Đồ án 2

Stroke prediction using machine learning model

Mentor: Võ Minh Thẩn
GV phụ trách: TS. Nguyễn Văn Vũ
Thực hiện: Nhóm 3

Giới thiệu đề tài

Đột quỵ là một trong những nguyên nhân gây tử vong hàng đầu trên toàn thế giới. Đây là căn bệnh nguy hiểm, có thể tước đoạt mạng sống của người bệnh một cách đột ngột nếu không được phát hiện và can thiệp kịp thời.

Việc chẩn đoán sớm là yếu tố then chốt để cứu sống bệnh nhân và giảm thiểu di chứng nặng nề. Tuy nhiên, không phải lúc nào các dấu hiệu cảnh báo cũng rõ ràng hoặc được nhận biết đúng lúc.

Vì lý do đó, nhóm chúng em đã thực hiện dự án này, nhằm hỗ trợ phát hiện sớm nguy cơ đột quỵ bằng mô hình học máy.

Dự án sử dụng các đặc trưng đầu vào như: *tuổi*, *giới tính*, *chỉ số đường huyết*, *huyết áp*, v.v... để huấn luyện mô hình. Kết quả, mô hình đạt độ chính xác tổng thể **84%** trong việc dự đoán nguy cơ đột quỵ.

Nội dung

1. Giới thiệu bài toán
2. Giới thiệu bộ dữ liệu
3. Cơ sở lý thuyết
4. Khám phá dữ liệu
5. Tiền xử lý dữ liệu
6. Lựa chọn mô hình và huấn luyện
7. Áp dụng kỹ thuật Stacking Model
8. Lưu và gọi lại mô hình thực hiện dự đoán
9. Tổng kết

1. Giới thiệu bài toán

- Bài toán phân lớp nhị phân: **Dự đoán nguy cơ đột quỵ**.
- Mỗi cá nhân được phân vào một trong hai lớp:
  - Lớp 1: Có nguy cơ đột quỵ (stroke $= 1$)
  - Lớp 0: Không có nguy cơ đột quỵ (stroke $= 0$)
- Dữ liệu gồm các đặc điểm như: tuổi, giới tính, chỉ số đường huyết, tiền sử bệnh, lối sống,...
- **Mục tiêu:** Hỗ trợ phát hiện sớm nguy cơ đột quỵ để can thiệp kịp thời.
- Góp phần tối ưu hoá điều trị và hỗ trợ bác sĩ trong phân loại, theo dõi bệnh nhân.

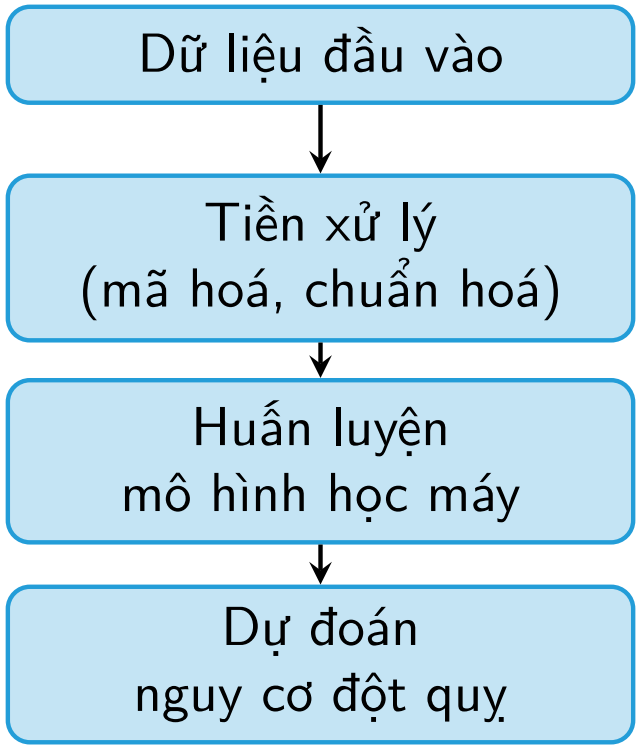
2. Giới thiệu bộ dữ liệu

- 5 110 bản ghi, 12 thuộc tính.
- Dữ liệu bao gồm các biến đầu vào (feature) và nhãn đầu ra (label).

Tên cột	**Ý nghĩa**
`id`	Mã định danh duy nhất của mỗi cá nhân
`gender`	Giới tính của cá nhân (`Male`, `Female`, hoặc `Other`)
`age`	Tuổi của cá nhân (tính theo năm)
`hypertension`	Tăng huyết áp (0 = Không, 1 = Có)
`heart_disease`	Bệnh tim (0 = Không, 1 = Có)
`ever_married`	Đã từng kết hôn hay chưa (`Yes`, `No`)
`work_type`	Loại hình công việc (`Private`, `Self-employed`, `Govt_job`, `children`, `Never_w`
`Residence_type`	Loại nơi cư trú (`Urban` = thành thị, `Rural` = nông thôn)
`avg_glucose_level`	Mức đường huyết trung bình (mg/dL)
`bmi`	Chỉ số khối cơ thể (Body Mass Index)
`smoking_status`	Tình trạng hút thuốc (`formerly smoked`, `never smoked`, `smokes`, `Unknown`)
`stroke`	Đã từng bị đột quỵ hay chưa (0 = Không, 1 = Có)

3.1 Học máy và bài toán phân lớp (nhị phân)

Mục tiêu: Xây dựng mô hình dự đoán nguy cơ đột quỵ (stroke)
Bài toán: Phân lớp nhị phân (0: Không bị, 1: Có nguy cơ)



Các đặc trưng như tuổi, giới tính, chỉ số đường huyết,... sẽ được dùng để huấn luyện mô hình phân loại.

3.2 Các thuật toán đã sử dụng

- **Logistic Regression:**
  Mô hình hồi quy tuyến tính cho phân lớp nhị phân, tính toán xác suất xảy ra của một sự kiện.
- **Random Forest:**
  Kết hợp nhiều cây quyết định, giúp giảm overfitting và tăng độ chính xác thông qua voting.
- **XGBoost:**
  Boosting mạnh mẽ, tối ưu hóa gradient và tốc độ huấn luyện, thường cho kết quả tốt trong các bài toán thực tế.
- **Stacking Ensemble:**
  Kết hợp đầu ra của các mô hình trên thành một mô hình tổng hợp (meta-model) để tăng hiệu suất.

Các mô hình riêng lẻ được đánh giá độc lập, sau đó tích hợp theo phương pháp Stacking để nâng cao độ chính xác.

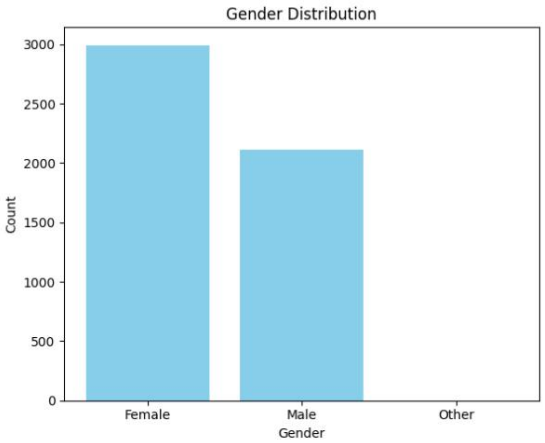
4.1 Xem thống kê các cột dữ liệu số

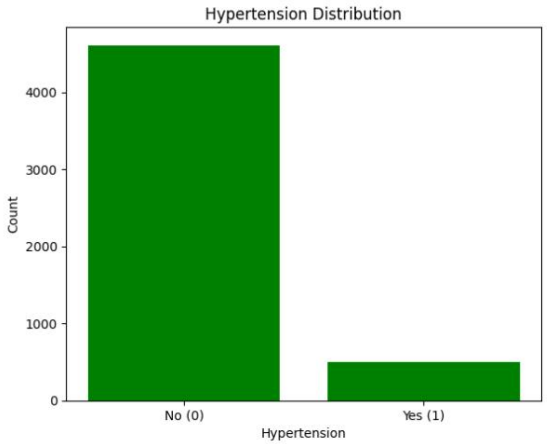
	count	mean	std	min	25%	50%	75%	max
id	5110.0	36517.829354	21161.721625	67.00	17741.250	36932.000	54682.00	72940.00
age	5110.0	43.226614	22.612647	0.08	25.000	45.000	61.00	82.00
hypertension	5110.0	0.097456	0.296607	0.00	0.000	0.000	0.00	1.00
heart_disease	5110.0	0.054012	0.226063	0.00	0.000	0.000	0.00	1.00
avg_glucose_level	5110.0	106.147677	45.283560	55.12	77.245	91.885	114.09	271.74
bmi	4909.0	28.893237	7.854067	10.30	23.500	28.100	33.10	97.60
stroke	5110.0	0.048728	0.215320	0.00	0.000	0.000	0.00	1.00

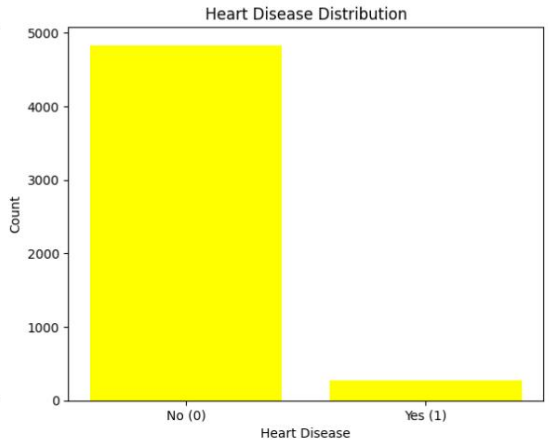
4.2 Xem thống kê các cột dữ liệu phân loại

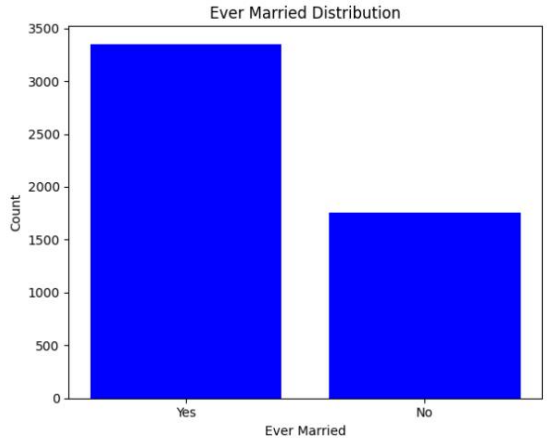
	gender	ever_married	work_type	Residence_type	smoking_status
count	5110	5110	5110	5110	5110
unique	3	2	5	2	4
top	Female	Yes	Private	Urban	never smoked
freq	2994	3353	2925	2596	1892

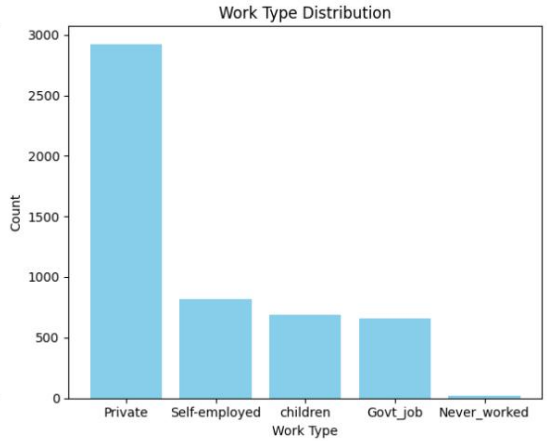
4.3 Biểu đồ tần suất cho các cột

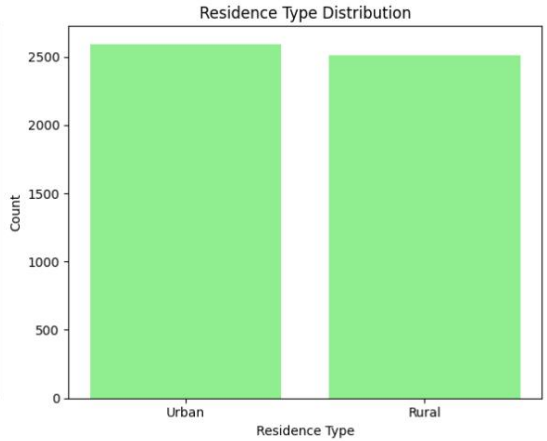




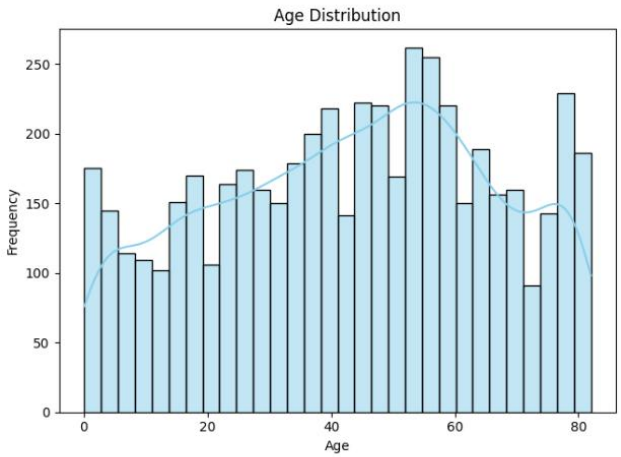


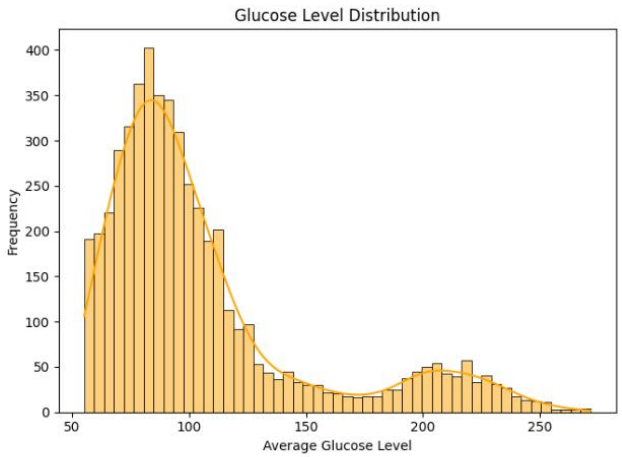


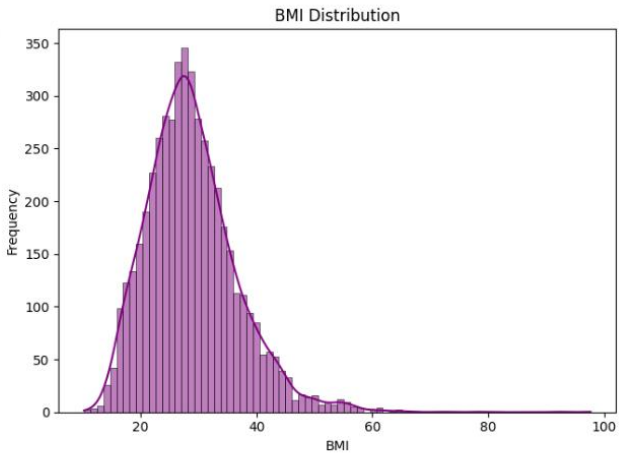




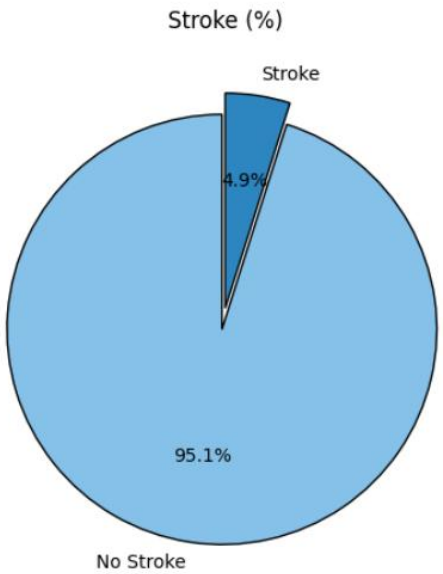
4.4 Biểu đồ tần suất cho các cột

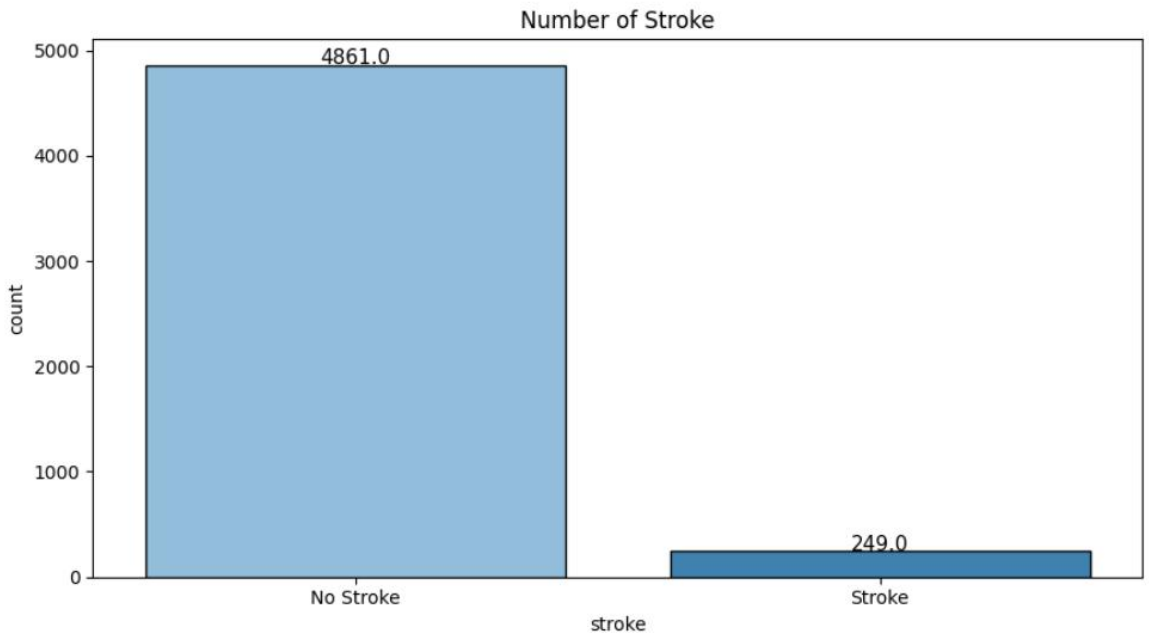




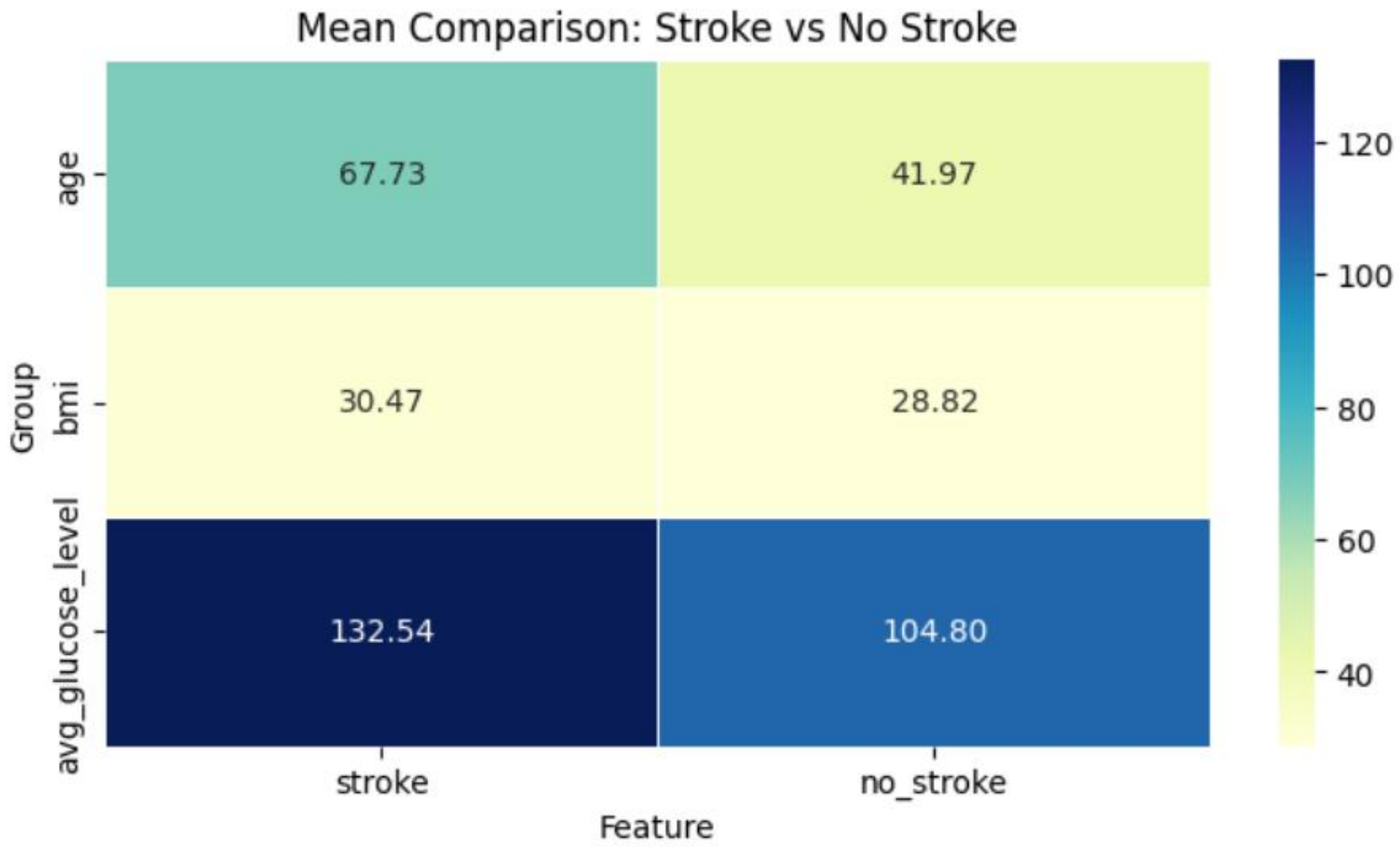


4.5 Phân bố nhãn

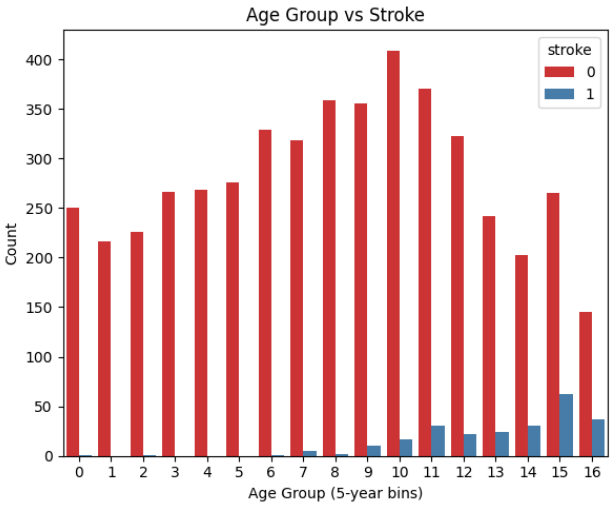


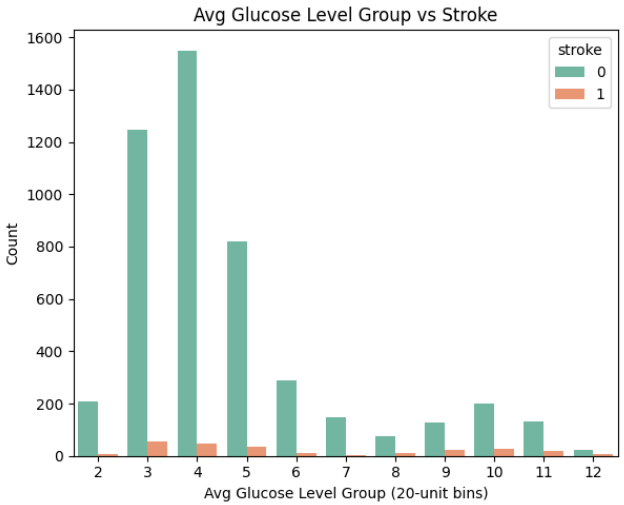


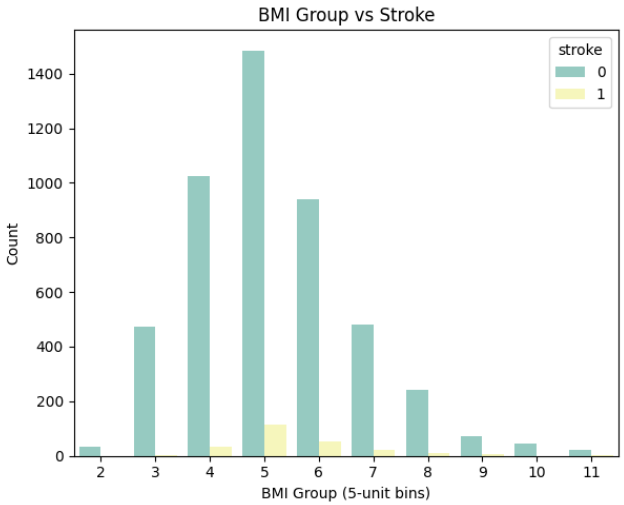
4.6 Trung bình Age, bmi, glucose giữa Stroke/No Stroke



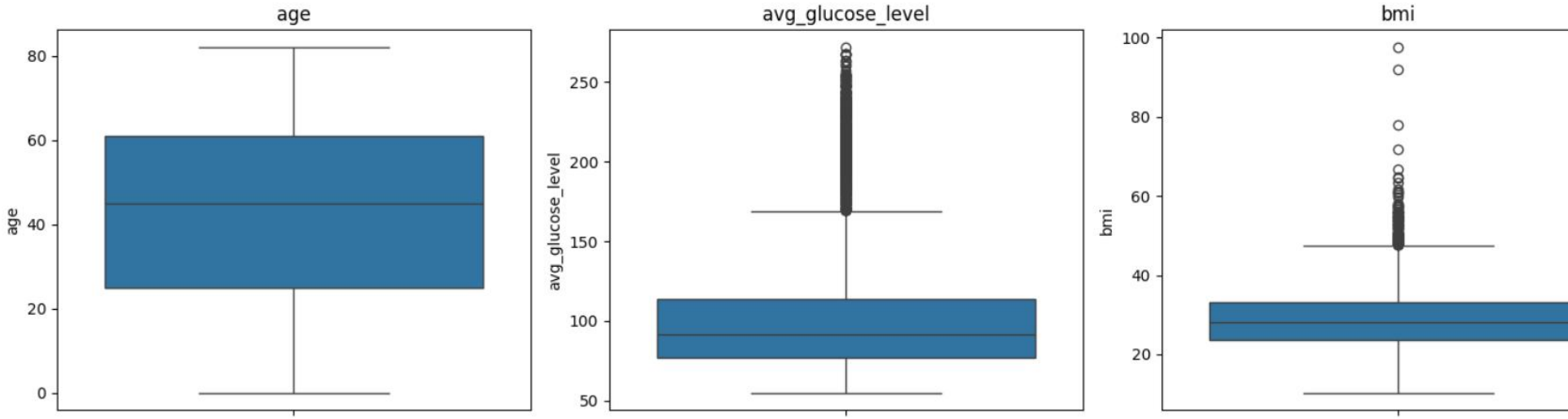
4.7 age, glucose, bmi vs stroke



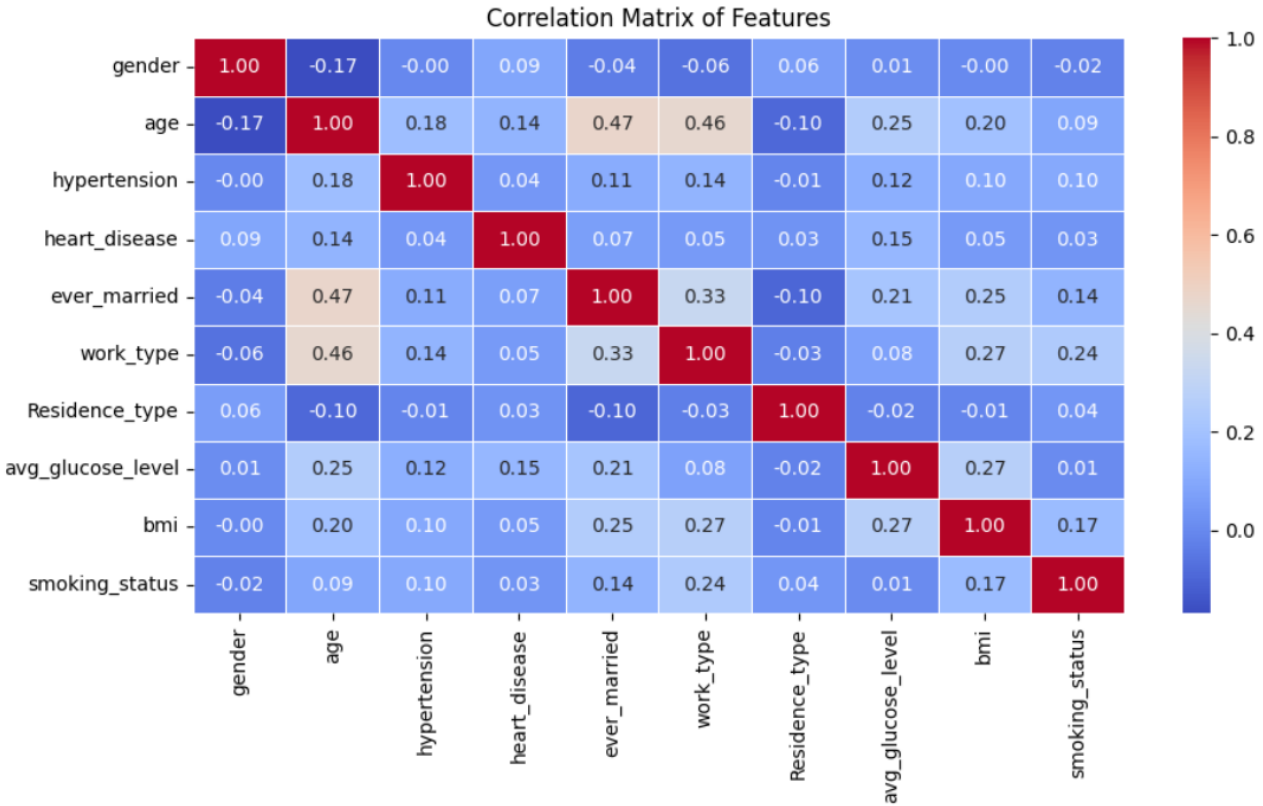




4.8 Outliers


age
age
0
20
40
60
80
avg_glucose_level
avg_glucose_level
50
100
150
200
250
bmi
bmi
20
40
60
80
100

4.9 Ma trận tương quan



Nhìn chung thì không có các cặp biến nào tương quan quá cao để dẫn tới tình trạng đa cộng tuyến.

5. Tiền xử lý dữ liệu

Điền giá trị thiếu
`bmi.fillna(mean)`

Ép kiểu
`age.astype(int)`

Chuẩn hoá văn bản
`work_type.strip().lower()`

Loại bỏ giá trị không hợp lệ
`gender == "Other"`

Xử lý ngoại lệ
`avg_glucose_level <= 250`

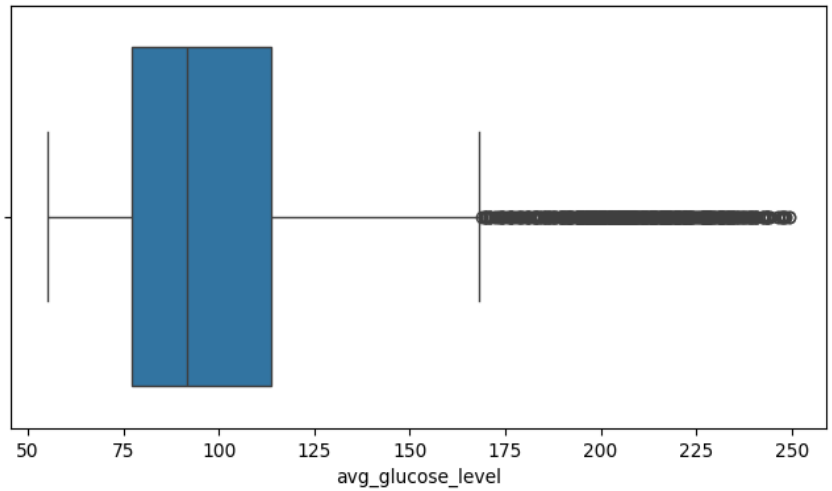
Xử lý ngoại lệ
`13 < bmi < 60`

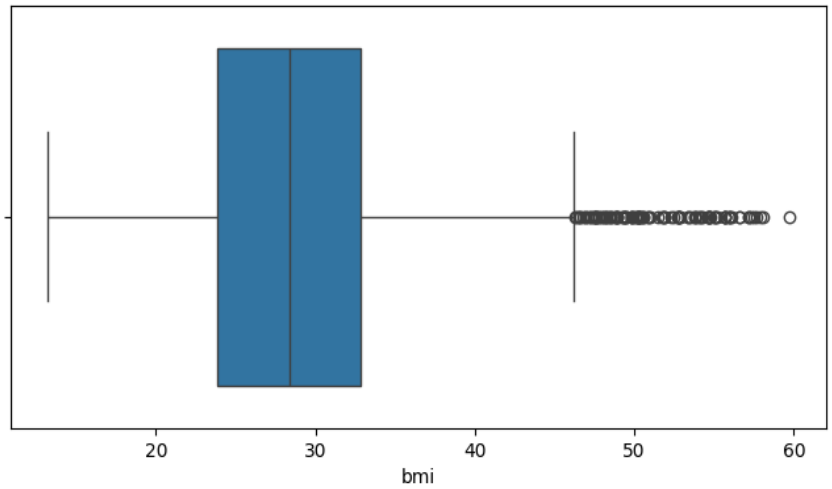
Tiền xử lý giúp loại bỏ nhiễu, đảm bảo dữ liệu sạch và phù hợp để huấn luyện mô hình.

Kết quả xử lý dữ liệu

- **Kiểm tra giá trị thiếu:**
  - `bmi: 0` — không có giá trị thiếu.
- **Chuẩn hoá cột `work_type`:**
  - Giá trị đã được chuyển thành chữ thường, không còn khoảng trắng thừa.
  - Các giá trị sau khi chuẩn hoá: `['private', 'self-employed', 'govt_job', 'children', 'never_worked']`
- **Xử lý cột `gender`:**
  - Đã loại bỏ giá trị không hợp lệ `'Other'`.
  - Kết quả còn lại: `['Male', 'Female']`

Kết quả xử lý ngoại lệ





5 Tiền xử lý dữ liệu

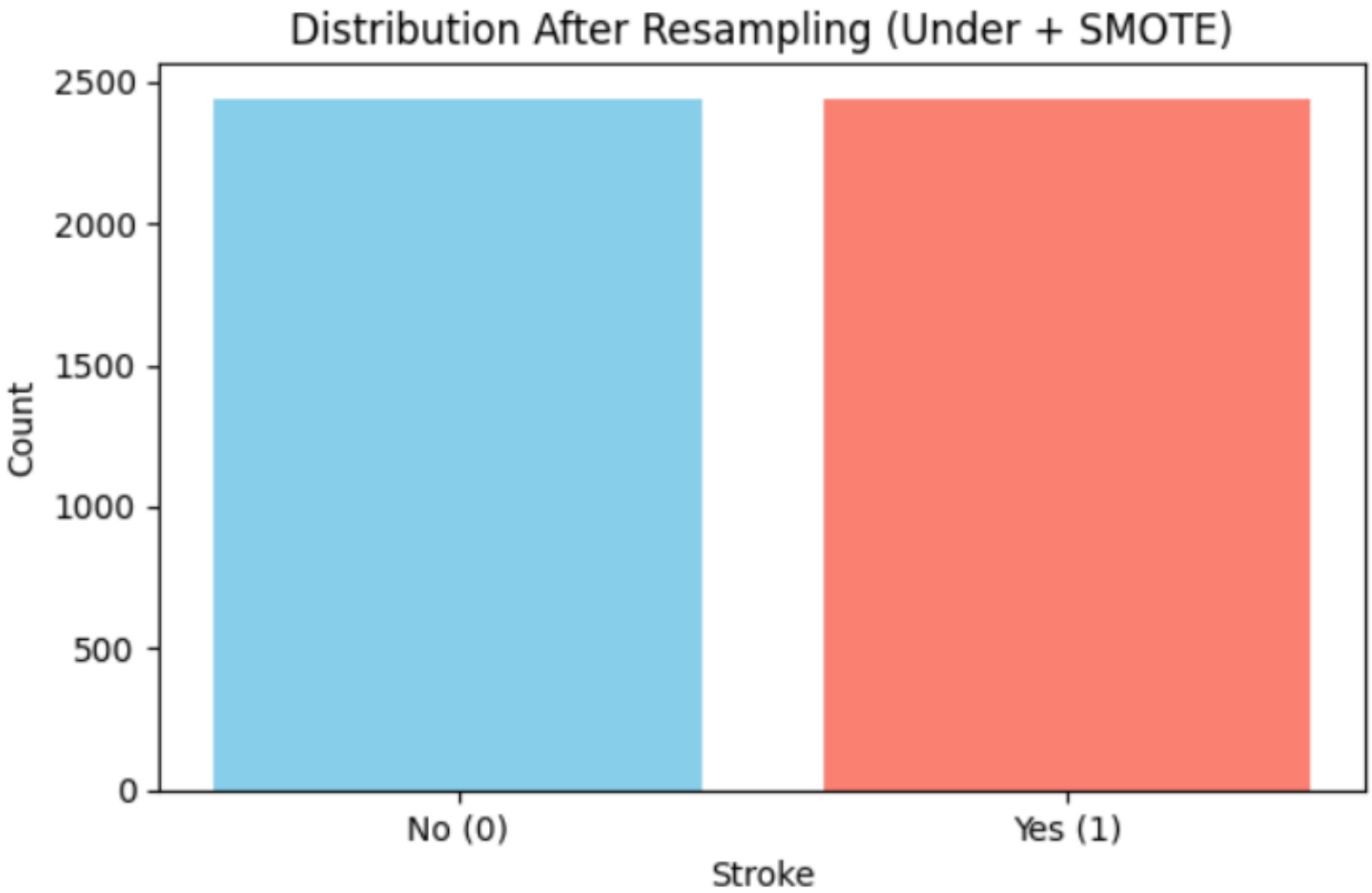
5.1 Xử lý mất cân bằng nhãn

Phương pháp kết hợp under-sampling và SMOTE để cân bằng số lượng giữa hai nhãn trong tập dữ liệu.

```
over = SMOTE(sampling_strategy = 1)
under = RandomUnderSampler(sampling_strategy = 0.1)
x = df1.loc[:,:'smoking_status']
y = df1.loc[:,'stroke']

steps = [('under', under),('over', over)]
pipeline = Pipeline(steps=steps)
x, y = pipeline.fit_resample(x, y)
Counter(y)
```

Kết quả sau xử lý mất cân bằng nhãn



5.2 Chia dữ liệu train/test

Tỷ lệ chia: 80% train — 20% test

- Dữ liệu được chia thành hai tập:
  - **Tập huấn luyện (Train set):** chiếm 80% tổng dữ liệu, được sử dụng để huấn luyện mô hình.
  - **Tập kiểm tra (Test set):** chiếm 20% tổng dữ liệu, được sử dụng để đánh giá hiệu quả mô hình trên dữ liệu chưa từng thấy.
- Việc chia dữ liệu giúp kiểm tra khả năng tổng quát hoá của mô hình và tránh overfitting.
- Phân chia đảm bảo tỷ lệ giữa hai nhãn được giữ nguyên (stratified splitting).

5.3 Mã hoá biến phân loại (Label Encoding)

Các biến phân loại đã được mã hoá như sau:

- **gender**: 1 = `Male`, 0 = `Female`
- **ever_married**: 1 = `Yes`, 0 = `No`
- **work_type**:
  - 0 = `children`
  - 1 = `govt_job`
  - 2 = `never_worked`
  - 3 = `private`
  - 4 = `self-employed`
- **Residence_type**: 1 = `Urban`, 0 = `Rural`
- **smoking_status**:
  - 0 = `Unknown`
  - 1 = `formerly smoked`
  - 2 = `never smoked`
  - 3 = `smokes`

Mã hoá giúp chuyển các biến phân loại sang dạng số để huấn luyện mô hình.

5.4 Chuẩn hoá dữ liệu

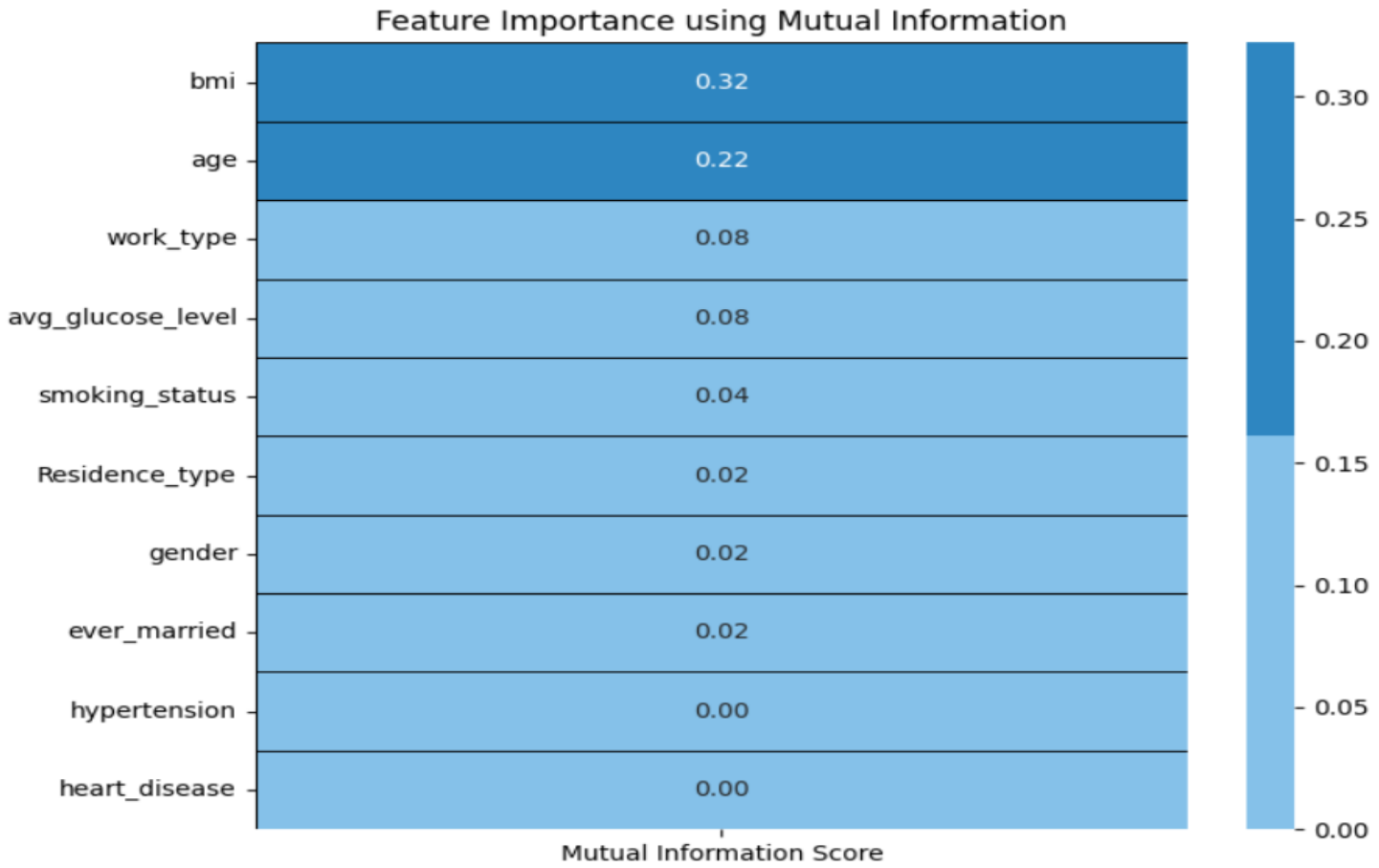
Đưa các đặc trưng về cùng một thang đo

```
from sklearn.preprocessing import MinMaxScaler,
    StandardScaler
ms = MinMaxScaler()
ss = StandardScaler()

x_train['age'] = ms.fit_transform(x_train[['age']])
x_test['age'] = ms.transform(x_test[['age']])
x_train['avg_glucose_level'] = ms.fit_transform(x_train[[
    'avg_glucose_level']])
x_test['avg_glucose_level'] = ms.transform(x_test[['
    avg_glucose_level']])
x_train['bmi'] = ss.fit_transform(x_train[['bmi']])
x_test['bmi'] = ss.transform(x_test[['bmi']])
```

Chuẩn hoá giúp cải thiện hiệu quả huấn luyện, đặc biệt với các mô hình như SVM, KNN, hoặc các thuật toán dùng Gradient Descent.

5.5 Tính toán độ quan trọng của các đặc trưng



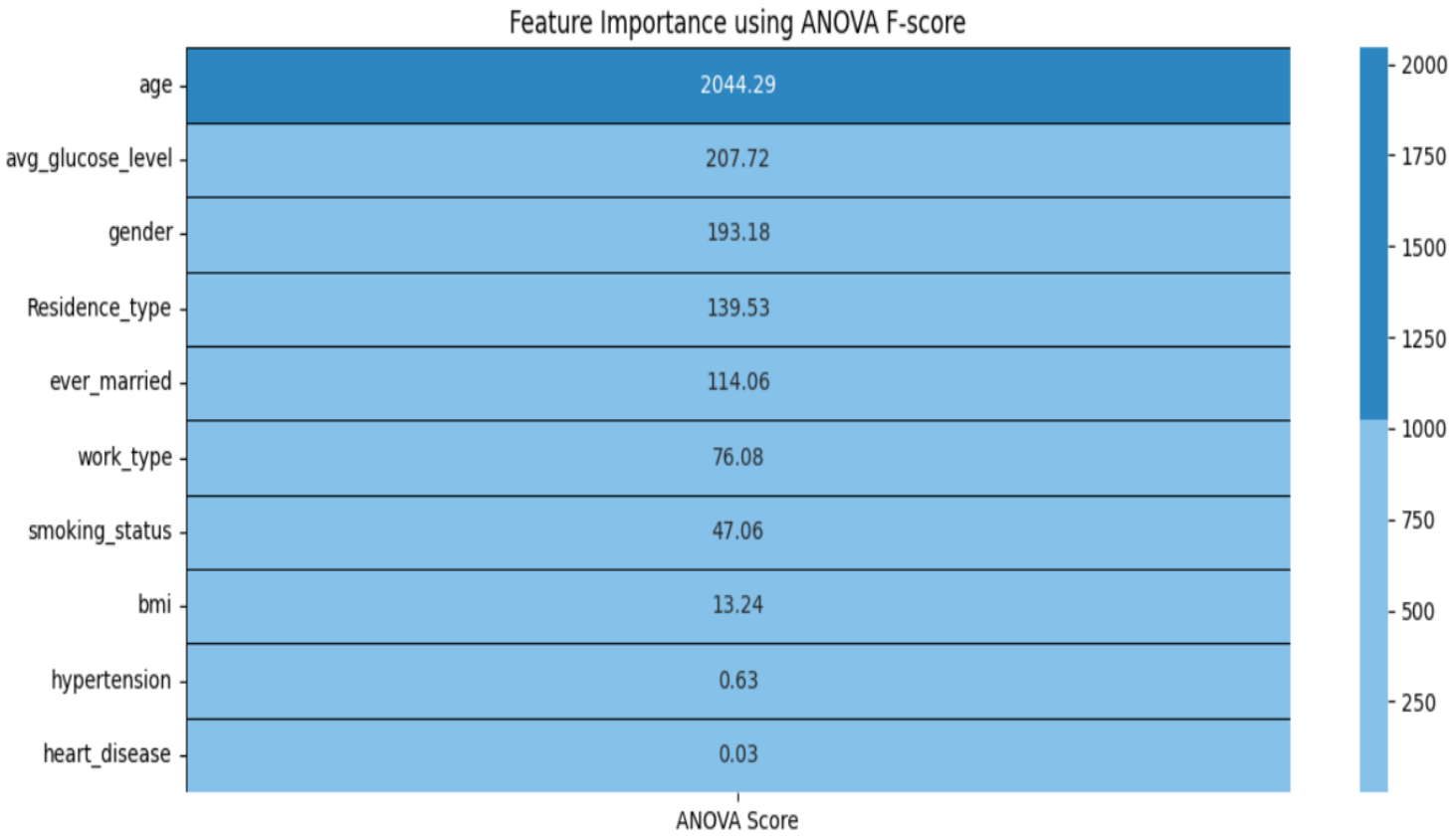
Độ quan trọng của đặc trưng tính bằng Mutual Information

Mutual Information (MI) đo lường mức độ phụ thuộc giữa đặc trưng và nhãn.

MI cao $\Rightarrow$ đặc trưng có ảnh hưởng lớn đến kết quả phân loại.

Giúp chọn lọc đặc trưng quan trọng, giảm nhiễu và cải thiện hiệu suất mô hình.

5.5 Tính toán độ quan trọng của các đặc trưng



Đánh giá độ quan trọng của đặc trưng bằng ANOVA F-score

ANOVA F-score là phương pháp thống kê đo lường sự khác biệt giữa các nhóm (nhãn) về giá trị trung bình của một đặc trưng.

F-score cao $\Rightarrow$ đặc trưng đó có khả năng phân tách tốt giữa các nhãn.

Giúp lựa chọn các đặc trưng có khả năng phân biệt cao, phục vụ hiệu quả cho các mô hình phân loại.

6 Lựa chọn mô hình và huấn luyện

Với bài toán dự đoán khả năng đột quỵ (stroke prediction), việc lựa chọn mô hình không chỉ dựa vào hiệu suất, mà còn phụ thuộc vào:

- **Bản chất của dữ liệu:** có mất cân bằng nhãn, chứa nhiều biến phân loại và nhiễu.
- **Đặc điểm bài toán:** phân loại nhị phân, cần khả năng diễn giải kết quả trong y tế.
- **Yêu cầu ứng dụng thực tế:** cần mô hình vừa hiệu quả vừa có thể giải thích được khi triển khai.

Dựa vào các tiêu chí trên bọn em lựa chọn 3 mô hình:Logistic Regression, Random Forest và XGBoost

6 Lựa chọn mô hình và huấn luyện

Vì sao chọn 3 mô hình Logistic Regression, Random Forest và XGBoost cho bài toán này ?

- **Logistic Regression:**
  - Mô hình tuyến tính đơn giản, dễ giải thích.
  - Thích hợp với dữ liệu nhị phân và cung cấp xác suất đầu ra.
- **Random Forest:**
  - Mô hình ensemble dựa trên nhiều cây quyết định.
  - Khả năng xử lý dữ liệu không tuyến tính, kháng nhiễu tốt.
  - Tự động xử lý tính năng không quan trọng và giảm overfitting.
- **XGBoost:**
  - Mô hình boosting mạnh mẽ, tối ưu tốc độ và hiệu suất.
  - Hoạt động tốt trên tập dữ liệu vừa và lớn.
  - Có khả năng xử lý mất cân bằng nhãn và tùy chỉnh nhiều tham số.

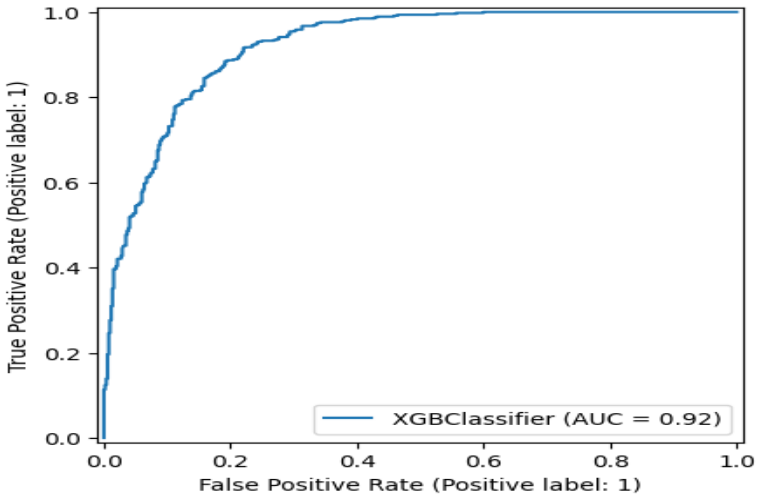
XGBoost Model

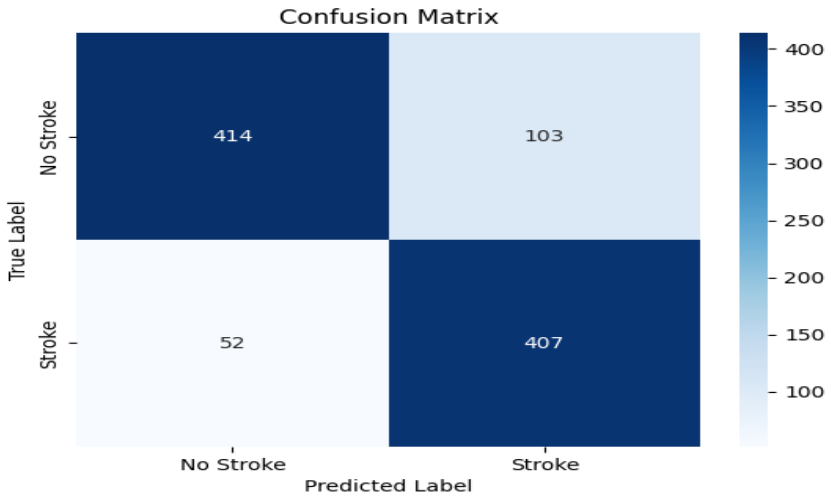
ROC AUC Score: 84.37%

	precision	recall	f1-score	support
0	0.89	0.80	0.84	517
1	0.80	0.89	0.84	459
accuracy			0.84	976
macro avg	0.84	0.84	0.84	976
weighted avg	0.85	0.84	0.84	976

6 Lựa chọn mô hình và huấn luyện

XGBoost Model



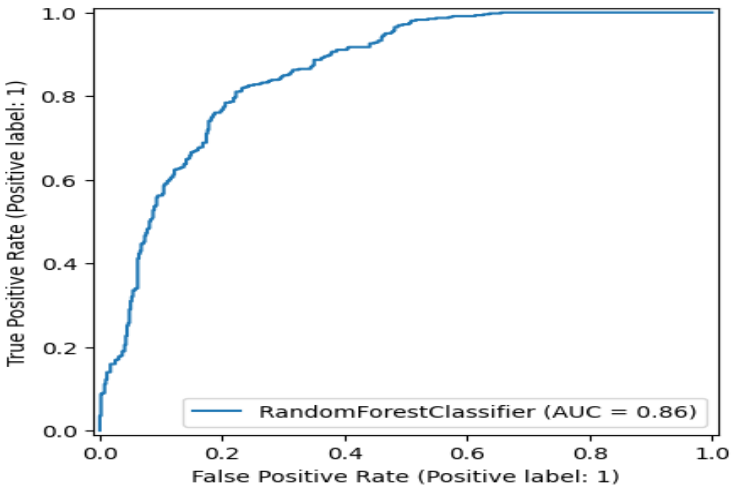


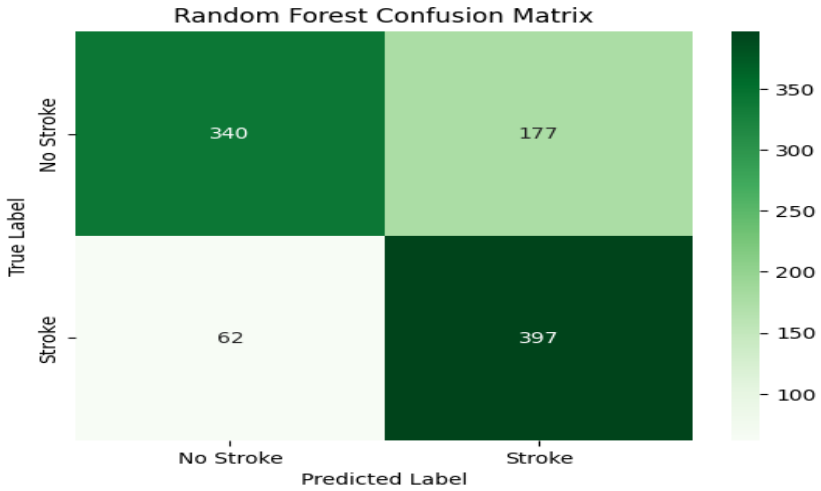
Random Forest

Random Forest:
ROC AUC Score: 76.13%

	precision	recall	f1-score	support
0	0.85	0.66	0.74	517
1	0.69	0.86	0.77	459
accuracy			0.76	976
macro avg	0.77	0.76	0.75	976
weighted avg	0.77	0.76	0.75	976

Random Forest



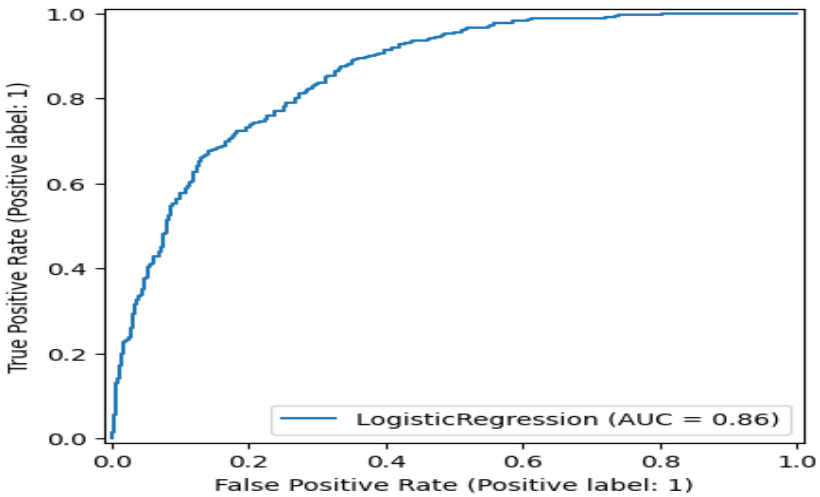


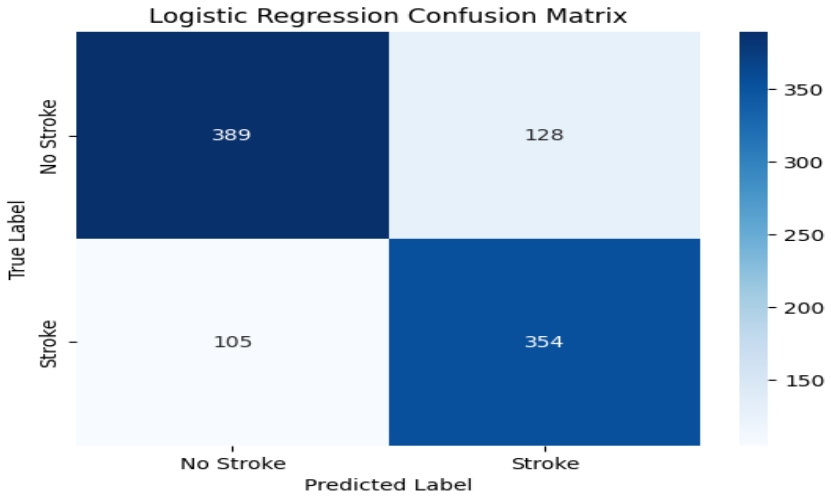
Logistic Regression

Logistic Regression:
ROC AUC Score: 76.18%

	precision	recall	f1-score	support
0	0.79	0.75	0.77	517
1	0.73	0.77	0.75	459
accuracy			0.76	976
macro avg	0.76	0.76	0.76	976
weighted avg	0.76	0.76	0.76	976

6 Lựa chọn mô hình và huấn luyện





7 Áp dụng kỹ thuật Stacking Model

Để tận dụng tối đa sức mạnh của cả ba mô hình và đạt được hiệu quả vượt trội, áp dụng Stacking Ensemble – một kỹ thuật kết hợp các mô hình cơ sở để tạo ra một mô hình dự đoán mạnh mẽ hơn.

Model Performance Summary:

	Model	ROC-AUC	F1-Score	Recall	Precision	Accuracy
0	XGBoost	0.920819	0.840041	0.886710	0.798039	0.841189
1	Random Forest	0.859020	0.768635	0.864924	0.691638	0.755123
2	Logistic Regression	0.857119	0.752391	0.771242	0.734440	0.761270
3	Stacking Ensemble	0.921784	0.838298	0.858388	0.819127	0.844262

8 Lưu và gọi lại mô hình thực hiện dự đoán

Dự đoán với nhãn 1 (labels = 1 1 1 1 1)

Mẫu	Label	Dự đoán	Xác suất (label=1)
1	1	1	0.9969
2	1	1	0.9745
3	1	1	0.9838
4	1	1	0.9692
5	1	1	0.9969

8 Lưu và gọi lại mô hình thực hiện dự đoán

Dự đoán với nhãn 0 (labels = 0 0 0 0 0)

Mẫu	Label	Dự đoán	Xác suất (label=1)
1	0	1	0.5780
2	0	0	0.0019
3	0	0	0.0016
4	0	0	0.0021
5	0	0	0.0027

8 Lưu và gọi lại mô hình thực hiện dự đoán

- Thực hiện dự đoán với dữ liệu giả lập(20 mẫu):
- Mô hình dự đoán đúng: 14/20 mẫu (70%)
- Mô hình dự đoán sai: 6/20 mẫu (30%)

Label:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Dự đoán nhãn (0/1):

1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	0	0	0	0

Xác suất dự đoán (label=1):

0.99	0.43	0.02	0.02	0.97	0.21	0.01	0.97	0.88	0.47
0.01	0.36	0.90	0.00	0.88	0.99	0.13	0.39	0.002	0.007

9. Tổng kết

Những kết quả đạt được từ dự án:

- **Tiền xử lý dữ liệu**: xử lý thiếu, loại bỏ ngoại lệ, chuẩn hóa và cân bằng bằng kỹ thuật kết hợp *under-sampling* và *SMOTE* – nâng cao chất lượng đầu vào cho mô hình.
- **Ba mô hình học máy** đã được triển khai: **Logistic Regression, Random Forest, XGBoost**.
  - Mô hình **XGBoost** đạt hiệu suất cao nhất với **ROC AUC 84.37%**.
  - Logistic Regression và Random Forest cũng cho kết quả khả quan (~76%), đặc biệt phù hợp cho mục tiêu diễn giải.
- Áp dụng **Stacking Model** nhằm kết hợp ưu điểm giữa các mô hình con — cho thấy tiềm năng cải thiện hiệu suất tổng thể trong thực tế.
- Thông qua dự án, nhóm đã phát triển các kỹ năng:
  - Làm việc với dữ liệu y tế và tiền xử lý nâng cao.
  - Triển khai và đánh giá nhiều mô hình học máy.
  - Tư duy phân tích và cải tiến hiệu quả mô hình.
  - Trình bày kết quả một cách trực quan và khoa học.

9 Tổng kết

Trong đề tài này, nhóm đã thực hiện toàn bộ quy trình xây dựng mô hình học máy nhằm dự đoán nguy cơ đột quỵ của bệnh nhân, từ bước khám phá và xử lý dữ liệu đến việc lựa chọn, huấn luyện và đánh giá mô hình.

Hướng phát triển:

- Sử dụng **tập dữ liệu thực tế và cập nhật hơn**, mở rộng thêm các đặc trưng (*features*) liên quan đến sinh học phân tử hoặc hình ảnh.
- Ứng dụng **mô hình học sâu (Deep Learning)** như CNN, RNN để xử lý dữ liệu ảnh y khoa hoặc chuỗi thời gian.
- Nghiên cứu khả năng giải thích mô hình (*model interpretability*) nhằm tăng tính minh bạch trong dự đoán.

Xin chân thành cảm ơn quý thầy/cô, anh/chị và các bạn đã lắng nghe!

Rất mong nhận được những góp ý quý báu để hoàn thiện hơn.