

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN HỌC: PHÂN TÍCH DỮ LIỆU**

**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI: PHÂN TÍCH NHỮNG NGUYÊN NHÂN**

**DẪN ĐẾN ĐỘT QUỴ**

**GVHD: Nguyễn Văn Thành**

**MÃ HP: DAAN436277\_23\_2\_01**

|  |  |
| --- | --- |
| **SVTH:** | **MSSV** |
| **Lê Hoàng Đức Duy** | **21133016** |
| **Nguyễn Tấn Khang** | **21133105** |
| **Nguyễn Thị Thùy Linh** | **21133051** |
| **Nguyễn Quang Nhất Linh** | **21133050** |
|  |  |

*TP. Hồ Chí Minh, tháng 05 năm 2024*

**NHẬN XÉT CỦA GIÁO VIÊN**

**ĐIỂM GV KÝ TÊN**

**LỜI CẢM ƠN**

Lời nói đầu tiên, nhóm chúng em xin được gửi đến thầy Nguyễn Văn Thành – giảng viên bộ môn *Phân tích dữ liệu* lời cảm ơn chân thành và sâu sắc nhất.

Nhóm thực hiện xin cảm ơn sự quan tâm và giúp đỡ tận tình của thầy trong suốt quá trình giảng dạy. Cảm ơn thầy đã luôn giải đáp những thắc mắc cũng như đưa ra những nhận xét, góp ý giúp nhóm thực hiện cải thiện chất lượng công việc của nhóm.

Vì khả năng còn hạn chế nên trong quá trình thực hiện báo cáo không tránh khỏi sai sót, kính mong nhận được những ý kiến đóng góp từ thầy để nhóm có thể cải thiện hơn sau này.

Nhóm thực hiện xin chân thành cảm ơn thầy!

Nhóm 6

MỤC LỤC

[**CHƯƠNG I. TỔNG QUAN** 5](#_Toc167321522)

[**1.1** **Lí do chọn đề tài** 5](#_Toc167321523)

[**1.2** **Nội dung cơ bản** 5](#_Toc167321524)

[**CHƯƠNG II. GIỚI THIỆU TẬP DỮ LIỆU** 6](#_Toc167321525)

[**2.1 Mô tả tập dữ liệu** 6](#_Toc167321526)

[**2.2 Thư viện** 6](#_Toc167321527)

[**2.3 Xem cấu trúc tập dữ liệu** 7](#_Toc167321528)

[**CHƯƠNG III. XỬ LÝ DỮ LIỆU** 10](#_Toc167321529)

[**3.1 Làm sạch dữ liệu** 10](#_Toc167321530)

[**3.2 Kiểm tra các cột** 11](#_Toc167321531)

[**3.3 EDA** 15](#_Toc167321532)

[**3.3.1 Xem xét phân phối của biến phân loại** 15](#_Toc167321533)

[**3.3.2 Xem xét phân phối của biến liên tục** 20](#_Toc167321534)

[**3.3.3 Mối tương quan giữa các biến định lượng đầu vào với nhau** 23](#_Toc167321535)

[**CHƯƠNG IV. MÔ HÌNH DỰ ĐOÁN** 26](#_Toc167321536)

[**4.1 Chuẩn bị** 26](#_Toc167321537)

[**4.2 Mô hình** 27](#_Toc167321538)

[**CHƯƠNG V. ĐÁNH GIÁ VÀ KẾT LUẬN** 32](#_Toc167321539)

[**THAM KHẢO** 33](#_Toc167321540)

# **CHƯƠNG I. TỔNG QUAN**

## **Lí do chọn đề tài**

Theo tình hình thế giới hiện nay, con người đang phải đối diện với các nguy hiểm từ các căn bệnh phức tạp kèm theo đó là tỷ lệ tử vong cao và đột quỵ là một trong số chúng. Theo thống kê, số lượng bệnh nhân đột quỵ trẻ ngày càng gia tăng. Hiện nay bệnh nhân đột quỵ dưới 40 tuổi đã chiếm 5% trong số bệnh nhân đột quỵ. Theo Tổ chức Y tế thế giới, mỗi năm có khoảng 6,5 triệu người tử vong do đột quỵ. Đây cũng là nguyên nhân gây tử vong cao thứ hai trên thế giới, chiếm khoảng 11% tổng số ca tử vong. Trung bình có 1 ca tử vong do đột quỵ sau mỗi 6 giây và cứ 6 người thì có 1 người bị đột quỵ. Nhận thấy tính nghiêm trọng của căn bệnh này với nhu cầu giúp ích trong việc phòng ngừa ngăn chặn bệnh đột quỵ, chúng em quyết định chọn đề tài “Phân tích nguyên nhân dẫn đến đột quỵ”

## **Nội dung cơ bản**

Gồm 2 vấn đề chính:

* Tìm hiểu tập dữ liệu và trực quan hóa phân tích yếu tố dẫn đến độ quỵ
* Sử dụng mô hình để dự đoán bệnh nhân đột quỵ

# **CHƯƠNG II. GIỚI THIỆU TẬP DỮ LIỆU**

## **2.1 Mô tả tập dữ liệu**

Dữ liệu được lấy trên Kaggle: <https://www.kaggle.com/code/joshuaswords/predicting-a-stroke-shap-lime-explainer-eli5/data>.

Tập dữ liệu này được sử dụng để dự đoán liệu một bệnh nhân có khả năng bị đột quỵ hay không dựa trên các thông số đầu vào như giới tính, tuổi tác, các bệnh khác nhau và tình trạng hút thuốc. Mỗi hàng trong dữ liệu cung cấp thông tin liên quan về bệnh nhân.

Dữ liệu có 510 dòng và 12 cột dữ liệu được mô tả như sau :

|  |  |
| --- | --- |
| **Tên cột** | **Ý nghĩa** |
| id | Định danh duy nhất |
| gender | "Male", "Female" hoặc "Other" |
| age | Tuổi của người bệnh |
| hypertension | 0 nếu bệnh nhân không bị tăng huyết áp, 1 nếu bệnh nhân bị tăng huyết áp |
| heart\_disease | 0 nếu bệnh nhân không mắc bất kỳ bệnh tim nào, 1 nếu bệnh nhân có bệnh tim |
| ever\_married | Đã từng kết hôn “No” hoặc “Yes” |
| work\_type | "children", "Govt\_jov", "Never\_worked", "Private" hoặc "Self-employed" |
| Residence\_type | "Rural" hoặc "Urban" |
| avg\_glucose\_level | Mức đường trung bình trong máu |
| bmi | Chỉ số khối cơ thể |
| smoking\_status | "formerly smoked", "never smoked", "smokes" or "Unknown" |
| stroke | 1 nếu bệnh nhân bị đột quỵ hoặc 0 nếu không |

## **2.2 Thư viện**

Các thư viện được sử dụng:

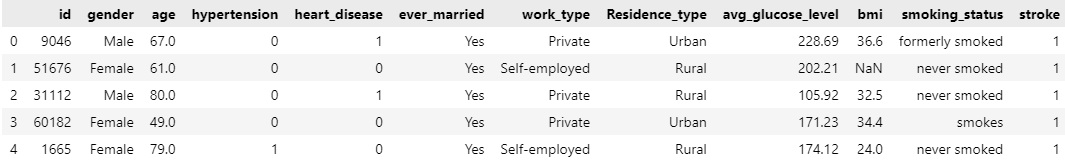
* pandas: pandas là một thư viện mạnh mẽ cho phân tích dữ liệu và xử lý dữ liệu được biểu diễn dưới dạng bảng. Nó cung cấp các cấu trúc dữ liệu như DataFrame và Series, cho phép bạn thực hiện các thao tác phức tạp như lọc, nhóm, và biến đổi dữ liệu dễ dàng.
* numpy: numpy là một thư viện cung cấp các công cụ cho việc làm việc với mảng và ma trận, làm cho việc tính toán số học trên dữ liệu số trở nên nhanh chóng và hiệu quả hơn.
* matplotlib.pyplot: matplotlib.pyplot là một thư viện trực quan hóa dữ liệu cơ bản trong Python. Nó cho phép bạn tạo ra các biểu đồ, đồ thị và hình ảnh đa dạng để trực quan hóa dữ liệu của mình.
* seaborn: seaborn là một thư viện trực quan hóa dữ liệu được xây dựng trên matplotlib, cung cấp các chức năng cao cấp hơn và giao diện thân thiện hơn để tạo ra các biểu đồ phức tạp một cách dễ dàng và nhanh chóng.
* GridSpec: GridSpec là một module trong matplotlib cho phép bạn tạo ra các lưới đa chiều phức tạp cho các biểu đồ.
* statsmodels.graphics.mosaicplot: statsmodels là một thư viện được sử dụng cho phân tích và kiểm định thống kê. Trong đó, module graphics.mosaicplot cung cấp các công cụ để tạo ra biểu đồ mosaic, một loại biểu đồ phân loại để hiển thị mối quan hệ giữa các biến phân loại.

Các thư viện này cung cấp một loạt các công cụ để xử lý, phân tích và trực quan hóa dữ liệu, giúp bạn hiểu sâu hơn về dữ liệu của mình và trình bày nó một cách dễ hiểu và hấp dẫn.

## **2.3 Xem cấu trúc tập dữ liệu**

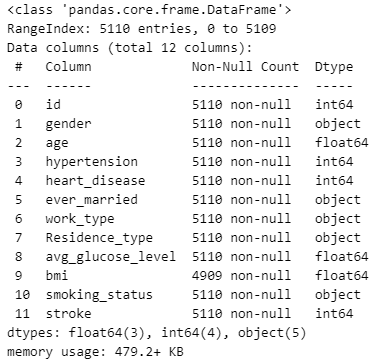
* Load dữ liệu

|  |
| --- |
| df = pd.read\_csv("./healthcare-dataset-stroke-data.csv")  df.head(5) |



* Cấu trúc

|  |
| --- |
| df.info() |



Từ cấu trúc của dữ liệu trên, ta thấy dự liệu có 3 cột thuộc kiểu float64, 4 cột thuộc kiểu int64 và 5 cột thuộc kiểu object. Nhưng cột ‘bmi’ có 4909 dòng, các dòng còn lại mang giá trị null, điều này gây bất lợi cho việc phân tích.

* Tổng quan về các đặc điểm chính

|  |
| --- |
| df.describe() |

A screenshot of a calculator

Description automatically generated

Dựa vào sự mô tả các cột trong tập dữ liệu, ta có cái nhìn tổng quát hơn :

+ Đa số người bệnh trong tập dữ liệu có số tuổi trung bình rơi vào khoảng 43 tuổi.

+ Có khoảng 9.75% người bệnh huyết áp và 5.41% người có bệnh tim.

+ Mức độ đường trung bình là 106.15 mg/dL là khá cao, có nguy cơ mắc bệnh tiểu đường.

+ BMI trung bình là 28.89 kg/m2 ở mức bình thường nhưng có 4.87% bệnh nhân bị đột quỵ

* Phân loại và đếm số lượng các cột số

|  |
| --- |
| from pandas.api.types import is\_numeric\_dtype  # Initialize counters  num\_numerical = 0  num\_non\_numerical = 0  # Iterate through columns  for c in df.columns:      if is\_numeric\_dtype(df[c]):          num\_numerical += 1      else:          num\_non\_numerical += 1  # Print the counts  print(f"Biến liên tục: {num\_numerical}")  print(f"Biến phân loại: {num\_non\_numerical}") |

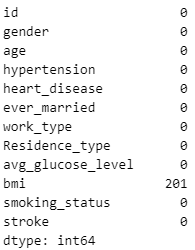
Trong tập dữ liệu, có 7 biến liên tục ( id, age, bmi, hypertension, heart\_disease, ever\_married, stroke) và 5 biến phân loại (gender, work\_type, Residence\_type, smoking\_status) phục phụ cho việc phân tích đột quỵ.

# **CHƯƠNG III. XỬ LÝ DỮ LIỆU**

## **3.1 Làm sạch dữ liệu**

* Kiểm tra giá trị null trong cột

|  |
| --- |
| null\_counts = df.isnull().sum()  print(null\_counts) |



Dữ liệu có một số giá trị thiếu trong cột "bmi", nhưng các cột khác không có giá trị thiếu. Việc xử lý dữ liệu thiếu này sẽ cần được thực hiện trước khi tiến hành các phân tích hoặc trực quan hóa dữ liệu để đảm bảo tính chính xác và đáng tin cậy của kết quả.

* Kiểm tra số lượng hàng trùng lặp

|  |
| --- |
| print(df.duplicated().sum()) |

Không có hàng nào trùng trong tập dữ liệu.

* Loại bỏ cột không cần thiết

|  |
| --- |
| df = df.drop("id", axis = 1) |

Cột “id” không có giá trị cho việc phân tích cũng như việc dự đoán đột quỵ.

* Chuyển đổi kiểu dữ liệu của cột

|  |
| --- |
| # Chuyển đổi cột "bmi" sang dạng số học để tính toán  df["bmi"] = pd.to\_numeric(df["bmi"], errors="coerce") |

Chuyển đổi cột "bmi" sang dạng số và xử lý các giá trị không hợp lệ thành NaN để thay thế giá trị null của cột 'bmi' bằng giá trị trung bình của các giá trị thực của cột.

|  |
| --- |
| # Tính giá trị trung bình của các giá trị số trong cột "bmi"  bmi\_mean = df["bmi"].mean()  # Thay thế giá trị null bằng giá trị trung bình  df.fillna({"bmi": bmi\_mean}, inplace=True) |

* Xem lại dữ liệu

|  |
| --- |
| print(df.info()) |

A screenshot of a computer

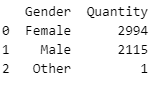
Description automatically generated

Sau khi xử lí xong, dữ liệu còn lại 11 cột và 5110 dòng.

## **3.2 Kiểm tra các cột**

* Kiểm tra giá trị cột gender

|  |
| --- |
| # Tính số lượng mẫu của cột "gender"  gender\_col = df["gender"].value\_counts().reset\_index()  # Chuyển đổi thành dataframe  gender\_col\_df = pd.DataFrame(gender\_col)  # Đặt tên cho các cột  gender\_col\_df.columns = ["Gender", "Quantity"]  #In kết quả  print(gender\_col\_df) |



Có 1 giá trị ngoại lai “Other” trong gender nên tiến hành xoá nó

|  |
| --- |
| # Tìm chỉ mục của các hàng có giới tính là "Other"  other\_index = df[df["gender"] == "Other"].index  # Xoá các hàng có giới tính là "Other" khỏi DataFrame  df.drop(other\_index, inplace=True)  # Đếm số lượng mẫu cho mỗi giới tính  gender\_counts = df["gender"].value\_counts().reset\_index()  # Chuyển đổi thành DataFrame  gender\_counts\_df = pd.DataFrame(gender\_counts)  # Đặt tên cho các cột  gender\_counts\_df.columns = ["Gender", "Quantity"]  # In kết quả  print(gender\_counts\_df) |

* Kiểm tra giá trị cột Residence\_type

|  |
| --- |
| # Tính số lượng mẫu của cột "Residence\_type"  residence\_type = df["Residence\_type"].value\_counts().reset\_index()  # Chuyển đổi thành dataframe  residence\_type\_df = pd.DataFrame(residence\_type)  # Đặt tên cho các cột  residence\_type\_df.columns = ["Place", "Quantity"]  # In kết quả  print(residence\_type\_df) |

A close up of words

Description automatically generated

* Kiểm tra cột ever\_married

|  |
| --- |
| # Tính số lượng mẫu của cột "ever\_married"  ever\_married\_col = df["ever\_married"].value\_counts().reset\_index()  # Chuyển đổi thành dataframe  ever\_married\_col\_df = pd.DataFrame(ever\_married\_col)  # Đặt tên cho các cột  ever\_married\_col\_df.columns = ["Status", "Quantity"]  # In kết quả  print(ever\_married\_col\_df) |

A close up of words

Description automatically generated

* Kiểm tra cột smoking\_status

|  |
| --- |
| # Tính số lượng mẫu của cột "smoking\_status"  smoking\_status\_col = df["smoking\_status"].value\_counts().reset\_index()  # Chuyển đổi thành dataframe  smoking\_status\_col\_df = pd.DataFrame(smoking\_status\_col)  # Đặt tên cho các cột  smoking\_status\_col\_df.columns = ["Status", "Quantity"]  # In kết quả  print(smoking\_status\_col\_df) |

A close up of words

Description automatically generated

Cột dữ liệu này có 1544 giá trị “Unknown”, vì thế ta thay thế các giá trị unknown thành các giá trị phổ biến

|  |
| --- |
| # Thay thế giá trị "Unknown" bằng giá trị phổ biến nhất  most\_common\_status = df["smoking\_status"].mode()[0]  df["smoking\_status"].replace("Unknown", most\_common\_status, inplace=True)  # Kiểm tra kết quả sau khi thay thế  print(df["smoking\_status"].value\_counts()) |

A close-up of a computer screen

Description automatically generated

Sau khi thay bằng gí trị phổ biến thì có 3436 dòng có gái trị “never smoked”

* Xử lí ngoại lệ

A graph of a bar graph

Description automatically generated with medium confidence

Các cột “bmi”, “avg\_glucose\_level” có nhiều ngoại lệ. Vì thế, loại bỏ ngoại lệ trên ta được:

+ Dữ liệu ban đầu : (5109, 11)

+ Dữ liệu sau khi bỏ ngoại lệ : (4390, 11)

## **3.3 EDA**

### **3.3.1 Xem xét phân phối của biến phân loại**

Tạo biểu đồ cho biến phân loại

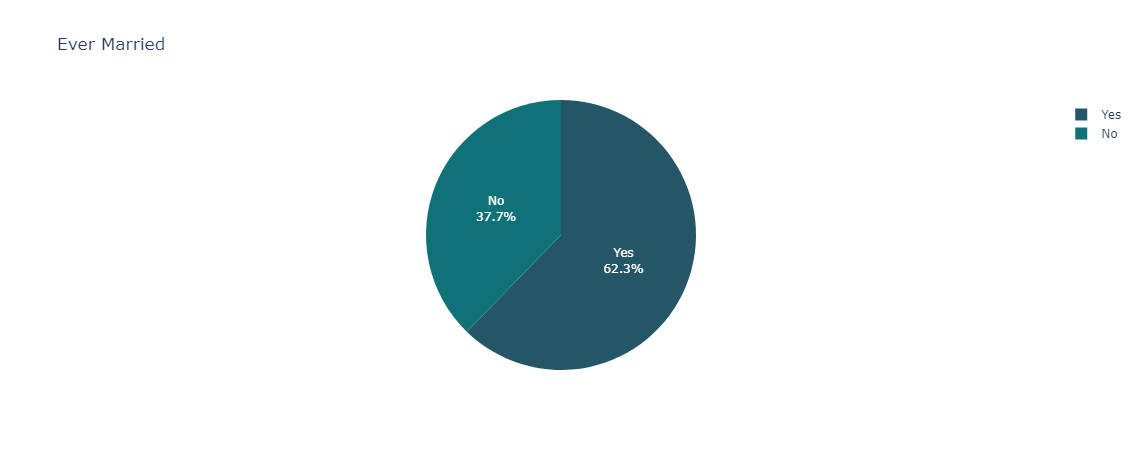
* **Giới tính**

**A graph of a patient

Description automatically generated**

Trong tập dữ liệu, đa số người bệnh là nữ giới, nam giới cũng bệnh gấp khoảng 2/3 lần. Có thể nữ giới bệnh nhiều xuất phát từ nhiểu nguyên nhân như ăn uống không hợp lí, sinh con, thẩm mĩ,…

* **Tình trạng hôn nhân**



Kết quả cho thấy rằng, đa số người bệnh đã lập gia đình chiếm hơn 50%.

* **Bệnh tim mạch**

**A blue circle with a number of percentages

Description automatically generated**

Có thể thấy rằng, người không mắc bệnh tim chiếm đa số trong tập dữ liệu.

* **Bệnh huyết áp**

**A blue circle with a number of percentages

Description automatically generated**

Người mắc bệnh huyết áp chiếm thiểu số trong tập dữ liệu.

* **Loại công việc**

**A pie chart with numbers and text

Description automatically generated**

Những người làm việc tư nhân thì mắc bệnh cao hơn những người làm những công việc còn lại. Có thể những người này làm việc trong môi trường căng thẳng, nhiều giờ, điều kiện không tốt ,… nên dễ bệnh.

* **Môi trường sống**

**A blue circle with white text

Description automatically generated**

* **Tình trạng hút thuốc lá**

A pie chart with numbers and a few different colored circles

Description automatically generated with medium confidence

Từ tập dữ liệu cho thấy, đa số người bệnh họ không hút thuốc, còn lại có hút nhưng chiếm số ít.

* Xét những tính chất (giới tính, công việc, nơi sống, hôn nhân, hút thuốc, huyết áp, bệnh tim) đối với đột quỵ

A group of gray and blue bars

Description automatically generated with medium confidence

Dựa vào các biểu đồ cột, ta nhận thấy một số đặc điểm nổi bật liên quan đến những người bị đột quỵ. Trước hết, người bị đột quỵ đa số là nữ giới. Nhóm người bệnh thường thuộc nhóm công việc tư nhân, có thể là do sự căng thẳng và áp lực công việc gây ra những tác động tiêu cực đến sức khỏe. Điều đáng chú ý là đa phần người bị đột quỵ không hút thuốc lá, cho thấy rằng có những yếu tố khác ngoài hút thuốc cũng đóng vai trò quan trọng trong nguy cơ đột quỵ. Ngoài ra, phần lớn bệnh nhân đột quỵ đã kết hôn, điều này có thể liên quan đến độ tuổi và giai đoạn cuộc sống khi nguy cơ đột quỵ tăng cao. Đa số bệnh nhân không mắc bệnh tim mạch và ít người mắc bệnh cao huyết áp, cho thấy rằng đột quỵ không nhất thiết liên quan trực tiếp đến những bệnh lý này. Tổng hợp lại, những yếu tố này giúp chúng ta hiểu rõ hơn về các đặc điểm chung của nhóm người bị đột quỵ, từ đó có thể đưa ra các biện pháp phòng ngừa và chăm sóc sức khỏe hiệu quả hơn.

### **3.3.2 Xem xét phân phối của biến liên tục**

Tạo biểu đồ cho từng biến liên tục

* **Tuổi tác**

**A graph with blue lines

Description automatically generated**

Trong tập dữ liệu, người bị đột quỵ có độ tuổi khác cao, chiếm tỉ lệ cao từ khoảng 75 tuổi trở lên. Ở độ tuổi này, người lớn tuổi có sức đề kháng yếu cũng như sức khỏe không còn tốt như còn trẻ nữa, dễ mắc các bệnh nền và tiềm ẩn nhiều nguy cơ cho sức khỏe.

* **Lượng đường trong máu**

**A graph of blue bars

Description automatically generated**

Do đối tượng phân bố khá đều, nên ta không thể kết luận gì giữa sự liên quan của mức đường huyết đối với đột quỵ.

* **Chỉ số cơ thể**

**A green bar graph with black text

Description automatically generated**

Chỉ số cơ thể của người bệnh càng cao thì có ảnh hưởng nhiều đến đột quỵ.

* **Đột quỵ**

A blue circle with a triangle in the center

Description automatically generated

Đa số bệnh nhân trong tập dữ liệu có sự chệnh lệch giữa người bệnh đột quỵ.

### **3.3.3 Mối tương quan giữa các biến định lượng đầu vào với nhau**

A screenshot of a computer

Description automatically generated

Từ biểu đồ HeatMap, cho thấy:

Tuổi tác có mối tương quan thuận mạnh với nguy cơ đột quỵ (r = 0,23). Điều này có nghĩa là nguy cơ đột quỵ tăng lên khi tuổi tác cao hơn.

Huyết áp cao và tình trạng hôn nhân có mối tương quan thuận vừa phải với nguy cơ đột quỵ (r = 0,09). Điều này có nghĩa là những người bị huyết áp cao, đã lập gia đình có nguy cơ đột quỵ cao hơn những người có huyết áp bình thường.

Chỉ số BMI có mối tương quan thuận vừa phải với nguy cơ đột quỵ (r = 0,03). Điều này có nghĩa là những người béo phì có nguy cơ đột quỵ cao hơn những người có cân nặng bình thường.

Mức đường huyết trung bình có mối tương quan nghịch vừa phải với nguy cơ đột quỵ (r = -0,03). Điều này có nghĩa là những người có mức đường huyết cao có nguy cơ đột quỵ thấp hơn những người có mức đường huyết bình thường.

Những mối tương quan này cho thấy rằng một số yếu tố nguy cơ nhất định có thể làm tăng nguy cơ đột quỵ. Điều quan trọng là phải nhận thức được những yếu tố nguy cơ này và thực hiện các bước để giảm thiểu chúng.

# **CHƯƠNG IV. MÔ HÌNH DỰ ĐOÁN**

## **4.1 Chuẩn bị**

Tập dữ liệu ban đầu đã bị mất cân bằng, vì thế việc thực hiện mô hình sẽ kém hiệu quả và không chính xác. Dùng phương pháp Smote (Synthetic Minority Over-sampling Technique) để khắc phục sự mất cân bằng đó.

Sau khi khắc phục khó khăn trên, tiến hành chia tập dữ liệu 70:30 để tiến hành thực hiện mô hình dự đoán.

Một số thư viện được sử dụng:

* LinearRegression: Là một mô hình hồi quy tuyến tính trong đó mối quan hệ giữa các biến đầu vào và biến mục tiêu được giả định là tuyến tính.
* LogisticRegression: Là một mô hình hồi quy logistic được sử dụng cho các tác vụ phân loại, trong đó biến phụ thuộc là một biến phân loại và được dự đoán dựa trên một hoặc nhiều biến độc lập.
* DecisionTreeRegressor: Một mô hình hồi quy dựa trên cây quyết định, nó tạo ra một loạt các câu hỏi được thiết lập để dự đoán giá trị đầu ra.
* DecisionTreeClassifier: Tương tự như DecisionTreeRegressor nhưng được sử dụng cho các tác vụ phân loại, trong đó biến phụ thuộc là một biến phân loại.
* RandomForestClassifier: Một mô hình phân loại dựa trên rừng ngẫu nhiên, kết hợp nhiều cây quyết định khác nhau để cải thiện hiệu suất và giảm overfitting.
* SVC (Support Vector Classifier): Là một mô hình phân loại dựa trên máy vector hỗ trợ, trong đó mục tiêu là tìm ra một ranh giới phân loại tối ưu giữa các lớp.
* classification\_report: Một báo cáo tổng hợp về hiệu suất của mô hình phân loại, bao gồm precision, recall, f1-score và support cho từng lớp.
* accuracy\_score: Một phương pháp đo lường hiệu suất của mô hình phân loại bằng cách tính tỷ lệ dự đoán chính xác trên tất cả các dự đoán.
* f1\_score: Một phương pháp đo lường trung hòa giữa precision và recall của mô hình phân loại.
* GridSearchCV: Một phương pháp tìm kiếm siêu tham số trong một không gian siêu tham số được định nghĩa trước đó, thông qua kiểm tra chéo mô hình trên các tập dữ liệu huấn luyện và kiểm tra.
* Pipeline: Một công cụ để kết hợp các bước tiền xử lý dữ liệu và mô hình hóa vào một luồng làm việc duy nhất.
* StandardScaler: Một phương pháp tiền xử lý dữ liệu để chuẩn hóa dữ liệu bằng cách loại bỏ trung bình và chia tỷ lệ biến đầu vào theo độ lệch chuẩn.
* LabelEncoder: Một phương pháp tiền xử lý dữ liệu để mã hóa biến phân loại thành giá trị số.

## **4.2 Mô hình**

* Chuẩn bị Modelling

|  |
| --- |
| df['gender'] = df['gender'].replace({'Male':0,'Female':1,'Other':-1}).astype(np.uint8)  df['Residence\_type'] = df['Residence\_type'].replace({'Rural':0,'Urban':1}).astype(np.uint8)  df['work\_type'] = df['work\_type'].replace({'Private':0,'Self-employed':1,'Govt\_job':2,'children':-1,'Never\_worked':-2}).astype(np.uint8) |

* Chia tập dữ liệu ra theo tỉ lệ 70:30

|  |
| --- |
| X  = df[['gender','age','hypertension','heart\_disease','work\_type','avg\_glucose\_level','bmi']]  y = df['stroke']  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.7, random\_state=42) |

* Sử dụng SMOTE
* SMOTE là một kỹ thuật over-sampling (tăng cường mẫu) cho các lớp thiểu số bằng cách tạo ra các mẫu tổng hợp mới dựa trên các mẫu hiện có.

|  |
| --- |
| from imblearn.over\_sampling import SMOTE, RandomOverSampler  from imblearn.under\_sampling import RandomUnderSampler  oversample = SMOTE()  X\_train\_resh, y\_train\_resh = oversample.fit\_resample(X\_train, y\_train) |

Để dự đoán người bị đột quỵ, ta tiến hành sử dụng 3 mô hình dự đoán là Random Forest, SVM, Logisitc Regression.

* **Random Forest**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **F1-score** | **support** |
| **0** | 0.97 | 0.91 | 0.94 | 1265 |
| **1** | 0.10 | 0.25 | 0.15 | 52 |
| **Accuracy** |  | | 0.89 | 1317 |
| **Macro avg** | 0.54 | 0.58 | 0.54 | 1317 |
| **Weighted avg** | 0.93 | 0.89 | 0.91 | 1317 |
| **Accuracy Score** | 0.8861047835990888 | | | |
| **F1 Score** | 0.11764705882352941 | | | |

Mô hình đạt độ chính xác tổng thể là 90%, cho thấy khả năng phân biệt tốt giữa người có nguy cơ đột quỵ cao và thấp.

Tỷ lệ người có nguy cơ cao chỉ chiếm 3.9% (52/1317), mô hình đạt recall tương đối thấp (0.25) cho lớp này.

Tỷ lệ giữa hai lớp (người có nguy cơ cao và thấp) chênh lệch khá lớn (1265/52), có thể ảnh hưởng đến hiệu suất của mô hình, đặc biệt là đối với lớp thiểu số.

Nhìn chung, mô hình Random Forest cho thấy tiềm năng trong việc dự đoán người đột quỵ với độ chính xác cao. Tuy nhiên, cần lưu ý đến vấn đề mất cân bằng dữ liệu và cải thiện hiệu suất cho lớp thiểu số để nâng cao độ tin cậy của mô hình.

* **SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **F1-score** | **support** |
| **0** | 0.97 | 0.76 | 0.85 | 1265 |
| **1** | 0.08 | 0.50 | 0.14 | 52 |
| **Accuracy** |  | | 0.75 | 1317 |
| **Macro avg** | 0.53 | 0.63 | 0.50 | 1317 |
| **Weighted avg** | 0.94 | 0.75 | 0.83 | 1317 |
| **Accuracy Score** | 0.7509491268033409 | | | |
| **F1 Score** | 0.1368421052631579 | | | |

Mô hình đạt độ chính xác (precision) 97% cho lớp người không có nguy cơ đột quỵ (lớp 0), cho thấy khả năng phân biệt tốt giữa hai lớp.

Mặc dù tỷ lệ người có nguy cơ cao (lớp 1) chỉ chiếm 3.9% (52/1265), mô hình vẫn đạt recall tương đối cao (0.50) cho lớp này.

Mô hình đạt độ chính xác tổng thể chỉ 75%, thấp hơn so với mô hình Random Forest (90%).

Nhìn chung, mô hình SVM cho thấy khả năng phân biệt tốt giữa người không có nguy cơ đột quỵ (lớp 0) và người có nguy cơ cao (lớp 1), đặc biệt là đối với lớp đa số. Tuy nhiên, độ chính xác tổng thể của mô hình thấp hơn so với mô hình Random Forest và cần cải thiện hiệu suất cho lớp thiểu số để nâng cao độ tin cậy của mô hình.

* **Logisitc Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **F1-score** | **support** |
| **0** | 0.98 | 0.78 | 0.87 | 1265 |
| **1** | 0.10 | 0.60 | 0.17 | 52 |
| **Accuracy** |  | | 0.77 | 1317 |
| **Macro avg** | 0.54 | 0.69 | 0.52 | 1317 |
| **Weighted avg** | 0.94 | 0.77 | 0.84 | 1317 |
| **Accuracy Score** | 0.7744874715261959 | | | |
| **F1 Score** | 0.17270194986072424 | | | |

Mô hình đạt độ chính xác (precision) 98% cho lớp người không có nguy cơ đột quỵ (lớp 0), cho thấy khả năng phân biệt tốt giữa hai lớp.

Dù tỷ lệ người có nguy cơ cao (lớp 1) chỉ chiếm 3.9% (52/1265), mô hình vẫn đạt độ thu hồi (recall) tương đối (0.60) cho lớp này.

Mô hình đạt độ chính xác tổng thể chỉ 77%, thấp hơn so với mô hình Random Forest (90%) và SVM (75%).

Mô hình Logistic Regression cho thấy khả năng phân biệt tốt giữa người không có nguy cơ đột quỵ (lớp 0) và người có nguy cơ cao (lớp 1), đặc biệt là đối với lớp đa số. Tuy nhiên, độ chính xác tổng thể của mô hình thấp hơn so với mô hình Random Forest và SVM và cần cải thiện hiệu suất cho lớp thiểu số để nâng cao độ tin cậy của mô hình.

* **So sánh mô hình Random Forest, SVM và Logistic Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô hình** | **Độ chính xác** | **F1 cho lớp 1** | **Ưu điểm** | **Nhược điểm** |
| **Random Forest** | 90% | 0.1477 | Độ chính xác cao, dễ áp dụng | Thiếu cân bằng dữ liệu |
| **SVM** | 75% | 0.1368 | Chịu ảnh hưởng thấp bởi lớp thiểu số | Độ chính xác tổng thể thấp |
| **Logistic Regression** | 77% | 0.1727 | Dễ áp dụng, dễ hiểu | Thiếu cân bằng dữ liệu, độ chính xác tổng thể thấp |

Mô hình trên ta có thể thấy mô hình RandomForest có độ chính xác cao nhất, nhưng recall và F1-score cho class 1 thấp, cho thấy có thể có nhiều trường hợp đột quỵ bị bỏ sót. Tiếp đến 2 mô hình Logistic Regression và SVM có recall tốt hơn, nhưng precision và F1-score của class 1 vẫn thấp.

Tất cả các mô hình đều gặp khó khăn trong việc phát hiện các trường hợp đột quỵ (class 1) và có xu hướng dự đoán sai cho class 1. Điều này có thể yêu cầu các cải tiến hoặc sử dụng các phương pháp khác nhau để cải thiện hiệu suất dự đoán trên tập dữ liệu này.

Vì thế, việc sử dụng SMOTE không hiệu quả nên ta thử đổi phương pháp sang Random Forest kết hợp với OverSampling và Random Forest kết hợp với UnderSampling.

UnderSampling là kỹ thuật giảm số lượng mẫu của lớp phổ biến để đạt được sự cân bằng với lớp thiểu số.

OverSampling là kỹ thuật tăng số lượng mẫu của lớp thiểu số để đạt được sự cân bằng với lớp phổ biến.

* **Random Forest kết hợp với OverSampling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **F1-score** | **support** |
| **0** | 1.00 | 0.94 | 0.97 | 2077 |
| **1** | 0.94 | 1.00 | 0.97 | 2067 |
| **Accuracy** |  | | 0.97 | 4144 |
| **Macro avg** | 0.97 | 0.97 | 4144 | 4144 |
| **Weighted avg** | 0.97 | 0.97 | 0.97 | 4144 |

Mô hình đạt được độ chính xác 94%, cho thấy khả năng phân loại dữ liệu chính xác cao. Độ chính xác cho cả hai lớp đều cao, đặc biệt là lớp 0 đạt 100%. Điểm F1 cho lớp 0 là 94%, cho thấy sự cân bằng tốt giữa độ chính xác và độ nhạy. Số lượng mẫu trong mỗi lớp tương đối cân bằng.

* **Random Forest kết hợp với UnderSampling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **F1-score** | **support** |
| **0** | 0.70 | 0.72 | 0.71 | 78 |
| **1** | 0.72 | 0.70 | 0.71 | 81 |
| **Accuracy** |  | | 0.71 | 159 |
| **Macro avg** | 0.71 | 0.71 | 0.71 | 159 |
| **Weighted avg** | 0.71 | 0.71 | 0.71 | 159 |

Mô hình đạt được độ chính xác 84%, cho thấy khả năng phân loại dữ liệu tương đối tốt. Độ chính xác cho lớp 0 cao hơn so với lớp 1. Điểm F1 cho lớp 0 là 87%, cho thấy sự cân bằng giữa độ chính xác và độ nhạy tương đối tốt. Số lượng mẫu trong mỗi lớp tương đối thấp, đặc biệt là lớp 1, có thể ảnh hưởng đến hiệu suất của mô hình.

🡺 Dựa trên các phân tích trên, **mô hình Random Forest kết hợp với OverSampling là lựa chọn tốt hơn cho bài toán dự đoán người đột quỵ.** Mô hình đạt được độ chính xác cao, điểm F1 cao và hỗ trợ cân bằng cho cả hai lớp.

# **CHƯƠNG V. ĐÁNH GIÁ VÀ KẾT LUẬN**

Dựa vào sự phân tích trên, những người bị đột quỵ đa số có độ tuổi, mắc bệnh tim, lượng đường trong cơ thể và chỉ số BMI cao hơn so với người không mắc bệnh.

Kết hợp với việc EDA để phân tích nguyên nhân dẫn đến đột quỵ, trong tương lai cần:

* Tổ chức các chương trình khám sức khỏe cộng đồng và tăng cường giáo dục về tầm quan trọng của việc kiểm tra sức khỏe định kỳ.
* Khuyến khích bệnh nhân tuân thủ liệu trình điều trị của bác sĩ, sử dụng thuốc đúng liều, và duy trì một lối sống lành mạnh (chế độ ăn uống cân bằng, tập thể dục đều đặn, và tránh stress).
* Tăng cường kiểm tra đường huyết thường xuyên, duy trì chế độ ăn uống lành mạnh, tập thể dục đều đặn, và tuân thủ phác đồ điều trị của bác sĩ. Khuyến khích bệnh nhân sử dụng các công nghệ hỗ trợ quản lý tiểu đường như máy đo đường huyết và ứng dụng di động.
* Tư vấn dinh dưỡng và lập kế hoạch ăn uống khoa học, khuyến khích tham gia các chương trình giảm cân và duy trì hoạt động thể chất thường xuyên. Đào tạo và cung cấp thông tin cho cộng đồng về lợi ích của việc duy trì cân nặng hợp lý.
* Đảm bảo rằng các cơ sở y tế có đủ trang thiết bị và nhân lực để xử lý kịp thời các trường hợp đột quỵ.

Việc giảm thiểu nguy cơ đột quỵ yêu cầu một cách tiếp cận toàn diện, bao gồm kiểm soát các yếu tố nguy cơ như bệnh tim, tiểu đường, và thừa cân. Đồng thời, cần tăng cường giáo dục cộng đồng và khuyến khích thay đổi lối sống lành mạnh. Điều này không chỉ giúp giảm nguy cơ đột quỵ mà còn cải thiện chất lượng cuộc sống tổng thể.

Dựa trên tập dữ liệu hiện có, mô hình Random Forest kết hợp với phương pháp oversampling đã chứng minh hiệu quả vượt trội trong việc dự đoán đột quỵ khi sử dụng tất cả các biến số. Mô hình này đạt độ chính xác cao, lên đến 96%. Khi xem xét các chỉ số precision, recall và F1-score, tỷ lệ phần trăm của mô hình Random Forest kết hợp với oversampling cũng cao hơn so với các mô hình khác. Điều này cho thấy mô hình này rất phù hợp cho việc dự đoán đột quỵ, mang lại kết quả đáng tin cậy và có giá trị trong việc hỗ trợ quyết định y tế.

# **THAM KHẢO**

[1]. Slide bài giảng của thầy Nguyễn Văn Thành

[2].<https://www.kaggle.com/code/joshuaswords/predicting-a-stroke-shap-lime-explainer-eli5/data> (1/5/2024)

[3]. <https://phamdinhkhanh.github.io/2020/02/17/ImbalancedData.html>, phamdinhkhanh, (5/5/2024)

[4]. <https://www.geeksforgeeks.org/support-vector-machine-algorithm/> ,(5/5/2024)

[5]. <https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html> , Tuấn Nguyễn, (6/5/2024)