tuổi để cải thiện tính tổng quát hóa.

# TÀI LIỆU THAM KHẢO

1. Dữ liệu

[Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)

1. Tài liệu

|  |  |
| --- | --- |
| [1] | Giáo trình Học máy cơ bản ( Nhà xuất bản khoa học và kỹ thuật) |
| [2] | Các slide bài thuyết trình Học máy cơ bản – các nhóm trình bày. |
| [3] | Các slide bài giảng Học máy cơ bản – thầy TS. Hoàng Quốc Việt |
| [4] | Deep Learning - Tác giả: Ian Goodfellow, Yoshua Bengio, Aaron Courville |
| [5] | Python Machine Learning - Tác giả: Sebastian Raschka, Vahid Mirjalili |
| [6] | [https://www.coursera.org/learn/machine-learning](http://www.coursera.org/learn/machine-learning) |
| [7] | https://scikit-learn.org/stable/index.html |
| [8] | [https://www.kaggle.com/](http://www.kaggle.com/) |
| [9] | https://chatgpt.com/ |