

## **ĐỒ ÁN MÔN HỌC**

# **ỨNG DỤNG HỒI QUY TRONG PHÂN TÍCH DỮ LIỆU: NGHIÊN CỨU TỪ 4 BỘ MẪU**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **THỐNG KÊ MÁY TÍNH VÀ ỨNG DỤNG**

Giảng viên hướng dẫn : ThS. Hứa Thị Phượng Vân

Sinh viên thực hiện :

2286400908-Nguyễn Ngọc Quỳnh Anh

2286400043-Lê Hoàng Gia Vĩ

2286400045-Hoàng Nguyên Vũ

2386400039-Phạm Tường Phát

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2025

## **ĐỒ ÁN MÔN HỌC**

# **ỨNG DỤNG HỒI QUY TRONG PHÂN TÍCH DỮ LIỆU: NGHIÊN CỨU TỪ 4 BỘ MẪU**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **THỐNG KÊ MÁY TÍNH VÀ ỨNG DỤNG**

Giảng viên hướng dẫn : ThS. Hứa Thị Phượng Vân

Sinh viên thực hiện :

2286400908-Nguyễn Ngọc Quỳnh Anh

2286400043-Lê Hoàng Gia Vĩ

2286400045-Hoàng Nguyên Vũ

2386400039-Phạm Tường Phát

Lớp: 22DKHA1

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TPHCM, Ngày..... Tháng ..... Năm 2025

**Giáo viên hướng dẫn**

(Ký tên, đóng dấu)

## **LỜI CAM ĐOAN**

Nhóm chúng tôi gồm Nguyễn Ngọc Quỳnh Anh, Lê Hoàng Gia Vĩ, Phạm Tường Phát và Hoàng Nguyên Vũ xin cam đoan rằng:

Tất cả thông tin và kết quả nghiên cứu trong bài báo cáo này đều trung thực và khách quan, được thu thập từ các nguồn đáng tin cậy, chính thống. Chúng tôi đã phân tích kỹ lưỡng các tài liệu và đảm bảo rằng mọi dữ liệu hoặc ý kiến trích dẫn đều được ghi rõ ràng nguồn gốc, tuân thủ đúng quy định về trích dẫn học thuật.

Chúng tôi cam kết không có hành vi sao chép hoặc sử dụng trái phép bất kỳ thông tin nào từ các nguồn khác. Bài báo cáo này là sản phẩm nghiên cứu độc lập của nhóm, chưa từng được công bố trước đây và tuân thủ đầy đủ các quy định của môn học. Chúng tôi sử dụng công cụ nghiên cứu một cách hợp lý và chính xác, đảm bảo tính khoa học và đạo đức trong quá trình thực hiện.

Nhóm chúng tôi hy vọng rằng bài báo cáo sẽ cung cấp cái nhìn sâu sắc về chủ đề “” và đóng góp tích cực vào phát triển của lĩnh vực này.

TPHCM, Ngày..... Tháng ..... Năm 2025

**Sinh viên**

Nguyễn Ngọc Quỳnh Anh

Lê Hoàng Gia Vĩ

Phạm Tường Phát

Hoàng Nguyên Vũ

# MỤC LỤC

CHƯƠNG 1 .....	9
1.1 DỮ LIỆU 1: MÔ HÌNH HỒI QUY ĐA BIẾN.....	10
1.1.1 Giới thiệu bộ dữ liệu .....	10
1.1.2 Phân tích và chọn mô hình .....	13
1.1.3 Nhận xét và kết luận.....	20
1.2 DỮ LIỆU 2: MÔ HÌNH HỒI QUY LOGISTIC.....	22
1.2.1 Giới thiệu bộ dữ liệu .....	22
1.2.2 Phân tích và chọn mô hình .....	26
1.2.3 Nhận xét và kết luận.....	32
1.3 DỮ LIỆU 3: HỒI QUY THÀNH PHẦN CHÍNH.....	34
1.3.1 Giới thiệu bộ dữ liệu .....	34
1.3.2 Phân tích và chọn mô hình .....	36
1.3.3 Nhận xét và kết luận.....	43
CHƯƠNG 2 .....	45
2.1 Thu thập và tiền xử lý dữ liệu.....	46
2.2 Mô hình. ....	52
2.3 Nhận xét và kết luận.....	54

# DANH MỤC HÌNH ẢNH

Hình 1.1: Một vài quan trắc đầu tiên và số chiều của bộ dữ liệu.....	11
Hình 1.2: Phân bố ban đầu của các biến.....	11
Hình 1.3: Phân bố của các biến so với biến phụ thuộc stroke.....	11
Hình 1.4: Biểu đồ boxplot của các biến so với biến phụ thuộc stroke.....	12
Hình 1.5: Điền thiếu bằng median.....	13
Hình 1.6: Loại bỏ outliers.....	14
Hình 1.7: Biểu đồ sau khi loại bỏ outliers.....	14
Hình 1.8: Mã hoá và chuẩn hoá dữ liệu.....	15
Hình 1.9: Biểu đồ KDE cho từng cột trước và sau khi chuẩn hóa.....	15
Hình 1.10: Mô hình ban đầu.....	16
Hình 1.11: Kết quả VIF.....	17
Hình 1.12: Mô hình sau khi các biến không có ý nghĩa thống kê.....	17
Hình 1.13: Kết quả huấn luyện mô hình.....	19
Hình 1.14: Confusion matrix.....	20
Hình 2.1: Dữ liệu trùng lặp.....	23
Hình 2.2: Một vài quan trắc đầu tiên và số chiều của bộ dữ liệu.....	24
Hình 2.3: Phân bố ban đầu của các biến.....	24
Hình 2.4: Phân bố của các biến so với biến phụ thuộc class.....	24
Hình 2.5: Biểu đồ boxplot của các biến so với biến phụ thuộc class.....	25
Hình 2.6: Chuẩn hoá RobustScaler.....	27
Hình 2.7: Biểu đồ KDE cho từng cột trước và sau khi chuẩn hóa.....	27
Hình 2.8: Mô hình ban đầu.....	28
Hình 2.9: Kết quả VIF.....	28
Hình 2.10: loại bỏ x4 (fConc) có giá trị VIF cao nhất 28.05.....	29
Hình 2.11: Mô hình sau khi loại bỏ đa cộng tuyến và các biến không có ý nghĩa thống kê.....	30

Hình 2.12: Kết quả huấn luyện mô hình.....	30
Hình 2.13: Confusion Matrix.....	31
Hình 2.14: Biểu đồ đường cong ROC .....	31
Hình 3.1: Mô tả dữ liệu.....	34
Hình 3.2: Dữ liệu trùng lặp.....	34
Hình 3.3: Dữ liệu thiếu .....	35
Hình 3.4: Một vài mẫu đầu tiên. ....	35
Hình 3.5: Phân bố ban đầu của các biến.....	36
Hình 3.6: Phân bố của các biến so với biến phụ thuộc satisfaction. ....	36
Hình 3.7: Phân bố của các biến so với biến phụ thuộc satisfaction. ....	38
Hình 3.8: Biểu đồ boxplot của các biến. ....	39
Hình 3.9: Xử lý NaN.....	39
Hình 3.10 Chuẩn hóa dùng Ordinal Encoder.....	39
Hình 3.11 Kết quả sau khi chuẩn hóa.....	40
Hình 3.12: Chuẩn hoá RobustScaler.....	40
Hình 3.13: Biểu đồ KDE sau khi chuẩn hoá. ....	40
Hình 3.14: Số thành phần chính .....	41
Hình 3.15: Mô hình sau khi giảm chiều .....	41
Hình 3.16: Kết quả huấn luyện mô hình.....	42
Hình 3.17: Confusion Matrix.....	43
Hình 4.1: Thu thập dữ liệu.....	46
Hình 4.2: Một vài quan trắc đầu tiên. ....	47
Hình 4.3: Loại bỏ các cột ít đóng góp và tạo biến phụ thuộc.....	48
Hình 4.4: Biểu đồ boxplot của các biến. ....	49
Hình 4.5: Mã hoá dữ liệu.....	49
Hình 4.6: Chuẩn hoá dữ liệu.....	50
Hình 4.7: Các biến sau khi chuẩn hoá. ....	50

Hình 4.8: Tỷ lệ phương sai giải thích. ....	51
Hình 4.9: Phương sai tích lũy. ....	51
Hình 4.10: Lựa chọn thành phần chính. ....	51
Hình 4.11: Mô hình sau khi giảm chiều ....	52
Hình 4.12: Kết quả huấn luyện mô hình.....	53
Hình 4.13: Confusion Matrix.....	54



# CHƯƠNG 1

## DỮ LIỆU CÓ SẴN

- Tên đề tài, nguồn gốc của dữ liệu, giới thiệu các biến.
- Mô hình chọn được, phân tích kết quả.
- Đưa ra những phương pháp / phân tích khác có thể giúp cho kết quả tốt hơn.
- Kết luận

## 1.1 DỮ LIỆU 1: MÔ HÌNH HỒI QUY ĐA BIẾN

### 1.1.1 Giới thiệu bộ dữ liệu

Bộ dữ liệu này được tìm thấy trên Kaggle, một cộng đồng trực tuyến về khoa học dữ liệu và học máy. Đây là bộ dữ liệu về **Dự đoán đột quỵ** (Stroke Prediction Dataset)[1], được sử dụng để phân tích và dự đoán khả năng mắc đột quỵ của bệnh nhân. Bộ dữ liệu này ghi lại thông tin chi tiết về các bệnh nhân, bao gồm các yếu tố liên quan đến sức khỏe và tình trạng bệnh lý của họ. Nó bao gồm tổng cộng 5.110 quan sát, mỗi quan sát đại diện cho một bệnh nhân, và có 12 biến đặc trưng. Những biến này bao gồm các thông tin quan trọng như giới tính, tuổi tác, các bệnh lý nền, tình trạng hút thuốc, và nhiều yếu tố khác có thể ảnh hưởng đến nguy cơ đột quỵ. Cụ thể các thông số bao gồm:

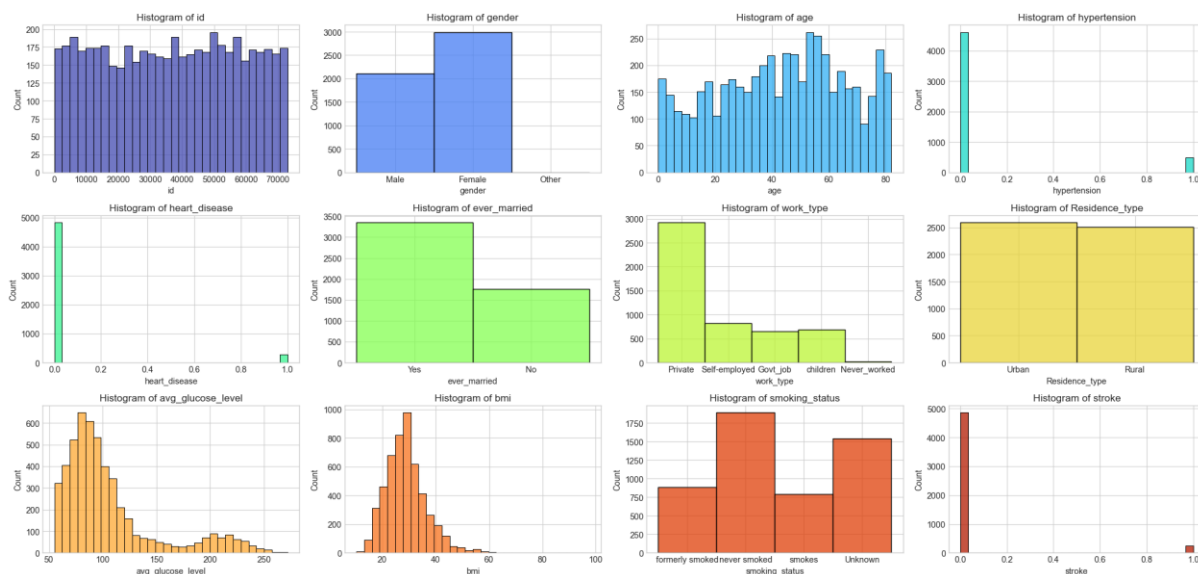
- Id: Mã định danh duy nhất.
- Gender: Giới tính bệnh nhân.
- Age: Tuổi của bệnh nhân.
- hypertension: 0 nếu bệnh nhân không bị tăng huyết áp, 1 nếu bệnh nhân bị tăng huyết áp.
- heart\_disease: 0 nếu bệnh nhân không có bệnh lý tim mạch, 1 nếu bệnh nhân có bệnh lý tim mạch.
- ever\_married: "Không" hoặc "Có".
- work\_type: "Trẻ em", "Công chức", "Chưa từng làm việc", "Tư nhân" hoặc "Tự làm chủ".
- Residence\_type: "Nông thôn" hoặc "Thành thị".
- avg\_glucose\_level: Mức đường huyết trung bình trong máu.
- bmi: Chỉ số khối cơ thể.
- smoking\_status: "Đã từng hút thuốc", "Chưa từng hút thuốc", "Đang hút thuốc" hoặc "Không rõ".

- stroke: 1 nếu bệnh nhân đã bị đột quỵ, 0 nếu không bị đột quỵ.

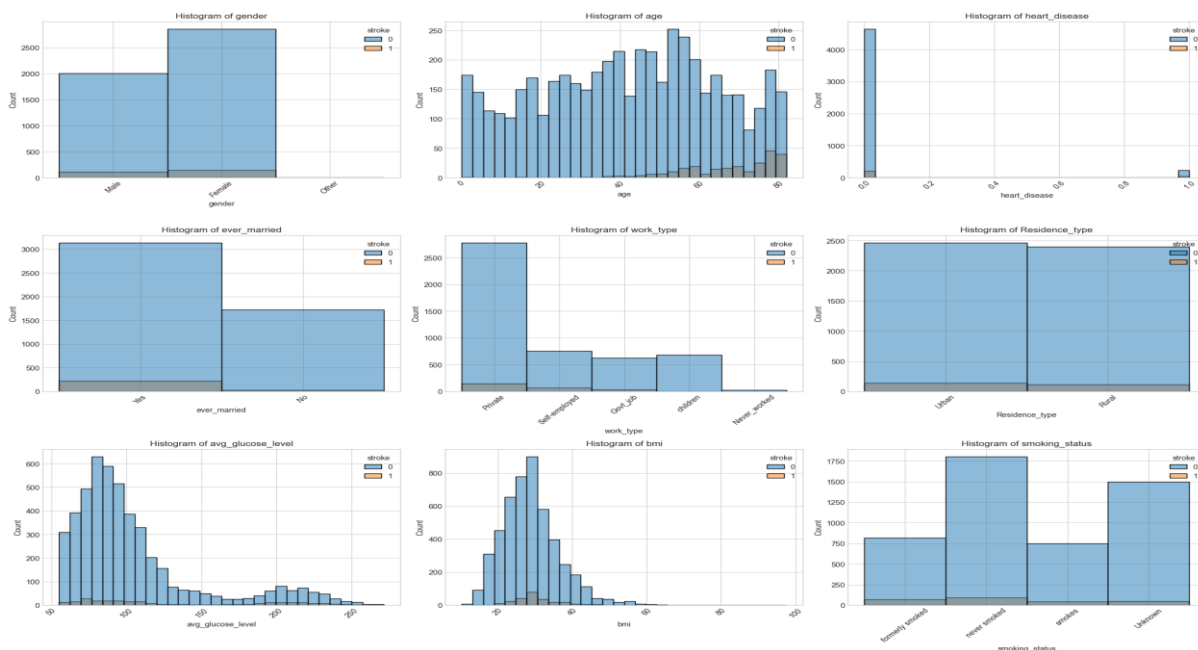
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Hình 1.1: Một vài quan trắc đầu tiên và số chiều của bộ dữ liệu.

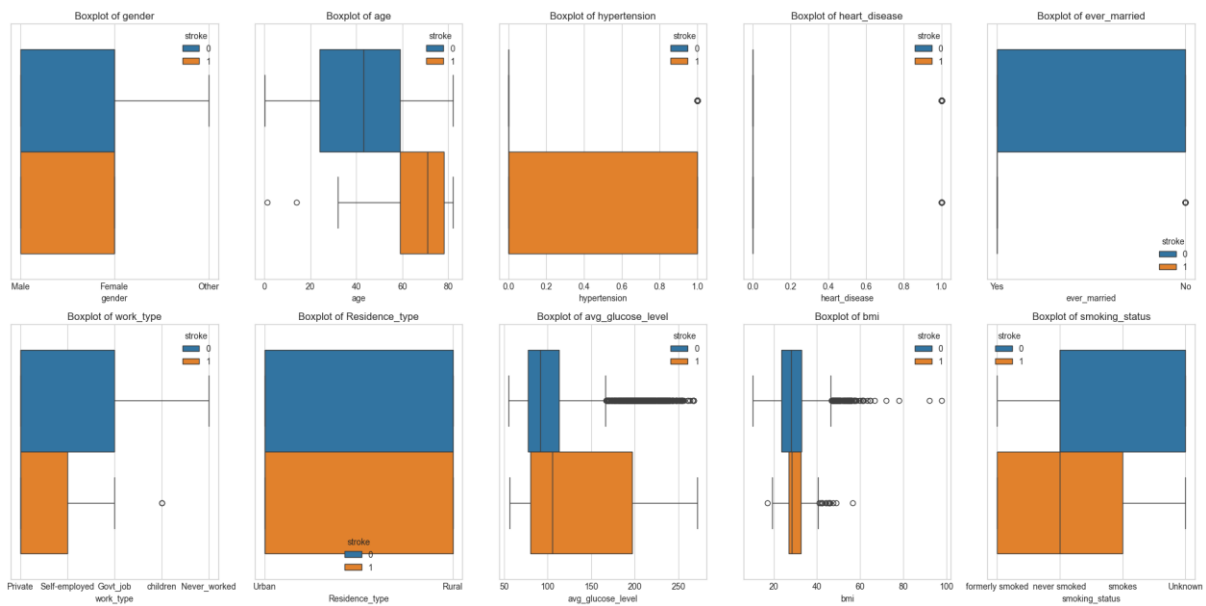
Phân bố ban đầu của các biến và trung bình tổng của từng biến theo biến phụ thuộc stroke.



Hình 1.2: Phân bố ban đầu của các biến



Hình 1.3: Phân bố của các biến so với biến phụ thuộc stroke.



Hình 1.4: Biểu đồ boxplot của các biến so với biến phụ thuộc stroke.

Biểu đồ phân phối tổng quát cho thấy rằng hầu hết các thuộc tính trong dữ liệu có sự phân phối rõ ràng, với một số đặc điểm nổi bật như:

- Đa số người tham gia khảo sát không bị tăng huyết áp, bệnh tim và không bị đột quỵ. Tuy nhiên, khi so sánh với các biểu đồ phân nhóm, tỷ lệ đột quỵ có xu hướng tăng đáng kể ở những người có bệnh lý nền như tăng huyết áp và bệnh tim.
- Người cao tuổi (60+) có nguy cơ bị đột quỵ cao hơn. Điều này được minh họa rõ ràng ở cả phân phối tuổi và các boxplot, khi nhóm tuổi lớn hơn có tỷ lệ đột quỵ cao hơn đáng kể.

Ảnh hưởng của nhân khẩu học (Giới tính, Tình trạng hôn nhân, Loại công việc): Phân tích về giới tính cho thấy nữ chiếm tỷ lệ cao hơn trong mẫu dữ liệu và cũng có tỷ lệ đột quỵ cao hơn so với nam giới, như biểu diễn trong biểu đồ tổng quát và phân nhóm. Tuy nhiên, boxplot không cho thấy sự khác biệt đáng kể về giá trị trung bình của tuổi hoặc BMI giữa các giới tính bị đột quỵ. Về tình trạng hôn nhân, phần lớn những người đã từng kết hôn có nguy cơ bị đột quỵ cao hơn, điều này có thể liên quan đến việc họ thuộc nhóm tuổi lớn hơn hoặc có nhiều yếu tố nguy cơ sức khỏe hơn. Xét về loại công việc, nhóm làm việc tự nhân và tự do chiếm tỷ lệ cao

trong dữ liệu và có tỷ lệ đột quỵ cao hơn, như biểu đồ phân nhóm và boxplot đã chỉ ra, mặc dù sự chênh lệch giữa các nhóm không quá lớn.

Ảnh hưởng của yếu tố sức khỏe (Tăng huyết áp, Bệnh tim, Đường huyết, BMI): Những người có tăng huyết áp hoặc bệnh tim chiếm tỷ lệ lớn trong nhóm bị đột quỵ. Đây là hai yếu tố nguy cơ lớn cho đột quỵ. Phân phối đường huyết cho thấy một số lượng đáng kể người có mức đường huyết rất cao. Phân tích theo nhân cho thấy đột quỵ thường xảy ra ở nhóm có đường huyết cao hơn. Boxplot cũng minh họa sự chênh lệch này, với nhiều giá trị ngoại lệ ở nhóm đột quỵ. Phân phối BMI gần chuẩn, nhưng nhóm bị đột quỵ thường có giá trị BMI cao hơn, minh họa mối liên hệ giữa béo phì và nguy cơ đột quỵ.

### 1.1.2 Phân tích và chọn mô hình

Vì mục đích của bài toán là phân loại bệnh nhân bị đột quỵ hoặc bệnh nhân không bị đột quỵ nên chúng tôi lựa chọn mô hình hồi quy Logistic cho bài toán này.

Sau khi rà soát bộ dữ liệu nhóm nhận thấy trong bộ dữ liệu có biến BMI bị khuyết thiếu dữ liệu. Tuy nhiên tỉ lệ thiếu rất ít chỉ chiếm khoảng 0.5% do đó nhóm quyết định điền dữ liệu khuyết thiếu bằng giá trị median để đảm bảo tính toàn vẹn của bộ dữ liệu. Vì số lượng dữ liệu thiếu rất ít, việc áp dụng phương pháp này sẽ không gây sai lệch đáng kể trong kết quả phân tích.

```
df['bmi'] = df.groupby('gender')['bmi'].transform(lambda x: x.fillna(x.median()))
```

Hình 1.5: Điền thiếu bằng median.

Sau khi xử lý dữ liệu thiếu, chúng tôi tiến hành xử lý outliers. Sau khi phân tích đặc điểm của các thuộc tính, chúng tôi quyết định giữ lại các giá trị ngoại lai (outliers) của thuộc tính của age, hypertension, heart\_disease,

stroke, avg\_glucose\_level, vì các giá trị này phù hợp cho việc dự đoán đột quỵ và có ý nghĩa về mặt y học. Ví dụ, đối với thuộc tính mức glucose trung bình (avg\_glucose\_level), một số nghiên cứu phân tích dữ liệu đã chỉ ra rằng bệnh nhân có đường huyết trung bình cao (ví dụ: **200–300 mg/dL**) có nguy cơ đột quỵ và các biến chứng nặng hơn so với những người có mức đường huyết bình thường (khoảng **70–140 mg/dL**). Tương tự, age, hypertension, heart\_disease và stroke cũng có giá trị ngoại lai nằm gần mức trung bình thường và không có bất thường, do đó không cần loại bỏ.

Ngược lại, đối với thuộc tính bmi, chúng tôi nhận thấy một số giá trị ngoại lai nằm ở mức vô lý, chẳng hạn như chỉ số bmi vượt quá 45, điều này không phù hợp với thực tế y học. Do đó, chúng tôi quyết định loại bỏ hoàn toàn các giá trị ngoại lai của thuộc tính này để đảm bảo tính chính xác và hợp lý của dữ liệu.

Chúng tôi sử dụng khoảng tứ phân vị để loại bỏ Outliers như sau:

```
Q1 = df['bmi'].quantile(0.25)
Q3 = df['bmi'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

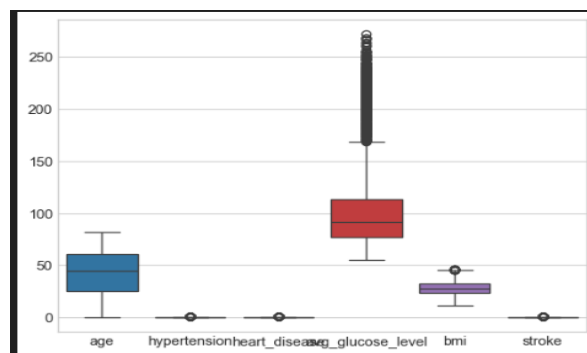
data_no_outliers = df[(df['bmi'] >= lower_bound) & (df['bmi'] <= upper_bound)]
rows_before = len(df)
rows_after = len(data_no_outliers)

rows_before, rows_after

✓ 0.0s
(5110, 4984)
```

Hình 1.6: Loại bỏ outliers.

Sau khi loại bỏ outliers ta được như hình sau:



Hình 1.7: Biểu đồ sau khi loại bỏ outliers.

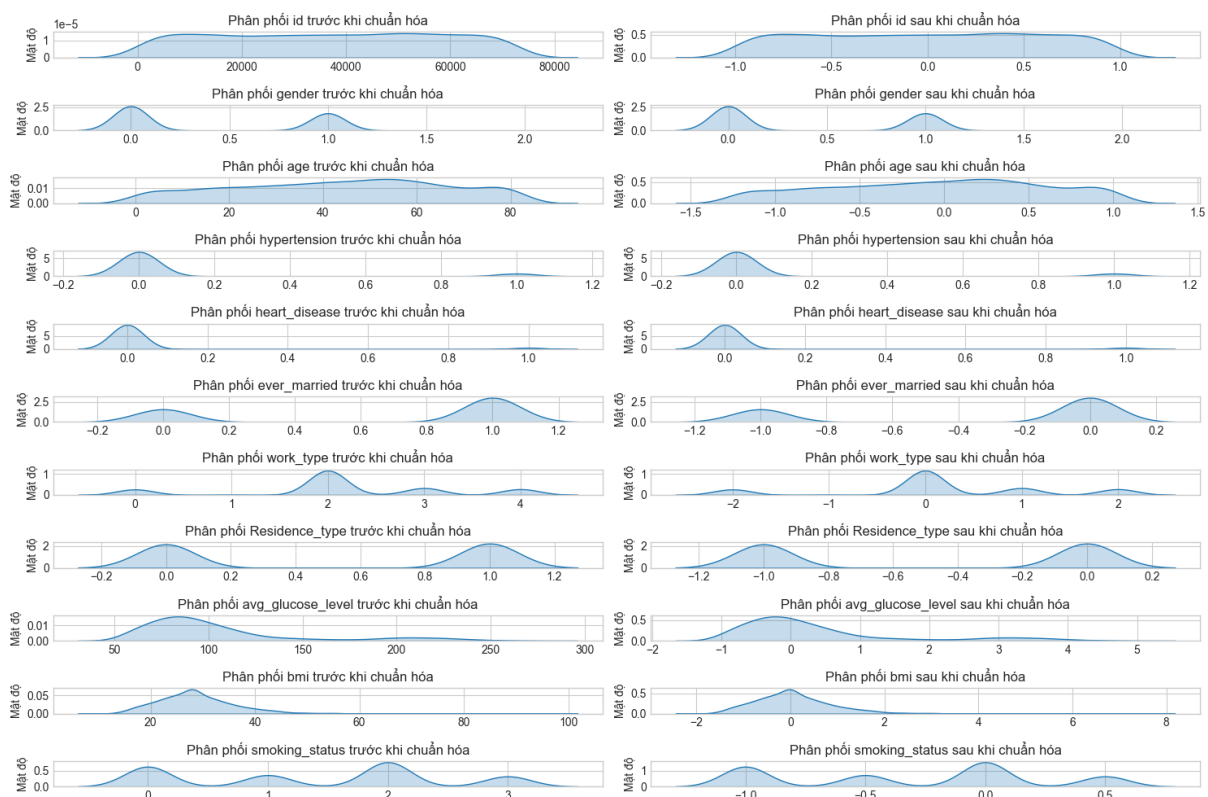
Sau đó nhóm tiếp tục mã hoá cho các biến phân loại có kiểu dữ liệu là ‘object’ về dạng số để chuẩn bị cho việc xây dựng mô hình hồi quy. Nhóm sử dụng LabelEncoder để mã hoá dữ liệu. Tiếp theo nhóm tiếp tục chuẩn hoá dữ liệu, đưa dữ liệu về cùng một thang đo tránh việc mô hình học máy gán trọng số cho các biến có giá trị lớn hơn dẫn tới việc bị thiên vị cho thuộc tính có giá trị lớn. Gây ảnh hưởng đến kết quả sau này khi xây dựng mô hình hồi quy.

```
# Mã hóa các cột phân loại
categorical_columns = df.select_dtypes(include=['object']).columns
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
df_original = df.copy()

# Áp dụng RobustScaler cho toàn bộ dataset
scaler = RobustScaler()
df.iloc[:, :-1] = scaler.fit_transform(df.iloc[:, :-1]) # Không áp dụng cho cột mục tiêu 'stroke'

df.head()
```

Hình 1.8: Mã hoá và chuẩn hoá dữ liệu.



Hình 1.9: Biểu đồ KDE cho từng cột trước và sau khi chuẩn hóa.

Mô hình ban đầu có các thông số như hình sau. Theo như mô hình ta có thể thấy rằng có 1 số thuộc tính không có ý nghĩa thống kê. Mặt khác có một số thuộc tính có ý nghĩa thống kê cao.

```
Optimization terminated successfully.
Current function value: 0.155535
Iterations 9
```

Logit Regression Results						
=====						
Dep. Variable:	stroke	No. Observations:	5110			
Model:	Logit	Df Residuals:	5099			
Method:	MLE	Df Model:	10			
Date:	Sun, 05 Jan 2025	Pseudo R-squ.:	0.2014			
Time:	14:13:21	Log-Likelihood:	-794.79			
converged:	True	LL-Null:	-995.19			
Covariance Type:	nonrobust	LLR p-value:	6.358e-80			
=====						
	coef	std err	z	P> z	[0.025	0.975]
-----						
const	-4.0279	0.172	-23.371	0.000	-4.366	-3.690
x1	0.0510	0.140	0.364	0.716	-0.224	0.326
x2	2.5227	0.191	13.178	0.000	2.147	2.898
x3	0.3896	0.164	2.374	0.018	0.068	0.711
x4	0.3203	0.190	1.688	0.091	-0.052	0.692
x5	-0.1889	0.219	-0.862	0.389	-0.619	0.241
x6	-0.0531	0.072	-0.735	0.462	-0.195	0.089
x7	0.0990	0.138	0.719	0.472	-0.171	0.369
x8	0.1523	0.044	3.466	0.001	0.066	0.238
x9	-0.0059	0.101	-0.058	0.954	-0.203	0.191
x10	0.0009	0.144	0.006	0.995	-0.281	0.283
=====						

Hình 1.10: Mô hình ban đầu

Tiến hành sử dụng phương pháp tính hệ số VIF để kiểm tra hiện tượng đa cộng tuyến trong mô hình này. Kết quả cho thấy tất cả các thuộc tính đều có giá trị VIF dưới 5 chứng tỏ không xảy ra hiện tượng đa cộng tuyến trong mô hình này.



	feature	VIF
0	x1	1.480447
1	x2	2.135948
2	x3	1.209098
3	x4	1.171113
4	x5	2.412230
5	x6	1.311718
6	x7	1.497101
7	x8	1.197649
8	x9	1.210303
9	x10	1.455448

Hình 1.11: Kết quả VIF

Tuy nhiên để chính xác hơn cần phải xem xét thêm các đặc trưng có ý nghĩa thống kê hay không. ( $P > |z|$ )  $< 0.05$ . Sau khi xem xét và loại bỏ từ từ thì chúng tôi phát hiện ra x1, x4, x5, x6, x7, x9, x10 đại diện cho gender, heart\_disease, ever\_married, work\_type, residence\_type, bmi, smoking\_status có ( $P > |z|$ )  $> 0.05$  không có ý nghĩa thống kê nên loại bỏ. Cuối cùng chúng tôi giữ lại các thuộc tính sau hypertension, age, avg\_glucose\_level đưa vào mô hình hồi quy logistic và thu được bảng thông số và kết quả sau.

```

Optimization terminated successfully.
Current function value: 0.156010
Iterations 9
  
```

Logit Regression Results						
Dep. Variable:	stroke	No. Observations:	5110			
Model:	Logit	Df Residuals:	5106			
Method:	MLE	Df Model:	3			
Date:	Tue, 07 Jan 2025	Pseudo R-squ.:	0.1989			
Time:	18:44:08	Log-Likelihood:	-797.21			
converged:	True	LL-Null:	-995.19			
Covariance Type:	nonrobust	LLR p-value:	1.672e-85			
	coef	std err	z	P> z	[0.025	0.975]
const	-4.0025	0.132	-30.319	0.000	-4.261	-3.744
x2	2.5411	0.182	13.944	0.000	2.184	2.898
x3	0.3845	0.162	2.368	0.018	0.066	0.703
x8	0.1604	0.042	3.779	0.000	0.077	0.244

Hình 1.12: Mô hình sau khi các biến không có ý nghĩa thống kê

Dựa trên kết quả hồi quy logistic, dưới đây là phần nhận xét tổng hợp: Mô hình hồi quy logistic được sử dụng để dự đoán khả năng bị đột quỵ (stroke) với biến phụ thuộc là stroke. Số lượng quan sát trong mô hình là 5105, cho thấy dữ liệu đủ lớn để đảm bảo tính ổn định của kết quả. Giá trị Pseudo R-squared là 0.1989, nghĩa là mô hình chỉ giải thích được khoảng 19,89% sự biến thiên của khả năng bị đột quỵ. Điều này cho thấy rằng có thể còn nhiều yếu tố khác ngoài các biến độc lập đang được xem xét có ảnh hưởng đến biến phụ thuộc.

Tất cả các biến giải thích trong mô hình (X2, X3, và X8) đều có giá trị P-value dưới 0.05, chứng tỏ chúng có ý nghĩa thống kê ở mức ý nghĩa 5%. Cụ thể:

- Biến X2 có hệ số 2.5411, cho thấy mối quan hệ tích cực đáng kể với khả năng bị đột quỵ.
- Biến X3 có hệ số 0.3845, là biến có tác động mạnh nhất trong số các biến giải thích.
- Biến X8 có hệ số 0.1604, tác động yếu hơn nhưng vẫn có ý nghĩa thống kê.

Hệ số giao điểm (const) là -4.0025, thể hiện giá trị log-odds của khả năng bị đột quỵ khi tất cả các biến độc lập bằng 0. Các khoảng tin cậy 95% của các hệ số đều không bao gồm giá trị 0, củng cố thêm rằng các biến đều có ý nghĩa thống kê.

Tóm lại, mô hình này cho thấy các biến X2, X3, và X8 có tác động tích cực và ý nghĩa đối với khả năng xảy ra đột quỵ. Tuy nhiên, giá trị Pseudo R-squared thấp chỉ ra rằng mô hình còn hạn chế trong việc giải thích biến động của dữ liệu.

```

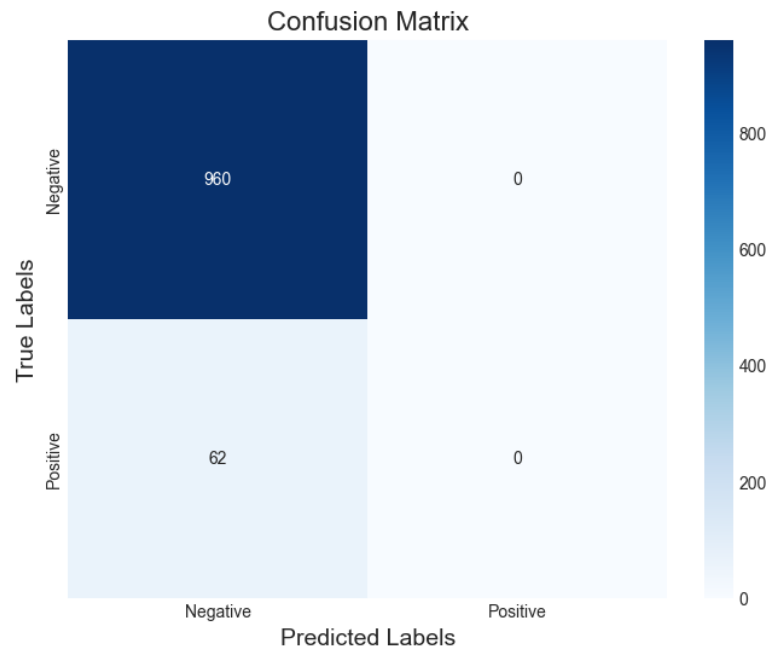
He so chan Intercept: [-3.99222363]
He so hoi quy ung voi tung dac trung Coefficients: [[2.42542491 0.35243037 0.15033073]]
Accuracy: 0.9393346379647749
Classification Report:

```

	precision	recall	f1-score	support
0	0.94	1.00	0.97	960
1	0.00	0.00	0.00	62
accuracy			0.94	1022
macro avg	0.47	0.50	0.48	1022
weighted avg	0.88	0.94	0.91	1022

*Hình 1.13: Kết quả huấn luyện mô hình*

Kết quả cho thấy mô hình có hệ số giao điểm (Intercept) là -3.99222363 và các hệ số hồi quy của các đặc trưng lần lượt là [2.42542491, 0.35243037, 0.15033073], phản ánh mức độ ảnh hưởng của các đặc trưng đến khả năng phân loại. Hiệu suất của mô hình được đánh giá qua chỉ số **ROC AUC** là 0.852436, cho thấy khả năng phân biệt giữa hai lớp ở mức khá. Độ chính xác tổng thể (Accuracy) đạt 94%, tuy nhiên, báo cáo phân loại cho thấy mô hình chỉ hoạt động tốt với lớp 0 (Precision: 0.94, Recall: 1.00, F1-score: 0.97) và hoàn toàn thất bại trong việc dự đoán lớp 1 (Precision, Recall, và F1-score đều là 0.00). Điều này chủ yếu do mất cân bằng dữ liệu, khi số lượng lớp 0 (960) vượt trội so với lớp 1 (62).



Hình 1.14: Confusion matrix

### 1.1.3 Nhận xét và kết luận

#### Nhận xét:

##### Hiệu suất mô hình:

- Mô hình đạt độ chính xác tổng thể (accuracy) cao, khoảng 94%, nhưng điều này có thể đánh lừa vì mô hình chủ yếu dự đoán tốt lớp 0 (không đột quỵ), trong khi hoàn toàn thất bại với lớp 1 (đột quỵ).
- precision, recall, và F1-score của lớp 1 đều bằng 0.00, cho thấy mô hình không nhận diện được bất kỳ trường hợp nào thuộc lớp 1.

##### Mất cân bằng dữ liệu:

- Biến stroke rất mất cân bằng, với 4861 giá trị thuộc lớp 0 (chiếm khoảng 95.1%) và 249 giá trị thuộc lớp 1 (chỉ 4.9%). Điều này khiến mô hình bị thiên lệch mạnh về lớp chiếm đa số (lớp 0), dẫn đến việc bỏ qua lớp thiểu số (lớp 1).

##### Ảnh hưởng của sự mất cân bằng:

- ROC AUC đạt 0.8524361, chỉ ra khả năng phân biệt giữa hai lớp của mô hình ở mức khá. Tuy nhiên, độ nhạy (recall) của lớp 1 bằng 0,

cho thấy mô hình không nhận diện được bất kỳ trường hợp nào thuộc lớp này.

- Hiệu suất tốt của lớp 0 (precision: 0.94, recall: 1.00) là do sự áp đảo về số lượng của lớp này trong dữ liệu.

### **Kết luận:**

- Nguyên nhân chính của vấn đề là sự mất cân bằng nghiêm trọng trong biên stroke, khi lớp 0 áp đảo so với lớp 1. Điều này khiến mô hình tập trung vào việc dự đoán lớp 0 và không học được đủ thông tin để nhận diện lớp 1.
- Mặc dù độ chính xác tổng thể cao, mô hình hiện tại không đáp ứng được yêu cầu dự đoán cả hai lớp. Kết quả này nhấn mạnh rằng độ chính xác không phải là chỉ số duy nhất để đánh giá hiệu quả của mô hình, đặc biệt khi dữ liệu mất cân bằng.

### **Đề xuất:**

- Cần xử lý mất cân bằng dữ liệu bằng các kỹ thuật như: Tăng cường dữ liệu lớp 1 (oversampling) bằng phương pháp SMOTE hoặc tạo dữ liệu tổng hợp. Giảm số lượng dữ liệu lớp 0 (undersampling) để cân bằng tỷ lệ các lớp. Sử dụng các thuật toán hỗ trợ dữ liệu mất cân bằng, như điều chỉnh trọng số lớp trong hàm mất mát.
- Đánh giá mô hình bằng các chỉ số bổ sung như F1-score, recall của lớp 1, và ROC AUC để có cái nhìn toàn diện hơn về hiệu quả dự đoán.

## 1.2 DỮ LIỆU 2: MÔ HÌNH HỒI QUY LOGISTIC

### 1.2.1 Giới thiệu bộ dữ liệu

Bộ dữ liệu "MAGIC Gamma Telescope" được lấy từ UCI Machine Learning Repository[2], một kho lưu trữ trực tuyến các bộ dữ liệu phục vụ nghiên cứu và học tập về học máy.

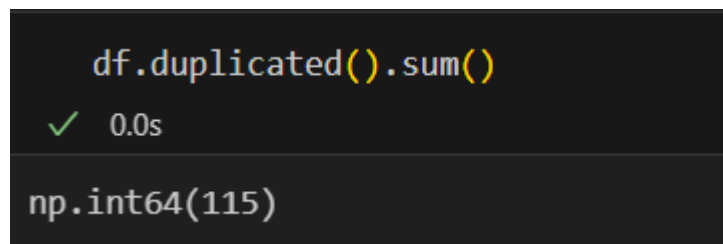
Bộ dữ liệu này được tạo ra bằng cách sử dụng chương trình mô phỏng Monte Carlo có tên Corsika, nhằm mô phỏng việc ghi nhận các hạt gamma năng lượng cao trong kính thiên văn Cherenkov khí quyển đặt trên mặt đất. Quá trình mô phỏng này giúp tạo ra các mẫu dữ liệu phản ánh đặc trưng của các sự kiện gamma và hadron, phục vụ cho việc phân loại và nghiên cứu trong lĩnh vực vật lý thiên văn.

Bộ dữ liệu bao gồm 19.020 mẫu với 10 thuộc tính liên tục và một thuộc tính phân loại (gồm hai lớp: gamma và hadron). Các thuộc tính này mô tả các đặc trưng hình học và cường độ của hình ảnh trận mưa, bao gồm:

- **fLength**: Trục lớn của hình ellipse (mm). Đặc trưng này đo độ dài của đối tượng theo chiều dài nhất.
- **fWidth**: Trục nhỏ của hình ellipse (mm). Đo chiều rộng của đối tượng theo chiều ngắn nhất.
- **fSize**: Logarithm cơ số 10 của tổng giá trị ánh sáng trên toàn bộ các pixel (số photon). Đặc trưng này đo tổng năng lượng ánh sáng mà đối tượng phát ra.
- **fConc**: Tỷ lệ của tổng hai pixel sáng nhất trên tổng giá trị ánh sáng (fSize). Đo mức độ tập trung ánh sáng tại khu vực sáng nhất.
- **fConc1**: Tỷ lệ của pixel sáng nhất trên tổng giá trị ánh sáng (fSize). Chi tiết hóa mức độ tập trung tại pixel sáng nhất.
- **fAsym**: Khoảng cách từ pixel sáng nhất đến tâm hình ellipse, chiếu lên trục lớn (mm). Đo sự bất đối xứng trong phân bố ánh sáng.

- **fM3Long**: Căn bậc ba của moment bậc ba dọc theo trục lớn (mm). Đo sự lệch trong cách ánh sáng phân bố theo chiều dài.
- **fM3Trans**: Căn bậc ba của moment bậc ba dọc theo trục nhỏ (mm). Đo sự lệch theo hướng ngang của ánh sáng.
- **fAlpha**: Góc giữa trục lớn của hình ellipse và vector hướng đến gốc tọa độ (độ). Đo hướng của đối tượng trong không gian.
- **fDist**: Khoảng cách từ gốc tọa độ đến tâm của hình ellipse (mm). Đo vị trí tổng thể của đối tượng so với gốc tọa độ.
- **class**: Nhãn phân loại của đối tượng: g gamma tín hiệu, h: hadron nhiễu nền.

Trong quá trình kiểm tra chất lượng dữ liệu, chúng tôi đã tiến hành rà soát toàn bộ bộ dữ liệu để xác định sự tồn tại của các dòng trùng lặp. Kết quả phân tích cho thấy có tổng cộng 115 dòng dữ liệu bị trùng lặp. Những dòng trùng lặp này có thể gây ảnh hưởng tiêu cực đến các bước phân tích và mô hình hóa sau này, làm giảm độ chính xác và hiệu quả của dự án. Do đó, để đảm bảo tính toàn vẹn và chất lượng dữ liệu, chúng tôi đã quyết định loại bỏ toàn bộ các dòng dữ liệu trùng lặp này. Sau khi thực hiện quá trình loại bỏ, bộ dữ liệu sẽ được kiểm tra lại để xác nhận rằng không còn bất kỳ trùng lặp nào, giúp duy trì độ tin cậy và sẵn sàng cho các bước xử lý tiếp theo. Bộ dữ liệu còn lại 18905 quan trắc.



```
df.duplicated().sum()
✓ 0.0s
np.int64(115)
```

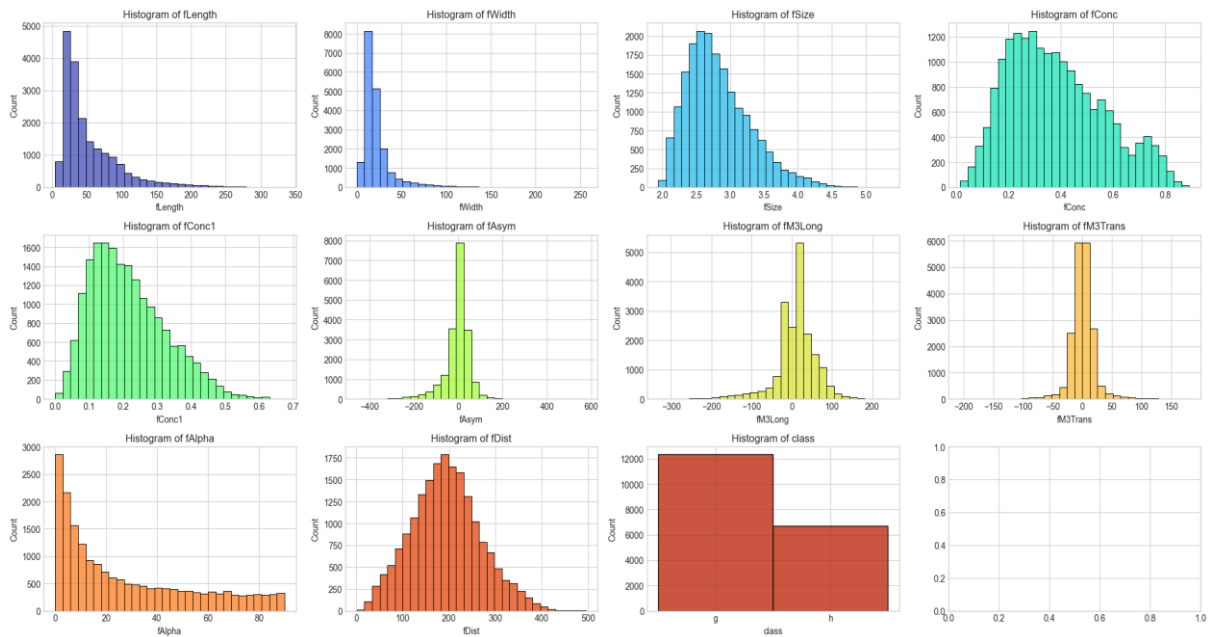
*Hình 2.1: Dữ liệu trùng lặp*

Bộ dữ liệu gồm 18.905 quan trắc và 11 cột, cung cấp thông tin cần thiết cho việc phân tích. Hình minh họa dưới đây hiển thị một số quan trắc đầu tiên, giúp tổng quan về cấu trúc và nội dung dữ liệu.

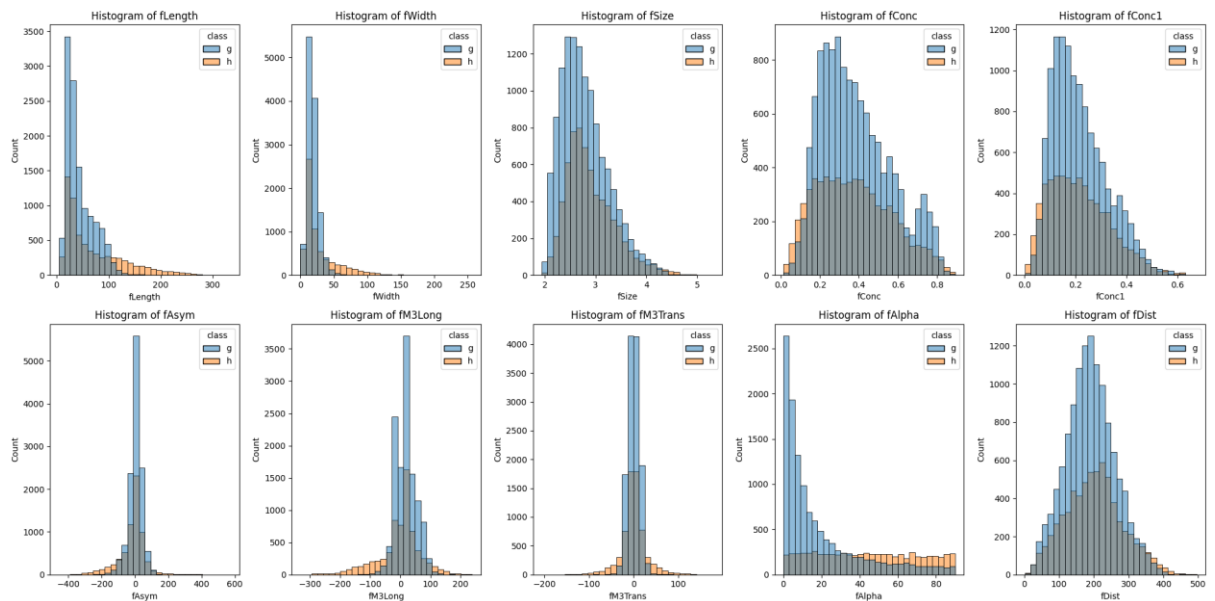
	fLength	fWidth	fSize	fConc	fConc1	fAsym	fM3Long	fM3Trans	fAlpha	fDist	class
0	28.7967	16.0021	2.6449	0.3918	0.1982	27.7004	22.0110	-8.2027	40.0920	81.8828	g
1	31.6036	11.7235	2.5185	0.5303	0.3773	26.2722	23.8238	-9.9574	6.3609	205.2610	g
2	162.0520	136.0310	4.0612	0.0374	0.0187	116.7410	-64.8580	-45.2160	76.9600	256.7880	g
3	23.8172	9.5728	2.3385	0.6147	0.3922	27.2107	-6.4633	-7.1513	10.4490	116.7370	g
4	75.1362	30.9205	3.1611	0.3168	0.1832	-5.5277	28.5525	21.8393	4.6480	356.4620	g

Hình 2.2: Một vài quan trắc đầu tiên và số chiều của bộ dữ liệu

Phân bố ban đầu của các biến và trung bình tổng của từng biến theo biến phụ thuộc class.

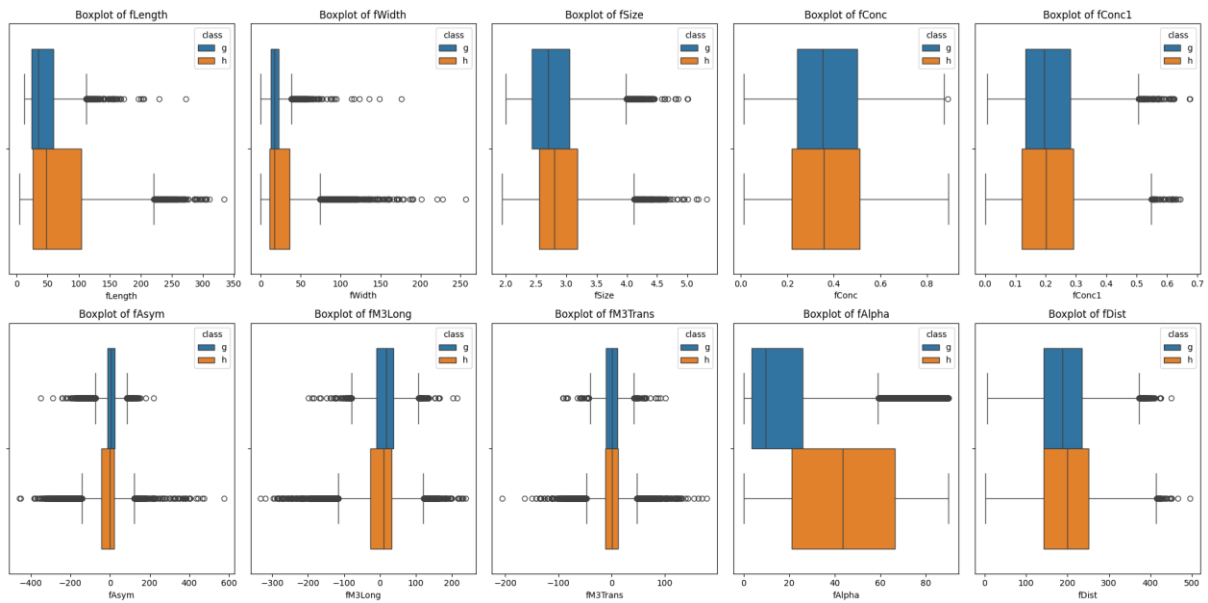


Hình 2.3: Phân bố ban đầu của các biến



Hình 2.4: Phân bố của các biến so với biến phụ thuộc class.





Hình 2.5: Biểu đồ boxplot của các biến so với biến phụ thuộc class.

Quan sát các biểu đồ trên chúng tôi nhận thấy một số điều như sau:

**fLength:** Biến này có phân bố lệch phải với phần lớn giá trị tập trung ở mức thấp. Sự phân biệt giữa hai lớp "g" và "h" thể hiện rõ ràng trên boxplot, với lớp "g" có giá trị trung bình lớn hơn lớp "h".

**fWidth:** Phân bố của biến này cũng lệch phải, tương tự như fLength, với phần lớn giá trị nhỏ. Boxplot cho thấy lớp "g" có phân bố rộng hơn và giá trị trung bình cao hơn lớp "h".

**fSize:** Biến này có phân bố gần chuẩn, tập trung trong khoảng 2.5–4.0. Trên boxplot, lớp "g" và "h" được phân tách khá rõ, với lớp "g" có giá trị trung bình cao hơn.

**fConc:** Phân phối của biến này gần chuẩn, với phần lớn giá trị trong khoảng 0.2–0.6. Boxplot cho thấy lớp "g" và "h" chồng lấn khá nhiều, nhưng lớp "g" có xu hướng có giá trị trung bình cao hơn một chút.

**fConc1:** Phân bố của biến này tương tự fConc, với giá trị tập trung quanh 0.2–0.4. Boxplot thể hiện sự chồng lấn đáng kể giữa hai lớp, không rõ ràng để phân biệt.

**fAsym**: Biến này có phân phối đối xứng quanh giá trị 0 nhưng với nhiều giá trị ngoại lai ở cả hai phía. Boxplot cho thấy sự chồng lấn gần như hoàn toàn giữa hai lớp, làm giảm giá trị phân biệt.

**fM3Long**: Phân phối tương tự fAsym, với tập trung quanh giá trị 0 và nhiều ngoại lai. Sự chồng lấn giữa hai lớp trên boxplot là rất lớn, khó có thể phân biệt rõ ràng.

**fM3Trans**: Biến này có phân phối gần chuẩn, tập trung quanh giá trị 0. Boxplot cho thấy sự khác biệt giữa hai lớp là rất nhỏ, với giá trị trung bình tương đương nhau.

**fAlpha**: Biến này có phân phối lệch phải, với phần lớn giá trị ở mức thấp. Boxplot cho thấy lớp "g" có giá trị nhỏ hơn lớp "h", thể hiện sự phân tách tương đối tốt giữa hai lớp.

**fDist**: Phân phối của biến này gần chuẩn, với giá trị tập trung quanh 200. Boxplot cho thấy sự phân tách khá tốt giữa lớp "g" và "h", với lớp "g" có giá trị trung bình lớn hơn.

Nhận xét tổng quan: Những đặc trưng như **fSize**, **fDist**, và **fAlpha** có tiềm năng phân biệt tốt giữa hai lớp. Tuy nhiên, một số đặc trưng như **fAsym**, **fM3Long**, và **fM3Trans** có sự chồng lấn lớn giữa hai lớp, làm giảm hiệu quả phân loại. Cần xử lý ngoại lai và cân nhắc kỹ khi lựa chọn đặc trưng cho mô hình phân loại.

### *1.2.2 Phân tích và chọn mô hình*

Chúng tôi chọn mô hình Logistic Regression cho bài toán phân loại tín hiệu gamma và nhiễu hadron.

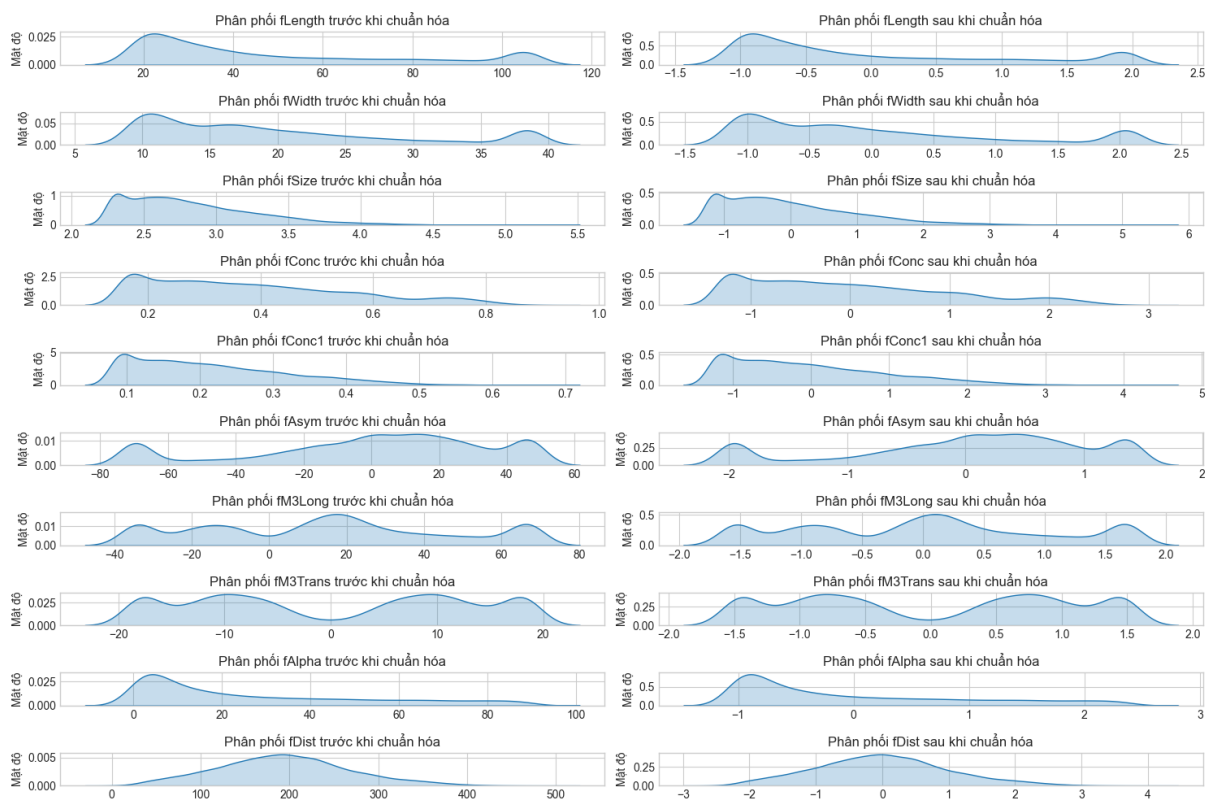
Trong bài toán phân loại tín hiệu gamma và nhiễu hadron, việc giữ lại tất cả các giá trị ngoại lai là cần thiết vì chúng mang ý nghĩa vật lý quan trọng và có thể đại diện cho các sự kiện năng lượng cao từ thiên văn. Các giá trị ngoại lai như **fLength**, **fWidth**, và **fSize** phản ánh kích thước, độ sáng, và

tổng năng lượng của sự kiện, giúp phân biệt tín hiệu gamma với nhiễu. Đặc biệt, chúng có thể đại diện cho các hiện tượng mạnh mẽ như **vụ nổ siêu tân tinh**, **bức xạ gamma từ hố đen siêu lớn**, hoặc **sự hợp nhất sao neutron**, cung cấp thông tin quý giá về vũ trụ. Loại bỏ ngoại lai sẽ làm mất thông tin quan trọng và giảm khả năng mô hình nhận diện chính xác. Do đó, các giá trị này nên được giữ lại, nhưng cần được kiểm soát qua chuẩn hóa hoặc giới hạn giá trị để đảm bảo hiệu suất mô hình.

Tiến hành chuẩn hoá dữ liệu trước khi tiến hành đưa bộ dữ liệu vào mô hình hồi quy. Sử dụng phương pháp RobustScaler để bỏ qua outliers.

```
# chuẩn hoá bằng robust scaler
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
df.iloc[:, :-1] = scaler.fit_transform(df.iloc[:, :-1])
df.head()
```

Hình 2.6: Chuẩn hoá RobustScaler.



Hình 2.7: Biểu đồ KDE cho từng cột trước và sau khi chuẩn hóa.

```

Optimization terminated successfully.
Current function value: 0.455592
Iterations 7

```

Logit Regression Results						
=====						
Dep. Variable:	class	No. Observations:	18905			
Model:	Logit	Df Residuals:	18894			
Method:	MLE	Df Model:	10			
Date:	Sat, 04 Jan 2025	Pseudo R-squ.:	0.2948			
Time:	10:55:06	Log-Likelihood:	-8613.0			
converged:	True	LL-Null:	-12213.			
Covariance Type:	nonrobust	LLR p-value:	0.000			
=====						
	coef	std err	z	P> z	[0.025	0.975]
-----						
const	-1.8102	0.028	-63.889	0.000	-1.866	-1.755
x1	1.3525	0.048	27.953	0.000	1.258	1.447
x2	0.0699	0.032	2.209	0.027	0.008	0.132
x3	0.3967	0.060	6.591	0.000	0.279	0.515
x4	0.0010	0.140	0.007	0.995	-0.274	0.276
x5	0.8266	0.119	6.945	0.000	0.593	1.060
x6	0.0035	0.019	0.179	0.858	-0.034	0.041
x7	-0.3492	0.026	-13.439	0.000	-0.400	-0.298
x8	-0.0125	0.025	-0.500	0.617	-0.062	0.037
x9	1.8153	0.034	52.797	0.000	1.748	1.883
x10	0.0534	0.030	1.800	0.072	-0.005	0.112
=====						

Hình 2.8: Mô hình ban đầu

Tiến hành sử dụng phương pháp tính hệ số VIF để kiểm tra hiện tượng đa cộng tuyến trong mô hình này. Kết quả cho thấy ở x3, x4, x5 ( $fSize$ ,  $fConc$ ,  $fConc1$ ) lớn hơn 5 chứng tỏ tồn tại hiện tượng đa cộng tuyến.

	feature	VIF
0	x1	3.361700
1	x2	3.489899
2	x3	5.082901
3	x4	28.056350
4	x5	22.969670
5	x6	1.284958
6	x7	1.247024
7	x8	1.002697
8	x9	1.341576
9	x10	1.356828

Hình 2.9: Kết quả VIF

Ta tiến hành loại bỏ các đa cộng tuyến như sau. Loại bỏ từ từ từng giá trị đa cộng tuyến sau đó xem xét các đặc trưng có ý nghĩa thống kê hay không.

```
X_1 = pd.DataFrame({'x1': x1, 'x2': x2, 'x3': x3, 'x5': x5, 'x6': x6, 'x7': x7, 'x8': x8, 'x9': x9, 'x10': x10})
print(calculate_vif(X_1))
```

	feature	VIF
0	x1	3.354834
1	x2	3.488598
2	x3	4.430091
3	x5	2.605865
4	x6	1.284953
5	x7	1.246189
6	x8	1.002685
7	x9	1.326874
8	x10	1.355750

Hình 2.10: loại bỏ x4 (fConc) có giá trị VIF cao nhất 28.05

Sau khi loại bỏ x4 xong tiến hành kiểm tra lại kết quả VIF lúc này các chỉ số VIF đều dưới 5 không còn hiện tượng đa cộng tuyến xảy ra nữa. Tuy nhiên để chính xác hơn cần phải xem xét thêm các đặc trưng có ý nghĩa thống kê hay không.  $(P > |z|) < 0.05$ . Sau khi xem xét và loại bỏ từ từ thì chúng tôi phát hiện ra x6, x8, x10 đại diện cho fAsym, fM3Trans, fDist có  $(P > |z|) > 0.05$  không có ý nghĩa thống kê nên loại bỏ.

Cuối cùng chúng tôi giữ lại các thuộc tính sau fLength, fWidth, fSize, fConcl, fM3Long, fAlpha đưa vào mô hình hồi quy logistic và thu được bảng thông số và kết quả sau.

```
Optimization terminated successfully.
Current function value: 0.455685
Iterations 7
```

Logit Regression Results						
Dep. Variable:	class	No. Observations:	18905			
Model:	Logit	Df Residuals:	18898			
Method:	MLE	Df Model:	6			
Date:	Sat, 04 Jan 2025	Pseudo R-squ.:	0.2946			
Time:	11:37:36	Log-Likelihood:	-8614.7			
converged:	True	LL-Null:	-12213.			
Covariance Type:	nonrobust	LLR p-value:	0.000			
	coef	std err	z	P> z	[0.025	0.975]
const	-1.8147	0.028	-65.362	0.000	-1.869	-1.760
x1	1.3740	0.045	30.345	0.000	1.285	1.463
x2	0.0692	0.032	2.184	0.029	0.007	0.131
x3	0.4140	0.055	7.505	0.000	0.306	0.522
x5	0.8409	0.045	18.565	0.000	0.752	0.930
x7	-0.3509	0.025	-13.872	0.000	-0.400	-0.301
x9	1.8025	0.033	53.830	0.000	1.737	1.868

*Hình 2.11: Mô hình sau khi loại bỏ đa cộng tuyến và các biến không có ý nghĩa thống kê*

Dựa vào bảng kết quả của mô hình Logit Regression trên, dưới đây là những nhận xét chi tiết:

Kết quả từ mô hình Logit Regression cho thấy biến phụ thuộc là class, với tổng số quan sát là 18,905. Mô hình được ước lượng bằng phương pháp Maximum Likelihood Estimation (MLE) và có giá trị Pseudo R-squared là 0.2946, nghĩa là mô hình giải thích được khoảng 29.46% biến thiên trong dữ liệu. Kết quả kiểm định tổng thể của mô hình, thông qua giá trị Log-Likelihood (-8614.7) và LLR p-value (0.000), khẳng định rằng mô hình có ý nghĩa thống kê ở mức ý nghĩa 5%.

Hệ số hồi quy của các biến độc lập như x1, x2, x3, x5, x7, và x9 đều có giá trị p-value nhỏ hơn 0.05, cho thấy các biến này có ý nghĩa thống kê trong việc giải thích biến phụ thuộc. Hệ số x1=1.3740 cho thấy khi x1 tăng thêm 1 đơn vị, log-odds (log của tỷ lệ xác suất) tăng lên 1.374, trong khi hệ số x7=-0.3509 phản ánh rằng khi x7 tăng thêm 1 đơn vị, log-odds giảm đi 0.3509. Đặc biệt, biến x9=1.8025 có ảnh hưởng mạnh nhất đến log-odds.

```

He so chan Intercept: [-1.79953135]
He so hoi quy ung voi tung dac trung Coefficients: [[ 1.38892417  0.05838024  0.38214339  0.82745869 -0.35185517  1.79908189]]
ROC AUC: 0.8421793704979545
False Positive Rate: [0.          0.          0.          ... 0.99879324 0.99879324 1.          ]
True Positive Rate: [0.00000000e+00 7.72200772e-04 1.12741313e-01 ... 9.99227799e-01
1.00000000e+00 1.00000000e+00]
Thresholds: [      inf 0.99996841 0.98308772 ... 0.0470001  0.046946  0.04252942]
Accuracy: 0.7884157630256546
Classification Report:

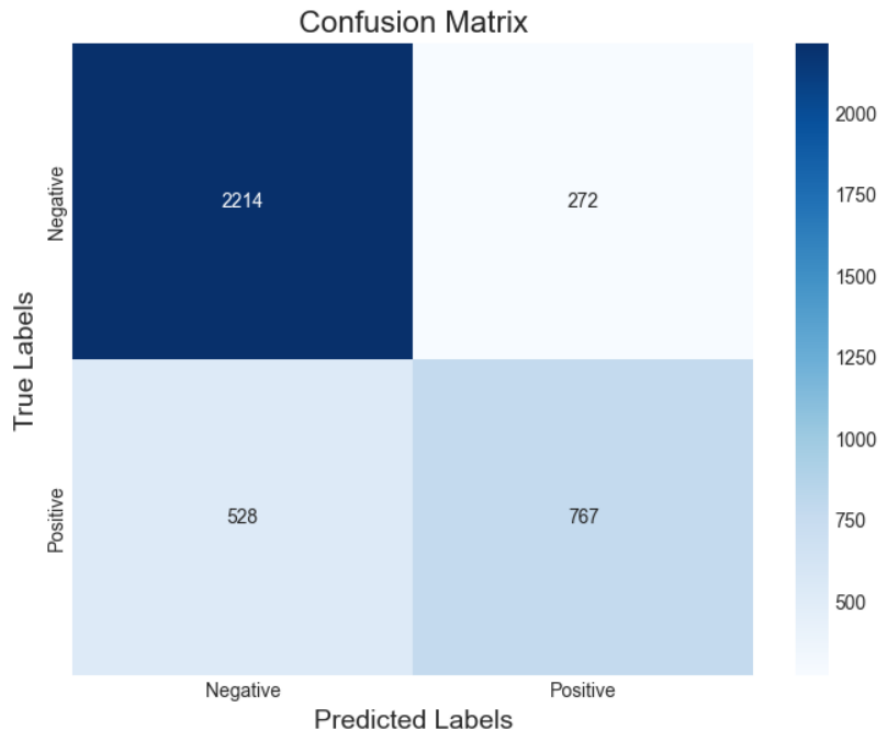
```

	precision	recall	f1-score	support
0	0.81	0.89	0.85	2486
1	0.74	0.59	0.66	1295
accuracy			0.79	3781
macro avg	0.77	0.74	0.75	3781
weighted avg	0.78	0.79	0.78	3781

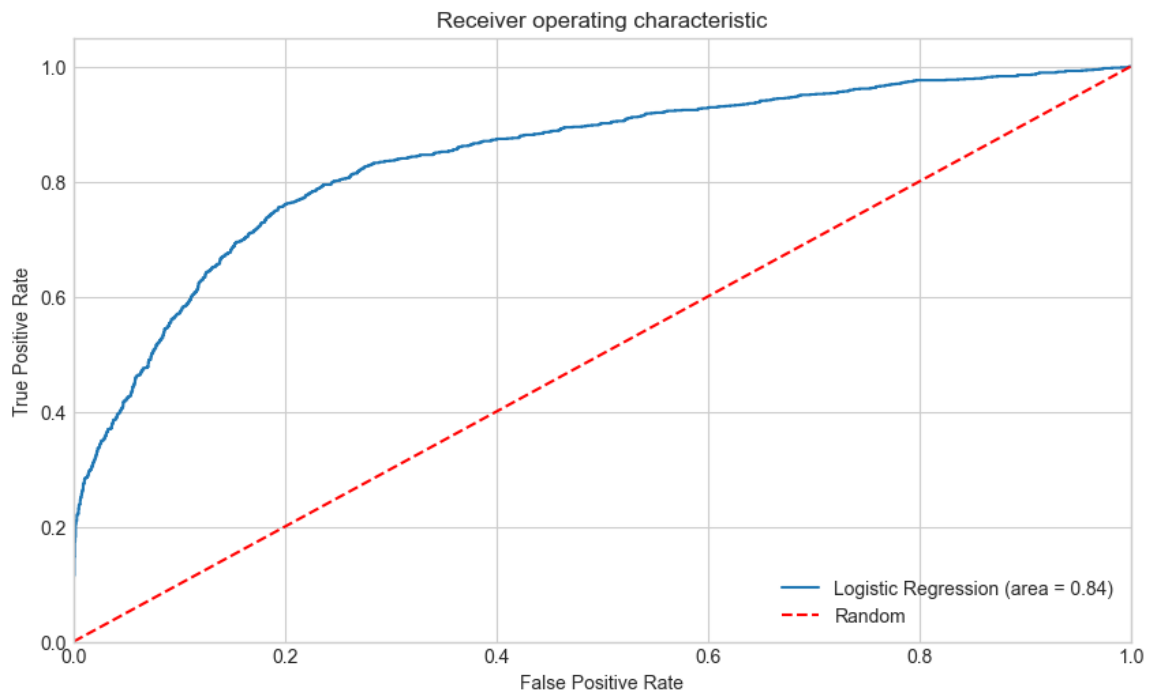
*Hình 2.12: Kết quả huấn luyện mô hình*

Kết quả cho thấy mô hình phân loại đạt độ chính xác (accuracy) 78.84%, với AUC-ROC là 0.842, chứng tỏ mô hình có khả năng phân biệt tốt giữa hai lớp. Trong báo cáo phân loại, lớp 0 có hiệu suất tốt hơn với precision

81%, recall 89%, và F1-score 85%, trong khi lớp 1 có precision 74%, recall 59%, và F1-score 66%. Điều này cho thấy mô hình hoạt động tốt hơn trong việc phân loại các trường hợp thuộc lớp 0, nhưng khả năng nhận diện các trường hợp lớp 1 còn hạn chế.



Hình 2.13: Confusion Matrix



Hình 2.14: Biểu đồ đường cong ROC

Đường cong ROC nằm trên đường chéo (Random Guess) và cong về góc trái trên, cho thấy mô hình phân biệt tốt giữa hai lớp.

AUC lớn hơn 0.8 chứng tỏ mô hình có hiệu suất phân loại khá tốt. AUC này thể hiện xác suất rằng mô hình sẽ xếp hạng một mẫu thuộc lớp gamma cao hơn một mẫu thuộc lớp hadron.

Đường cong gần như tiếp cận góc trái trên của biểu đồ, điều này thể hiện rằng mô hình đạt TPR cao (tỷ lệ phân loại đúng tín hiệu gamma) trong khi vẫn giữ FPR thấp (ít nhiều bị nhầm là tín hiệu).

### *1.2.3 Nhận xét và kết luận*

#### **Điểm mạnh:**

- Pseudo R-squared = 0.2946: Mức này nằm trong khoảng 0.2 - 0.4, được coi là tốt đối với mô hình hồi quy logistic. Nó cho thấy mô hình giải thích được khoảng 29.46% sự biến thiên của biến mục tiêu.
- AUC = 0.84: Giá trị này cao, cho thấy mô hình có khả năng phân biệt tốt giữa hai lớp (class 0 và class 1).
- Độ chính xác (Accuracy) = 78.84%: Đây là một mức khá tốt, đặc biệt khi dữ liệu có tính phức tạp.
- Tất cả các biến trong mô hình đều có ý nghĩa thống kê (P-value < 0.05), điều này làm cho mô hình có giá trị trong việc phân tích ý nghĩa của các biến.

#### **Hạn chế:**

- Recall thấp (59%) ở lớp 1: Điều này cho thấy mô hình bỏ sót khá nhiều giá trị thực sự thuộc lớp 1 (False Negative = 528).
- Hiệu suất không cân bằng giữa lớp 0 và lớp 1: Mô hình hoạt động tốt hơn đáng kể ở lớp 0 so với lớp 1, dẫn đến sự thiên lệch trong kết quả.



- Pseudo R-squared = 0.2946: Mặc dù mức này là tốt cho logistic regression, nhưng nó cũng chỉ ra rằng mô hình chỉ giải thích được một phần nhỏ biến thiên của dữ liệu.

### **Kết luận:**

Mô hình hồi quy logistic là phù hợp ở mức cơ bản, đặc biệt khi muốn phân tích ý nghĩa của các biến độc lập hoặc cần một mô hình đơn giản để thực hiện phân loại. Tuy nhiên, nếu mục tiêu chính là dự đoán chính xác hơn, đặc biệt là đối với lớp 1, mô hình logistic chưa phải là lựa chọn tối ưu.

### **Đề xuất:**

Tiếp tục dùng hồi quy logistic:

- Sử dụng các kỹ thuật xử lý mất cân bằng dữ liệu như class weight, oversampling (SMOTE) hoặc undersampling để cải thiện Recall cho lớp 1.
- Bổ sung thêm các biến giải thích (feature engineering) để cải thiện Pseudo R-squared.

Thử nghiệm các mô hình phi tuyến:

- Random Forest hoặc Gradient Boosting (XGBoost, LightGBM): Các mô hình này thường đạt hiệu suất dự đoán cao hơn, đặc biệt khi dữ liệu mất cân bằng.
- SVM với kernel phi tuyến: Thích hợp nếu dữ liệu không quá lớn.

## 1.3 DỮ LIỆU 3: HỒI QUY THÀNH PHẦN CHÍNH

### 1.3.1 Giới thiệu bộ dữ liệu

Bộ dữ liệu “**Airline Passenger Satisfaction**” được tìm thấy trên Kaggle[3], một cộng đồng trực tuyến về khoa học dữ liệu và học máy. Bộ dữ liệu này chứa một cuộc khảo sát mức độ hài lòng của hành khách hàng không. Bộ dữ liệu này chứa **103,904 mẫu** và **25 đặc trưng**, phản ánh kết quả khảo sát mức độ hài lòng của hành khách hàng không. Các đặc trưng bao gồm thông tin cá nhân như giới tính, độ tuổi, loại khách hàng; thông tin về chuyến bay như khoảng cách, thời gian trễ; và các đánh giá dịch vụ như wifi, thức ăn, giải trí, độ thoải mái, và vệ sinh. Chi tiết như sau:

Tên đặc trưng	Tên tiếng Việt	Mô tả
id	Mã định danh	Mã định danh duy nhất cho mỗi hành khách.
Gender	Giới tính	Giới tính của hành khách (Nam hoặc Nữ).
Customer Type	Loại khách hàng	Phân loại khách hàng: Khách hàng trung thành hoặc không trung thành.
Age	Tuổi	Tuổi của hành khách.
Type of Travel	Loại chuyến đi	Mục đích chuyến đi: Cá nhân hay Công tác.
Class	Hạng ghế	Hạng ghế: Hạng Thương gia, Hạng Phổ thông đặc biệt, Hạng Phổ thông.
Flight Distance	Khoảng cách chuyến bay	Khoảng cách của chuyến bay (đậm).
Inflight wifi service	Dịch vụ wifi trên máy bay	Mức độ hài lòng với dịch vụ wifi trên máy bay (1-5).
Departure/Arrival time convenient	Thời gian khởi hành/đến thuận tiện	Mức độ hài lòng với sự thuận tiện của thời gian khởi hành/đến (1-5).
Ease of Online booking	Dễ dàng đặt vé trực tuyến	Mức độ hài lòng với việc đặt vé trực tuyến (1-5).
Gate location	Vị trí cổng lên máy bay	Mức độ hài lòng với vị trí cổng lên máy bay (1-5).
Food and drink	Thức ăn và đồ uống	Mức độ hài lòng với chất lượng thức ăn và đồ uống (1-5).
Online boarding	Lên máy bay trực tuyến	Mức độ hài lòng với quy trình lên máy bay trực tuyến (1-5).
Seat comfort	Sự thoải mái của ghế ngồi	Mức độ hài lòng với sự thoải mái của ghế ngồi (1-5).
Inflight entertainment	Giải trí trên máy bay	Mức độ hài lòng với các dịch vụ giải trí trên máy bay (1-5).
On-board service	Dịch vụ trên máy bay	Mức độ hài lòng với dịch vụ trên máy bay (1-5).
Leg room service	Dịch vụ chỗ để chân	Mức độ hài lòng với không gian để chân (1-5).
Baggage handling	Xử lý hành lý	Mức độ hài lòng với việc xử lý hành lý (1-5).
Check-in service	Dịch vụ làm thủ tục	Mức độ hài lòng với dịch vụ làm thủ tục (1-5).
Inflight service	Dịch vụ trong chuyến bay	Mức độ hài lòng với dịch vụ trong chuyến bay (1-5).
Cleanliness	Sự sạch sẽ	Mức độ hài lòng với sự sạch sẽ của máy bay (1-5).
Departure Delay in Minutes	Thời gian trễ khởi hành (phút)	Thời gian trễ khi khởi hành tính bằng phút.
Arrival Delay in Minutes	Thời gian trễ đến (phút)	Thời gian trễ khi đến tính bằng phút.
satisfaction	Sự hài lòng	Mức độ hài lòng tổng thể của hành khách (Hài lòng hoặc Không hài lòng).

Hình 3.1: Mô tả dữ liệu

Trong quá trình kiểm tra chất lượng dữ liệu, chúng tôi đã tiến hành rà soát toàn bộ bộ dữ liệu để xác định sự tồn tại của các dòng trùng lặp. Kết quả cho thấy không có hiện tượng trùng lặp dữ liệu trong bộ dữ liệu này.

```
print(f'Số dữ liệu bị trùng là: {df_airplane.duplicated().sum()}')  
✓ 0.0s  
Số dữ liệu bị trùng là: 0
```

Hình 3.2: Dữ liệu trùng lặp

Sau khi rà soát trùng lặp thì chúng tôi tiếp tục tiến hành rà soát toàn bộ dữ liệu để kiểm tra xem có dữ liệu thiếu hay không. Chúng tôi thu được kết quả sau. Trong bộ dữ liệu chỉ có cột Arrival Delay in Minutes bị thiếu 310 giá trị.

```
# kiểm tra null
for col in df_airplane.columns:
    nan_col = df_airplane[col].isna().sum()
    print(f'Cột {col} có số dữ liệu NaN là: {nan_col}')
```

✓ 0.0s

Cột Gender có số dữ liệu NaN là: 0  
 Cột Customer Type có số dữ liệu NaN là: 0  
 Cột Age có số dữ liệu NaN là: 0  
 Cột Type of Travel có số dữ liệu NaN là: 0  
 Cột Class có số dữ liệu NaN là: 0  
 Cột Flight Distance có số dữ liệu NaN là: 0  
 Cột Inflight wifi service có số dữ liệu NaN là: 0  
 Cột Departure/Arrival time convenient có số dữ liệu NaN là: 0  
 Cột Ease of Online booking có số dữ liệu NaN là: 0  
 Cột Gate location có số dữ liệu NaN là: 0  
 Cột Food and drink có số dữ liệu NaN là: 0  
 Cột Online boarding có số dữ liệu NaN là: 0  
 Cột Seat comfort có số dữ liệu NaN là: 0  
 Cột Inflight entertainment có số dữ liệu NaN là: 0  
 Cột On-board service có số dữ liệu NaN là: 0  
 Cột Leg room service có số dữ liệu NaN là: 0  
 Cột Baggage handling có số dữ liệu NaN là: 0  
 Cột Checkin service có số dữ liệu NaN là: 0  
 Cột Inflight service có số dữ liệu NaN là: 0  
 Cột Cleanliness có số dữ liệu NaN là: 0  
 Cột Departure Delay in Minutes có số dữ liệu NaN là: 0  
 Cột Arrival Delay in Minutes có số dữ liệu NaN là: 310  
 Cột satisfaction có số dữ liệu NaN là: 0

Hình 3.3: Dữ liệu thiếu

Bộ dữ liệu gồm 103,904 mẫu và 25 cột, cung cấp thông tin cần thiết cho việc phân tích. Hình minh họa dưới đây hiển thị một số mẫu đầu tiên, giúp tổng quan về cấu trúc và nội dung dữ liệu.

```
df_airplane.iloc[:, 0:12].head()
```

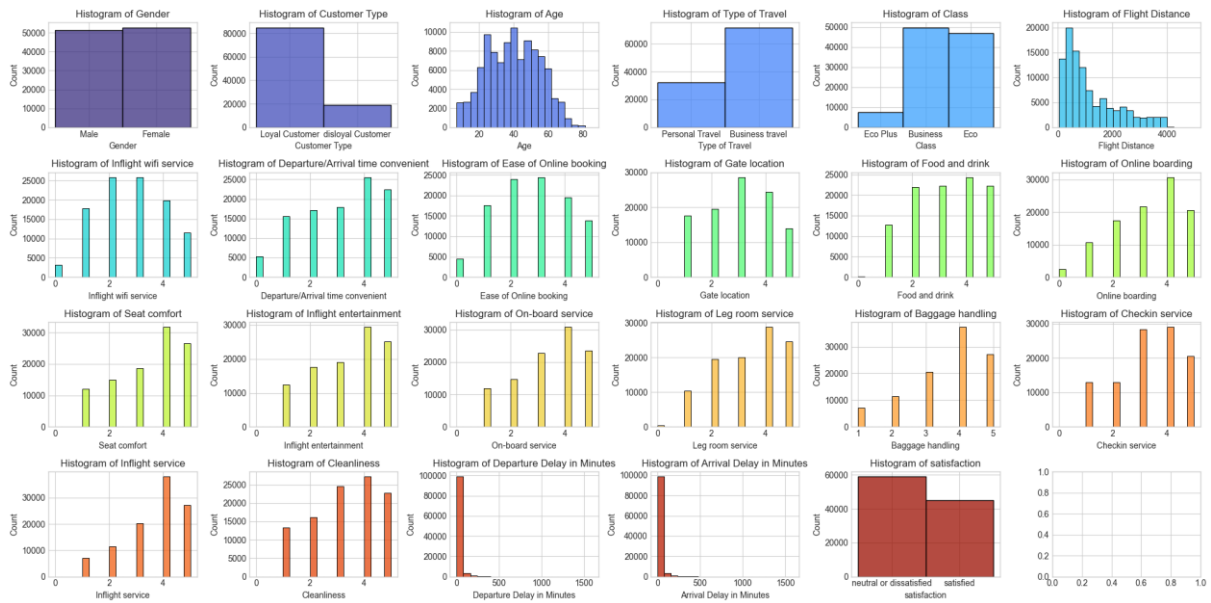
✓ 0.0s

	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding
0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	1	5	3
1	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	3	1	3
2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	2	5	5
3	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	5	2	2
4	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	3	4	5

Hình 3.4: Một vài mẫu đầu tiên.

### 1.3.2 Phân tích và chọn mô hình

Phân bố ban đầu của các biến và trung bình tổng của từng biến theo biến phụ thuộc satisfaction.



Hình 3.5: Phân bố ban đầu của các biến



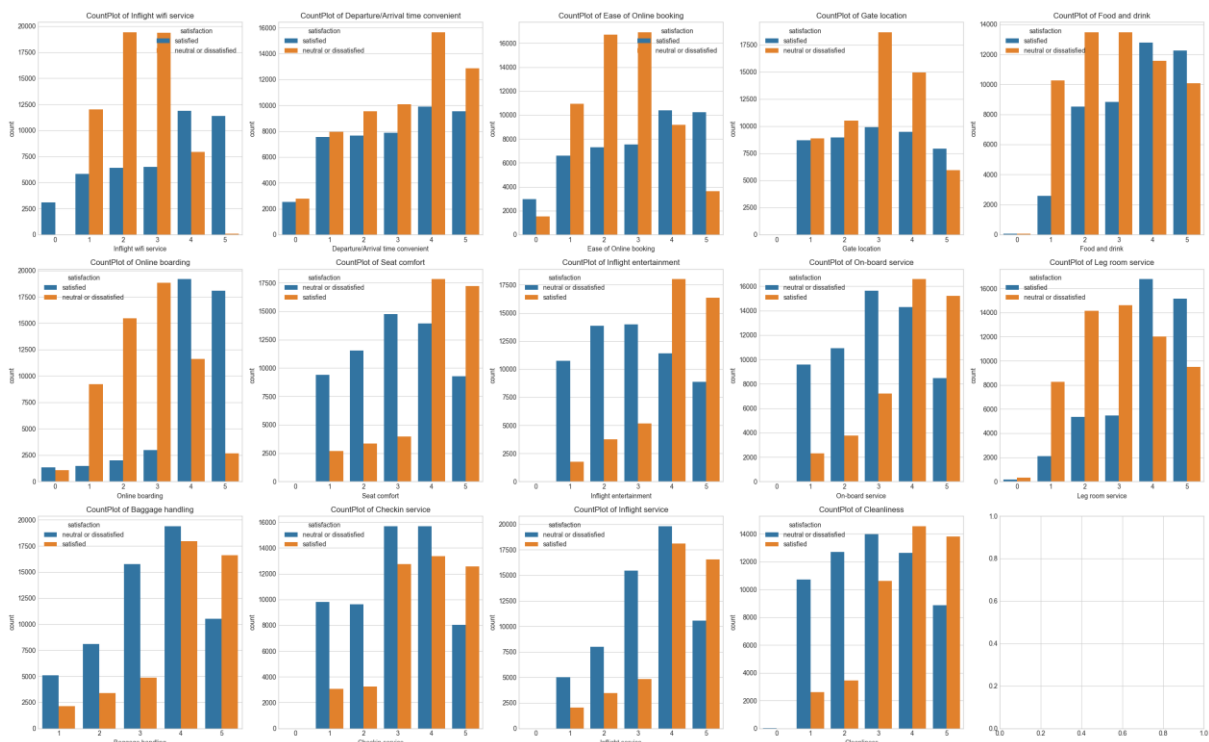
Hình 3.6: Phân bố của các biến so với biến phụ thuộc satisfaction.

Quan sát các biểu đồ trên chúng tôi nhận thấy một số điều như sau:

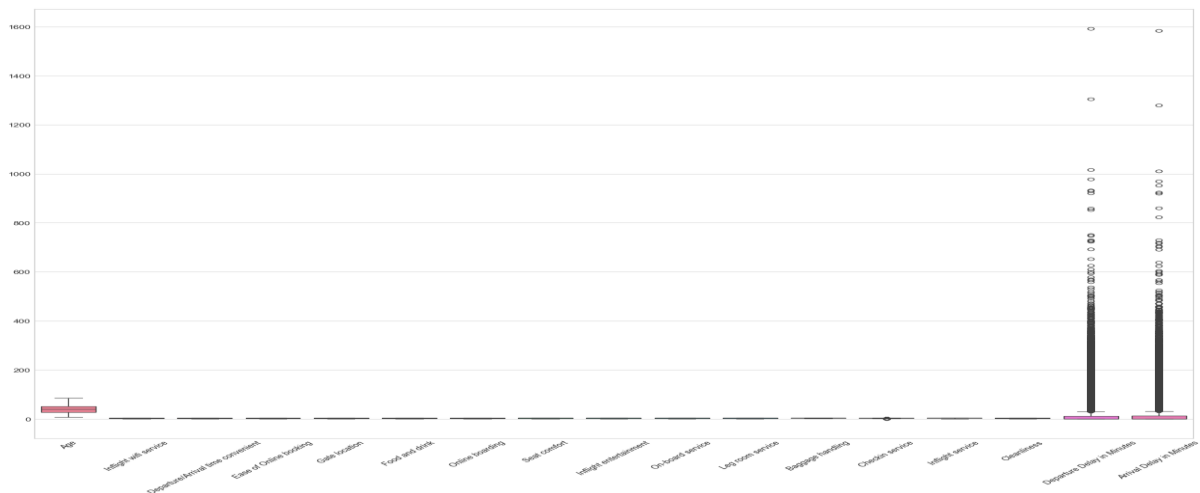
- Giới tính: Số lượng nam và nữ gần như bằng nhau, cho thấy hãng phục vụ đồng đều hai giới.

- Loại khách hàng: Khách hàng trung thành chiếm tỷ lệ lớn, chúng tôi hãy giữ chân khách hiệu quả.
- Tuổi: Hành khách chủ yếu từ 20–50 tuổi, tập trung ở nhóm 30–40. Nhóm trẻ (<20) và lớn tuổi (>60) chiếm tỷ lệ nhỏ.
- Loại chuyến đi: Đa số là công tác, ít chuyến đi cá nhân. Hãng có thể cải thiện để thu hút khách cá nhân.
- Hạng ghế: Business Class là phân khúc chính; Eco Class phổ biến nhưng thấp hơn. Eco Plus cần chiến lược để thu hút khách.
- Khoảng cách chuyến bay: Chủ yếu dưới 2.000 dặm, tập trung vào chặng ngắn và trung bình.
- Wifi trên máy bay: Đánh giá phân tán, cần cải thiện chất lượng để đáp ứng đồng đều hơn.
- Thời gian khởi hành/đến: Nhiều đánh giá thấp về sự thuận tiện thời gian, cần cải thiện để tăng hài lòng.
- Đặt vé trực tuyến: Được đánh giá cao, nên duy trì và phát triển thêm.
- Vị trí cổng: Đánh giá trung bình, không phải vấn đề lớn nhưng có thể tối ưu.
- Thức ăn và đồ uống: Đánh giá đa dạng, cần nâng cao chất lượng để đáp ứng kỳ vọng.
- Lên máy bay trực tuyến: Đánh giá tốt, cần duy trì và phát triển.
- Ghế ngồi: Đánh giá trung bình, nên cải thiện thiết kế ghế để tăng sự thoải mái.
- Giải trí trên máy bay: Được đánh giá tốt, nên tiếp tục đầu tư để giữ lợi thế.
- Dịch vụ trên máy bay: Đánh giá trung bình đến cao, cần cải thiện để giảm đánh giá tiêu cực.

- Chỗ để chân: Đánh giá trung bình, nên tối ưu không gian để tăng hài lòng.
- Xử lý hành lý: Được đánh giá cao, nên duy trì chất lượng.
- Làm thủ tục: Đa phần hài lòng, cần tiếp tục giữ vững dịch vụ.
- Dịch vụ trong chuyến bay: Đánh giá trung bình, cần cải thiện để tạo ấn tượng tốt hơn.
- Sạch sẽ: Đánh giá cao, duy trì tiêu chuẩn này sẽ là lợi thế.
- Trễ khởi hành: Phần lớn đúng giờ, cần giảm các chuyến bị trễ nhiều.
- Trễ khi đến: Tương tự, cần tối ưu để giảm bất tiện cho khách.
- Sự hài lòng: Đa số hài lòng, nhưng cần cải thiện các yếu tố như trễ chuyến, wifi, và chỗ ngồi.



Hình 3.7: Phân bố của các biến so với biến phụ thuộc satisfaction.



Hình 3.8: Biểu đồ boxplot của các biến.

Để đảm bảo tính toàn vẹn của bộ dữ liệu, nhóm chúng em quyết định xử lý các giá trị thiếu bằng cách thay thế bằng giá trị trung bình (mean). Vì số lượng dữ liệu thiếu rất ít, việc áp dụng phương pháp này sẽ không gây sai lệch đáng kể trong kết quả phân tích.

```
for col in df_airplane.columns:
    nan_col = df_airplane[col].isna().sum()
    if nan_col > 0:
        if nan_col/len(df_airplane)>0.2:
            df_airplane = df_airplane.drop(columns=col)
        else:
            df_airplane =df_airplane.fillna(df_airplane[col].mean())
```

Hình 3.9: Xử lý NaN

Để mô hình học máy học và hiểu được dữ liệu thì cần phải chuẩn hóa các đặc trưng phân loại (Categorical Feature) thành dạng số.

```
encode = OrdinalEncoder()
df_airplane[categorical_col] = encode.fit_transform(df_airplane[categorical_col])
df_airplane = pd.DataFrame(df_airplane, columns=['Gender', 'Customer Type', 'Age', 'Type of Travel', 'Class',
'Flight Distance', 'Inflight wifi service',
'Departure/Arrival time convenient', 'Ease of Online booking',
'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
'Inflight entertainment', 'On-board service', 'Leg room service',
'Baggage handling', 'Checkin service', 'Inflight service',
'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
'satisfaction'])
```

Hình 3.10 Chuẩn hóa dùng Ordinal Encoder



Các đặc trưng phân loại đã được chuyển về số quan sát kết quả dưới hình

Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Inflight entertainment	On board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes	Arrival Delay in Minutes	satisfaction	
0	1.0	0.0	13	1.0	2.0	460	3	4	3	1	5	4	3	4	4	5	5	25	18.0	0.0
1	1.0	1.0	25	0.0	0.0	235	3	2	3	3	1	5	3	1	4	1	1	6.0	0.0	0.0
2	0.0	0.0	26	0.0	0.0	1142	2	2	2	2	5	4	3	4	4	4	5	0	0.0	1.0
3	0.0	0.0	25	0.0	0.0	562	2	5	5	5	2	2	5	3	1	4	2	11	9.0	0.0
4	1.0	0.0	61	0.0	0.0	214	3	3	3	3	3	3	4	4	3	3	3	0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
103899	0.0	1.0	23	0.0	1.0	192	2	1	2	3	2	3	1	4	2	3	2	3	0.0	0.0
103900	1.0	0.0	49	0.0	0.0	2347	4	4	4	4	5	5	5	5	5	5	4	0	0.0	1.0
103901	1.0	1.0	30	0.0	0.0	1995	1	1	1	3	4	3	2	4	5	5	4	7	14.0	0.0
103902	0.0	1.0	22	0.0	1.0	1000	1	1	1	5	1	4	5	1	5	4	1	0	0.0	0.0
103903	1.0	0.0	27	0.0	0.0	1723	1	3	3	3	1	1	1	4	4	3	1	0	0.0	0.0

Hình 3.11 Kết quả sau khi chuẩn hóa

Vì bộ dữ liệu có quá nhiều đặc trưng và có mối tương quan cao nên chúng tôi quyết định sử dụng PCA phân tích thành phần chính để biến đổi dữ liệu về không gian có số chiều nhỏ hơn mà vẫn giữ được nhiều thông tin nhất có thể của bộ dữ liệu. Sau đó tiến hành xây dựng mô hình hồi quy Logistic.

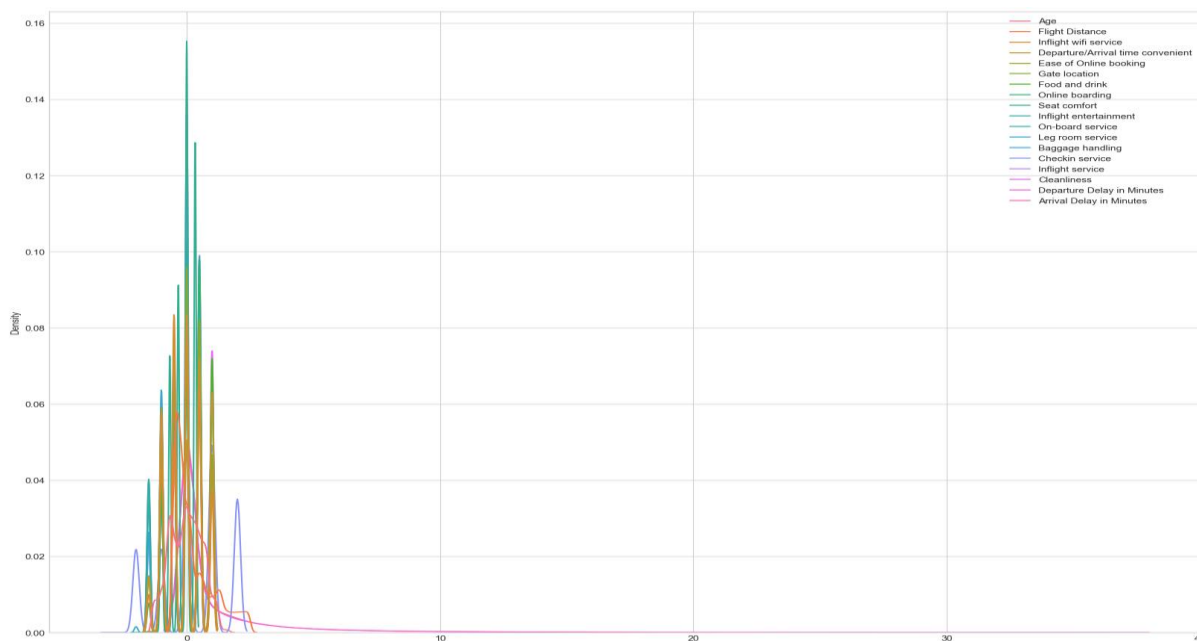
Tiến hành chuẩn hoá dữ liệu trước khi tiến hành đưa bộ dữ liệu vào mô hình hồi quy. Sử dụng phương pháp RobustScaler.

```

scaler = RobustScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = pd.DataFrame(X_scaled, columns=df_airplane.columns[0:-1])
X_scaled

```

Hình 3.12: Chuẩn hoá RobustScaler.



Hình 3.13: Biểu đồ KDE sau khi chuẩn hoá.



Bắt đầu dùng PCA để giảm số chiều của bộ data. Sau khi giảm chiều bằng PCA ta thu được số thành phần chính gồm 8 thành phần.

```
# Lua chọn thành phần chính tỷ lệ > 0.9
n_components = np.argmax(cumsum_explained_var >=0.9) + 1
print(f'Số lượng thành phần chính được chọn : {n_components}')
```

So lượng thành phần chính được chọn : 8

Hình 3.14: Số thành phần chính

Tiến hành đưa bộ dữ liệu vào mô hình hồi quy Logistic ta thu được bảng thông số và các kết quả sau:

```
Optimization terminated successfully.
      Current function value: 0.392697
      Iterations 7
```

Logit Regression Results						
Dep. Variable:	satisfaction	No. Observations:	103904			
Model:	Logit	Df Residuals:	103895			
Method:	MLE	Df Model:	8			
Date:	Tue, 07 Jan 2025	Pseudo R-squ.:	0.4261			
Time:	19:01:34	Log-Likelihood:	-40803.			
converged:	True	LL-Null:	-71094.			
Covariance Type:	nonrobust	LLR p-value:	0.000			
	coef	std err	z	P> z	[0.025	0.975]
const	-0.5092	0.009	-54.962	0.000	-0.527	-0.491
x1	-0.0476	0.002	-22.204	0.000	-0.052	-0.043
x2	1.1998	0.008	147.051	0.000	1.184	1.216
x3	-0.7664	0.009	-89.137	0.000	-0.783	-0.750
x4	-0.6643	0.009	-75.029	0.000	-0.682	-0.647
x5	0.4239	0.010	42.806	0.000	0.404	0.443
x6	1.0140	0.011	92.170	0.000	0.992	1.036
x7	-0.7818	0.013	-58.346	0.000	-0.808	-0.756
x8	0.2009	0.013	15.082	0.000	0.175	0.227

Hình 3.15: Mô hình sau khi giảm chiều

Kết quả hồi quy logistic cung cấp một cái nhìn chi tiết về các yếu tố ảnh hưởng đến sự hài lòng của hành khách (**satisfaction**). Mô hình sử dụng phương pháp Logit Regression với tối ưu hóa bằng Maximum Likelihood Estimation (MLE) và được huấn luyện trên 103,904 quan sát, đảm bảo dữ liệu đủ lớn để đưa ra kết quả đáng tin cậy. Giá trị **Pseudo R-squared** là

0.4261, cho thấy mô hình giải thích được khoảng 42.61% sự biến thiên của biến phụ thuộc.

Hầu hết các biến giải thích trong mô hình đều có ý nghĩa thống kê với **p-value < 0.05**, cho thấy chúng có mối quan hệ đáng kể với sự hài lòng.

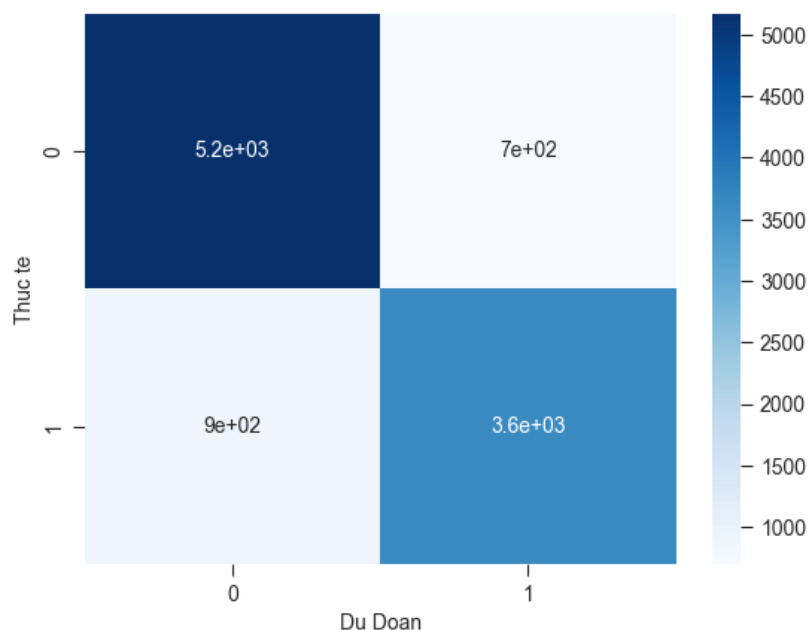
Kết quả cũng cho thấy giá trị **LLR p-value = 0.000**, khẳng định mô hình tổng thể có ý nghĩa thống kê và các biến giải thích có mối quan hệ đáng kể với biến phụ thuộc.

```
He so chan Intercept: [-0.51145017]
He so hoi quy ung voi tung dac trung Coefficients: [[-0.04862975  1.19948429 -0.76768049 -0.66148885  0.42245609  1.01860259
-0.77976729  0.19942922]]
Accuracy: 0.8460205947454528
Classification Report:
```

	precision	recall	f1-score	support
0.0	0.85	0.88	0.87	5868
1.0	0.84	0.80	0.82	4523
accuracy			0.85	10391
macro avg	0.84	0.84	0.84	10391
weighted avg	0.85	0.85	0.85	10391

*Hình 3.16: Kết quả huấn luyện mô hình*

Kết quả dự đoán tốt với độ chính xác (**accuracy**) đạt 84.60%. Các hệ số hồi quy cho thấy tác động của từng đặc trưng đến xác suất dự đoán sự hài lòng (**satisfaction**), trong đó các hệ số dương như **1.19948** thể hiện tác động tích cực, còn các hệ số âm như **-0.7676** phản ánh ảnh hưởng tiêu cực đến sự hài lòng. Báo cáo phân loại cho thấy hiệu suất tốt với **precision** lần lượt là 85% (lớp 0 - không hài lòng) và 84% (lớp 1 - hài lòng), cùng với **recall** đạt 88% cho lớp 0 và 80% cho lớp 1. F1-score cân bằng giữa hai lớp, lần lượt là 87% và 82%, trong khi trung bình F1-score (macro avg và weighted avg) đạt 84% 85%, phản ánh mô hình hoạt động ổn định trên cả hai lớp. Dữ liệu tương đối cân bằng giữa lớp "hài lòng" (9,068 mẫu) và "không hài lòng" (11,173 mẫu), giúp mô hình không bị thiên lệch.



Hình 3.17: Confusion Matrix

Ma trận nhầm lẫn cho thấy mô hình dự đoán hoạt động khá tốt, đặc biệt với lớp "không hài lòng" (lớp 0), khi dự đoán đúng khoảng 5200 mẫu và chỉ nhầm lẫn 700 mẫu vào lớp "hài lòng". Đối với lớp "hài lòng" (lớp 1), mô hình dự đoán đúng 3600 mẫu nhưng vẫn bỏ sót khoảng 900 mẫu (False Negative), cho thấy một lượng đáng kể hành khách hài lòng bị dự đoán nhầm là "không hài lòng".

Mô hình hiện tại có độ chính xác cao hơn ở lớp "không hài lòng" so với lớp "hài lòng".

### 1.3.3 Nhận xét và kết luận

#### Hiệu suất tổng thể của mô hình:

- Mô hình dự đoán có độ chính xác (**accuracy**) cao, đạt **84.60%**, với F1-score đồng đều giữa hai lớp, phản ánh khả năng phân loại tốt.
- Ma trận nhầm lẫn cho thấy mô hình hoạt động tốt hơn ở lớp "không hài lòng" với số lượng dự đoán đúng lớn (5200), trong khi lớp "hài lòng" vẫn còn bỏ sót 700 mẫu, dẫn đến độ nhạy (**recall**) thấp hơn ở lớp này.

### Tác động của các đặc trưng:

- Một số đặc trưng có tác động mạnh mẽ đến sự hài lòng, như **x2** (hệ số dương lớn) làm tăng khả năng hài lòng và **x3, x7** (hệ số âm) làm giảm mạnh xác suất hài lòng.
- Thời gian trễ (Departure/Arrival Delay) là một yếu tố quan trọng với nhiều outliers, ảnh hưởng tiêu cực đến sự hài lòng. Điều này nhấn mạnh sự cần thiết phải cải thiện lịch trình bay để giảm thiểu trễ chuyến.

### Cần cải thiện:

- Cần tập trung giảm **False Negative** (dự đoán nhầm hành khách hài lòng thành không hài lòng) để nâng cao độ nhạy ở lớp "hài lòng".
- Chất lượng dịch vụ như wifi, sự thoải mái ghế ngồi, và thời gian khởi hành/đến cần được tối ưu để tăng mức độ hài lòng tổng thể.

Mô hình hiện tại có độ chính xác và hiệu suất tốt, đặc biệt với lớp "không hài lòng". Tuy nhiên, cần cải thiện độ nhạy ở lớp "hài lòng" để giảm thiểu các trường hợp dự đoán sai. Điều này có thể đạt được bằng cách tối ưu ngưỡng dự đoán và cải thiện các yếu tố dịch vụ ảnh hưởng tiêu cực như thời gian trễ và sự thoải mái trên máy bay. Nhìn chung, mô hình có tiềm năng ứng dụng cao trong việc phân tích và dự đoán sự hài lòng của hành khách.

## **CHƯƠNG 2**

### **DỮ LIỆU TỰ THU THẬP**

- Tên đề tài, nguồn gốc của dữ liệu, giới thiệu các biến.
- Mô hình chọn được, phân tích kết quả.
- Đưa ra những phương pháp / phân tích khác có thể giúp cho kết quả tốt hơn.
- Kết luận

## 2.1 Thu thập và tiền xử lý dữ liệu.

Chúng tôi thực hiện thu thập dữ liệu từ trang web [Visual Crossing](#)[4], một nền tảng cung cấp dữ liệu thời tiết toàn cầu bao gồm thông tin lịch sử, dự báo và thời tiết hiện tại. Dữ liệu được thu thập thông qua phương pháp sử dụng API (Application Programming Interface), cho phép truy xuất thông tin tự động và nhanh chóng. Bằng cách sử dụng API của Visual Crossing, tôi có thể gửi các yêu cầu HTTP với các tham số cụ thể như vị trí, thời gian, và loại dữ liệu mong muốn để nhận được phản hồi dưới dạng JSON hoặc CSV. Phương pháp này không chỉ đảm bảo độ chính xác và tính cập nhật của dữ liệu mà còn tiết kiệm thời gian so với các phương pháp thu thập thủ công. Dữ liệu thu thập được sẽ được sử dụng để phân tích và nghiên cứu trong các ứng dụng liên quan đến thời tiết.

```
import requests
import json

API_KEY = "5MVU7E3V7P956YKGHJDSRSAEX"
BASE_URL = "https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/"
LOCATION = "Ho Chi Minh"
START_DATE = "2024-12-01"
END_DATE = "2025-01-05"
UNIT_GROUP = "metric"
INCLUDE = "hours,days"

url = f"{BASE_URL}{LOCATION}/{START_DATE}/{END_DATE}?unitGroup={UNIT_GROUP}&include={INCLUDE}&key={API_KEY}&contentType=json"
response = requests.get(url)

if response.status_code == 200:
    data = response.json()
    with open("weather_data.json", "w", encoding="utf-8") as file:
        json.dump(data, file, indent=4, ensure_ascii=False)
    print("Dữ liệu thời tiết đã được lưu vào 'weather_data.json'")
else:
    print(f"Không thể lấy dữ liệu: {response.status_code} - {response.text}")
```

Hình 4.1: Thu thập dữ liệu.

Sau khi hoàn thành quá trình thu thập dữ liệu, chúng tôi đã thu được một bộ dữ liệu bao gồm 407 quan trắc và 18 đặc trưng sau.

- Temp: Nhiệt độ hiện tại (độ C).
- Feelslike: Nhiệt độ cảm nhận được (độ C).
- Humidity: Độ ẩm không khí (%).
- Dew: Điểm sương (độ C) - nhiệt độ tại đó hơi nước ngưng tụ thành sương.

- Precip: Lượng mưa (mm).
- Precipprob: Xác suất xảy ra mưa (%).
- Snow: Lượng tuyết rơi (mm).
- Snowdepth: Độ sâu lớp tuyết (mm).
- Preciptype: Loại hình giáng thủy (mưa, tuyết, mưa đá, v.v.).
- Windgust: Cơn gió mạnh nhất (km/h).
- Windspeed: Tốc độ gió trung bình (km/h).
- Winddir: Hướng gió (độ từ Bắc, 0° - 360°).
- Pressure: Áp suất khí quyển (hPa).
- Visibility: Tầm nhìn xa (km).
- Cloudcover: Mức độ che phủ của mây (%).
- Uvindex: Chỉ số tia cực tím (UV).
- Severerisk: Mức độ rủi ro thời tiết nghiêm trọng (thang điểm).
- Conditions: Mô tả điều kiện thời tiết (như mây, nắng, mưa).

temp	feelslike	humidity	dew	precip	precipprob	snow	snowdepth	preciptype	windgust	windspeed	winddir	pressure	visibility	cloudcover	uvindex	severerisk	con
24.6	24.6	71.98	19.2	0.0	0	0	0	NaN	7.6	9.1	20.0	1009.5	10.9	80.0	0	10	f
24.0	24.0	73.63	19.0	0.0	0	0	0	NaN	9.4	8.6	24.0	1009.0	10.0	70.9	0	10	f
23.0	23.0	78.21	19.0	0.0	0	0	0	NaN	8.3	7.6	12.0	1009.0	10.0	27.0	0	10	f
23.0	23.0	78.76	19.1	0.0	0	0	0	NaN	8.3	6.1	4.0	1009.0	10.9	88.0	0	10	f
23.0	23.0	78.21	19.0	0.0	0	0	0	NaN	10.4	5.4	18.0	1009.4	10.0	72.5	0	10	f

*Hình 4.2: Một vài quan trắc đầu tiên.*

Mục tiêu của chúng tôi trong việc thu thập bộ dữ liệu này là sử dụng các chỉ số và đặc trưng để xây dựng một mô hình dự đoán chất lượng không khí, phân loại thành hai trạng thái: "Tốt" hoặc "Kém". Để đảm bảo mô hình đạt hiệu quả cao, chúng tôi quyết định loại bỏ những thuộc tính không có ý nghĩa hoặc ít đóng góp vào việc xác định chất lượng không khí, chẳng hạn như các cột liên quan đến tuyết, bao gồm snow, snowdepth, precipiype, và precipprob.

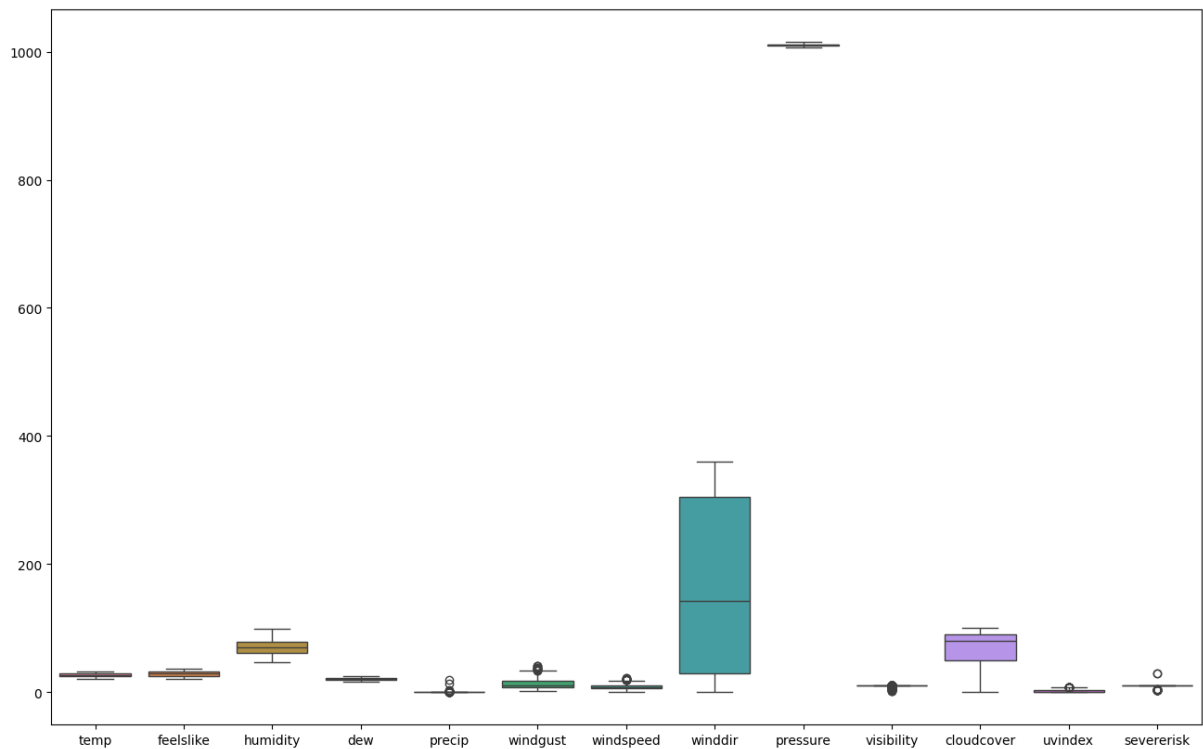
Bên cạnh đó, chúng tôi đã tạo thêm một biến mục tiêu mới có tên là `air_quality_adjusted`, được xây dựng dựa trên các thuộc tính quan trọng như `humidity`, `uvindex`, và `conditions`. Quy tắc phân loại được điều chỉnh để mở rộng tiêu chí đánh giá trạng thái "Tốt" nhằm phản ánh chính xác hơn chất lượng không khí. Cụ thể, nếu độ ẩm (`humidity`) dưới 80%, chỉ số UV (`uvindex`) dưới 5, và điều kiện thời tiết (`conditions`) nằm trong nhóm "Clear", "Partially cloudy", hoặc "Overcast", thì chất lượng không khí được phân loại là "Tốt". Ngược lại, các trường hợp còn lại sẽ được phân loại là "Kém".

```
def classify_air_quality_adjusted(row):  
    if (row['humidity'] < 80 and row['uvindex'] < 5 and  
        row['conditions'] in ['Clear', 'Partially cloudy', 'Overcast']):  
        return 'Tốt'  
    else:  
        return 'Kém'  
  
data['air_quality_adjusted'] = data.apply(classify_air_quality_adjusted, axis=1)  
  
data.isna().sum()  
data = data.drop(columns=["Unnamed: 0", "preciptype", "snowdepth", "snow", "precipprob"])  
data['precip'].value_counts()
```

*Hình 4.3: Loại bỏ các cột ít đóng góp và tạo biến phụ thuộc.*

Sau khi loại bỏ các đặc trưng không đóng góp đáng kể vào mục tiêu phân tích của bộ dữ liệu, chúng tôi tiến hành kiểm tra sự hiện diện của các giá trị ngoại lai (outliers). Qua đánh giá, các giá trị này chủ yếu nằm gần ngưỡng của khoảng tứ phân vị (IQR), cho thấy khả năng cao chúng không phải là ngoại lai thực sự. Để đảm bảo tính toàn vẹn của dữ liệu và tránh mất mát thông tin quan trọng, chúng tôi quyết định giữ lại toàn bộ các giá trị này, thay vì loại bỏ một cách không cần thiết.





*Hình 4.4: Biểu đồ boxplot của các biến.*

Tiếp theo, chúng tôi tiến hành mã hóa các biến phân loại sang dạng số để phù hợp với yêu cầu của mô hình học máy, vốn hoạt động dựa trên dữ liệu dạng số. Các biến phân loại cần mã hóa bao gồm conditions và air\_quality\_adjusted. Việc chuyển đổi này là một bước quan trọng trong quy trình tiền xử lý dữ liệu, nhằm chuẩn bị bộ dữ liệu sẵn sàng cho quá trình xây dựng và huấn luyện mô hình.

```

categorical_cols = ["conditions", "air_quality_adjusted"]
categorical_cols

encode = OrdinalEncoder()
data[categorical_cols] = encode.fit_transform(data[categorical_cols])
data

```

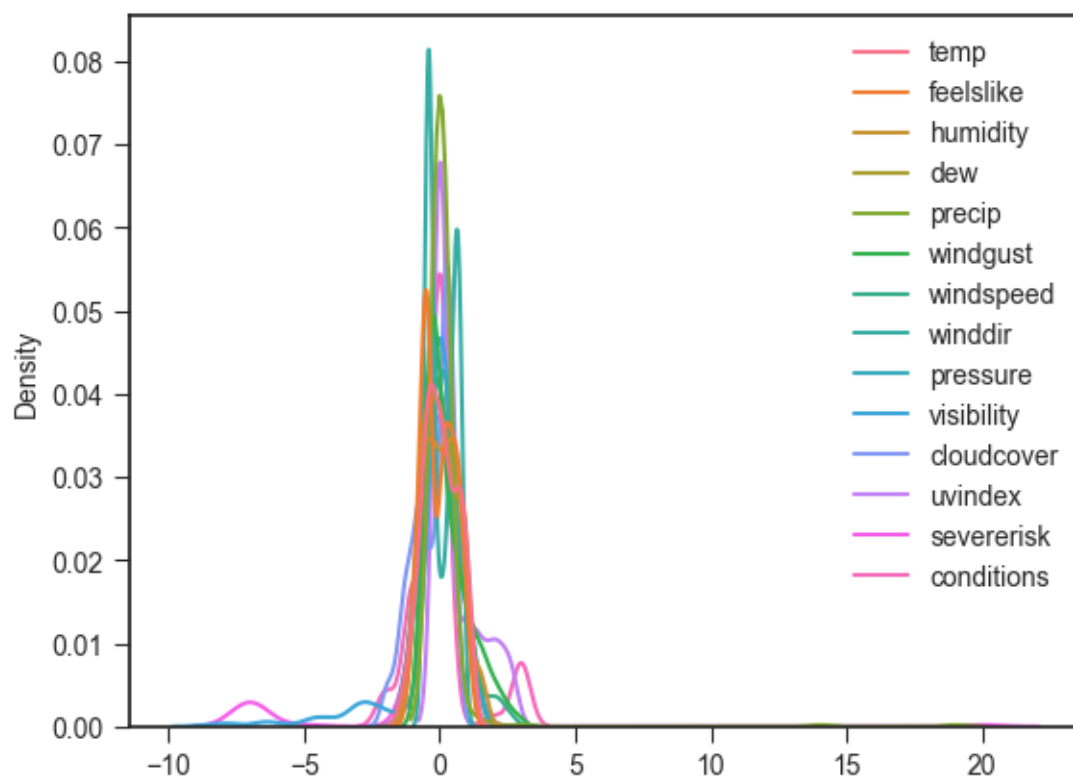
*Hình 4.5: Mã hoá dữ liệu.*

Tiếp theo, chúng tôi thực hiện bước chuẩn hóa dữ liệu để đưa các đặc trưng về cùng một thang đo. Điều này rất quan trọng trong học máy, vì bản chất của các thuật toán học máy là gán trọng số cho các biến đầu vào. Nếu không chuẩn hóa, các đặc trưng có giá trị lớn hơn sẽ được mô hình ưu tiên

hơn, dẫn đến sự thiên vị và ảnh hưởng đến hiệu suất mô hình. Để khắc phục, chúng tôi sử dụng **RobustScaler**, một phương pháp chuẩn hóa hiệu quả với dữ liệu chứa giá trị ngoại lai (outliers), vì nó sử dụng trung vị (median) và khoảng tứ phân vị (IQR) để chuẩn hóa, giảm thiểu tác động của các giá trị ngoại lai.

```
scaler = RobustScaler()
X_scaled = scaler.fit_transform(x)
X_scaled = pd.DataFrame(X_scaled, columns=data.columns[0:-1])
X_scaled
```

Hình 4.6: Chuẩn hoá dữ liệu.



Hình 4.7: Các biến sau khi chuẩn hoá.

Sau khi chuẩn hoá xong chúng tôi tiếp tục tiến hành sử dụng PCA để giảm chiều dữ liệu để loại bỏ các thông tin dư thừa hoặc không quan trọng mà vẫn giữ được phần lớn thông tin cần thiết.

PCA hoạt động bằng cách chuyển đổi các đặc trưng ban đầu thành một tập hợp các thành phần chính mới, trong đó các thành phần này được sắp

xếp theo thứ tự giảm dần về mức độ đóng góp thông tin. Chỉ một số ít thành phần đầu tiên, chứa phần lớn phương sai của dữ liệu, được giữ lại để sử dụng trong mô hình dự đoán.

Đầu tiên, PCA được áp dụng trên tập dữ liệu đã chuẩn hóa để tính toán tỷ lệ phương sai giải thích của từng thành phần. Điều này giúp chúng tôi hiểu được mức độ đóng góp thông tin của từng thành phần vào tập dữ liệu.

```
# Tỷ lệ phương sai
explained_var = pca.explained_variance_ratio_ # khoản tin cậy
print(f'Tỷ lệ phương sai giải thích {explained_var}')
```

*Hình 4.8: Tỷ lệ phương sai giải thích.*

Sau đó, phương sai tích lũy được tính toán để xác định tổng lượng thông tin mà các thành phần chính giữ lại. Chúng tôi lựa chọn số lượng thành phần chính sao cho phương sai tích lũy đạt ít nhất 90%, đảm bảo rằng phần lớn thông tin quan trọng trong dữ liệu ban đầu được giữ lại.

```
# chọn trên phương sai tích lũy
cumsum_explained_var = np.cumsum(pca.explained_variance_ratio_)
print(f'Phương sai tích lũy: {cumsum_explained_var}')
```

*Hình 4.9: Phương sai tích lũy.*

```
# Lựa chọn thành phần chính tỷ lệ > 0.9
n_components = np.argmax(cumsum_explained_var >= 0.9) + 1
print(f'Số lượng thành phần chính được chọn : {n_components}')
```

So lượng thành phần chính được chọn : 7

*Hình 4.10: Lựa chọn thành phần chính.*

Kết quả cho thấy số lượng thành phần chính được chọn là 7, tương ứng với các thành phần giữ lại ít nhất 90% phương sai. Việc lựa chọn này giúp giảm đáng kể số chiều dữ liệu, tăng hiệu quả tính toán, đồng thời giữ lại thông tin cần thiết để xây dựng mô hình dự đoán chính xác và ổn định.

Việc giảm chiều dữ liệu bằng PCA không chỉ làm giảm độ phức tạp của mô hình mà còn tăng hiệu quả tính toán và giảm nguy cơ quá khớp (overfitting). Điều này đặc biệt hữu ích khi xử lý các tập dữ liệu lớn và có nhiều đặc trưng tương quan với nhau. PCA giúp tập trung vào những thông tin quan trọng nhất, đảm bảo mô hình hoạt động hiệu quả và ổn định hơn.

Sau khi tiến hành PCA xong, bộ dữ liệu được đưa vào mô hình hồi quy Logistic thông số thu được thể hiện qua các kết quả sau:

## 2.2 Mô hình.

```
Optimization terminated successfully.
Current function value: 0.428295
Iterations 9
```

Logit Regression Results						
=====						
Dep. Variable:	air_quality_adjusted	No. Observations:	407			
Model:	Logit	Df Residuals:	399			
Method:	MLE	Df Model:	7			
Date:	Mon, 06 Jan 2025	Pseudo R-squ.:	0.3780			
Time:	20:43:08	Log-Likelihood:	-174.32			
converged:	True	LL-Null:	-280.24			
Covariance Type:	nonrobust	LLR p-value:	3.541e-42			
=====						
	coef	std err	z	P> z	[0.025	0.975]
-----						
const	-1.4780	0.314	-4.709	0.000	-2.093	-0.863
x1	-0.1505	0.068	-2.200	0.028	-0.285	-0.016
x2	-0.2774	0.135	-2.058	0.040	-0.542	-0.013
x3	-8.7297	1.312	-6.651	0.000	-11.302	-6.157
x4	-8.3904	1.431	-5.863	0.000	-11.195	-5.585
x5	6.9061	1.180	5.853	0.000	4.594	9.219
x6	2.4004	0.323	7.428	0.000	1.767	3.034
x7	-0.4793	0.229	-2.096	0.036	-0.927	-0.031
=====						

Hình 4.11: Mô hình sau khi giảm chiều

Kết quả hồi quy logistic cung cấp những thông tin chi tiết quan trọng về các yếu tố ảnh hưởng đến biến mục tiêu "chất lượng không khí điều chỉnh" (air\_quality\_adjusted). Mô hình sử dụng phương pháp Hồi quy Logit và được huấn luyện trên một tập dữ liệu gồm 407 quan sát, đảm bảo đủ độ lớn mẫu để đưa ra kết quả đáng tin cậy và chính xác.

Giá trị Pseudo R-squared đạt mức 0.3780, cho thấy mô hình có thể giải thích khoảng 37.80% sự biến thiên của biến phụ thuộc, phản ánh mức độ giải thích khá hợp lý cho sự thay đổi của chất lượng không khí. Đặc biệt, hầu hết các biến giải thích trong mô hình đều có ý nghĩa thống kê (với  $p\text{-value} < 0.05$ ), khẳng định mối quan hệ rõ rệt và có sự ảnh hưởng đáng kể giữa các yếu tố này và biến mục tiêu. Điều này cũng chứng minh rằng mô hình tổng thể là phù hợp với dữ liệu, đồng thời xác nhận khả năng dự báo của mô hình trong việc phân tích và hiểu rõ các yếu tố tác động đến chất lượng không khí.

```

He so chan Intercept: [-0.10700385]
He so hoi quy ung voi tung dac trung Coefficients: [[-0.05966694  0.0594603 -2.53772548 -1.41306207  1.05959538  1.22623186
0.06626928]]
Accuracy: 0.7682926829268293
Classification Report:

```

	precision	recall	f1-score	support
0.0	0.79	0.64	0.71	36
1.0	0.75	0.87	0.81	46
accuracy			0.77	82
macro avg	0.77	0.75	0.76	82
weighted avg	0.77	0.77	0.76	82

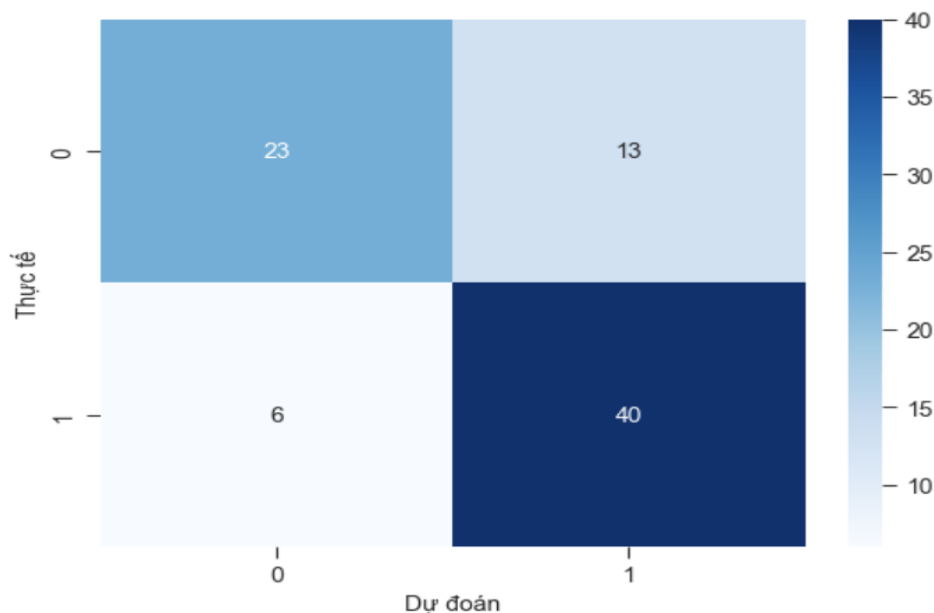
*Hình 4.12: Kết quả huấn luyện mô hình*

Kết quả phân tích cho thấy mô hình dự đoán đạt độ chính xác (accuracy) 77%, phản ánh khả năng phân loại khá tốt của mô hình đối với chất lượng không khí. Các hệ số hồi quy cung cấp thông tin về mức độ ảnh hưởng của từng đặc trưng đến xác suất dự đoán chất lượng không khí. Cụ thể, các hệ số dương (ví dụ: 1.2262) thể hiện tác động tích cực, trong khi các hệ số âm (ví dụ: -2.5377) phản ánh ảnh hưởng tiêu cực đến khả năng dự đoán chất lượng không khí.

Báo cáo phân loại cung cấp các chỉ số đánh giá chi tiết về hiệu suất mô hình. Precision đạt 79% đối với lớp 0 (kém) và 75% đối với lớp 1 (tốt), cho thấy độ chính xác cao khi mô hình phân loại các quan sát đúng vào từng nhóm. Về recall, mô hình đạt 64% đối với lớp 0 (kém) và 87% đối với lớp

1 (tốt), chỉ ra rằng mô hình có xu hướng nhạy cảm hơn với lớp "tốt" (lớp 1). F1-score, là chỉ số cân bằng giữa precision và recall, đạt 71% cho lớp 0 và 81% cho lớp 1, với giá trị trung bình (macro và weighted average) là 76%, thể hiện hiệu suất ổn định và khả năng phân loại khá tốt giữa hai nhóm.

Dữ liệu giữa các lớp "tốt" (36 mẫu) và "kém" (46 mẫu) có sự phân bố tương đối cân bằng, giúp đảm bảo mô hình không bị lệch trong quá trình dự đoán, đảm bảo tính khách quan và chính xác cao trong các kết quả phân loại.



Hình 4.13: Confusion Matrix

## 2.3 Nhận xét và kết luận

### Nhận xét

Giá trị Pseudo R-squared (0.3780) cho thấy mô hình có khả năng giải thích gần 38% sự biến thiên của biến mục tiêu. Đây là mức trung bình, có thể chấp nhận được trong bối cảnh dữ liệu phức tạp, nhưng vẫn còn dư địa để cải thiện.

Kết quả này cũng cho thấy mô hình phù hợp với dữ liệu, nhưng các yếu tố bên ngoài chưa được đưa vào mô hình có thể đóng vai trò quan trọng trong việc nâng cao độ chính xác.

Precision và recall của lớp "tốt" cao hơn lớp "kém", phản ánh mô hình ưu tiên nhận diện chính xác các trường hợp có chất lượng không khí tốt hơn.

Ngược lại, với lớp "kém", recall chỉ đạt 64%, cho thấy mô hình bỏ sót khá nhiều trường hợp có chất lượng không khí kém, điều này cần được chú ý nếu mục tiêu là phát hiện đầy đủ các trường hợp thuộc nhóm này.

Sự phân bố dữ liệu giữa hai nhóm "tốt" và "kém" khá cân bằng, giúp giảm nguy cơ lệch nhãn trong quá trình huấn luyện. Tuy nhiên, kết quả phân loại lại cho thấy sự mất cân bằng nhẹ về hiệu suất giữa hai nhóm.

Các hệ số hồi quy có ý nghĩa thống kê với hầu hết các biến ( $p\text{-value} < 0.05$ ), chỉ ra rằng các yếu tố này có ảnh hưởng đáng kể đến chất lượng không khí.

### **Kết luận:**

Mô hình hoạt động tốt với độ chính xác 77% và hiệu suất khá cân bằng giữa hai lớp.

Precision và recall của lớp 1 (tốt) cao hơn, cho thấy mô hình tập trung mạnh vào việc nhận diện các trường hợp có chất lượng không khí tốt, nhưng khả năng phát hiện chính xác lớp 0 (kém) còn hạn chế (recall 64%). Ma trận nhầm lẫn cho thấy cần cải thiện khả năng phân loại lớp 0 để giảm thiểu số lượng dự đoán sai.

Với kết quả này, mô hình có thể được sử dụng, nhưng nếu mục tiêu là cân bằng hoàn toàn giữa hai lớp, các phương pháp tối ưu hóa khác cần được xem xét để thay đổi ngưỡng dự đoán hoặc sử dụng thuật toán phân loại bổ sung.

## NGUỒN GỐC BỘ DỮ LIỆU

- [1] “Stroke Prediction Dataset.” Accessed: Jan. 05, 2025. [Online]. Available: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>
- [2] R. Bock, “MAGIC Gamma Telescope.” UCI Machine Learning Repository, 2004. doi: 10.24432/C52C8B.
- [3] “Airline Passenger Satisfaction.” Accessed: Jan. 04, 2025. [Online]. Available: <https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction>
- [4] “Weather Data Services | Visual Crossing.” Accessed: Jan. 06, 2025. [Online]. Available: <https://www.visualcrossing.com/weather/weather-data-services>