

MINDX - DA39

STROKE PREDICTION

NHÓM 3T:

1. Nguyễn Anh Triệu
2. Vương Thanh Tùng
3. Từ Thị Bích Tuyền

NGUỒN: [WWW.KAGGLE.COM/DATASETS/FEDESORIANO/STROKE-PREDICTION-DATASET/DATA](https://www.kaggle.com/fe-desoriano/stroke-prediction-dataset/data)



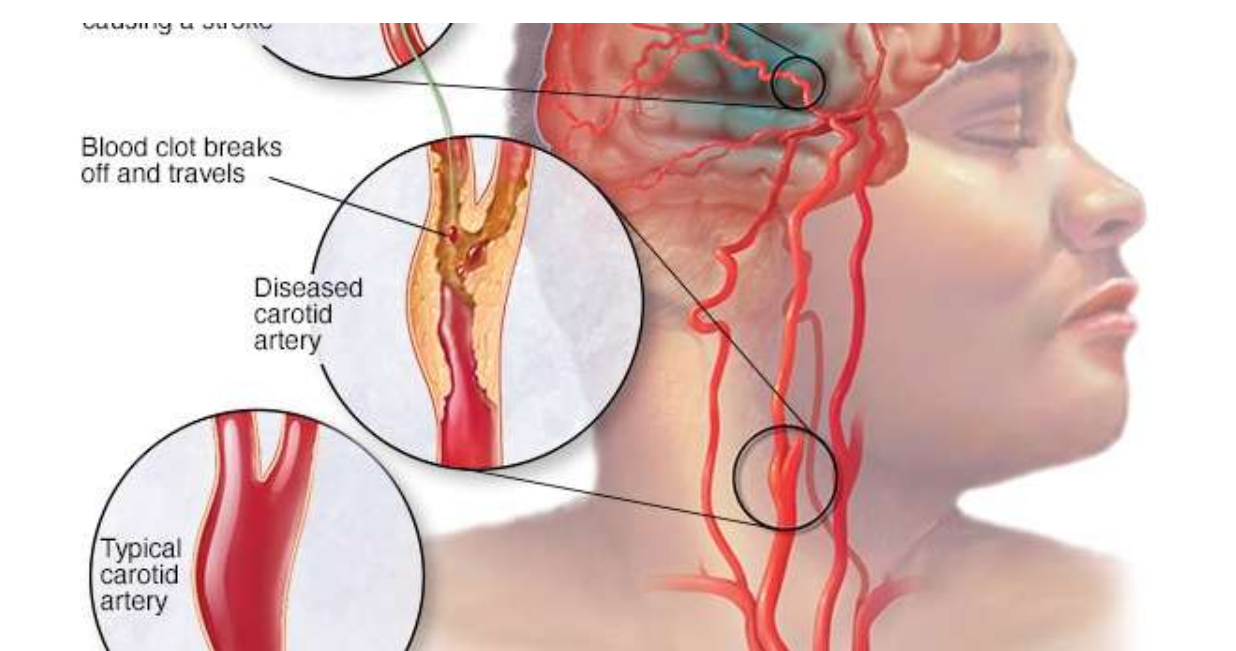
1.VẤN ĐỀ HIỆN NAY



Understanding



2. SƠ LƯỢC VỀ BỆNH ĐỘT QUÝ



3.CÁC TRIỆU CHỨNG CỦA CĂN BỆNH



4.NGUYÊN NHÂN MẮC BỆNH



1. VẤN ĐỀ HIỆN NAY



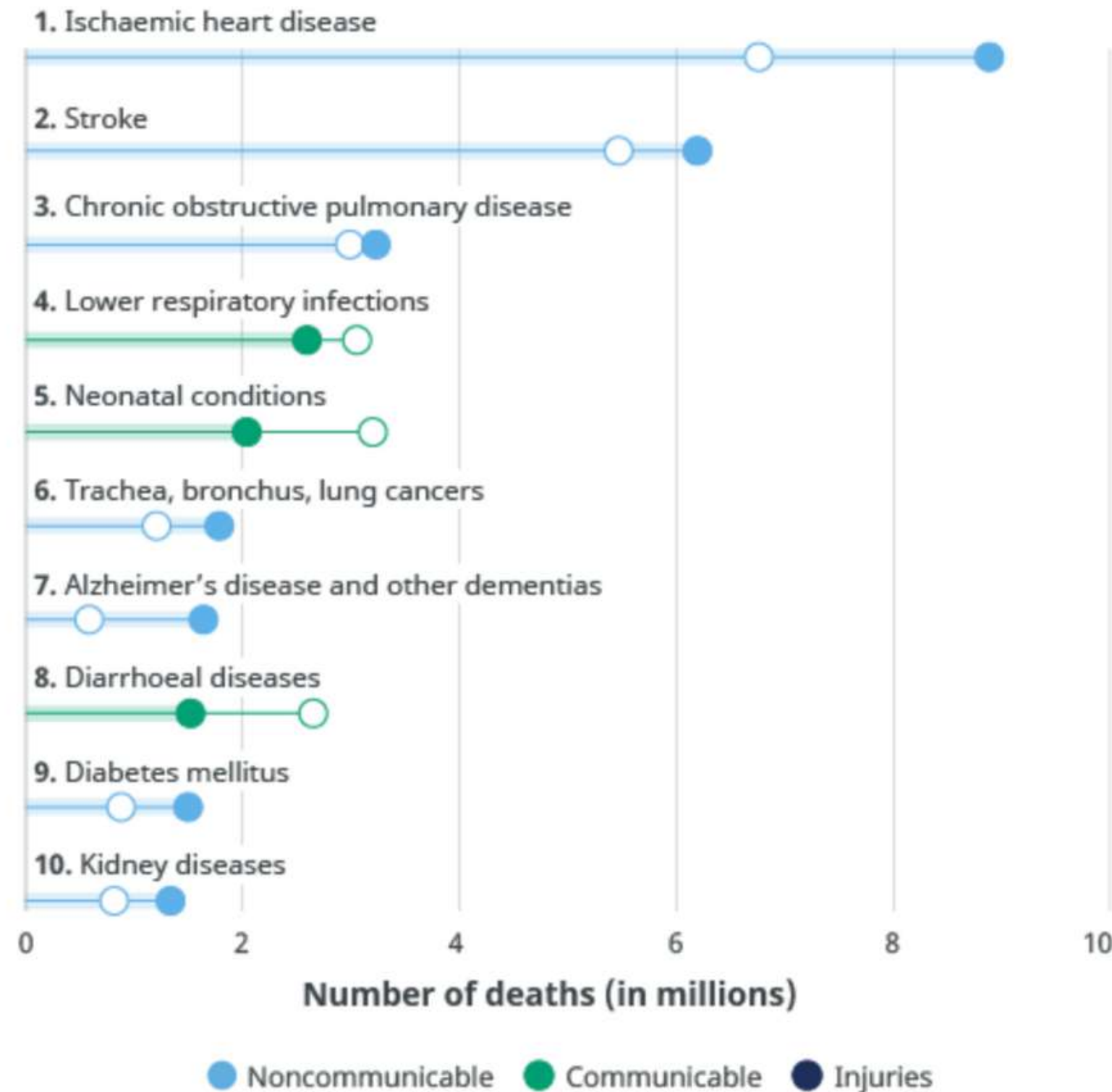
Theo Tổ chức Y tế Thế giới (WHO), đột quỵ là nguyên nhân gây tử vong hàng đầu **thứ hai** trên toàn cầu, chiếm khoảng 11% tổng số ca tử vong.

Việc dự đoán đúng khả năng đột quỵ có thể giúp phát hiện sớm bệnh và thực hiện các biện pháp phòng ngừa, điều trị kịp thời. Điều này có thể giảm nguy cơ và tác động của đột quỵ, cải thiện cơ hội phục hồi và tăng khả năng sống sót của người bệnh. Do đó, việc đưa ra dự đoán chính xác về khả năng đột quỵ có thể có tầm quan trọng quyết định đối với sức khỏe và cuộc sống của một cá nhân.

<https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>

Leading causes of death globally

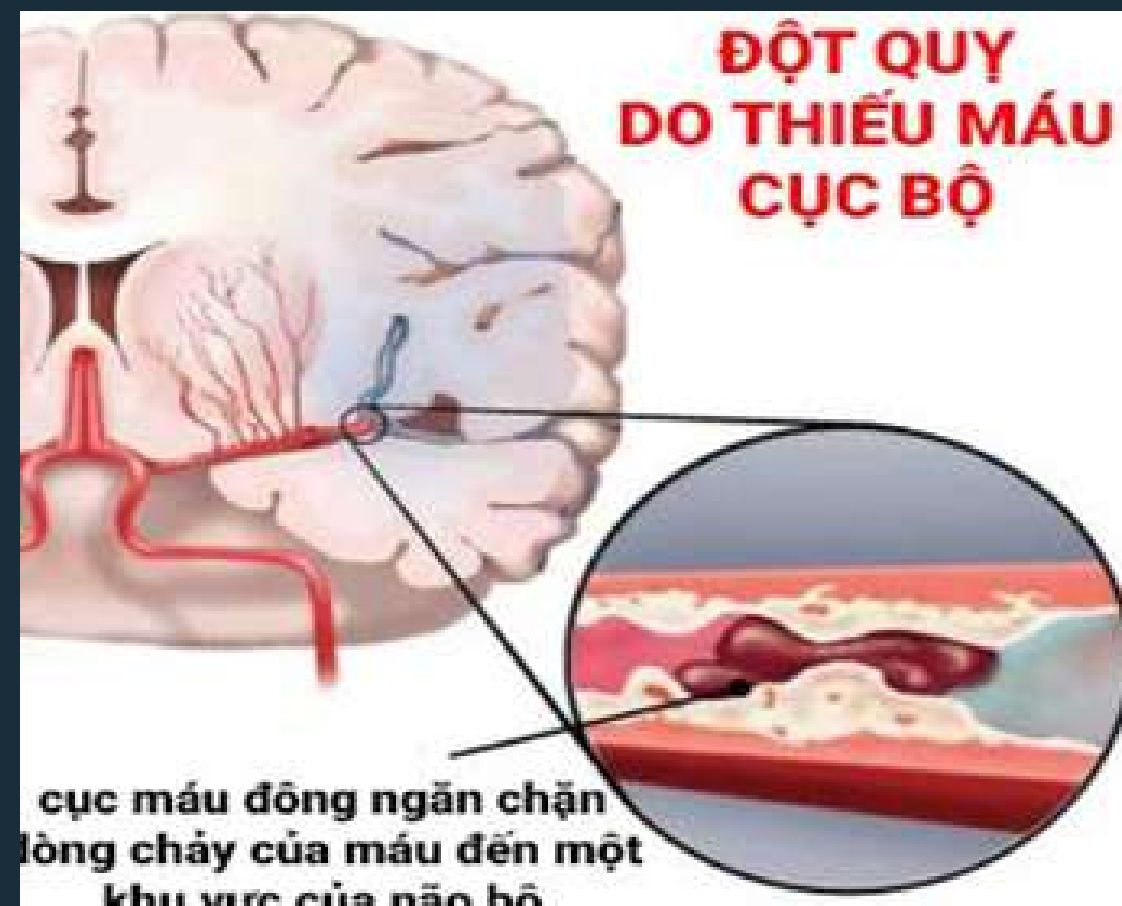
○ 2000 ● 2019



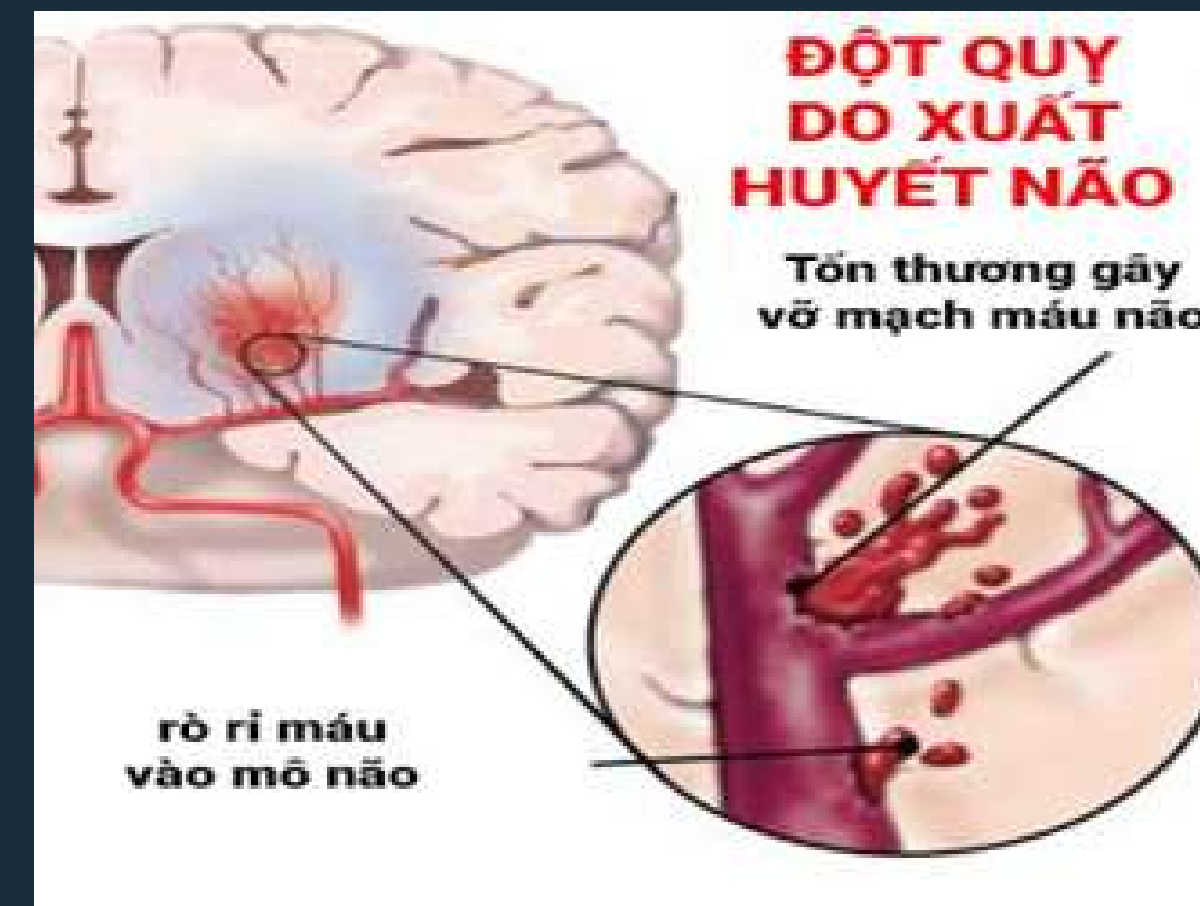
Source: WHO Global Health Estimates.

2. SƠ LƯỢC VỀ BỆNH ĐỘT QUY

Đột quy (stroke) còn gọi là tai biến mạch máu não thường xảy ra đột ngột khi nguồn máu cung cấp cho não bị tắc nghẽn, gián đoạn hoặc suy giảm. Khi đó, não bị thiếu oxy, dinh dưỡng và các tế bào não bắt đầu chết trong vòng vài phút. Người bị đột quy có nguy cơ tử vong cao nếu không được phát hiện và cấp cứu kịp thời. Đây là một trong những bệnh lý thần kinh nguy hiểm và phổ biến nhất.



Thiếu máu cục bộ dẫn đến đột quy xảy ra do tình trạng tắc nghẽn trong động mạch. Theo thống kê, hiện nay có đến khoảng 85% các ca bệnh đột quy thuộc nhóm này. Tuy nhiên, có nhiều trường hợp vẫn chưa tìm ra nguyên nhân chính xác dẫn đến tình trạng này.



Tình trạng xuất huyết (chảy máu) não, là do vỡ mạch máu não làm chảy máu vào nhu mô não, khoang dưới nhện hay não thất. Có khoảng 15% các trường hợp bệnh đột quy hiện nay là do xuất huyết não.

3. CÁC TRIỆU CHỨNG CỦA CĂN BỆNH

1. Dấu hiệu về thị lực: Thị lực đột ngột giảm, mờ dần cả 2 mắt hoặc một mắt, tuy nhiên biểu hiện này không rõ rệt nên người bên cạnh khó có thể nhận ra.



2. Dấu hiệu ở mắt: Mắt đột nhiên có biểu hiện không cân xứng, miệng méo, nhân trung hơi lệch so với bình thường, nếp mũi và má bên yếu bị rủ xuống.



3. Dấu hiệu ở tay: Tay đột nhiên tê mỏi, khó cử động. Ngoài ra người bệnh sẽ khó khăn trong đi lại, không thể nhấc chân lên được.



6. Dấu hiệu ở thần kinh: Người bị đột quy có thể sẽ cảm thấy đau đầu dữ dội. Đây là một triệu chứng rất nặng và khá phổ biến của bệnh đột quy. Nhất là người có tiền sử đau nửa đầu.

5. Dấu hiệu qua nhận thức: Người bị đột quy đột nhiên cảm thấy rối loạn trí nhớ, không nhận thức được, mắt mờ, tai ù không nghe rõ người xung quanh.



4. Dấu hiệu qua giọng nói: Người bị đột quy có thể gặp triệu chứng nói ngọng bất thường, môi lưỡi tê cứng, miệng mở khó, phải cố gắng lắm mới có thể nói được.



4. NGUYÊN NHÂN MẮC BỆNH



- Người bị các bệnh lý tim mạch như hở van tim, rung tâm nhĩ, nhịp tim không đều, suy tim,...;
- Người bị tăng huyết áp;
- Người bị tiểu đường;
- Người bị rối loạn Lipid máu;
- Tiền sử cá nhân hoặc gia đình đã từng bị đột quỵ, thiếu máu não thoáng qua hoặc bệnh tim;
- Lạm dụng các chất kích thích như uống nhiều rượu, sử dụng ma túy;
- Người hút thuốc lá chủ động hoặc hít phải khói thuốc lá thụ động, gặp tình trạng khói thuốc lá dẫn đến mỡ tích tụ trong động mạch, tăng nguy cơ máu đông;
- Người thừa cân, béo phì, ít vận động tập thể dục;
- Chế độ ăn uống không hợp lý, lượng Cholesterol cao.
- Về tuổi tác, người trong nhóm tuổi từ 55 tuổi trở lên có nguy cơ cao hơn;
- Phụ nữ có nguy cơ bị đột quỵ thấp hơn so với nam giới;
- Việc sử dụng thuốc tránh thai hay các liệu pháp điều chỉnh hormone, thay đổi nội tiết tố cũng làm tăng nguy cơ gây bệnh.

Report

I. GIỚI THIỆU DATASET

II. CLEANING DATA

III. EXPLORATION DATA ANALYSIS

IV. MODEL PREDICTION

V. SUGGESTION

I. GIỚI THIỆU DATASET

- Bộ dữ liệu này được sử dụng để dự đoán xem một bệnh nhân có khả năng bị đột quỵ hay không dựa trên các thông số đầu vào như giới tính, tuổi, các bệnh lý khác nhau và tình trạng hút thuốc. Mỗi hàng trong dữ liệu cung cấp thông tin liên quan về bệnh nhân.
- Gồm 12 cột và 5110 dòng, có 201 giá trị NULL ở cột BMI

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   id                    5110 non-null  int64  
 1   gender                5110 non-null  object  
 2   age                   5110 non-null  float64 
 3   hypertension          5110 non-null  int64  
 4   heart_disease         5110 non-null  int64  
 5   ever_married          5110 non-null  object  
 6   work_type             5110 non-null  object  
 7   Residence_type        5110 non-null  object  
 8   avg_glucose_level     5110 non-null  float64 
 9   bmi                   4909 non-null  float64 
10   smoking_status        5110 non-null  object  
11   stroke                5110 non-null  int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

df.isnull().sum()

```
id                0
gender            0
age              0
hypertension      0
heart_disease     0
ever_married      0
work_type         0
Residence_type    0
avg_glucose_level 0
bmi              201
smoking_status    0
stroke            0
dtype: int64
```


I. GIỚI THIỆU DATASET

Kiểm tra thống kê dữ liệu

df.describe()							
	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

df.describe(include=object)							
	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		


I. GIỚI THIỆU DATASET

1. Cột age có min = 0.08 cần kiểm tra outlier
=> Tạo thêm cột age_group bao gồm child(0-16), young adult(17-30), middle-aged adult(31-45), old-aged adult (>45)

2. BMI:

- * Dưới 18,5 : nhẹ cân
- * 18,5-24,9 : bình thường
- * 25-29,9 : thừa cân
- * 30 trở lên : béo phì

=> Tạo thêm cột weight bao gồm các thông số: underweight(nhẹ cân), healthy(bình thường), overweight(thừa cân), obesity(béo phì) để xác định mức độ ảnh hưởng của cân nặng tới đột quỵ

3. avg_glucose_level: chỉ số đường huyết được xét theo lúc đói, sau ăn, trước khi đi ngủ và bất kỳ trong trường hợp bộ data không đề cập tới thì chúng ta coi thông số đường huyết được đo bất kỳ

- * Dưới 70: hạ đường huyết
- * 70-140: bình thường
- * 140-200: tiền tiểu đường
- * 200 trở lên: Tiểu đường

=> Tạo thêm cột diabete bao gồm các thông số: hypoglycemia(hạ đường huyết), healthy(bình thường), prediabete(tiền tiểu đường), diabete(Tiểu đường) để xác định mức độ ảnh hưởng của bệnh tiểu đường tới đột quỵ

II. CLEANING DATA

Xoá cột thừa, tạo thêm cột age_group, weight và diabetes

```
# Xóa cột id:  
df = df.drop('id', axis=1)
```

```
# Tạo thêm cột age_group  
df.insert(2, 'age_group', np.where(df['age'] < 16, 'child',  
                                   np.where(df['age'] <= 30, 'young adult',  
                                             np.where(df['age'] <= 45, 'middle-aged adult', 'old-aged adult'))))
```

```
# Tạo thêm cột weight  
df.insert(10, 'weight', np.where(df['bmi'] < 18.5, 'underweight',  
                                  np.where(df['bmi'] <= 24.9, 'healthy',  
                                            np.where(df['bmi'] <= 29.9, 'overweight', 'obesity'))))
```

```
# Tạo thêm cột diabetes  
df.insert(9, 'diabetes', np.where(df['avg_glucose_level'] < 70, 'hypoglycemia',  
                                   np.where(df['avg_glucose_level'] <= 140, 'healthy',  
                                             np.where(df['avg_glucose_level'] <= 200, 'prediabetes', 'diabetes'))))
```


II. CLEANING DATA

Điền Null cho cột BMI bằng giá trị BMI trung bình của từng nhóm tuổi

```
mean_old = df[df['age_group']=='old-aged adult']['bmi'].mean()
mean_middle = df[df['age_group']=='middle-aged adult']['bmi'].mean()
mean_young = df[df['age_group']=='young adult']['bmi'].mean()
mean_child = df[df['age_group']=='child']['bmi'].mean()
```

```
def filling_bmi(age_group_bmi):
    age_group = age_group_bmi[0]
    bmi = age_group_bmi[1]
    if pd.isnull(bmi):
        if 'old-aged adult' in age_group:
            return mean_old
        if 'middle-aged adult' in age_group:
            return mean_middle
        if 'young adult' in age_group:
            return mean_young
        if 'child' in age_group:
            return mean_child
    else:
        return bmi
```

```
df['bmi']=df[['age_group','bmi']].apply(filling_bmi,axis = 1)
```



II. CLEANING DATA

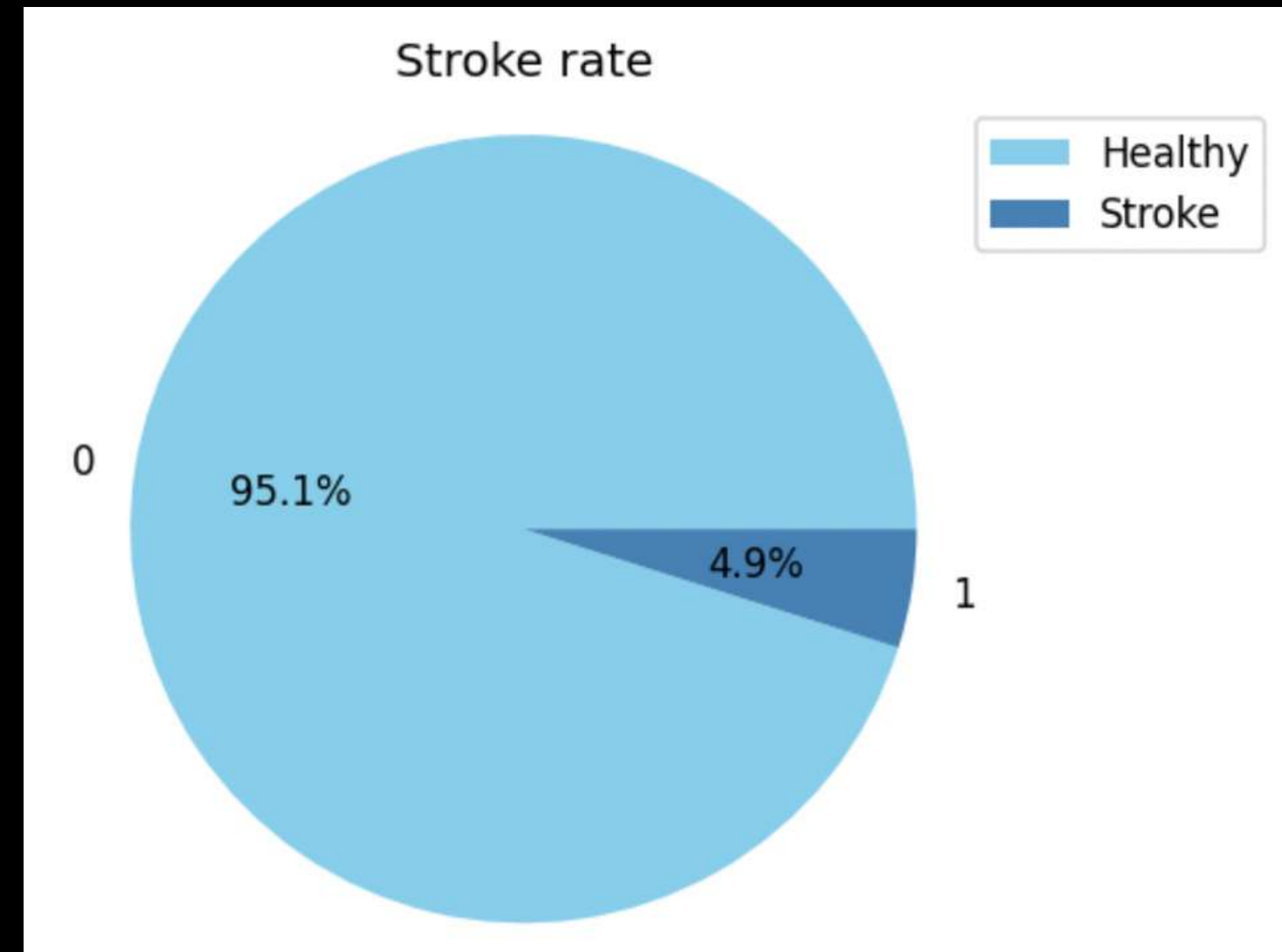
Mã hoá data chuẩn bị dữ liệu cho bước xây dựng mô hình dự báo

```
#Mã hoá data cho bước xây dựng mô hình
df.replace(to_replace = ['Other', 'Male', 'Female'], value = [0,1,2], inplace = True)
df.replace(to_replace = ['child', 'young adult', 'middle-aged adult', 'old-aged adult'], value = [0,1,2,3], inplace = True)
df.replace(to_replace = ['Yes', 'No'], value = [1,0], inplace = True)
df.replace(to_replace = ['Never_worked', 'children', 'Private', 'Self-employed', 'Govt_job'], value = [0,1,2,3,4], inplace = True)
df.replace(to_replace = ['Urban', 'Rural'], value = [1,0], inplace = True)
df.replace(to_replace = ['hypoglycemia', 'healthy', 'prediabete', 'diabete'], value = [0,1,2,3], inplace = True)
df.replace(to_replace = ['underweight', 'healthy', 'overweight', 'obesity'], value = [0,1,2,3], inplace = True)
df.replace(to_replace = ['Unknown', 'never smoked', 'formerly smoked', 'smokes'], value = [0,1,2,3], inplace = True)
df.sample(5)
```

	gender	age	age_group	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	diabete	
1670	1	45.0	2	0	1	1	2	0	93.77	1	31.37
4754	1	52.0	3	0	0	1	2	0	67.92	0	31.10
2042	1	8.0	0	0	0	0	1	1	104.51	1	20.60
3406	1	24.0	1	0	0	0	2	0	200.14	3	37.70
3892	1	37.0	2	0	0	1	4	1	173.97	2	26.30

III. EDA

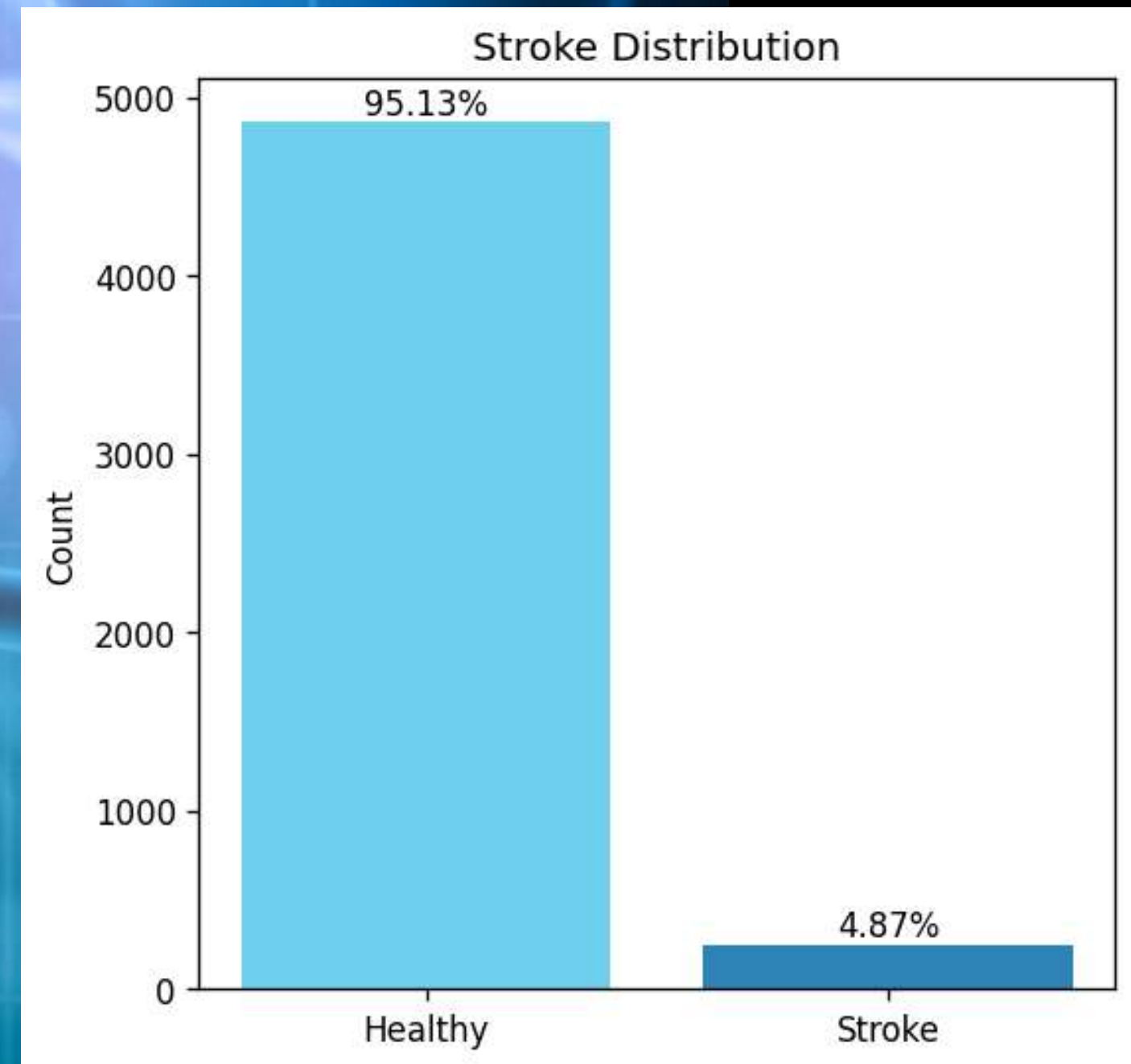
Tỷ lệ đột quỵ xấp xỉ 5%



Không chỉ gây nguy hiểm về tính mạng, đột quỵ còn để lại di chứng nặng nề, ảnh hưởng rất nhiều đến sức khỏe, sinh hoạt, tinh thần của người bệnh. Các chuyên gia y tế cũng đã cảnh báo, bệnh đột quỵ có xu hướng gia tăng và ngày càng trẻ hóa. Do vậy, mọi người cần chủ động tìm hiểu những dấu hiệu phát hiện sớm, cách phòng tránh cũng như xử trí khi đột quỵ xảy ra.

III. EDA

Dữ liệu bị mất cân bằng cần xử lý data bằng cách xử dụng Over Sampling hoặc Under Sampling để đạt được mô hình dự báo tốt nhất.



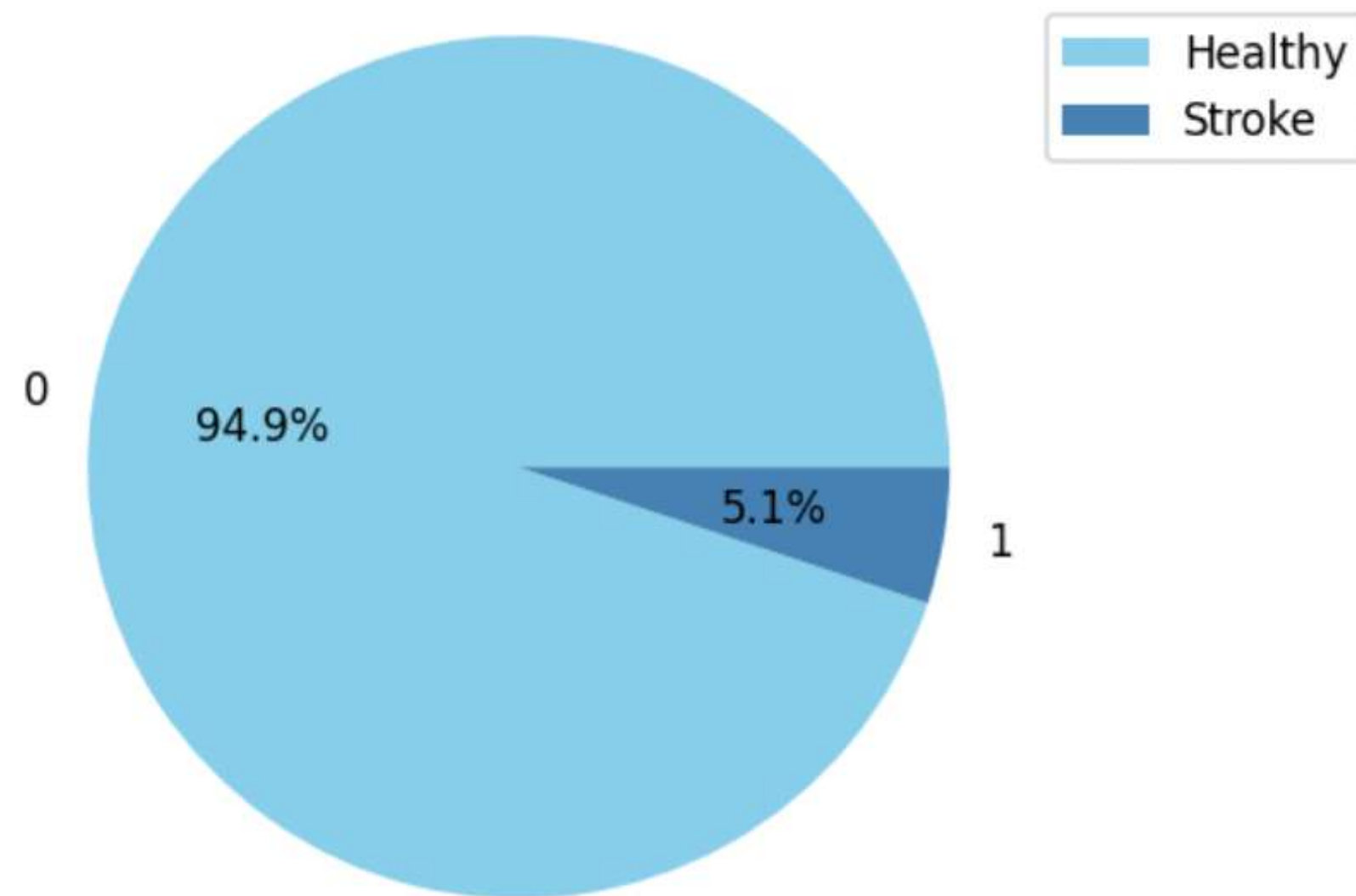
III. EDA

Có thể thấy tỷ lệ bị đột quỵ của nam giới cao hơn nữ giới

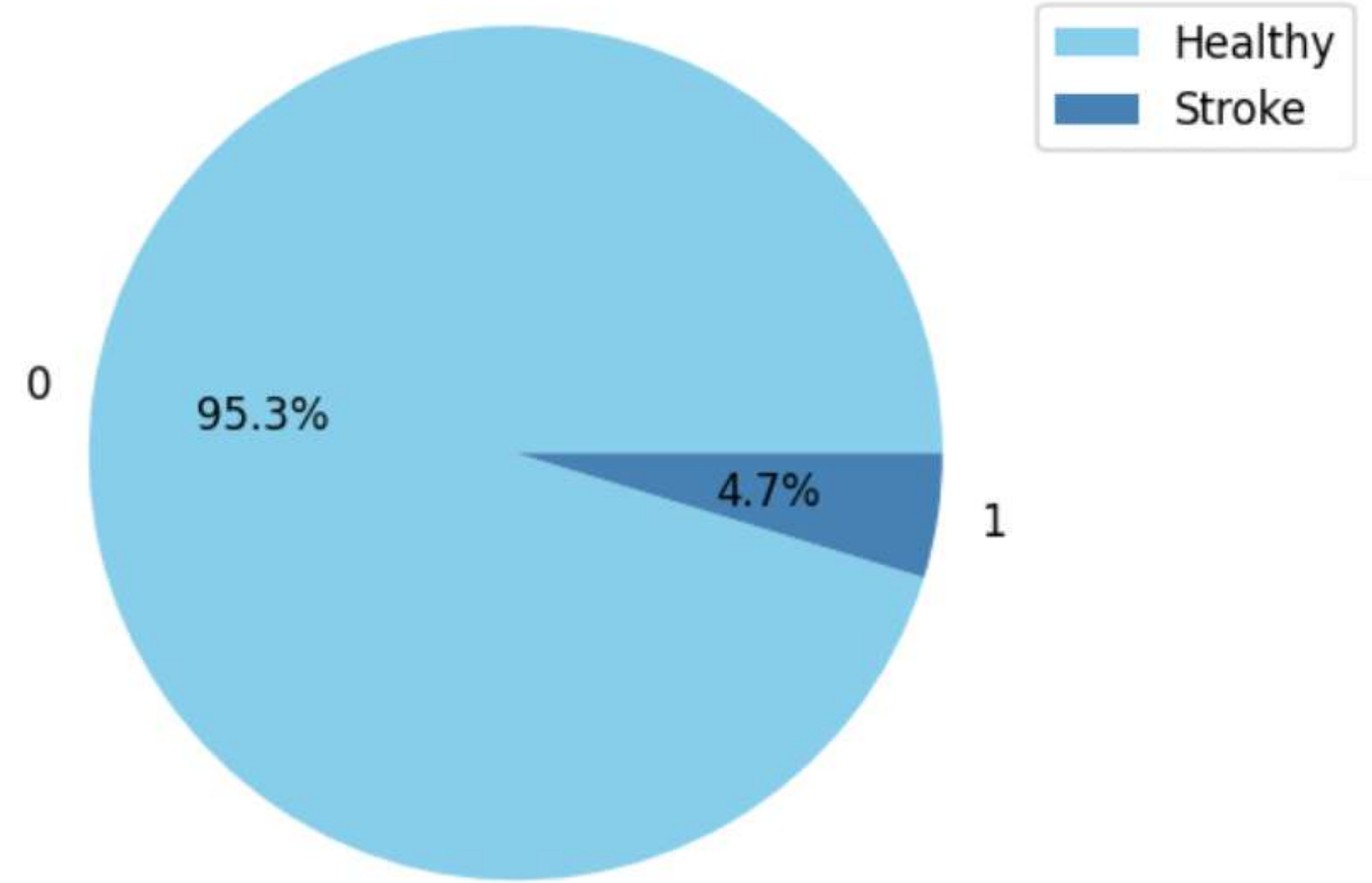
* Trung bình cứ 1000 người nam thì có 51 người bị đột quỵ

* Trung bình cứ 1000 người nữ thì có 47 người bị đột quỵ

Male Distribution



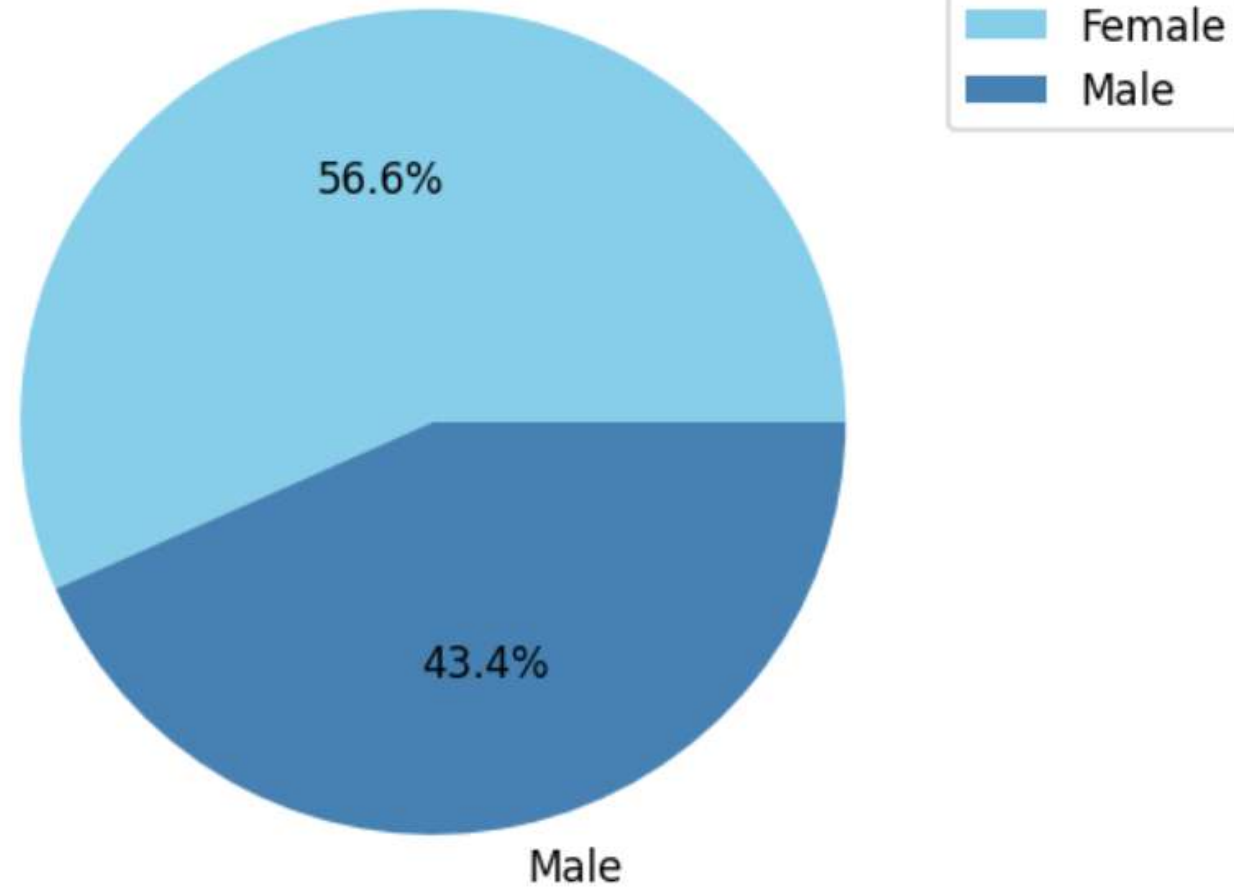
Female Distribution



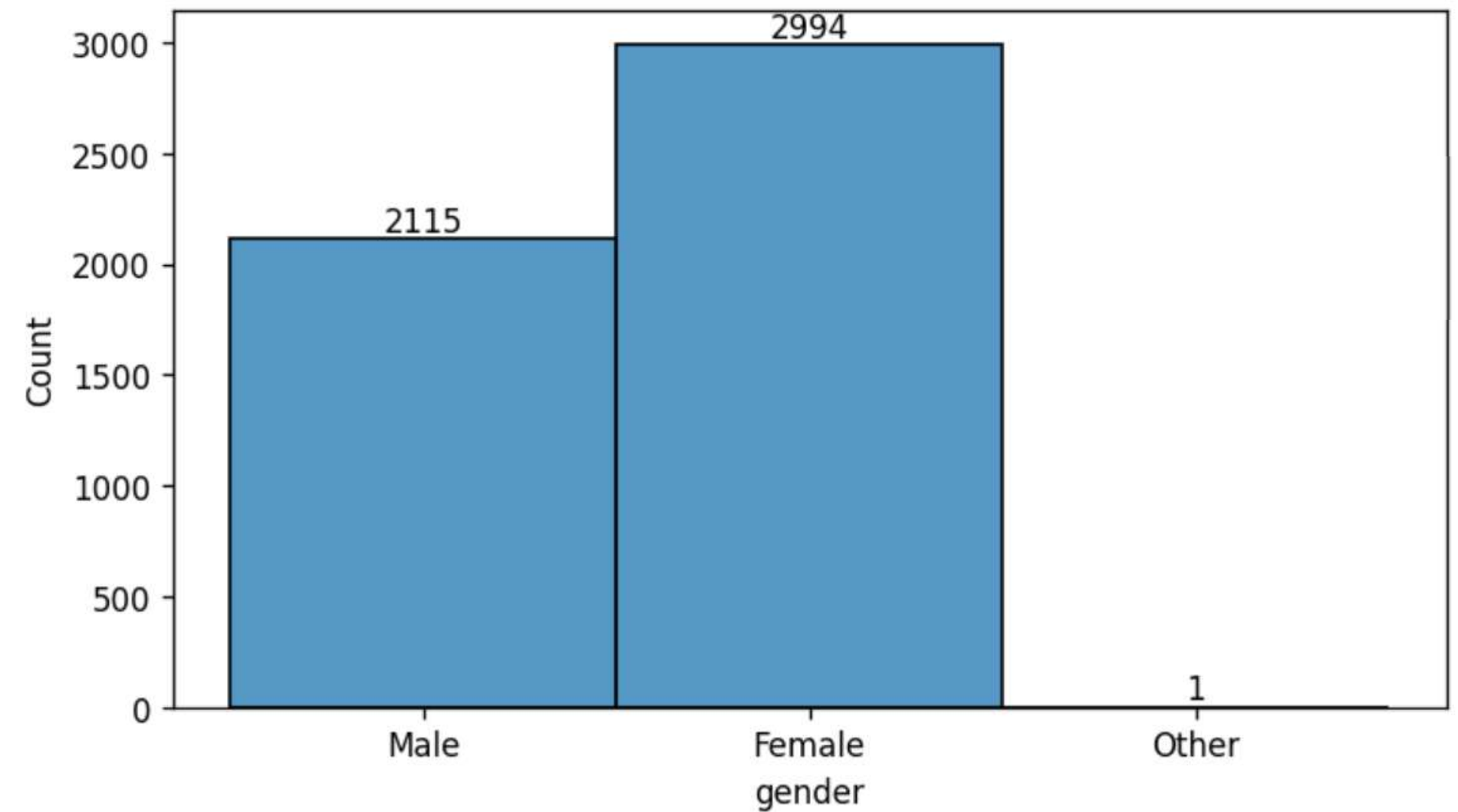
III. EDA

☰ Biểu đồ bên dưới cho thấy sự mất cân bằng giữa giới tính trong mẫu dữ liệu.

Stroke rate over gender

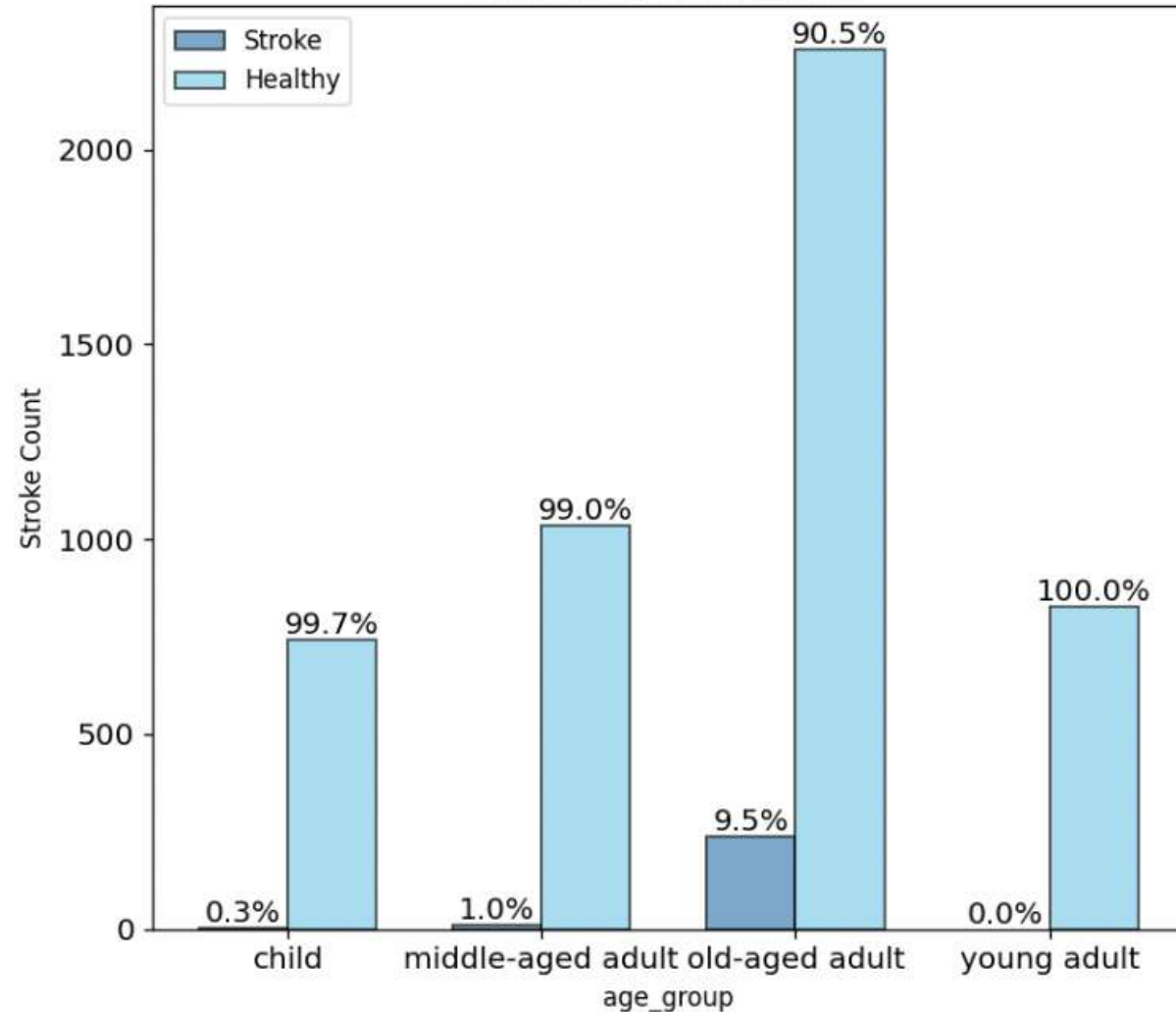


Gender Distribution

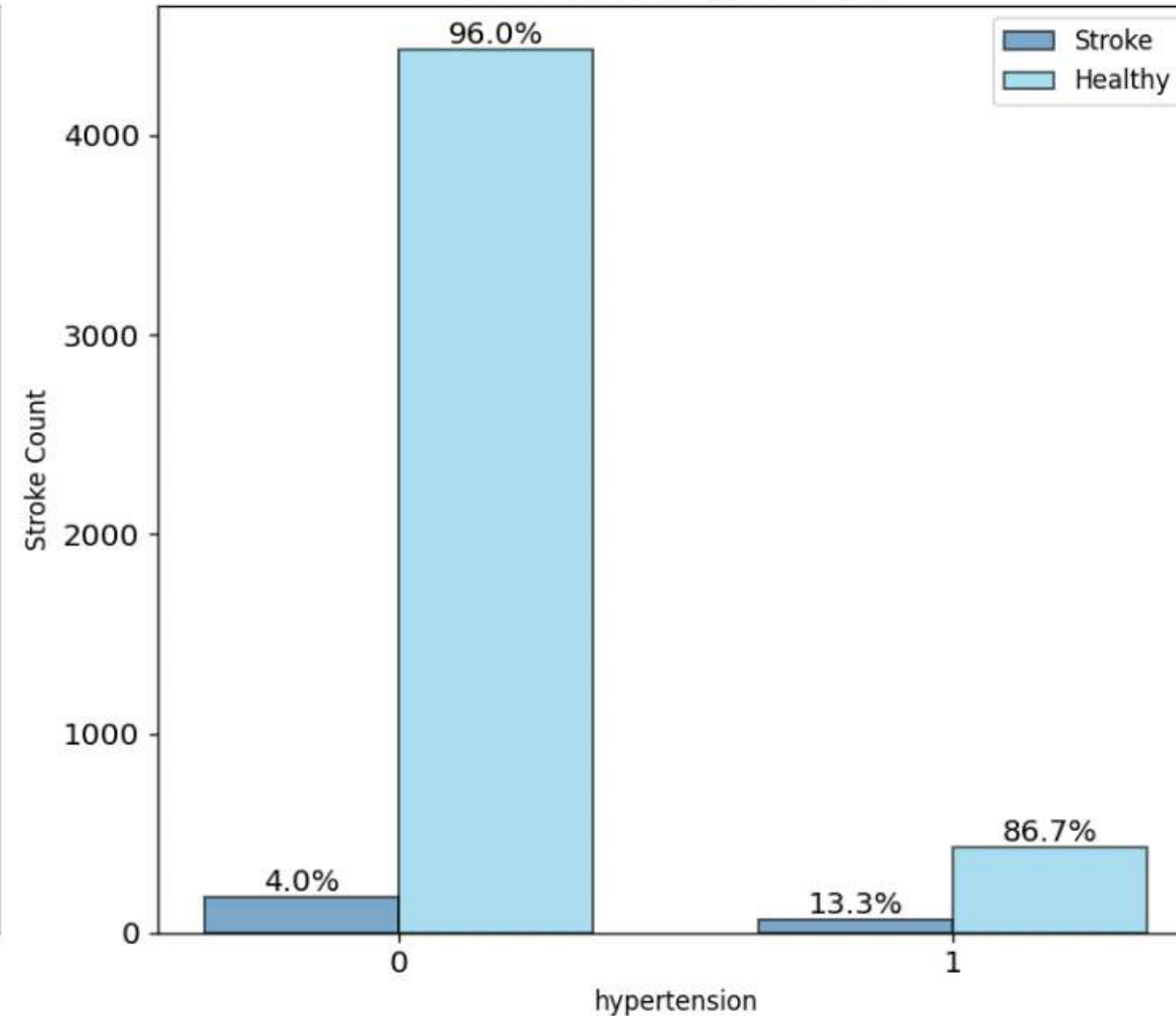


III. EDA

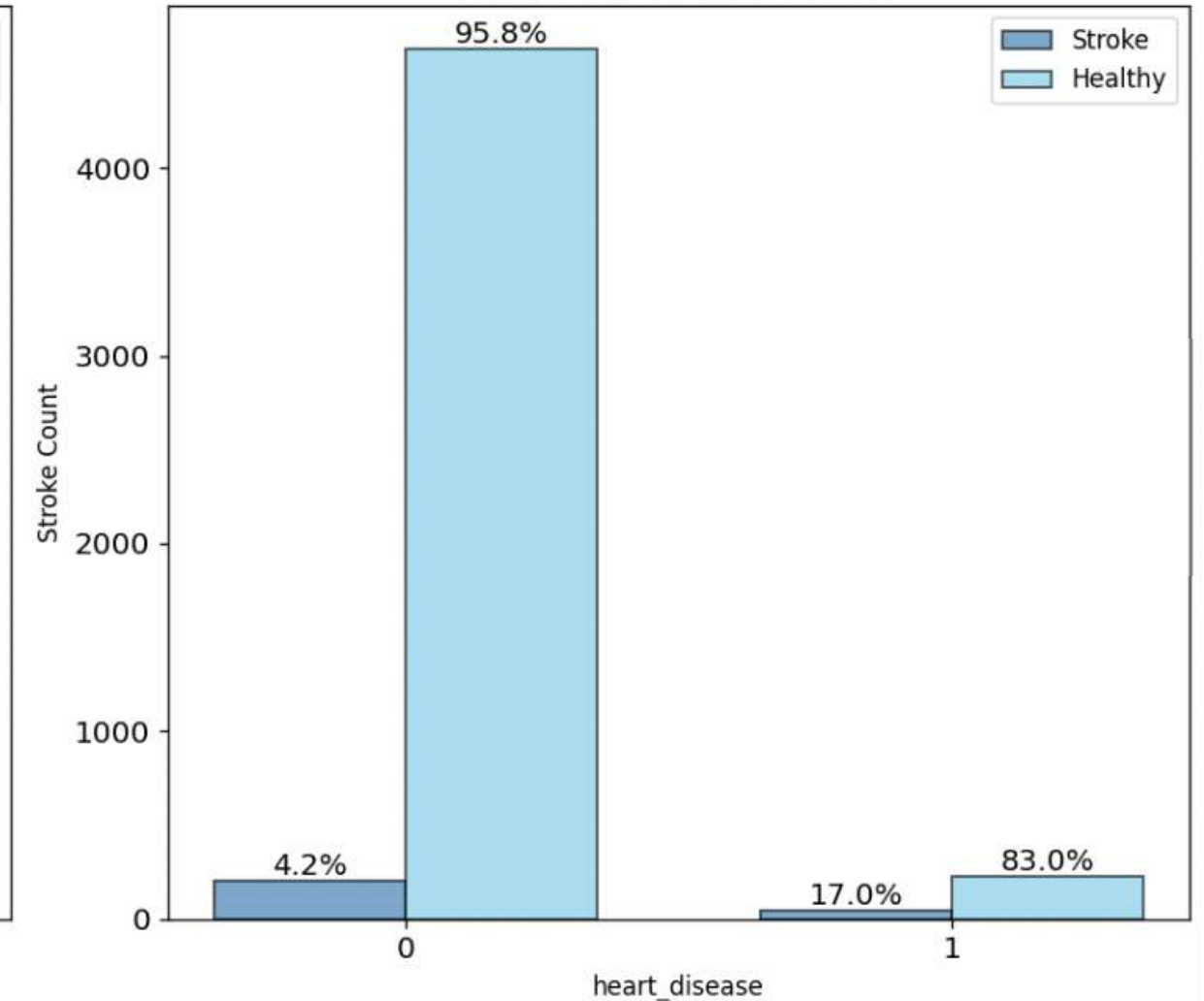
Stroke Rate by age_group



Stroke Rate by hypertension



Stroke Rate by heart_disease



Đột quỵ xảy ra chủ yếu ở nhóm người già (old_aged) với tỷ lệ 9.5%, tiếp theo đó là lứa tuổi trung niên (middle_aged) với tỷ lệ 1%. Tuy nhiên cũng lưu ý thêm về sự mất cân bằng dữ liệu, mẫu có nhiều giá trị thuộc class old_age hơn các class khác

Tỷ lệ bị đột quỵ của những người từng bị tăng huyết áp (13.3%) cao hơn gấp 4 lần so với người không bị cao huyết áp (4%). Huyết áp cao là 1 trong những nguyên nhân gây đột quỵ. Cũng như biểu đồ nhóm tuổi, chúng ta cũng cần cân nhắc về sự mất cân bằng dữ liệu giữa nhóm tăng huyết áp và không.

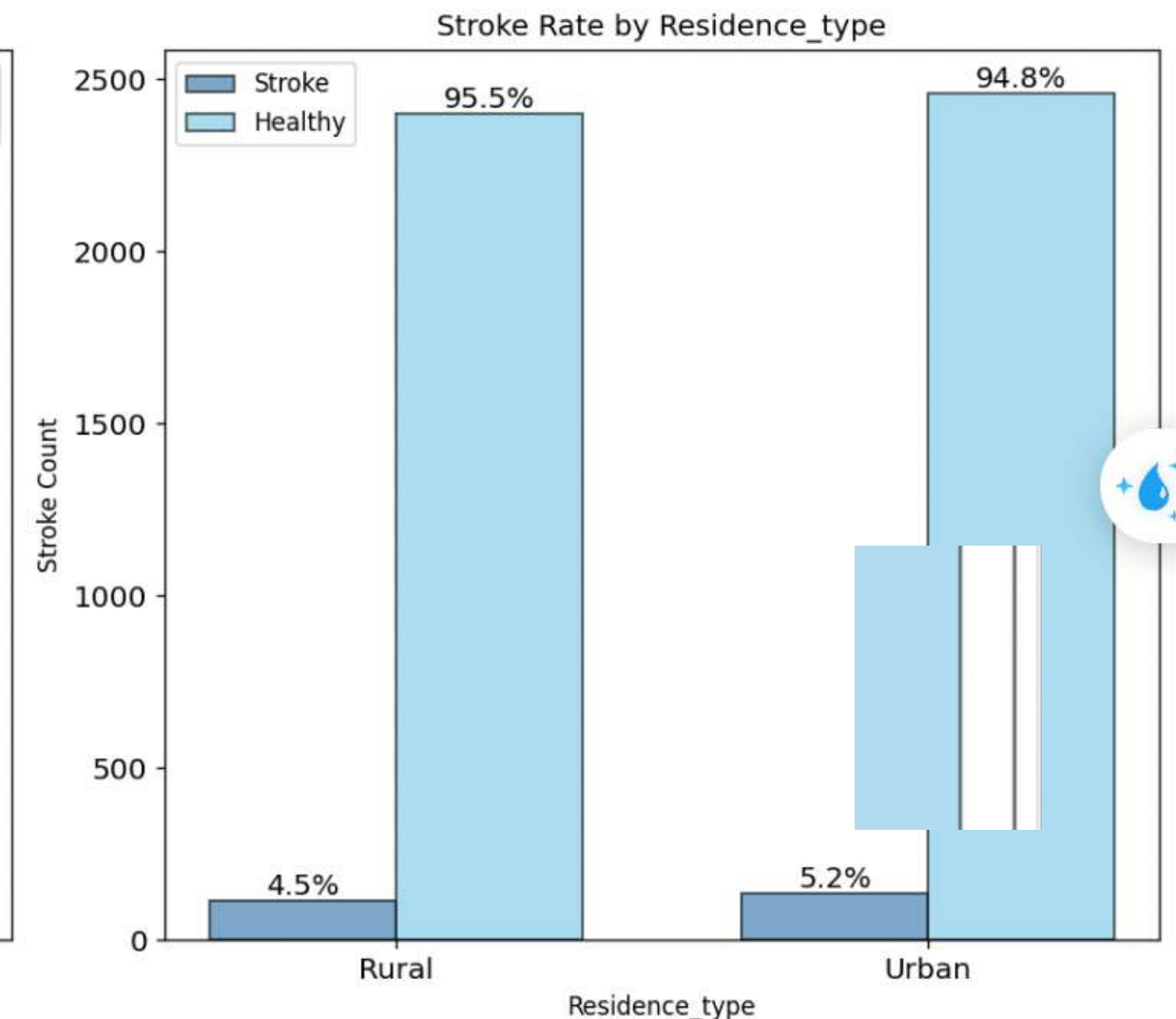
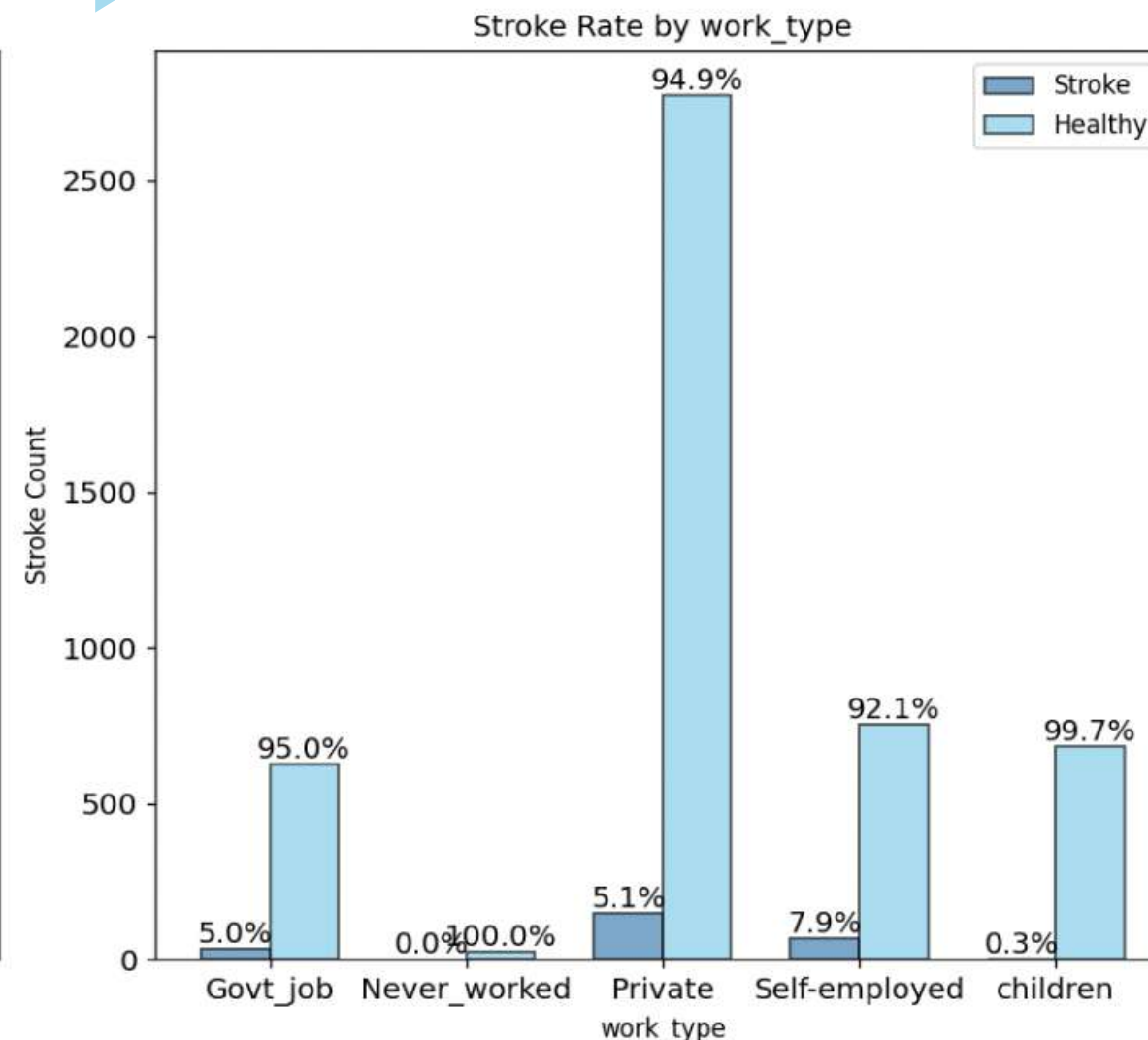
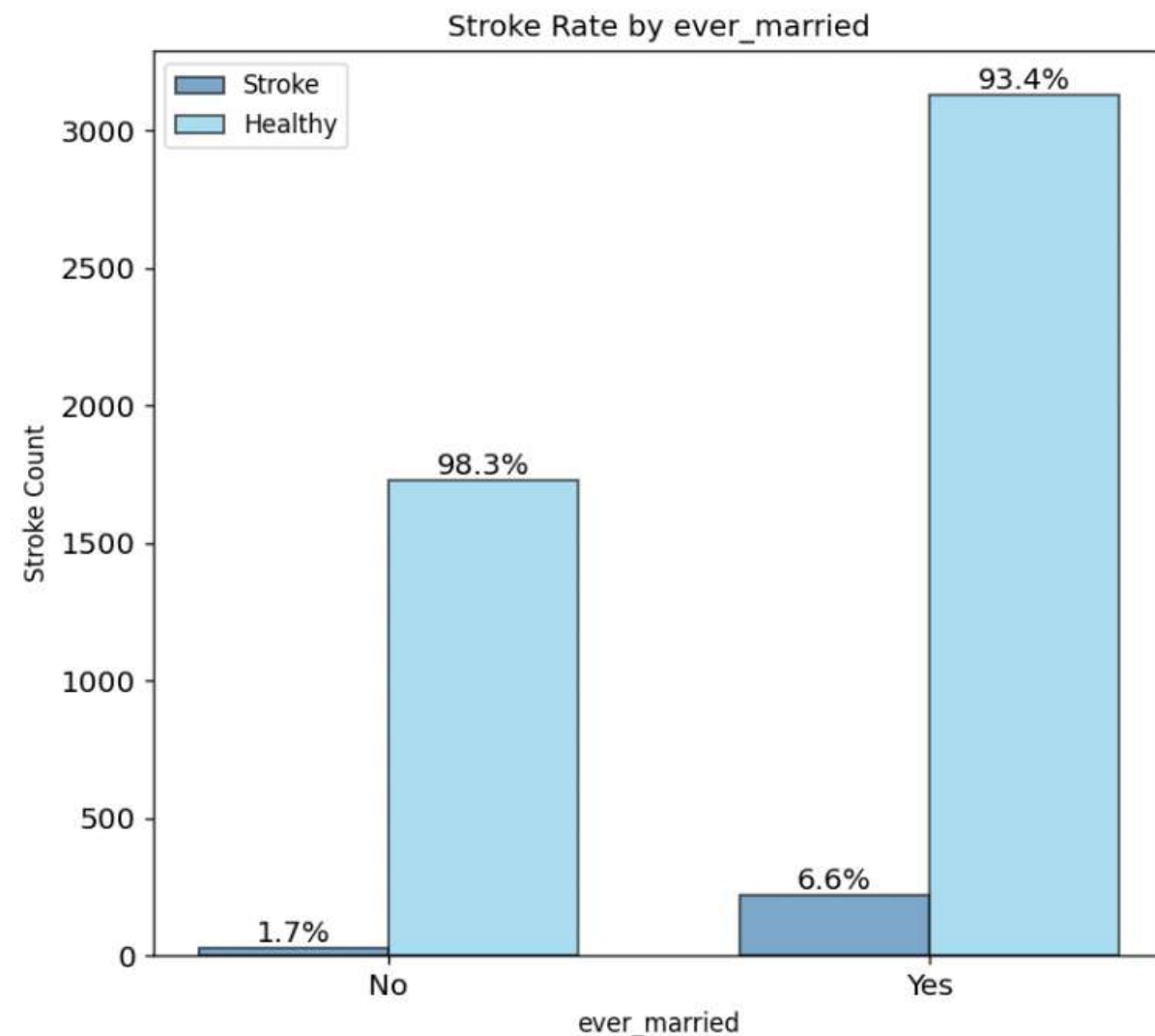
Người từng mắc bệnh về tim mạch (17%) có tỷ lệ đột quỵ cao gấp hơn 4 lần so với người khỏe mạnh (4.18%). Có sự mất cân bằng trong dữ liệu, mẫu lệch về class 0 nhiều hơn class 1.

III. EDA

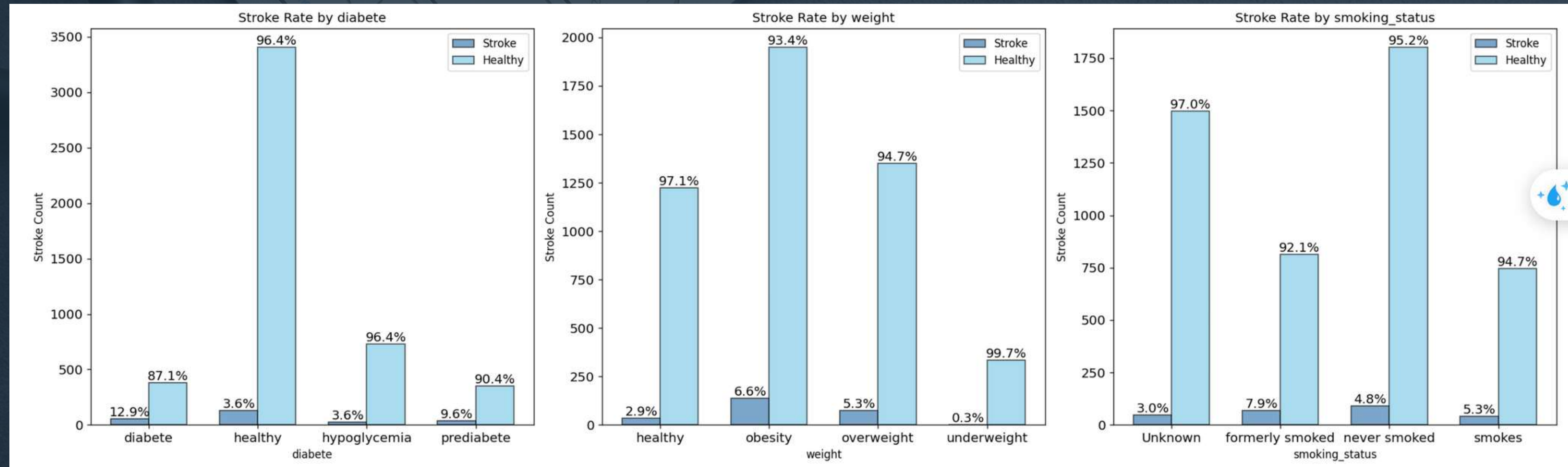
Người đã kết hôn (có tuổi) có khả năng đột quỵ cao hơn người chưa kết hôn (người trẻ độc thân). Mẫu lệch về class 'Yes' nhiều hơn class 'No'

Các nhóm người có nguy cơ mắc cao hơn các nhóm còn lại là self_employed (8%), private (5%), govt_job (5%) đều là những nhóm người làm việc dưới áp lực cao (áp lực cũng là 1 trong những nguyên nhân gây đột quỵ). Lưu ý mẫu lệch nhiều về class 'Private'

Nhóm người thành thị (Urban) có tỷ lệ mắc cao hơn đôi chút nhưng không đáng kể so với nhóm người nông thôn. Dữ liệu phân bố đồng đều ở 2 class 'Rural' và 'Urban'



III. EDA



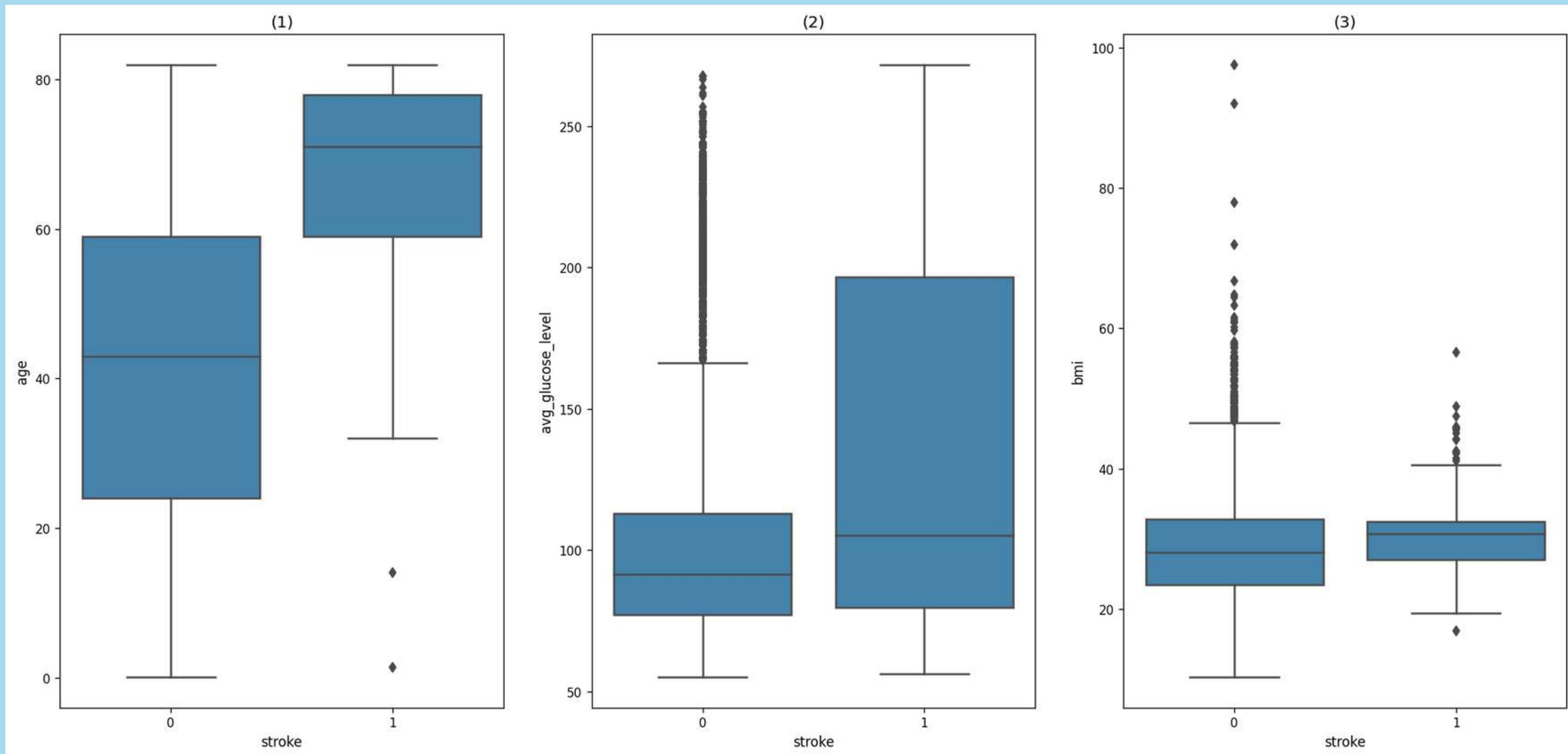
Những người bị tiểu đường hoặc tiền tiểu đường có nguy cơ đột quỵ cao hơn so với nhóm người khỏe mạnh và đường huyết thấp. Chỉ số đường huyết tỷ lệ thuận với nguy cơ đột quỵ. Mẫu lệch về class 'Healthy'.

Người bị thừa cân và béo phì có nguy cơ đột quỵ cao hơn người khỏe mạnh và người gầy.

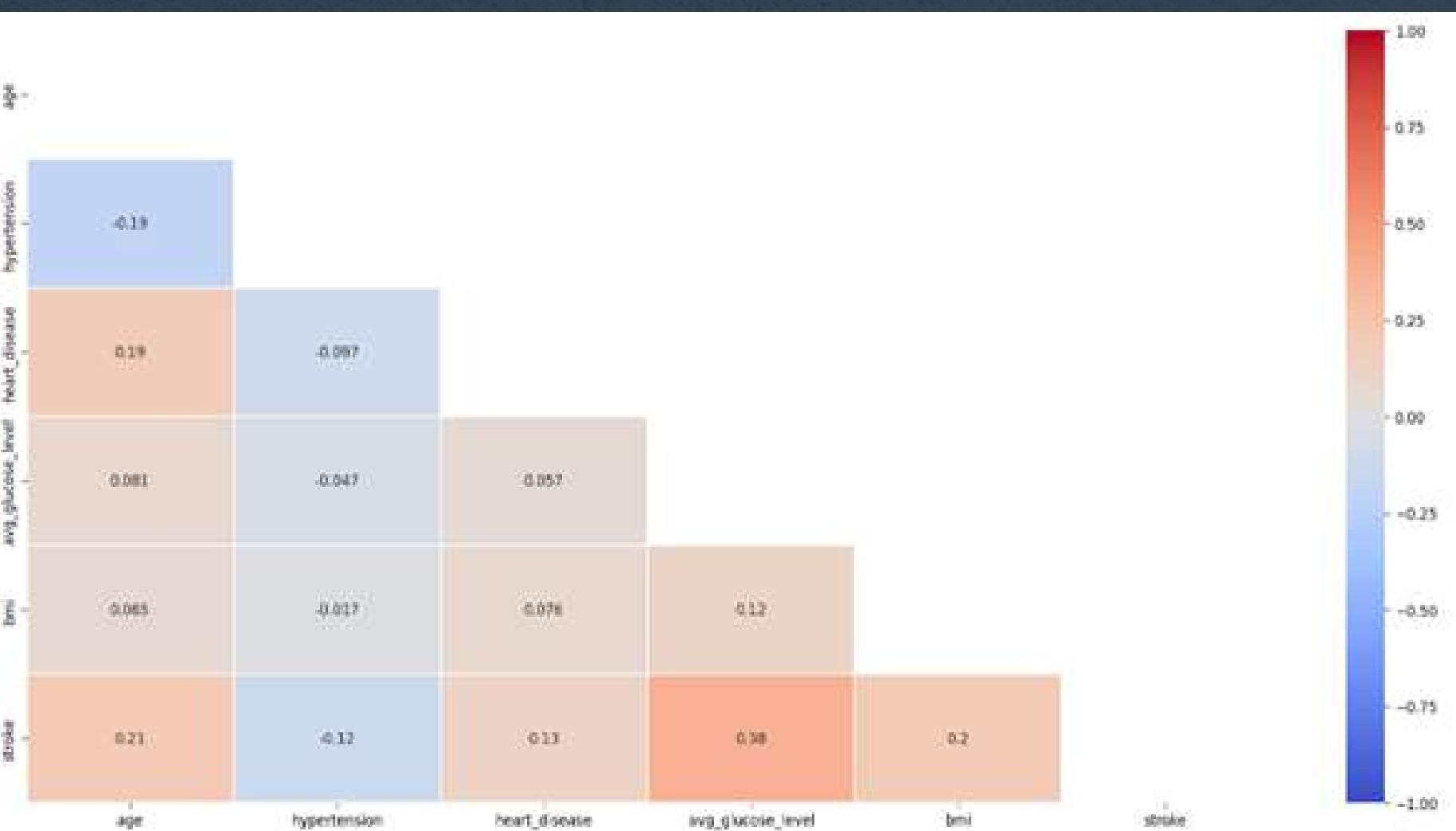
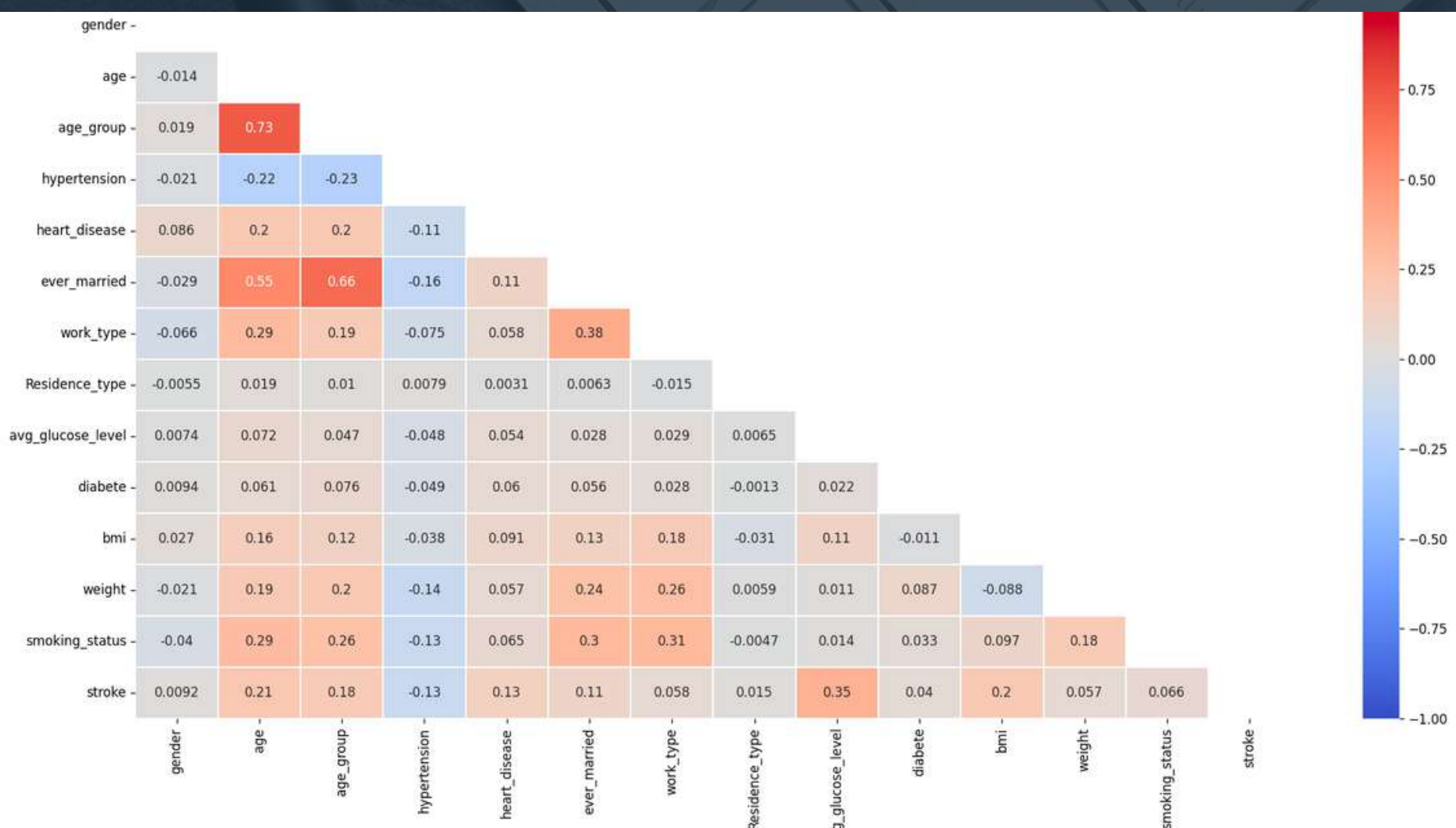
Người hút thuốc và đã từng hút thuốc có tỷ lệ mắc cao hơn nhóm người chưa hút bao giờ.

III. EDA

Kiểm tra outlier của các biến 'age', 'avg_glucose_level', 'bmi'.



III. EDA



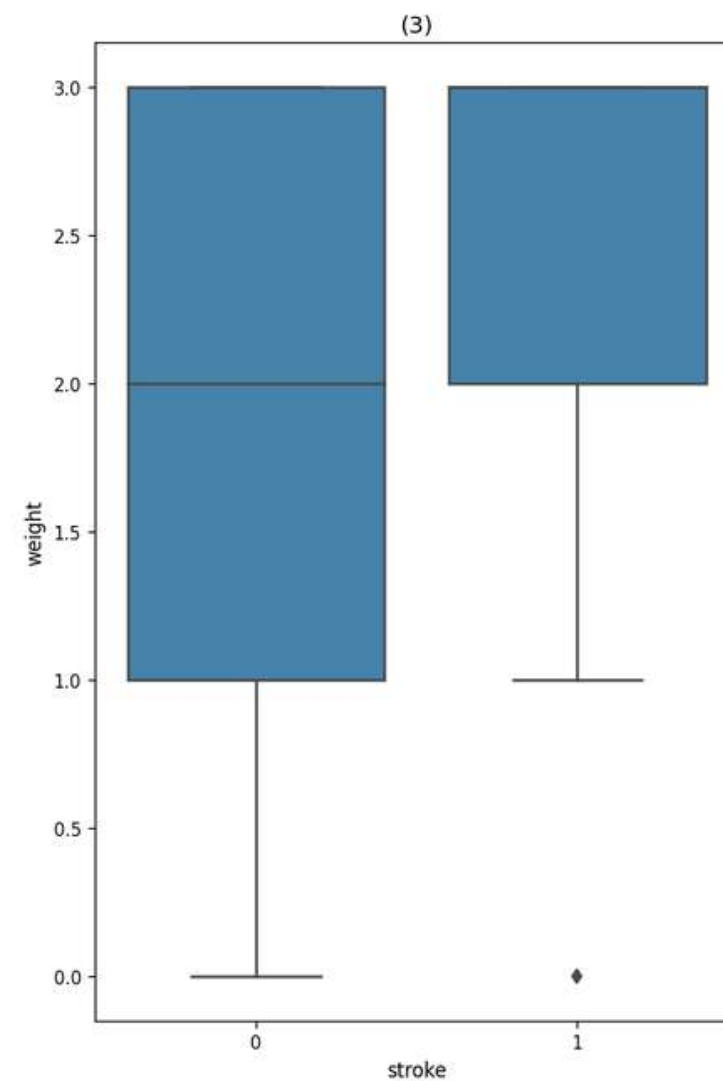
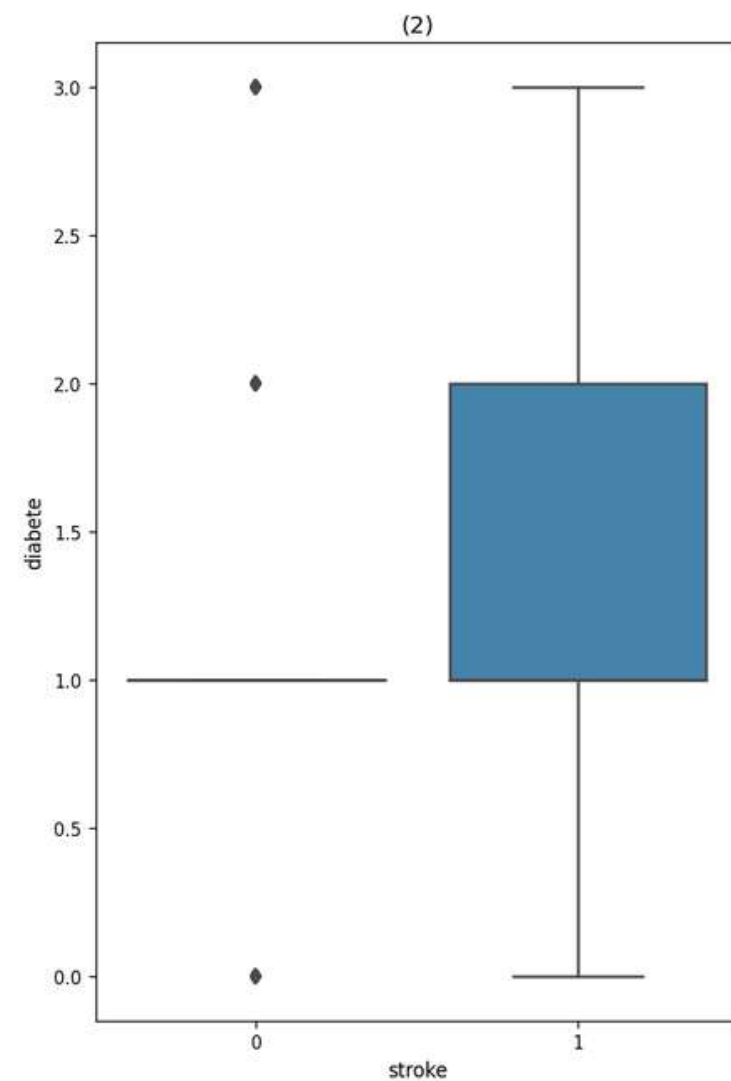
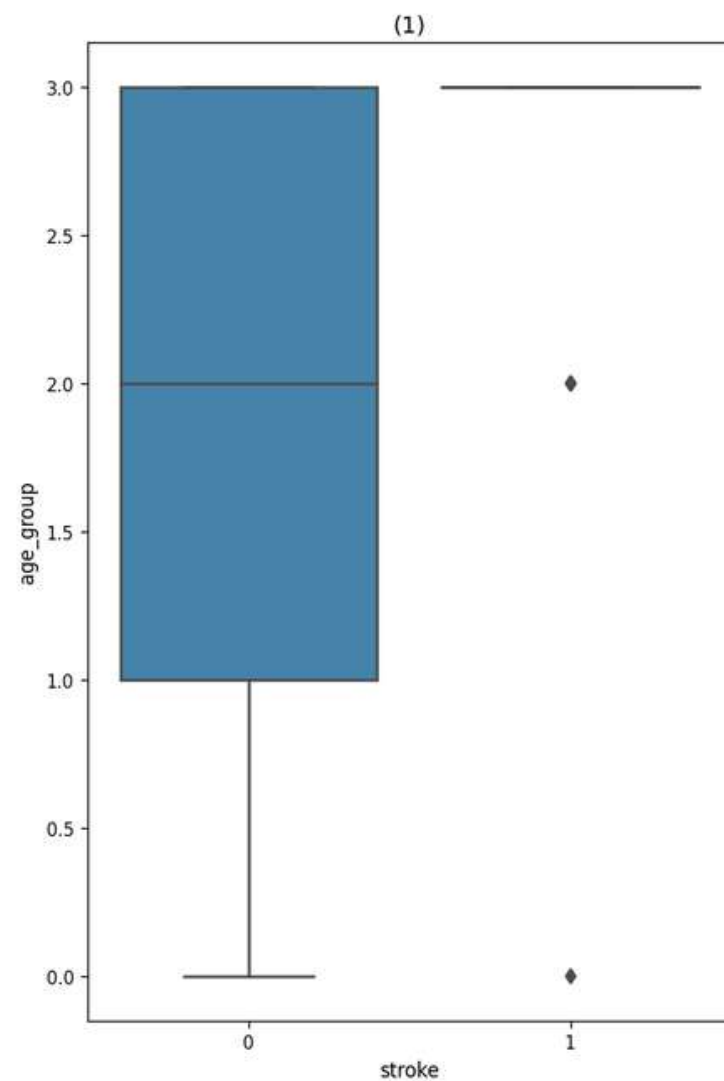
Nguy cơ đột quỵ có tương quan cao với **độ tuổi**, **huyết áp**, **bệnh tim mạch**, **lượng đường huyết** và **chỉ số khối cơ thể (BMI)**. Các biến này sẽ được lựa chọn để đưa vào mô hình dự báo.

IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

Do dữ liệu có outlier nên trước khi đi xây dựng mô hình dự báo, cần xử lý outlier và mất cân bằng dữ liệu.

```
df_drop_outlier = df[df['age_group'] >= 1]
df_drop_outlier
```

```
# xử lý mất cân bằng data
from imblearn.over_sampling import SMOTE
Oversample = SMOTE()
X_train, y_train = Oversample.fit_resample(X_train, y_train)
```



Medicine is the science and practice of treating, preventing



IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

1. Logistic Regression

Xây dựng mô hình

```
#Xây dựng mô hình
from sklearn.linear_model import LogisticRegression #import thư viện
#1 Import mô hình LogisticRegression gán cho biến "model"
model = LogisticRegression()
#2.Training model với training set
model.fit(X_train, y_train) #Training model
```

```
▼ LogisticRegression
LogisticRegression()
```

```
] y_pred = model.predict(X_test) #Dự báo cho tập test
y_pred
```

```
array([1, 0, 0, ..., 0, 0, 0])
```

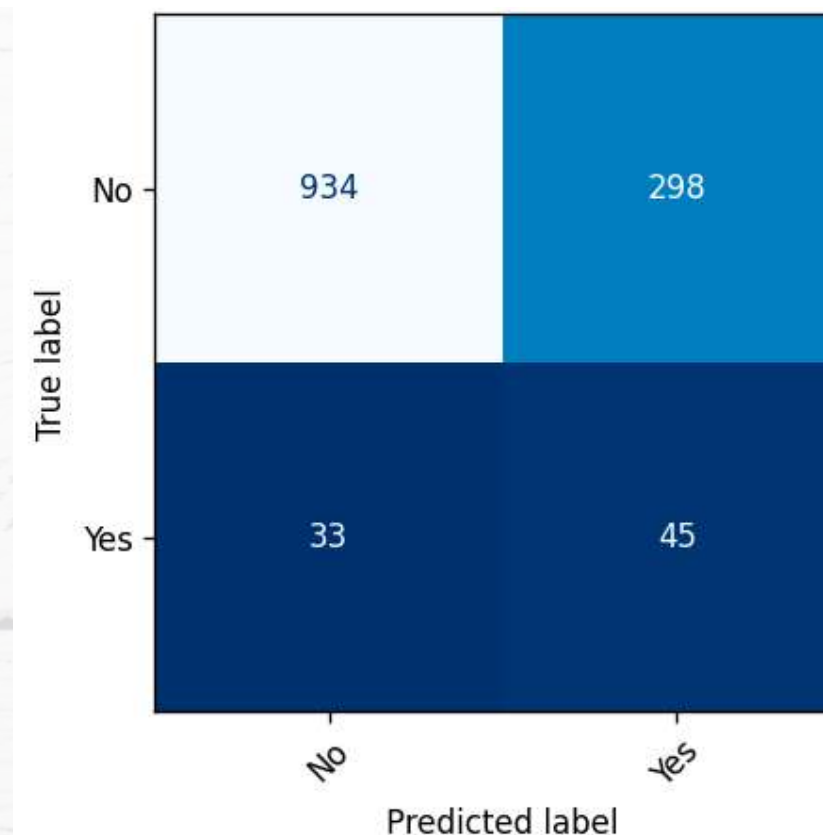
```
] #Đánh giá mô hình lần 2
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred) #Tính độ chính xác của mô hình

0.7473282442748092
```

Đánh giá mô hình

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.97	0.76	0.85	1232
1	0.13	0.58	0.21	78
accuracy			0.75	1310
macro avg	0.55	0.67	0.53	1310
weighted avg	0.92	0.75	0.81	1310

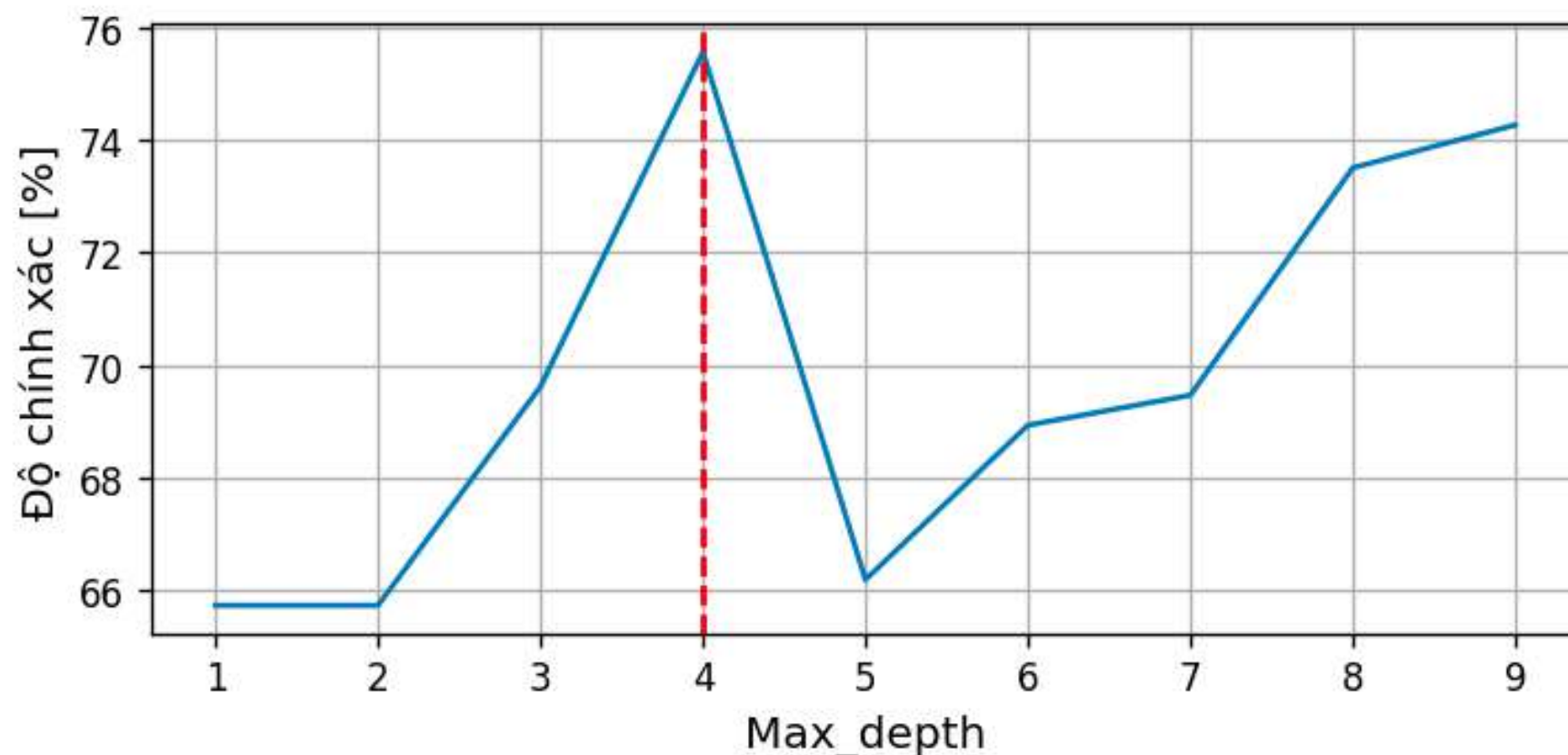


IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

2. Decision Tree Classification

Sử dụng vòng lặp for để tìm max_depth, xây dựng mô hình

```
# Sử dụng vòng lặp for để tìm max_depth
score_total=[]
for i in range(1,10):
    model_tree = tree.DecisionTreeClassifier(max_depth=i)
    model_tree.fit(X_train, y_train)
    y_pred=model_tree.predict(X_test)
    score_total.append(accuracy_score(y_test,y_pred)*100)
```



Đánh giá mô hình

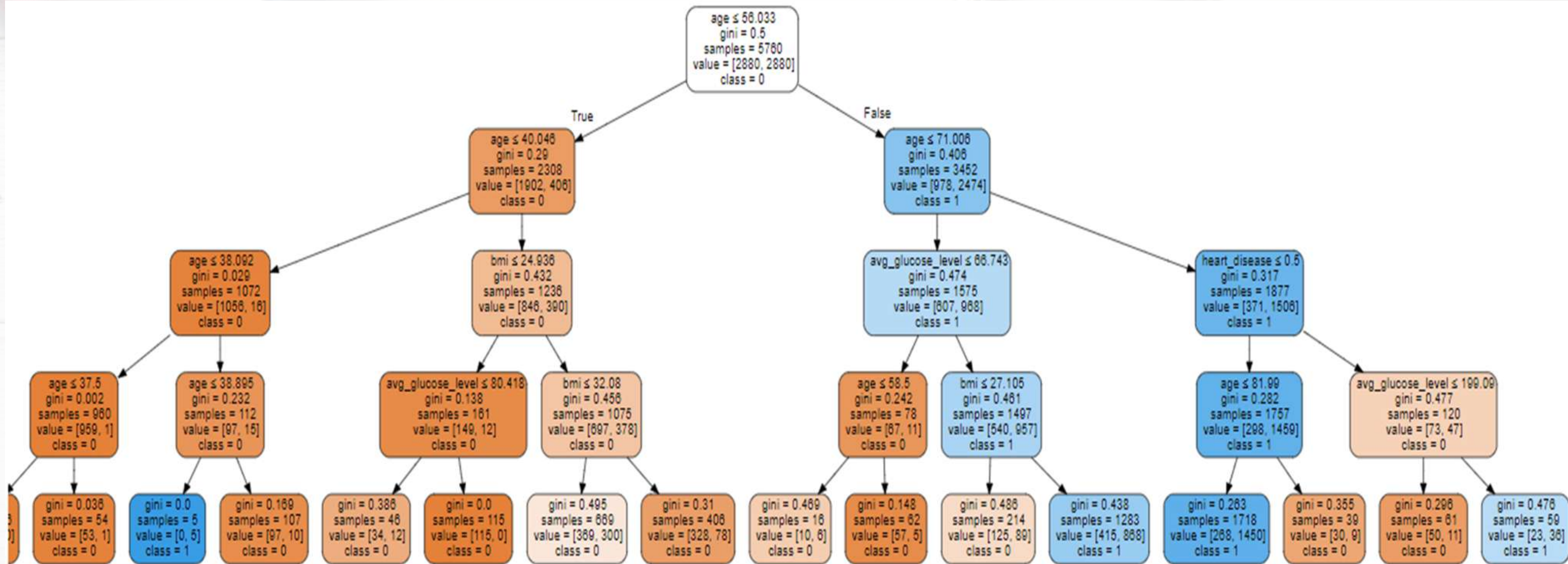
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.97	0.75	0.85	1239
1	0.12	0.56	0.19	71
accuracy			0.74	1310
macro avg	0.54	0.66	0.52	1310
weighted avg	0.92	0.74	0.81	1310

True label	Predicted label	
	No	Yes
No	933	306
Yes	31	40

IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

2. Decision Tree Classification



IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

3. Random Forest Classification

Sử dụng vòng lặp for để tìm hyperparameter và xây dựng mô hình

```
#hyperparameters turning
#from sklearn.model_selection import RandomizedSearchCV
rf = RandomForestClassifier()
from sklearn.model_selection import GridSearchCV
grid_space = {'max_depth': [1,2,3,4,5,6,7,8,9,10, None],
              'n_estimators': [5,10,15,20,25,30],
              'criterion': ['gini', 'entropy'],
              }
rf_grid = GridSearchCV(rf, param_grid=grid_space, scoring='accuracy')
model_grid = rf_grid.fit(X_train, y_train)
```

+ Mã + Văn bản

```
model_grid.best_params_
```

```
{'criterion': 'entropy', 'max_depth': None, 'n_estimators': 30}
```

```
y_pred=model_grid.predict(X_test)
accuracy_score(y_test,y_pred)
```

```
0.8778625954198473
```

Đánh giá mô hình

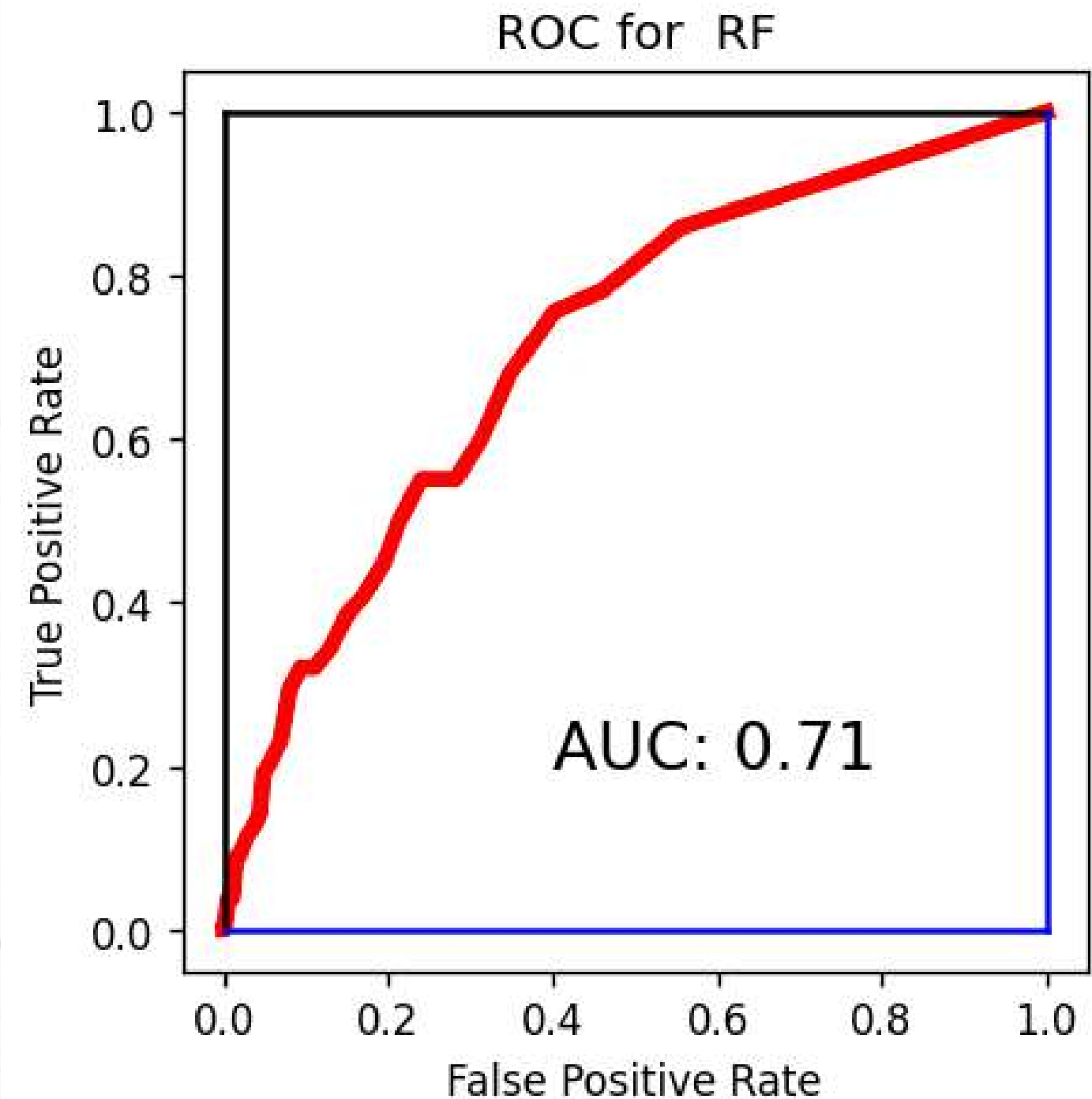
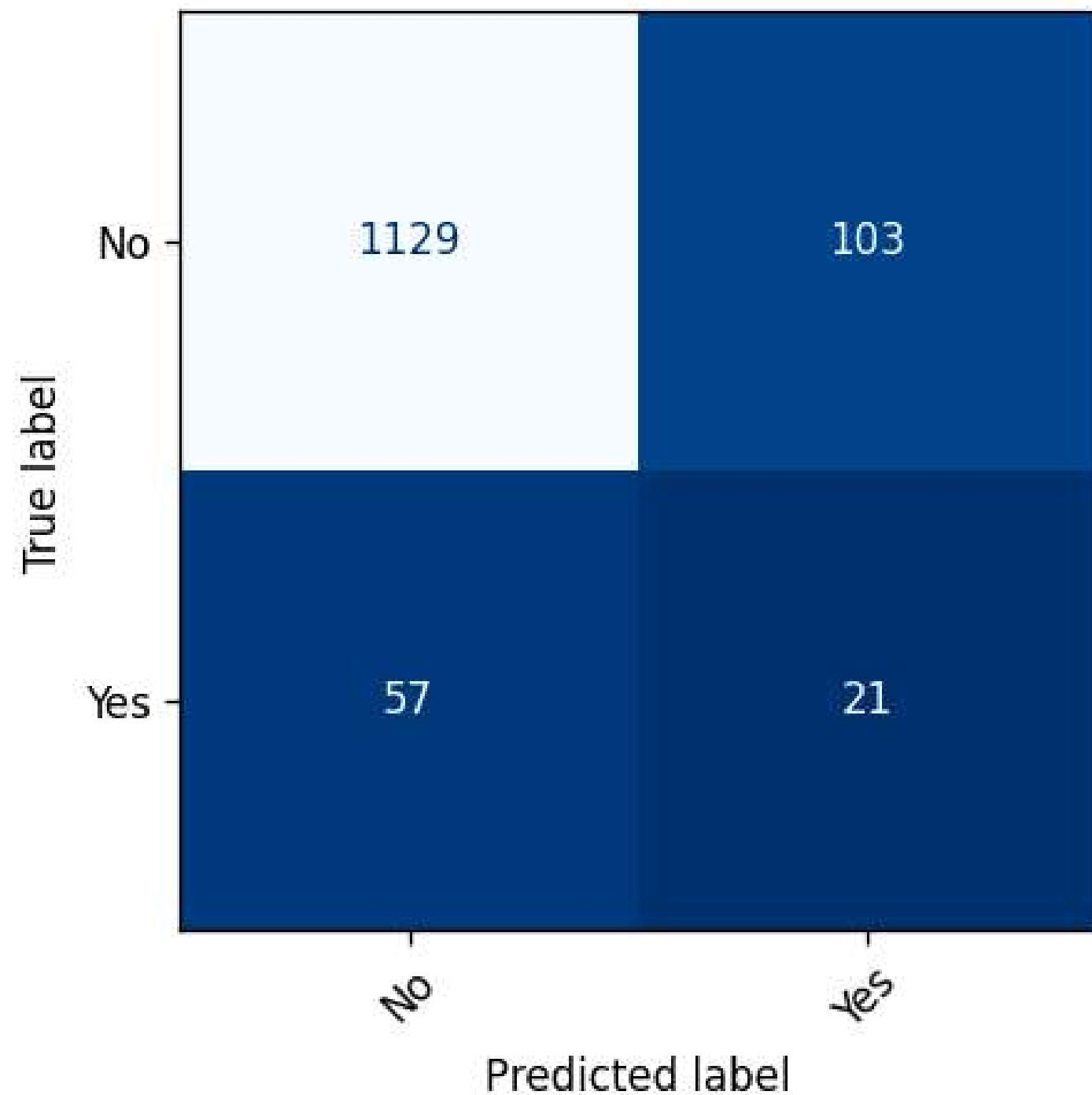
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.95	0.92	0.93	1232
1	0.17	0.27	0.21	78
accuracy			0.88	1310
macro avg	0.56	0.59	0.57	1310
weighted avg	0.91	0.88	0.89	1310

IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

3. Random Forest Classification

Đánh giá mô hình

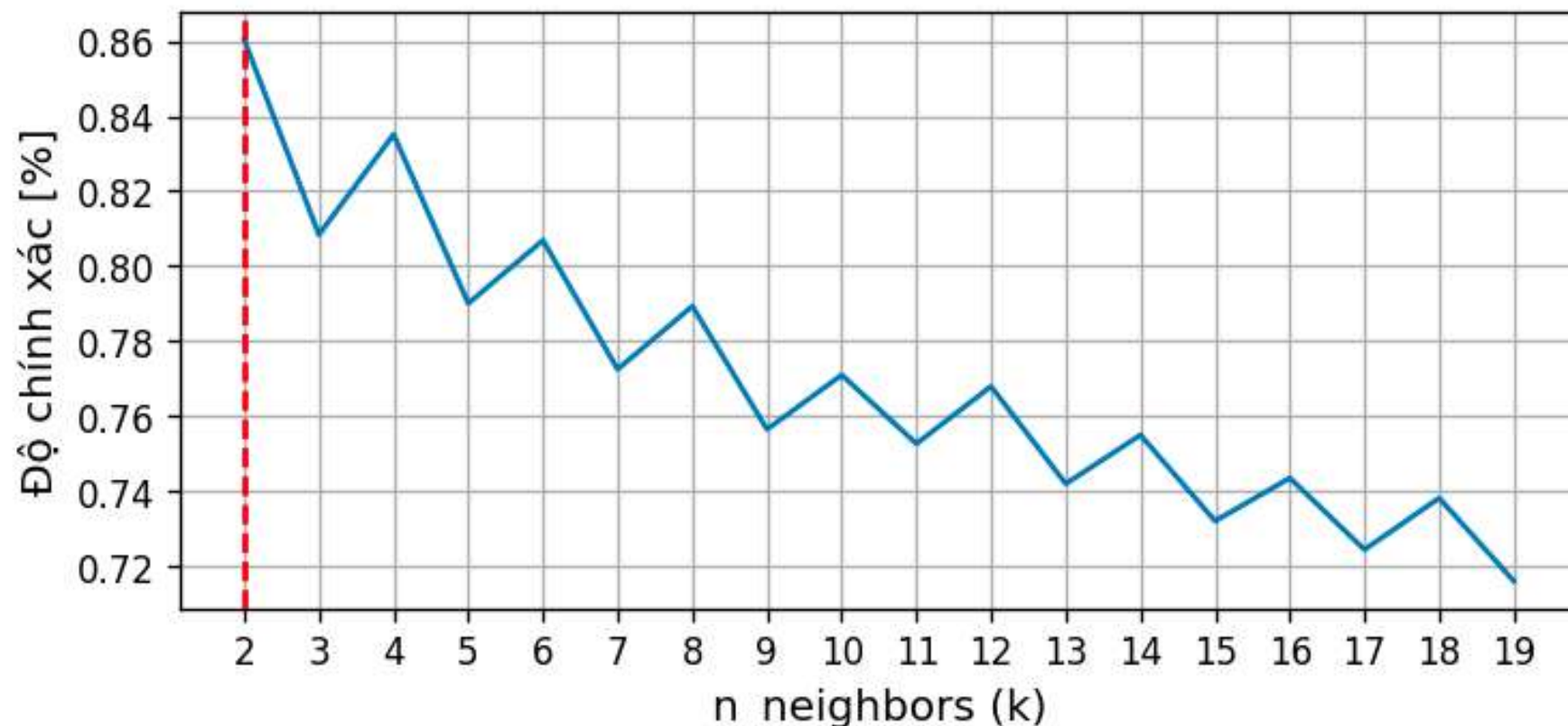


IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

4. K Nearest Neighbour - Classification

Dùng vòng lặp for tìm k tối ưu và xây dựng mô hình dự báo

```
ac=[] #Tạo list rỗng chứa giá trị của độ chính xác
for k in range (2,20,1):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    ac.append(metrics.accuracy_score(y_test, y_pred))
    print("Accuracy at k =",k,metrics.accuracy_score(y_test, y_pred))
```



Đánh giá mô hình

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.95	0.89	0.92	1232
1	0.16	0.33	0.21	78
accuracy			0.85	1310
macro avg	0.56	0.61	0.57	1310
weighted avg	0.91	0.85	0.88	1310

True label	Predicted label	
	No	Yes
No	1094	138
Yes	52	26

IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

5. Support Vector Machine - Classification

Dùng vòng lặp for tìm k tối ưu và xây dựng mô hình dự báo

```
#Xây dựng mô hình
from sklearn.svm import SVC
model = SVC(kernel='linear', C= 0.001, gamma=0.0001) # Sử dụng Linear Kernel
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

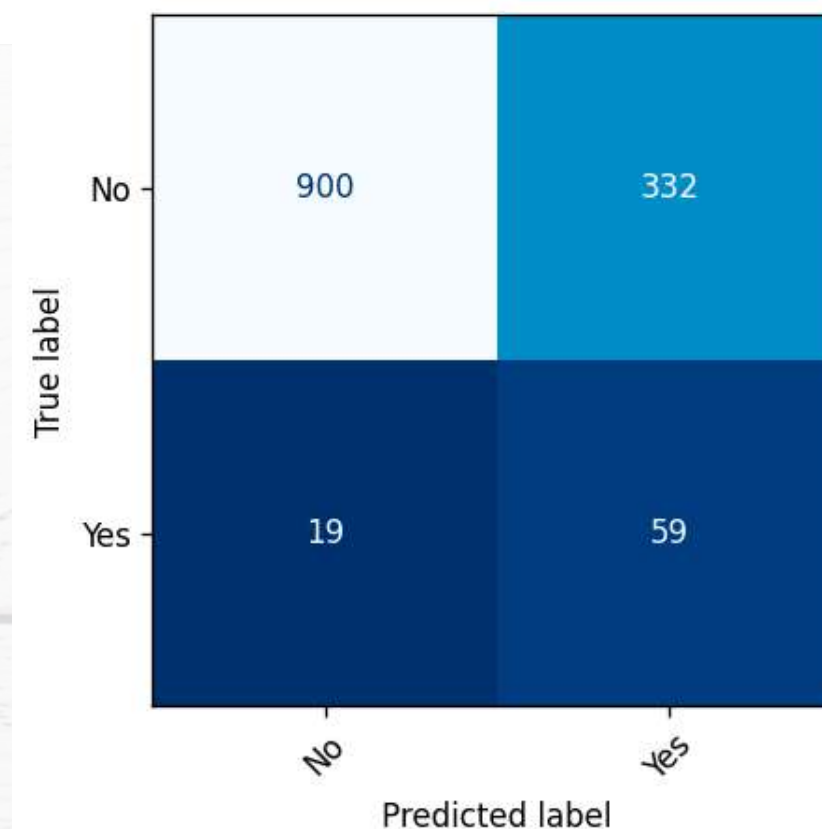
```
#Đánh giá độ chính xác của mô hình
from sklearn import metrics
from sklearn.metrics import classification_report
metrics.accuracy_score(y_test, y_pred)
```

0.73206106870229

Đánh giá mô hình

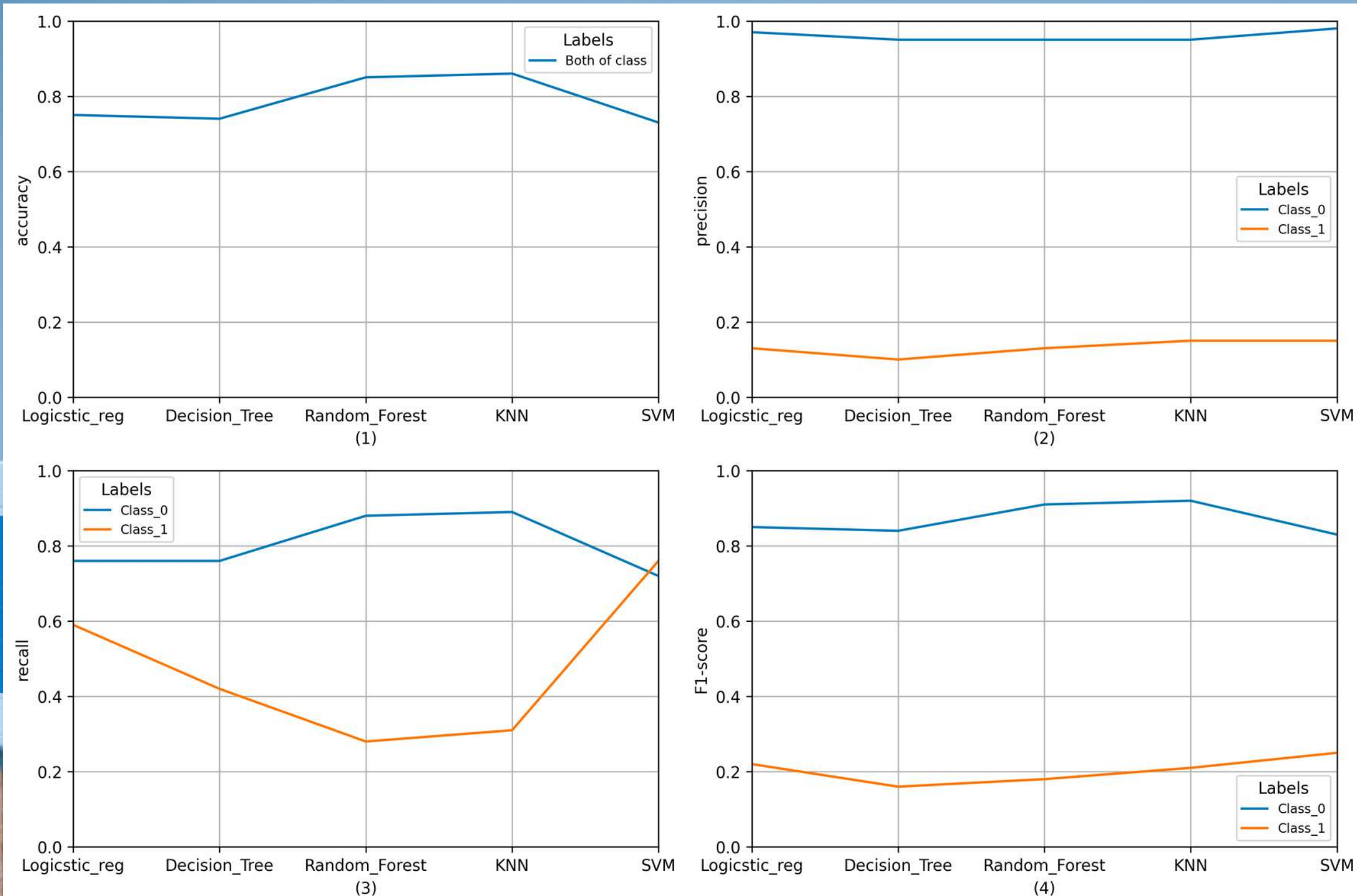
```
#In ra báo cáo đánh giá của mô hình
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	0.73	0.84	1232
1	0.15	0.76	0.25	78
accuracy			0.73	1310
macro avg	0.57	0.74	0.54	1310
weighted avg	0.93	0.73	0.80	1310



IV. XÂY DỰNG MÔ HÌNH DỰ BÁO

Trong các mô hình trên thì mô hình SVM có tỷ lệ dự đoán sai class 1 thấp nhất, recall cao nhất => nên dùng mô hình SVM để đưa vào dự báo.

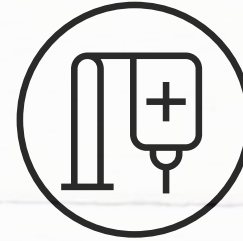


V. ĐỀ XUẤT



Sàng lọc định kỳ

- **Độ tuổi:** Tổ chức các chương trình sàng lọc định kỳ cho nhóm người ở độ tuổi cao, đặc biệt là những người 50 tuổi trở lên.
- **Huyết áp:** Kiểm tra huyết áp định kỳ để phát hiện và quản lý cao huyết áp, đặc biệt là ở nhóm người trưởng thành.



Quản lý yếu tố nguy cơ

- **Bệnh tim mạch:** Tăng cường chăm sóc cho những người có bệnh tim mạch, bao gồm kiểm soát cholesterol và theo dõi các chỉ số tim mạch.
- **Lượng đường huyết:** Đối với người có tiểu đường, duy trì mức đường huyết ổn định thông qua quản lý đường và kiểm soát ăn uống.



Chương trình giáo dục và thay đổi lối sống

- **Chăm sóc sức khỏe cá nhân:** Tăng cường giáo dục về quan trọng của lối sống lành mạnh và thay đổi thói quen ăn uống.
- **Chỉ số khối cơ thể (BMI):** Khuyến khích duy trì một BMI khỏe mạnh thông qua chế độ dinh dưỡng cân đối và hoạt động thể chất.

V. ĐỀ XUẤT



Chăm sóc toàn diện cho người cao tuổi

Đánh giá sức khỏe định kỳ: Cung cấp chăm sóc sức khỏe định kỳ và đánh giá tình trạng sức khỏe của người cao tuổi, tập trung vào các yếu tố nguy cơ đột quỵ.



Theo dõi và quản lý điều trị

- **Hệ thống theo dõi y tế:** Phát triển hệ thống theo dõi y tế để ghi lại và theo dõi thông tin về huyết áp, cholesterol, đường huyết, và BMI của người bệnh.
- **Kế hoạch điều trị cá nhân hóa:** Phát triển kế hoạch điều trị cá nhân hóa dựa trên các yếu tố riêng biệt của từng người, nhằm giảm thiểu rủi ro đột quỵ.



Thank You

