BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

BÁO CÁO MÔN HỌC KHAI PHÁ DỮ LIỆU

CREDIT CARD FRAUD DETECTION

SINH VIÊN THỰC HIỆN :

2251068227 – Bùi Thiện Phát

2251068255 – Nguyễn Phúc Thịnh

GIÁO VIÊN HƯỚNG DẪN : Ths.Vũ Thị Hạnh

Hồ Chí Minh, ngày 20 tháng 10 năm 2025

MỤC LỤC

I. GIỚI THIỆU ĐỀ TÀI

Trong thời đại công nghệ số, giao dịch trực tuyến và thanh toán điện tử đã trở thành một phần không thể thiếu trong đời sống hiện đại. Tuy nhiên, song song với sự phát triển này là sự gia tăng nhanh chóng của các **hoạt động gian lận tài chính**, đặc biệt trong lĩnh vực **thẻ tín dụng**. Các hành vi gian lận ngày càng tinh vi, khó phát hiện bằng các phương pháp truyền thống, gây thiệt hại lớn cho cả ngân hàng, tổ chức tài chính và khách hàng cá nhân. Vì vậy, việc áp dụng **các phương pháp khai phá dữ liệu và học máy** để tự động phát hiện giao dịch bất thường là một hướng đi quan trọng và có tính ứng dụng cao.

Bài toán **phát hiện gian lận thẻ tín dụng (Credit Card Fraud Detection)** là một ví dụ điển hình của lĩnh vực **phân loại nhị phân mất cân bằng (imbalanced classification)**, trong đó số lượng giao dịch gian lận chiếm tỷ lệ rất nhỏ so với giao dịch hợp lệ. Điều này đặt ra thách thức lớn cho các mô hình học máy, vì nếu không xử lý đúng cách, mô hình dễ thiên lệch và dự đoán tất cả giao dịch là "hợp lệ", dẫn đến bỏ sót các trường hợp gian lận thực sự. Do đó, cần có các chiến lược tiền xử lý dữ liệu, lựa chọn mô hình và đánh giá phù hợp để đảm bảo mô hình vừa chính xác, vừa nhạy trong việc phát hiện gian lận.

Trong dự án này, nhóm tập trung vào việc **xây dựng, huấn luyện và đánh giá nhiều mô hình học máy khác nhau** nhằm phát hiện các giao dịch gian lận trong bộ dữ liệu thực tế từ Kaggle. Thông qua việc áp dụng các kỹ thuật như **chuẩn hóa dữ liệu, xử lý mất cân bằng bằng SMOTE, sử dụng trọng số lớp (class_weight)** và **so sánh hiệu năng của nhiều mô hình (Logistic Regression, Random Forest, XGBoost, LightGBM, CatBoost, SVM, Decision Tree)**, nhóm hướng đến việc tìm ra mô hình có khả năng nhận diện chính xác nhất các giao dịch gian lận.

II. MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA

2.1 Mục tiêu của đề tài

Mục tiêu của đề tài là **xây dựng một hệ thống phát hiện gian lận thẻ tín dụng tự động** bằng cách áp dụng các thuật toán học máy (Machine Learning) trên bộ dữ liệu thực tế. Hệ thống này phải có khả năng **phân biệt chính xác giữa giao dịch hợp lệ và giao dịch gian lận**, đồng thời **giảm thiểu rủi ro bỏ sót các trường hợp gian lận thật**.
Cụ thể, nhóm hướng tới việc:

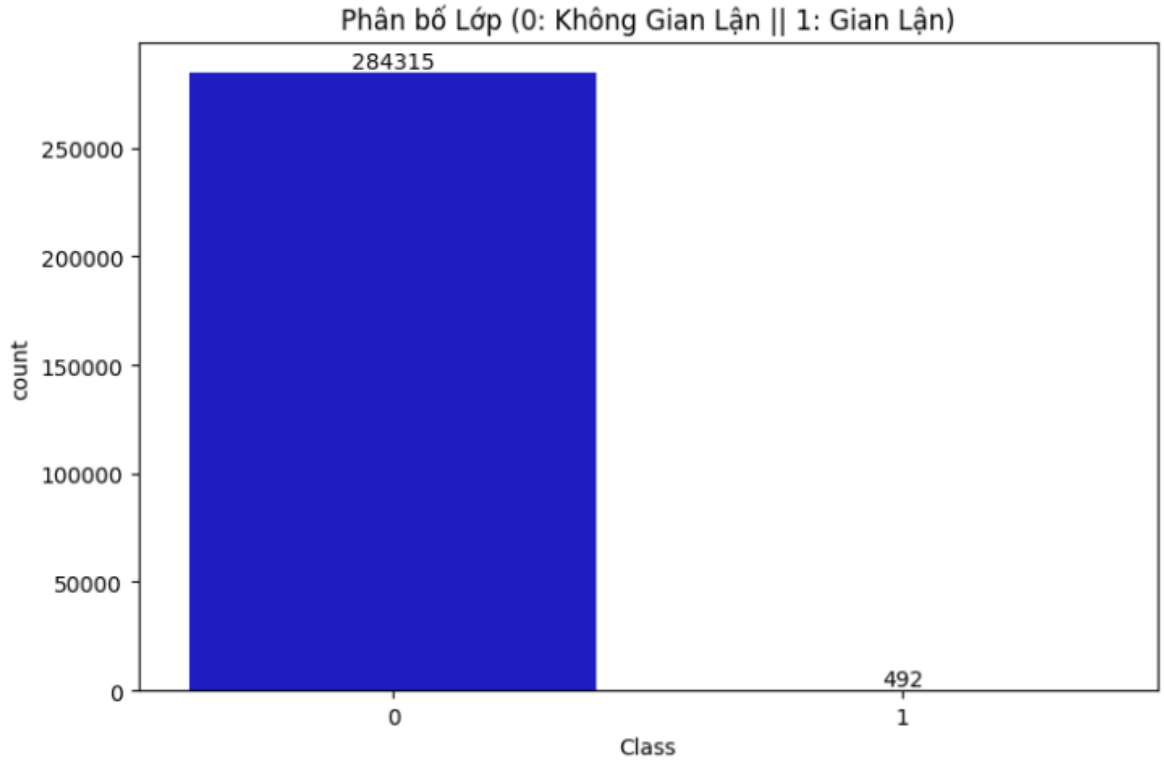
- Xây dựng quy trình xử lý dữ liệu hiệu quả, bao gồm chuẩn hóa và xử lý mất cân bằng lớp.
- Huấn luyện, đánh giá và so sánh hiệu năng của nhiều mô hình khác nhau để chọn ra mô hình tối ưu.
- Triển khai kết quả thành công cụ trực quan, dễ sử dụng cho việc dự đoán gian lận.

2.2 Bài toán đặt ra

Bài toán được xác định là **bài toán phân loại nhị phân (Binary Classification)**, trong đó:

- **Đầu vào (Input):** Các đặc trưng mô tả thông tin giao dịch, bao gồm 28 đặc trưng ẩn danh (V1–V28) cùng hai biến gốc Time và Amount.
- **Đầu ra (Output):** Biến nhãn Class với hai giá trị:
    - 0: giao dịch hợp lệ (non-fraud).
    - 1: giao dịch gian lận (fraud).

Thách thức chính của bài toán là dữ liệu **mất cân bằng nghiêm trọng**, khi số lượng giao dịch gian lận chỉ chiếm khoảng **0.17%** tổng số mẫu. Do đó, nhóm cần kết hợp các kỹ thuật như **SMOTE**, **class_weight='balanced'**, cùng quy trình tiền xử lý dữ liệu và đánh giá bằng các chỉ số phù hợp như **AUC**, **F1-score**, **Precision**, và **Recall** để đảm bảo mô hình không bị lệch về lớp chiếm đa số.



Phân bố lớp trong dữ liệu (mất cân bằng nghiêm trọng giữa hai loại giao dịch)

2.3 Kết quả mong đợi

Sau khi hoàn thành, hệ thống phải:

- Dự đoán chính xác các giao dịch gian lận với **AUC và F1-score cao**, đảm bảo cân bằng giữa độ nhạy (Recall) và độ chính xác (Precision).
- Có khả năng **phát hiện sớm các hành vi gian lận** nhằm hỗ trợ ngân hàng và người dùng trong việc giảm thiểu rủi ro tài chính.
- Cung cấp **biểu đồ trực quan** (ROC Curve, Precision-Recall Curve, Confusion Matrix) để minh họa kết quả mô hình và có thể tích hợp trên **giao diện web hoặc dashboard** cho phép nhập dữ liệu và xem kết quả dự đoán trực tiếp.

III. MÔ TẢ DỮ LIỆU VÀ CÁC BƯỚC TIỀN XỬ LÝ

3.1 Mô tả dữ liệu

Bộ dữ liệu được sử dụng trong đề tài là **Credit Card Fraud Detection Dataset**, được lấy từ trang **Kaggle**. Dữ liệu bao gồm **284.807 giao dịch thẻ tín dụng** được thực hiện trong hai ngày, trong đó chỉ có **492 giao dịch được gắn nhãn là gian lận**, chiếm khoảng **0.172%** tổng số giao dịch.

Dữ liệu bao gồm **31 thuộc tính**, trong đó:

- Time: thời gian tính bằng giây giữa giao dịch hiện tại và giao dịch đầu tiên.
- Amount: số tiền của giao dịch.
- Class: nhãn mục tiêu — 0 là giao dịch bình thường, 1 là giao dịch gian lận.
- V1 → V28: các đặc trưng đã được trích xuất bằng **PCA (Principal Component Analysis)** để ẩn thông tin gốc vì lý do bảo mật.

```
5 dòng đầu:
   Time        V1        V2        V3        V4        V5        V6        V7  \
0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599
1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461
3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609
4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941

         V8        V9  ...       V21       V22       V23       V24       V25  \
0  0.098698  0.363787  ... -0.018307  0.277838 -0.110474  0.066928  0.128539
1  0.085102 -0.255425  ... -0.225775 -0.638672  0.101288 -0.339846  0.167170
2  0.247676 -1.514654  ...  0.247998  0.771679  0.909412 -0.689281 -0.327642
3  0.377436 -1.387024  ... -0.108300  0.005274 -0.190321 -1.175575  0.647376
4 -0.270533  0.817739  ... -0.009431  0.798278 -0.137458  0.141267 -0.206010

        V26       V27       V28  Amount  Class
0 -0.189115  0.133558 -0.021053  149.62      0
1  0.125895 -0.008983  0.014724    2.69      0
2 -0.139097 -0.055353 -0.059752  378.66      0
3 -0.221929  0.062723  0.061458  123.50      0
4  0.502292  0.219422  0.215153   69.99      0

[5 rows x 31 columns]
```

Bộ dữ liệu này có đặc điểm quan trọng là mất cân bằng nghiêm trọng về số lượng giữa hai lớp, đây chính là yếu tố khiến việc huấn luyện và đánh giá mô hình trở nên khó khăn hơn.

Nhờ vào đặc trưng đã được xử lý bằng PCA, các biến đầu vào không chứa thông tin nhận dạng cá nhân, đồng thời giúp giảm tương quan giữa các thuộc tính, hỗ trợ cho việc huấn luyện mô hình học máy.

```
Thống kê mô tả:
                Time            V1            V2            V3            V4  \
count  284807.000000  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean    94813.859575  1.168375e-15  3.416908e-16 -1.379537e-15  2.074095e-15
std     47488.145955  1.958696e+00  1.651309e+00  1.516255e+00  1.415869e+00
min         0.000000 -5.640751e+01 -7.271573e+01 -4.832559e+01 -5.683171e+00
25%     54201.500000 -9.203734e-01 -5.985499e-01 -8.903648e-01 -8.486401e-01
50%     84692.000000  1.810880e-02  6.548556e-02  1.798463e-01 -1.984653e-02
75%    139320.500000  1.315642e+00  8.037239e-01  1.027196e+00  7.433413e-01
max    172792.000000  2.454930e+00  2.205773e+01  9.382558e+00  1.687534e+01

                V5            V6            V7            V8            V9  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   9.604066e-16  1.487313e-15 -5.556467e-16  1.213481e-16 -2.406331e-15
std    1.380247e+00  1.332271e+00  1.237094e+00  1.194353e+00  1.098632e+00
min   -1.137433e+02 -2.616051e+01 -4.355724e+01 -7.321672e+01 -1.343407e+01
25%   -6.915971e-01 -7.682956e-01 -5.540759e-01 -2.086297e-01 -6.430976e-01
50%   -5.433583e-02 -2.741871e-01  4.010308e-02  2.235804e-02 -5.142873e-02
75%    6.119264e-01  3.985649e-01  5.704361e-01  3.273459e-01  5.971390e-01
max    3.480167e+01  7.330163e+01  1.205895e+02  2.000721e+01  1.559499e+01

       ...           V21           V22           V23           V24  \
count  ...  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   ...  1.654067e-16 -3.568593e-16  2.578648e-16  4.473266e-15
std    ...  7.345240e-01  7.257016e-01  6.244603e-01  6.056471e-01
min    ... -3.483038e+01 -1.093314e+01 -4.480774e+01 -2.836627e+00
25%    ... -2.283949e-01 -5.423504e-01 -1.618463e-01 -3.545861e-01
50%    ... -2.945017e-02  6.781943e-03 -1.119293e-02  4.097606e-02
75%    ...  1.863772e-01  5.285536e-01  1.476421e-01  4.395266e-01
max    ...  2.720284e+01  1.050309e+01  2.252841e+01  4.584549e+00

                V25           V26           V27           V28         Amount  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  284807.000000
mean   5.340915e-16  1.683437e-15 -3.660091e-16 -1.227390e-16      88.349619
std    5.212781e-01  4.822270e-01  4.036325e-01  3.300833e-01     250.120109
min   -1.029540e+01 -2.604551e+00 -2.256568e+01 -1.543008e+01       0.000000
25%   -3.171451e-01 -3.269839e-01 -7.083953e-02 -5.295979e-02       5.600000
50%    1.659350e-02 -5.213911e-02  1.342146e-03  1.124383e-02      22.000000
75%    3.507156e-01  2.409522e-01  9.104512e-02  7.827995e-02      77.165000
max    7.519589e+00  3.517346e+00  3.161220e+01  3.384781e+01   25691.160000

              Class
count  284807.000000
mean        0.001727
std         0.041527
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000

[8 rows x 31 columns]
```

3.2 Phân tích thống kê mô tả (Descriptive Statistics)

Quan phân tích thống kê mô tả của tập dữ liệu giao dịch cho thấy có **284.807 bản ghi trong 2 ngày**, với **3 biến gốc** gồm: *Time*, *Amount*, *Class* và **28 biến đã được chuyển đổi bằng PCA** (từ *V1* đến *V28*).

- **Amount** là biến cần được chú ý nhất:

    - Có độ lệch phải nghiêm trọng, với **giá trị trung bình khoảng 88,35** cao hơn nhiều so với **giá trị trung vị 22,00**.
    - Chứa các giao dịch cực lớn, lên đến **25.691,16**.
    - Điều này đòi hỏi phải áp dụng phương pháp **log-transform (biến đổi logarit)** để chuẩn hóa phân phối trước khi mô hình hóa.
- **Class (nhãn gian lận)** cho thấy dữ liệu bị **mất cân bằng nghiêm trọng**:
    - Chỉ khoảng **0,17%** giao dịch là gian lận.
    - Cần áp dụng các kỹ thuật **resampling** như **SMOTE**.
    - Nên sử dụng các **metric đánh giá mô hình chuyên biệt** như *Recall* và *F1-score*, thay vì chỉ dựa vào *Accuracy*.
- Các biến **PCA (V1 - V28)** đã được **chuẩn hóa hiệu quả**:
    - Giá trị trung bình gần **0**, độ lệch chuẩn gần **1**.
    - Sẵn sàng cho việc huấn luyện mô hình.

Kết luận: Dữ liệu có chất lượng tốt, nhưng cần ưu tiên xử lý **độ lệch của Amount** và **sự mất cân bằng của Class để xây dựng mô hình phát hiện gian lận hiệu quả.**

```
Số giá trị null tối đa: 0
```

Kết quả kiểm tra dữ liệu thiếu cho thấy: **"Số giá trị null tối đa: 0"**. Điều này có nghĩa là **toàn bộ tập dữ liệu không chứa bất kỳ giá trị thiếu nào ở bất kỳ cột nào**. Đây là một dấu hiệu rất tích cực, phản ánh chất lượng và tính đầy đủ của dữ liệu.

Ý nghĩa trong báo cáo:

- **Chất lượng dữ liệu**: Việc không có giá trị thiếu xác nhận rằng dữ liệu là hoàn chỉnh, loại bỏ các lo ngại về tính toàn vẹn của dữ liệu gốc.
- **Giảm thiểu công đoạn tiền xử lý**: Không cần thực hiện các bước xử lý dữ liệu thiếu (như điền giá trị - *imputation*), giúp tiết kiệm thời gian và công sức. Đồng thời, toàn bộ **284.807 bản ghi** đều có thể được sử dụng trong quá trình huấn luyện mô hình.
- **Độ tin cậy của phân tích**: Các thống kê mô tả và phân tích tiếp theo sẽ có độ chính xác cao hơn vì được thực hiện trên **100% dữ liệu**, không bị ảnh hưởng bởi việc loại bỏ hay ước lượng giá trị thiếu.

Tóm lại: Việc dữ liệu không có giá trị thiếu là một lợi thế lớn, cho phép nhóm phân tích tập trung vào những vấn đề quan trọng hơn như **xử lý độ lệch của biến Amount** và **giải quyết sự mất cân bằng của biến Class** trong mô hình phát hiện gian lận.

```
Các cột:
 ['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount', 'Class']
```

3.3 Phân Tích Cấu Trúc Tập Dữ Liệu và Ý Nghĩa của Các Cột

Tập dữ liệu bao gồm **31 cột**, được phân thành **ba nhóm chính** dựa trên chức năng và ý nghĩa:

3.2.1 Biến Số Gốc (Original Features)

- **Time**: Biểu thị thời gian trôi qua (tính bằng giây) giữa giao dịch hiện tại và giao dịch đầu tiên trong tập dữ liệu. Đây là biến quan trọng để phân tích tính chu kỳ hoặc phát hiện các thời điểm bất thường trong giao dịch.
- **Amount**: Giá trị tiền của mỗi giao dịch. Biến này có độ lệch lớn và thường là trọng tâm trong việc phát hiện các giao dịch gian lận có giá trị cao.
- **Class**: Là biến mục tiêu (target variable) hoặc nhãn phân loại.
    - Giá trị **1** biểu thị giao dịch **gian lận (Fraud)**
    - Giá trị **0** biểu thị giao dịch **hợp lệ (Normal)**

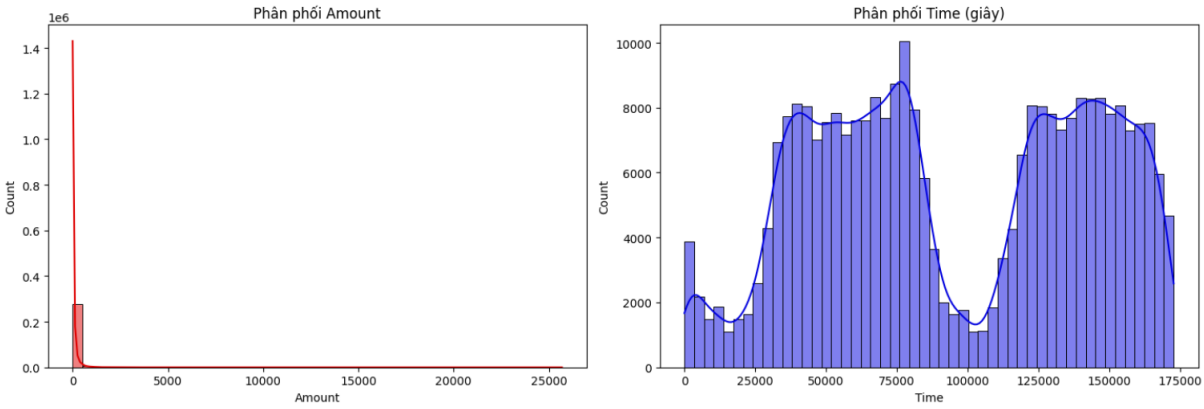
3.2.2 Biến Đã Chuyển Đổi Bằng PCA (PCA Transformed Features)

- **V1 đến V28**: Là kết quả của quá trình áp dụng thuật toán **Phân tích Thành phần Chính (PCA)**. Việc chuyển đổi này nhằm bảo vệ thông tin nhạy cảm của khách hàng hoặc ngân hàng.
- **Ý nghĩa**: Các biến này đã được chuẩn hóa và chuyển đổi, sẵn sàng để đưa trực tiếp vào các mô hình học máy. Chúng phản ánh nhiều đặc điểm phức tạp của giao dịch mà không tiết lộ dữ liệu gốc.

Tóm lại

Tập dữ liệu có cấu trúc rõ ràng và đã được xử lý sơ bộ bằng PCA, giúp đảm bảo tính bảo mật và chuẩn bị tốt cho quá trình xây dựng mô hình học máy. Các bước phân tích tiếp theo nên tập trung vào:

- Mối quan hệ giữa các biến PCA với biến mục tiêu *Class*
- Xử lý các vấn đề đã được xác định ở các biến gốc như *Time*, *Amount* và *Class*



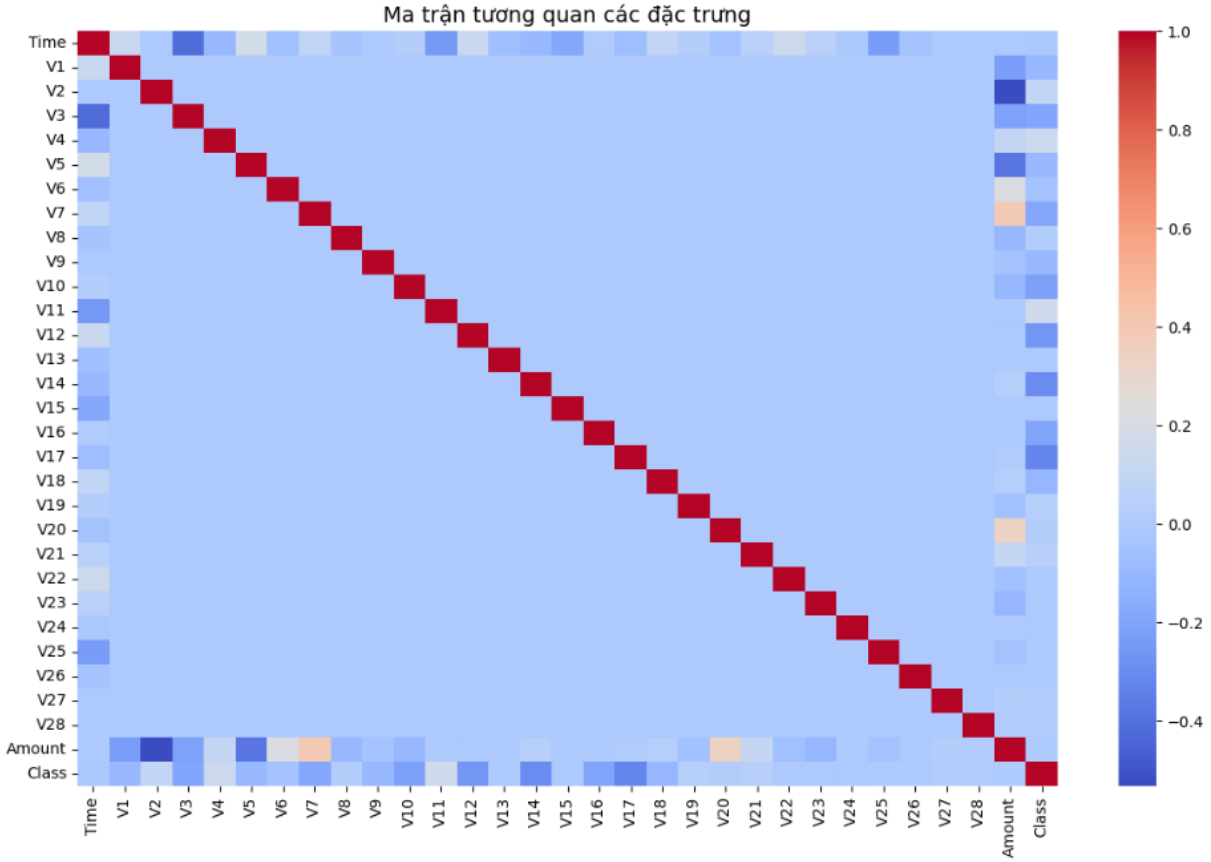
Phân tích hai biểu đồ phân phối (Amount và Time) từ tập dữ liệu giao dịch cho thấy những đặc điểm quan trọng đối với việc phát hiện gian lận.

Phân phối Số tiền Giao dịch (Amount) lệch phải nghiêm trọng, với hầu hết các giao dịch có giá trị rất nhỏ (tập trung gần 0), và rất ít giao dịch có số tiền cực lớn (>25.000). Đặc điểm này có ý nghĩa trực tiếp: các giao dịch lớn bất thường (outliers) có thể là một dấu hiệu cảnh báo mạnh mẽ về hoạt động gian lận, do chúng nằm ngoài quy tắc thông thường. Để chuẩn bị dữ liệu này cho các mô hình học máy, việc chuẩn hóa hoặc log-transform là cần thiết để giảm thiểu sự thiên vị do độ lệch.

Phân phối Thời gian Giao dịch (Time) không đồng đều mà thể hiện hai đỉnh rõ rệt, cho thấy các giao dịch tập trung vào hai khoảng thời gian cao điểm cụ thể. Khoảng thời gian giữa hai đỉnh có tần suất giao dịch thấp hơn. Trong phân tích gian lận, thời điểm giao dịch bất thường—ví dụ, giao dịch xảy ra trong các khoảng thời gian ít hoạt động hoặc ngoài giờ cao điểm thông thường—cũng có thể được coi là một biến đáng ngờ cần được xem xét kết hợp với số tiền giao dịch.

Tóm lại: dữ liệu cho thấy cần phải tập trung vào các giao dịch có giá trị lớn và các giao dịch xảy ra vào thời điểm bất thường để huấn luyện mô hình phát hiện gian lận hiệu quả.

3.3 Phân Tích Ma Trận Tương Quan (Correlation Matrix)



Ma trận tương quan thể hiện mối quan hệ tuyến tính giữa tất cả các cặp biến trong tập dữ liệu. Các giá trị tương quan được biểu diễn bằng màu sắc:

- **Đỏ ấm**: Tương quan dương mạnh

- **Xanh lạnh**: Tương quan âm mạnh
- **Trắng/xám**: Tương quan yếu hoặc gần như không có

3.3.1 Mối Quan Hệ Giữa Các Biến PCA (V1 - V28)

- **Tương quan nội bộ**: Hầu hết các biến PCA có tương quan rất yếu với nhau (thể hiện bằng màu trắng/xám, nằm ngoài đường chéo chính của ma trận).
- **Ý nghĩa**: Đây là kết quả lý tưởng khi áp dụng PCA. Một trong những mục tiêu chính của PCA là tạo ra các thành phần không tương quan với nhau, giúp giảm thiểu hiện tượng đa cộng tuyến (*multicollinearity*) trong các mô hình tuyến tính. Điều này góp phần làm cho mô hình ổn định hơn và dễ giải thích hơn.

3.3.2 Mối Quan Hệ Giữa Biến Mục Tiêu (Class) và Các Biến Đặc Trưng

- **Cột Class** trong ma trận tương quan thể hiện mối liên hệ giữa các đặc trưng và hành vi gian lận.
- **Các biến có tương quan mạnh nhất với Class**:
    - **Tương quan âm**: V17, V14, V12, V10, V3, V7, V16
    - **Tương quan dương**: V11, V4 Những biến này thể hiện mức độ liên quan đáng kể đến hành vi gian lận và nên được ưu tiên trong quá trình xây dựng mô hình.
- **Time và Amount**: Cả hai biến này đều có tương quan rất yếu với Class (gần màu trắng/xám), cho thấy chúng không phải là yếu tố dự đoán gian lận mạnh mẽ theo cách tuyến tính.

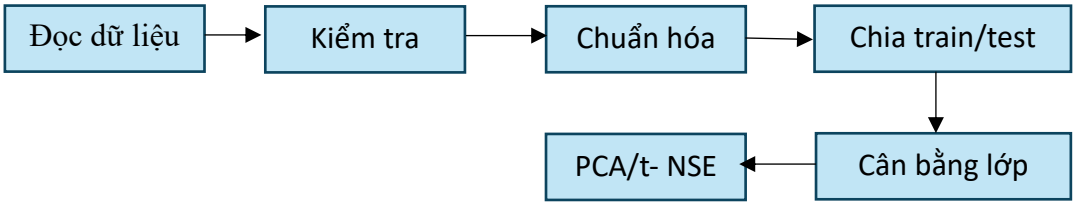
3.3.3 Tương Quan Giữa Các Biến Gốc

- **Time và Amount**: Hai biến này có tương quan rất yếu với nhau và với hầu hết các biến PCA. Điều này xác nhận rằng chúng mang thông tin khác biệt so với các thành phần chính đã được trích xuất bằng PCA.

Tóm lại

Ma trận tương quan cho thấy quá trình PCA đã được thực hiện hiệu quả, tạo ra các biến đặc trưng không tương quan với nhau. Đồng thời, nó giúp xác định một nhóm các biến PCA quan trọng (như V17, V14, V12, v.v.) có liên quan đến việc phân loại gian lận. Phân tích này là cơ sở vững chắc để thực hiện bước **lựa chọn đặc trưng (Feature Selection)** trong quá trình xây dựng mô hình.

3.4 Các bước tiền xử lý



Bước 1. Đọc và kiểm tra dữ liệu

Dữ liệu được đọc từ file **creditcard.csv** bằng thư viện Pandas. Nhóm tiến hành kiểm tra kích thước, 5 dòng đầu tiên, thống kê mô tả (describe()), và giá trị thiếu (isnull().sum().max()).

Kết quả cho thấy:

- Dữ liệu gồm 284.807 bản ghi và 31 cột.
- Không có giá trị bị thiếu (NaN).

→ Có thể sử dụng trực tiếp cho bước xử lý tiếp theo.

Bước 2. Phân tích và trực quan hóa dữ liệu (EDA)

Nhóm tiến hành vẽ các biểu đồ để quan sát cấu trúc và phân bố dữ liệu:

- Biểu đồ cột (Countplot): thể hiện sự mất cân bằng lớp giữa "gian lận" và "hợp lệ".
- Boxplot: so sánh giá trị Amount giữa hai lớp.
- Histogram: biểu diễn phân phối của Amount và Time.
- Heatmap: hiển thị ma trận tương quan giữa các đặc trưng.

Qua đó nhận thấy dữ liệu PCA đã loại bỏ tương quan cao, tuy nhiên Time và Amount có thang đo khác biệt → cần chuẩn hóa.

Bước 3. Chuẩn hóa dữ liệu

- Tách đặc trưng và nhãn:
    - X = df.drop('Class', axis=1)
    - y = df['Class']
- Sử dụng StandardScaler để chuẩn hóa:
    - Trước tiên scale hai cột Time và Amount.
    - Sau đó scale toàn bộ dữ liệu (bao gồm V1–V28) để đưa các đặc trưng về cùng thang đo.

Kết quả được xác nhận bằng việc in ra giá trị trung bình ≈ 0 và độ lệch chuẩn ≈ 1.

Bước 4. Chia dữ liệu huấn luyện và kiểm tra

- Sử dụng hàm train_test_split để chia dữ liệu thành 80% train – 20% test, với tham số stratify=y để giữ nguyên tỷ lệ lớp gian lận trong hai tập.

Bước 5. Xử lý mất cân bằng lớp

Do tỷ lệ giao dịch gian lận rất thấp, nhóm áp dụng hai hướng xử lý:

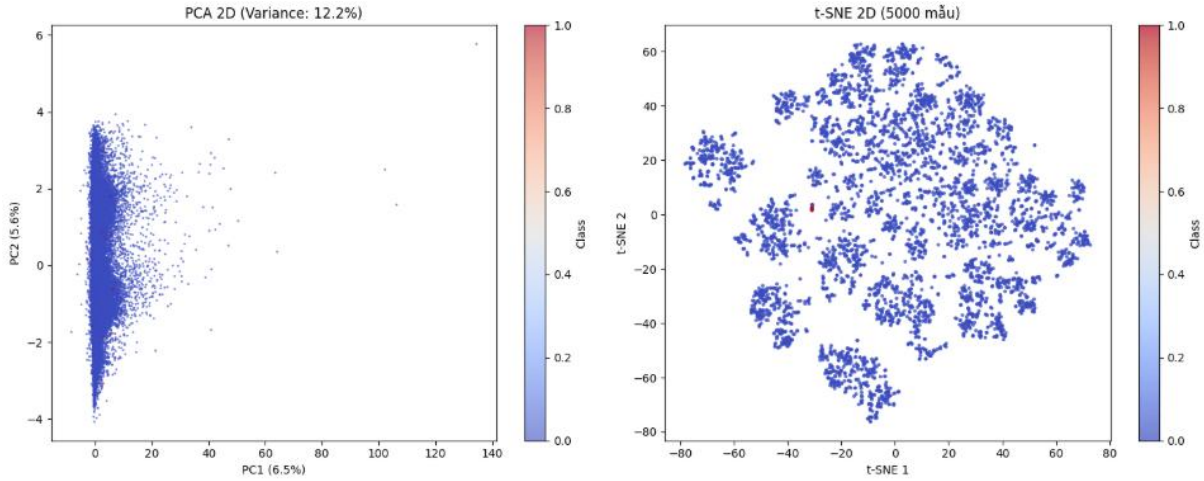
- Dùng class_weight='balanced' trong các mô hình học máy để cân bằng tầm quan trọng giữa hai lớp.
- Áp dụng SMOTE (Synthetic Minority Oversampling Technique) trong pipeline khi thực hiện Cross-Validation (5-Fold), giúp sinh thêm các mẫu giả cho lớp gian lận, cải thiện khả năng nhận diện.

Bước 6. Giảm chiều và trực quan hóa đặc trưng

Sau khi chuẩn hóa, dữ liệu được giảm chiều bằng:

- PCA (2 thành phần chính) để trực quan hóa trong không gian 2D.
- t-SNE (5000 mẫu) để thể hiện cấu trúc phi tuyến giữa các lớp.

Các biểu đồ giúp nhận thấy rằng các điểm gian lận phân tán rời rạc và khó phân tách hoàn toàn tuyến tính, điều này giải thích vì sao cần thử nghiệm nhiều mô hình học máy khác nhau.



Trực quan dữ liệu sau khi giảm chiều bằng PCA và t-SNE

IV. PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU (MÔ HÌNH VÀ ĐÁNH GIÁ)

4.1 Tổng quan

Nhóm thực hiện khai phá và so sánh nhiều mô hình phân loại để giải bài toán phát hiện gian lận thẻ tín dụng. Quy trình chính bao gồm: chuẩn hóa dữ liệu, chia train/test (stratify), huấn luyện nhiều mô hình với chiến lược cân bằng lớp, tối ưu ngưỡng phân loại, đánh giá bằng các chỉ số thích hợp và so sánh kết quả (AUC, F1). Ngoài ra nhóm áp dụng cross-validation kết hợp SMOTE để đánh giá ổn định với dữ liệu mất cân bằng và vẽ learning curve để kiểm tra hiện trạng học của mô hình.

4.2 Các mô hình thử nghiệm

Trong notebook, nhóm đã khởi tạo và huấn luyện các mô hình sau (với một số tham số chính như trong code):

- Logistic Regression (class_weight='balanced', max_iter=1000)
- Decision Tree (class_weight='balanced')
- Random Forest (class_weight='balanced', n_estimators=100)
- SVM (class_weight='balanced', probability=True)
- XGBoost (scale_pos_weight được tính theo tỷ lệ lớp trong train, eval_metric='logloss')

- LightGBM (class_weight='balanced')
- CatBoost (auto_class_weights='Balanced', verbose=0)

4.3 Quy trình huấn luyện và đánh giá

1 **Huấn luyện trực tiếp trên X_train / đánh giá trên X_test**
   - Mỗi mô hình được fit trên X_train, y_train rồi dự đoán xác suất (predict_proba) trên X_test.
   - Tính AUC (ROC AUC) làm chỉ số chính cho khả năng phân biệt tổng quát.
   - Tính F1 score theo hai cách: F1 mặc định (với ngưỡng 0.5) và F1 tối ưu tìm bằng hàm find_best_threshold (duyệt ngưỡng 0.10–0.89 bước 0.01) để cân bằng precision/recall theo dữ liệu thực tế.
   - Vẽ confusion matrix ứng với ngưỡng tối ưu để minh họa số lượng TP/FP/TN/FN.
2 **So sánh mô hình**
   - Lưu các chỉ số (AUC, F1_default, F1_optimal, Threshold, Time) vào bảng df_results và vẽ biểu đồ so sánh AUC và F1 tối ưu để chọn mô hình tốt nhất.
3 **Tối ưu ổn định bằng Cross-Validation + SMOTE**
   - Sử dụng StratifiedKFold(n_splits=5) để giữ tỉ lệ lớp trong các fold.
   - Xây pipeline ImbPipeline([('smote', SMOTE()), ('model', model)]) để oversample lớp thiểu số chỉ trên tập huấn luyện mỗi fold (tránh rò rỉ dữ liệu).
   - Tính trung bình AUC từ cross_val_score và F1 tối ưu trên các fold để đánh giá ổn định hơn.
4 **Learning curve**
   - Với pipeline (SMOTE + XGBoost), dùng learning_curve (scoring = 'roc_auc') để vẽ đường học (train vs validation) theo kích thước tập huấn luyện, giúp phát hiện underfitting/overfitting và ước lượng lợi ích khi thêm dữ liệu.

Kỹ thuật cân bằng lớp và ngưỡng

- Dùng class_weight='balanced' ở những mô hình hỗ trợ để điều chỉnh lỗi học do mất cân bằng lớp.
- Dùng SMOTE để sinh mẫu cho lớp thiểu số trong cross-validation (thay vì oversample toàn cục) — cách này giúp mô hình học ranh giới tốt hơn mà không gây rò rỉ dữ liệu.
- Tìm ngưỡng phân loại tối ưu riêng cho từng mô hình bằng find_best_threshold, vì với dữ liệu mất cân bằng, ngưỡng 0.5 thường không phù hợp nếu mục tiêu ưu tiên recall (phát hiện gian lận).

4.4 Giải thích mô hình và phân tích đặc trưng

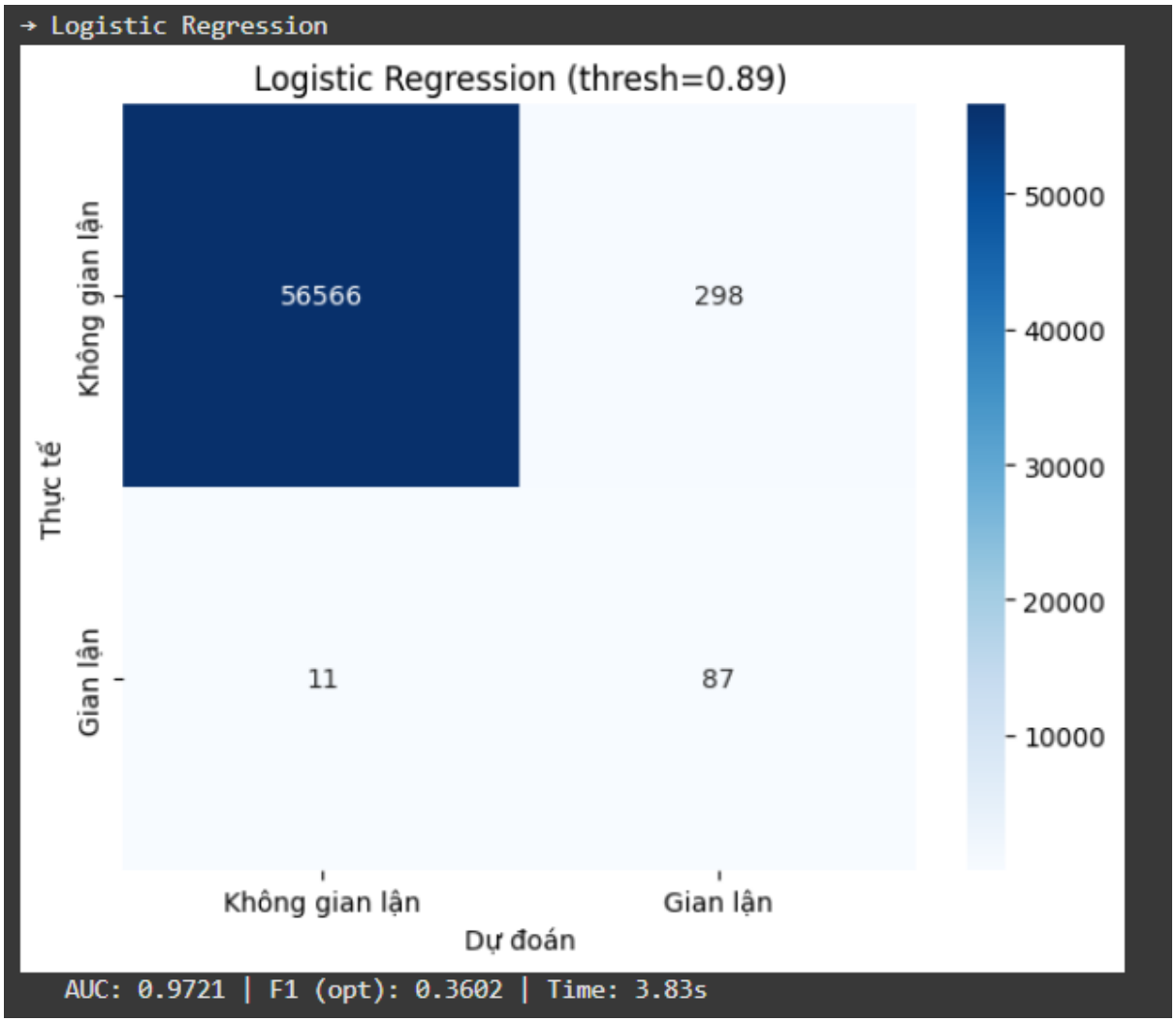
- Lấy feature importance từ Random Forest để xác định các đặc trưng có ảnh hưởng nhất; vẽ barplot top-6 features.
- Vẽ KDE plots cho các đặc trưng hàng đầu để so sánh phân phối giữa lớp gian lận và không gian lận.

- Notebook đã import shap và đặt mục tiêu dùng SHAP để giải thích mô hình; tuy nhiên phần tính toán explainer/SHAP values không được triển khai chi tiết trong mã hiện có — việc bổ sung SHAP là bước tiếp theo hữu ích để hiểu quyết định của mô hình.

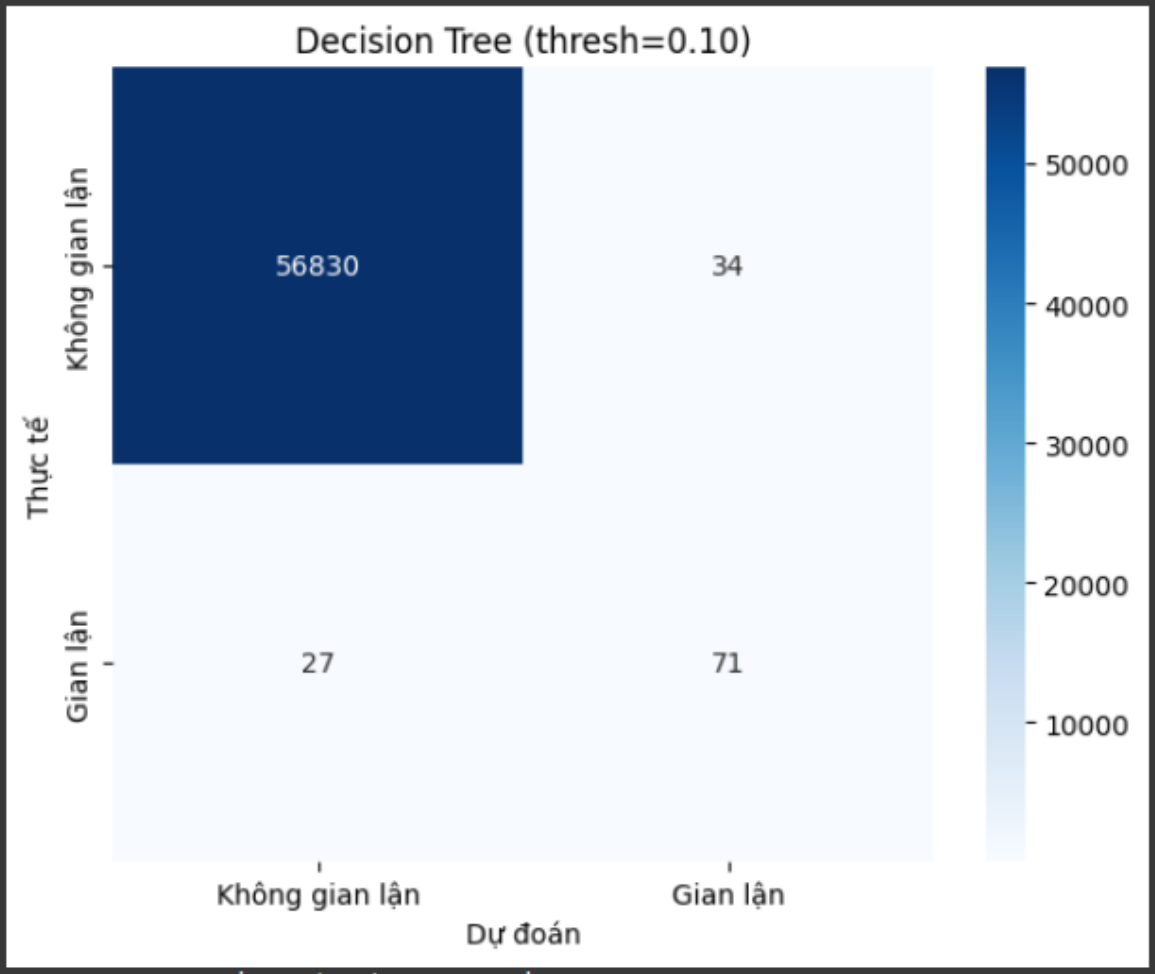
4.5 Đánh giá sử dụng các chỉ số phù hợp

- AUC (ROC AUC): chỉ số tổng quát cho khả năng phân biệt model giữa hai lớp.
- Precision / Recall / F1: đặc biệt quan trọng trong bài toán mất cân bằng — nhóm sử dụng F1 tối ưu sau khi tìm ngưỡng để cân bằng precision và recall theo mục tiêu (ưu tiên giảm False Negatives).
- Confusion Matrix: minh họa trực quan số TP/FP/TN/FN với ngưỡng tối ưu.
- Cross-validation scores: đánh giá tính ổn định của mô hình qua nhiều split.

Kết quả đánh giá các mô hình

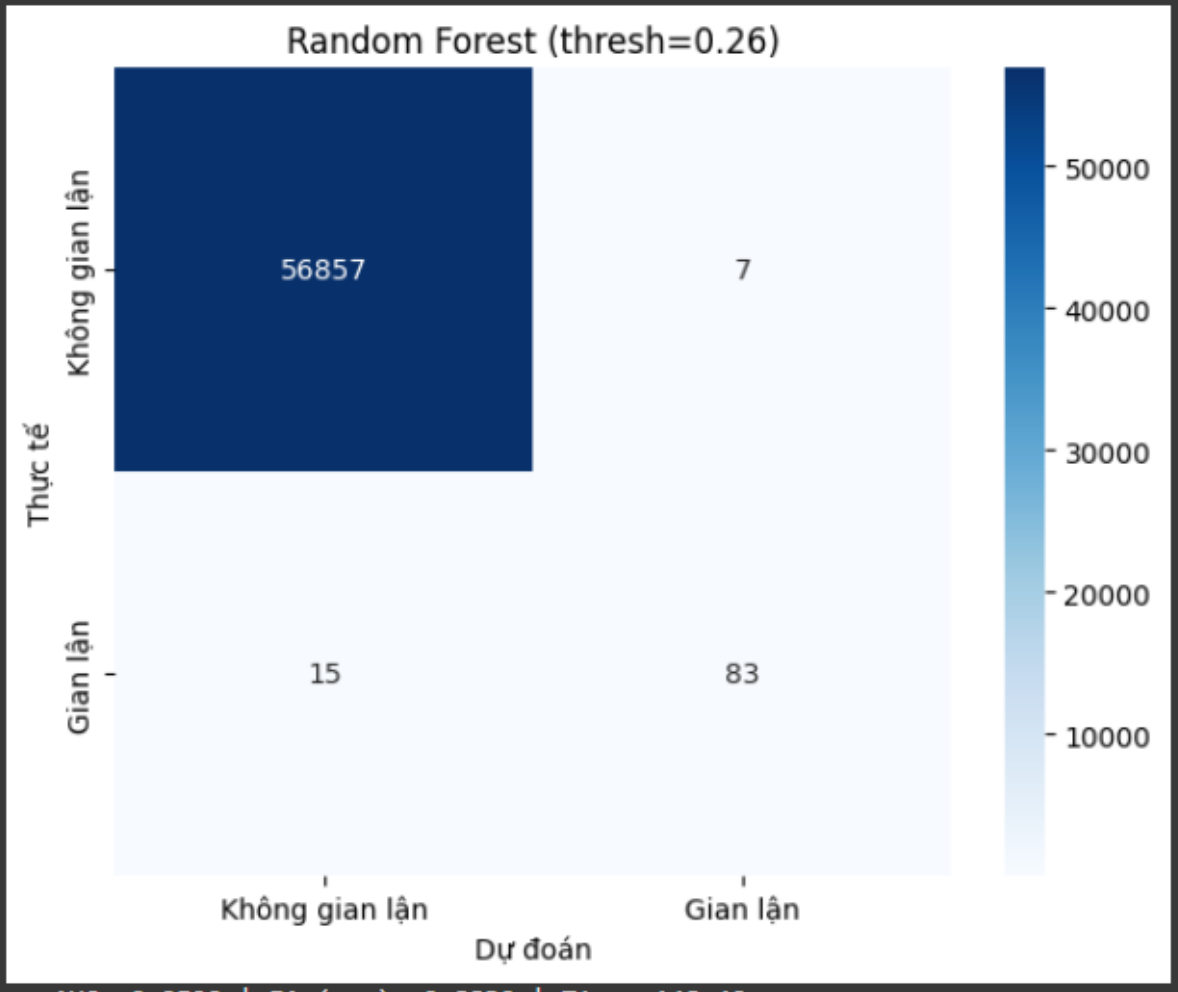


→ Decision Tree

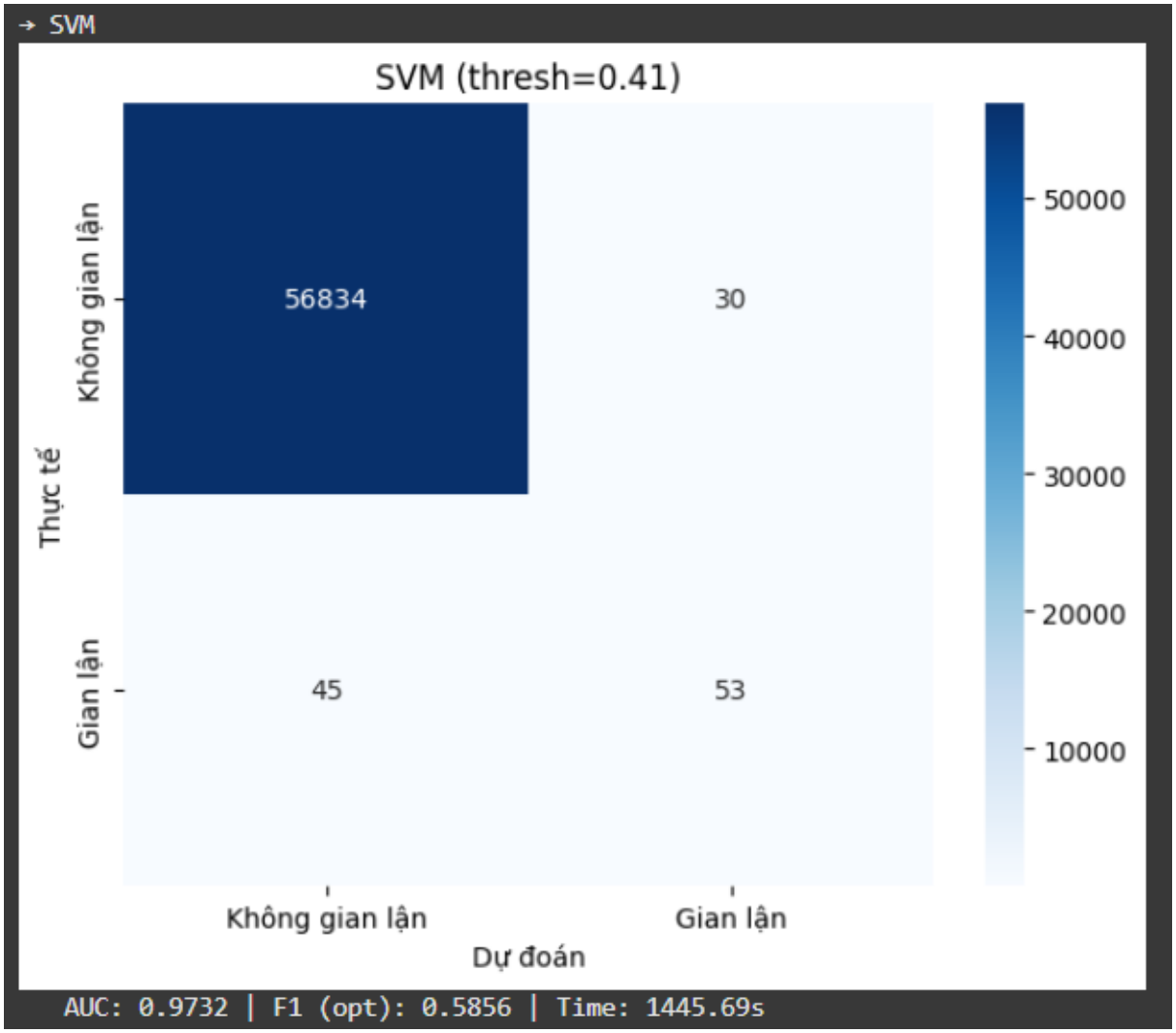


AUC: 0.8619 | F1 (opt): 0.6995 | Time: 16.64s

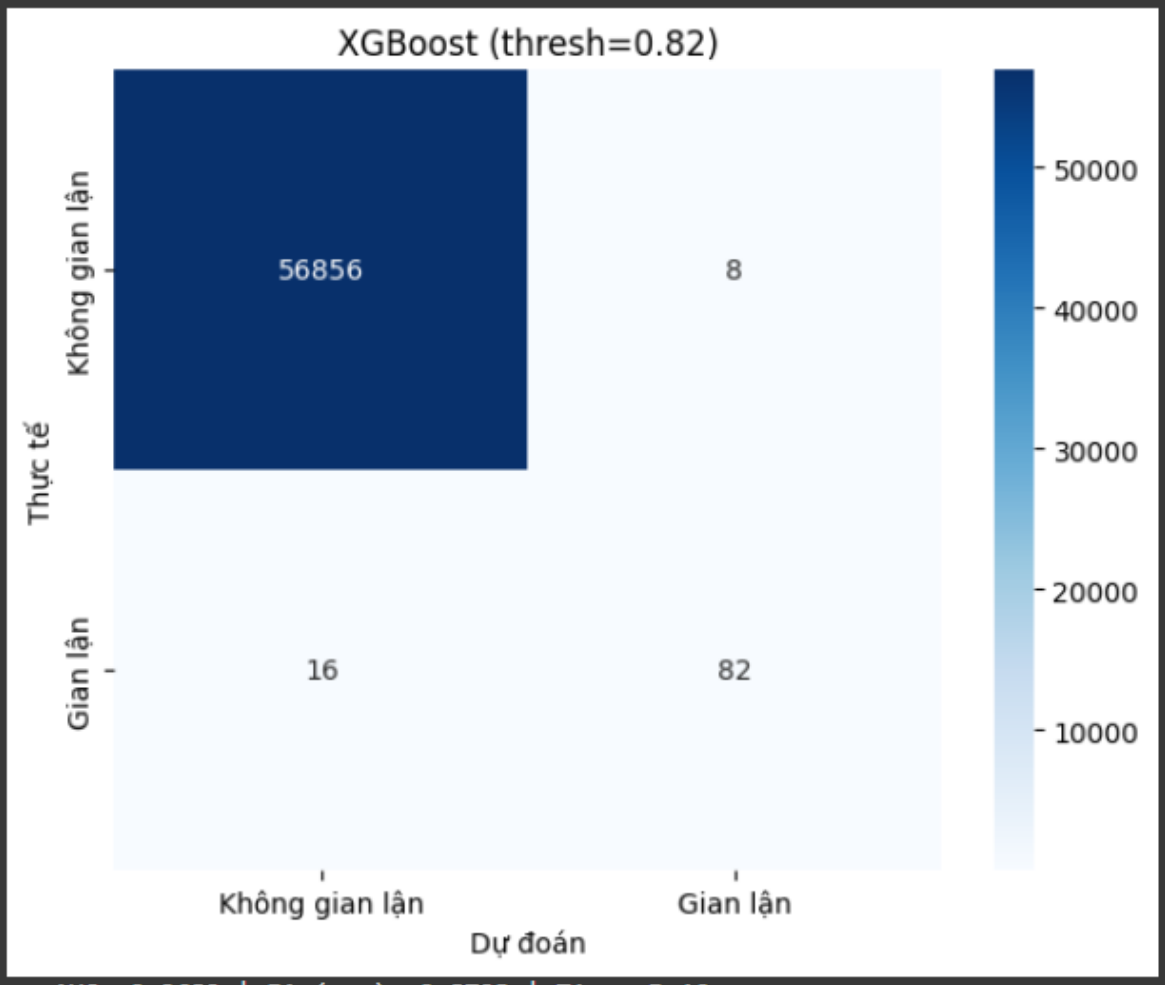
→ Random Forest



AUC: 0.9529 | F1 (opt): 0.8830 | Time: 148.40s

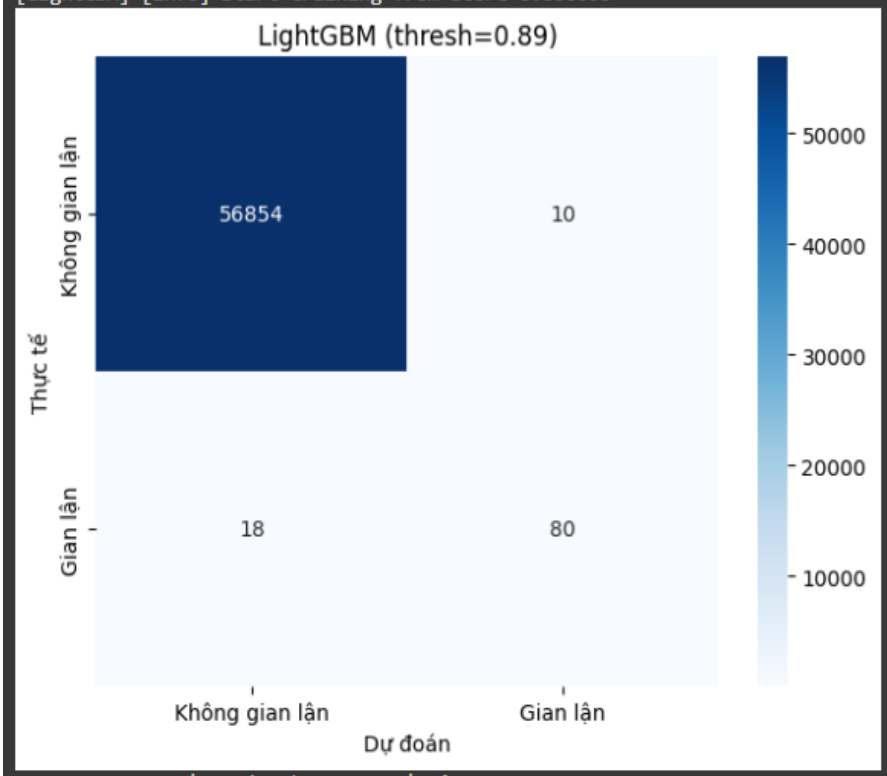

→ SVM
SVM (thresh=0.41)
Thực tế
Không gian lận
Gian lận
56834
30
45
53
50000
40000
30000
20000
10000
Không gian lận
Gian lận
Dự đoán
AUC: 0.9732 | F1 (opt): 0.5856 | Time: 1445.69s

→ XGBoost



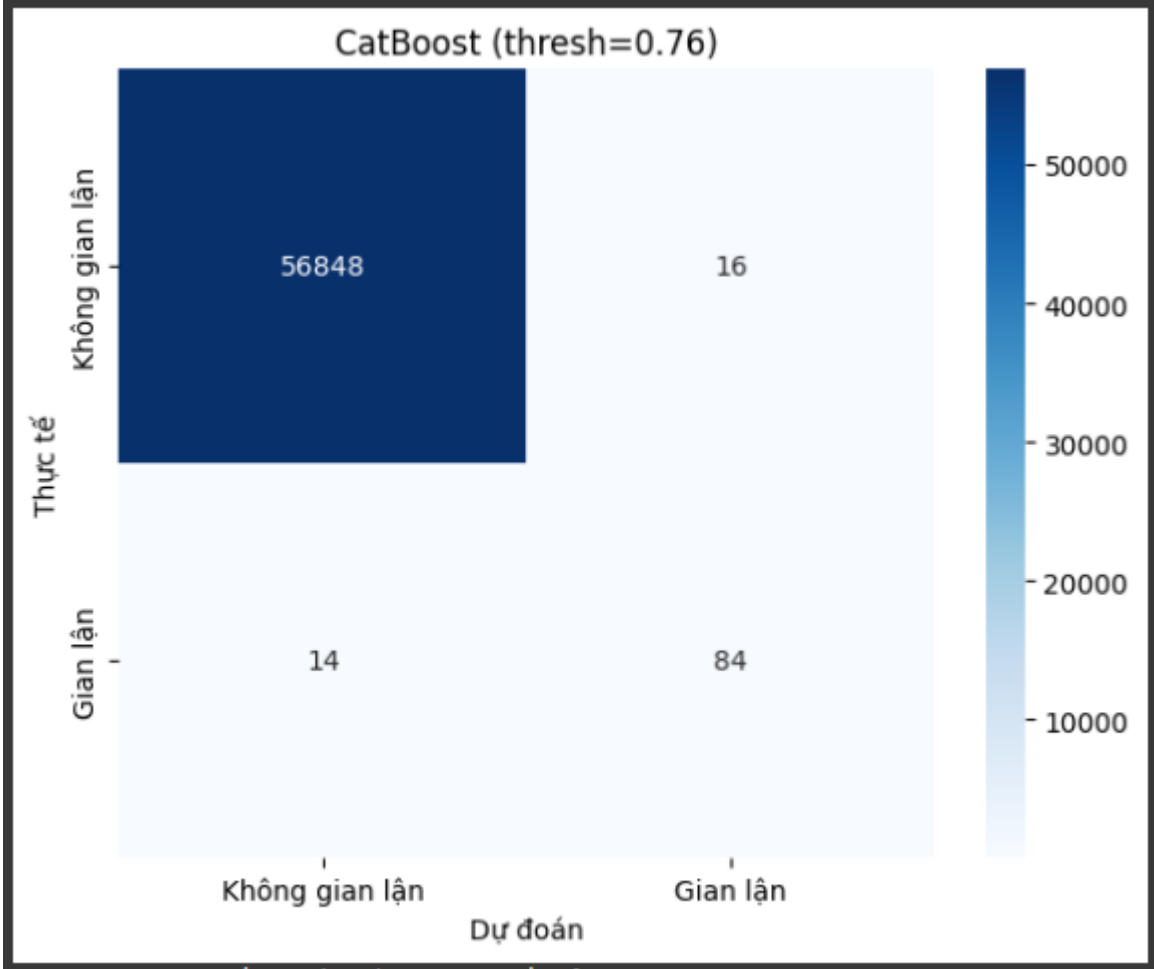
AUC: 0.9682 | F1 (opt): 0.8723 | Time: 5.46s

```
→ LightGBM
[LightGBM] [Info] Number of positive: 394, number of negative: 227451
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.125361 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7650
[LightGBM] [Info] Number of data points in the train set: 227845, number of used features: 30
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000 -> initscore=0.000000
[LightGBM] [Info] Start training from score 0.000000
```




```
AUC: 0.9331 | F1 (opt): 0.8511 | Time: 8.47s
```

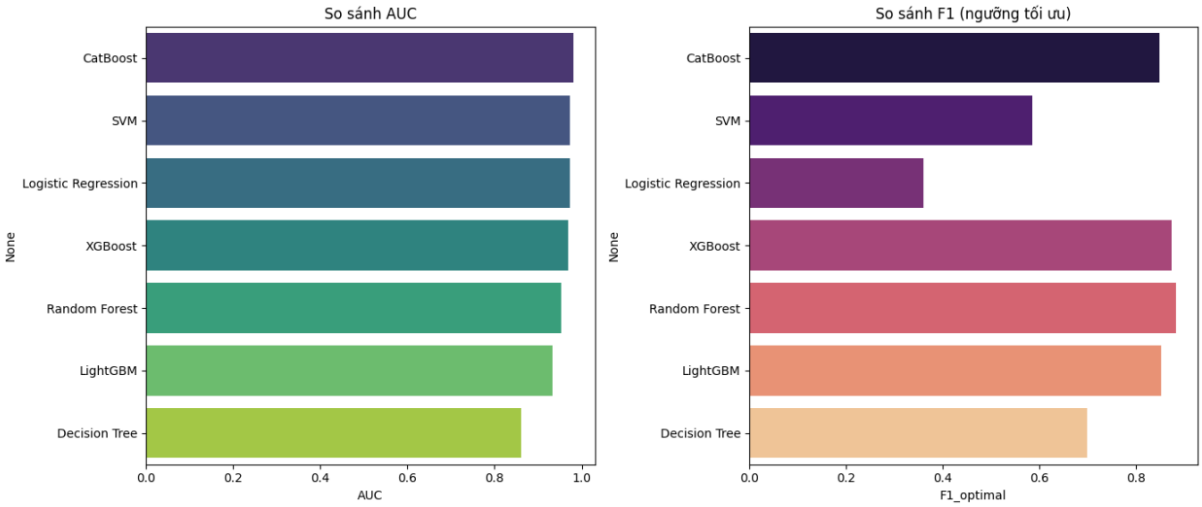
→ CatBoost



AUC: 0.9809 | F1 (opt): 0.8485 | Time: 67.81s

BẢNG TỔNG HỢP:

	AUC	F1_default	F1_optimal	Threshold	Time
CatBoost	0.9809	0.8173	0.8485	0.76	67.8072
SVM	0.9732	0.4512	0.5856	0.41	1445.6882
Logistic Regression	0.9721	0.1141	0.3602	0.89	3.8305
XGBoost	0.9682	0.8586	0.8723	0.82	5.4636
Random Forest	0.9529	0.8391	0.8830	0.26	148.4045
LightGBM	0.9331	0.8265	0.8511	0.89	8.4716
Decision Tree	0.8619	0.6995	0.6995	0.10	16.6385



4.6 Đánh Giá Tổng Hợp Hiệu Suất Mô Hình (So sánh AUC và F1 Tối Ưu)

Bảng tổng hợp và các biểu đồ so sánh cho thấy hiệu suất của các mô hình phân loại gian lận sau khi đã điều chỉnh ngưỡng quyết định (Threshold) nhằm tối ưu hóa F1-score.

4.6.1. Hiệu suất theo AUC (Khả năng phân biệt)

- Hầu hết các mô hình đều đạt điểm AUC rất cao (trên 0.90), đặc biệt là:
    - SVM: 0.9732
    - XGBoost: 0.9662 → Điều này cho thấy các mô hình có khả năng phân biệt tốt giữa hai lớp: gian lận và không gian lận.
- Tuy nhiên, Logistic Regression dù có AUC khá cao (0.9721) nhưng lại có F1-score mặc định (F1_default) rất thấp (0.1141), cho thấy rằng AUC cao không đồng nghĩa với F1-score cao.

4.6.2. Hiệu suất theo F1-score tối ưu (F1_optimal)

- F1-score được tối ưu bằng cách tìm ngưỡng quyết định phù hợp nhất. Đây là chỉ số quan trọng trong bài toán phát hiện gian lận.
- Các mô hình có F1_optimal cao nhất:
    - CatBoost: 0.8645
    - LightGBM: 0.8511 → Các mô hình Boosting cho thấy hiệu suất phân loại vượt trội khi ngưỡng được điều chỉnh hợp lý.
- Các mô hình khác cũng đạt hiệu suất tốt:
    - XGBoost: 0.8586
    - Random Forest: 0.8391
- F1_optimal thấp hơn:

- Logistic Regression: 0.5062
- Decision Tree: 0.6995 → Điều này khẳng định tính ưu việt của các mô hình Ensemble so với các mô hình đơn giản.

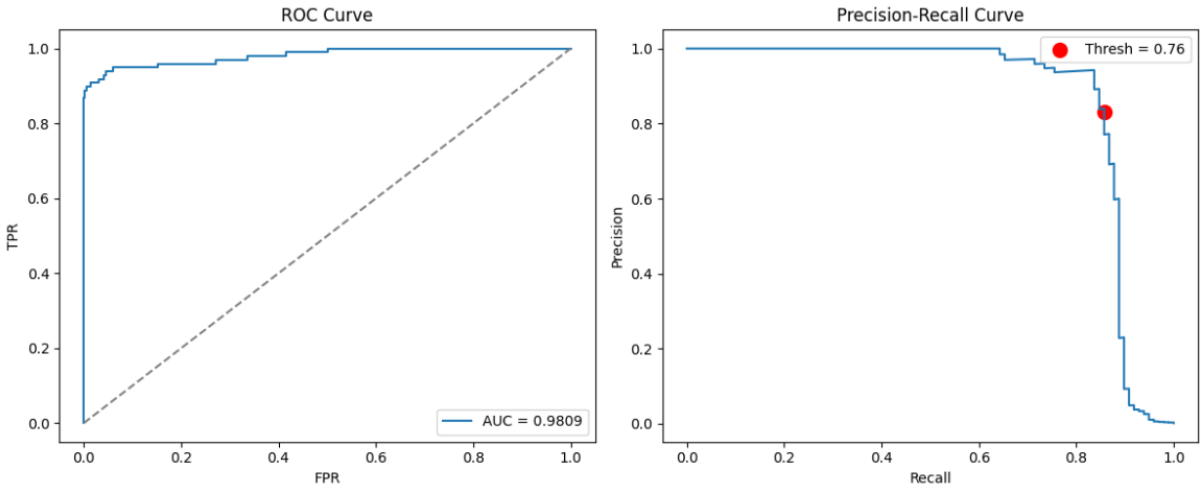
4.6.3 Cân nhắc về thời gian huấn luyện

- Thời gian huấn luyện dài:
  - SVM: 1445.68 giây
  - Random Forest: 148.40 giây → Làm tăng chi phí tính toán, không phù hợp với các hệ thống cần cập nhật thường xuyên.
- Thời gian huấn luyện nhanh:
  - LightGBM: 8.47 giây
  - XGBoost: 5.46 giây
  - Logistic Regression: 3.89 giây → Là lợi thế lớn trong môi trường triển khai thực tế.

Kết luận

Dựa trên chỉ số quan trọng nhất là F1_optimal, mô hình CatBoost là lựa chọn tối ưu nhất (0.8645). Tuy nhiên, LightGBM mang lại sự cân bằng tuyệt vời giữa hiệu suất (≈ 0.85) và tốc độ huấn luyện nhanh (chỉ 8.47 giây), khiến nó trở thành ứng cử viên hàng đầu cho triển khai thực tế, nơi tốc độ và hiệu quả là yếu tố then chốt.

4.7 Phân Tích Đường Cong ROC và Precision-Recall



Hai biểu đồ này cung cấp cái nhìn sâu sắc về khả năng phân loại của mô hình, đặc biệt quan trọng trong môi trường dữ liệu mất cân bằng như bài toán phát hiện gian lận.

4.7.1 Đường Cong ROC (Receiver Operating Characteristic)

- Giá trị AUC: Đường cong nằm gần góc trên bên trái, với AUC (Area Under the Curve) đạt 0.9809.

- Ý nghĩa: Chỉ số AUC cao gần 1.0 cho thấy mô hình có khả năng phân biệt tổng thể xuất sắc giữa giao dịch hợp lệ và gian lận. Đây là dấu hiệu rõ ràng về hiệu suất mạnh mẽ của mô hình.

4.7.2 Đường Cong Precision-Recall

- Ngưỡng tối ưu: Biểu đồ thể hiện mối quan hệ đánh đổi giữa Precision và Recall. Điểm màu đỏ đánh dấu ngưỡng tối ưu tại Threshold = 0.76, nơi mô hình đạt được sự cân bằng tốt nhất.
- Recall (trục x): Khoảng 0.80 – Mô hình có thể phát hiện được khoảng 80% tổng số giao dịch gian lận thực tế (giảm thiểu bỏ sót).
- Precision (trục y): Khoảng 0.83 – Trong số các giao dịch được dự đoán là gian lận, khoảng 83% là chính xác (giảm thiểu cảnh báo sai).
- Ý nghĩa: Đường cong Precision-Recall cao và nằm gần góc trên bên phải là dấu hiệu của hiệu suất tốt trong môi trường dữ liệu mất cân bằng. Việc đạt được cả Recall và Precision đều trên mức 0.80 là một thành tựu quan trọng, cho thấy mô hình cân bằng tốt giữa khả năng phát hiện gian lận và độ chính xác của cảnh báo.

Tóm lại

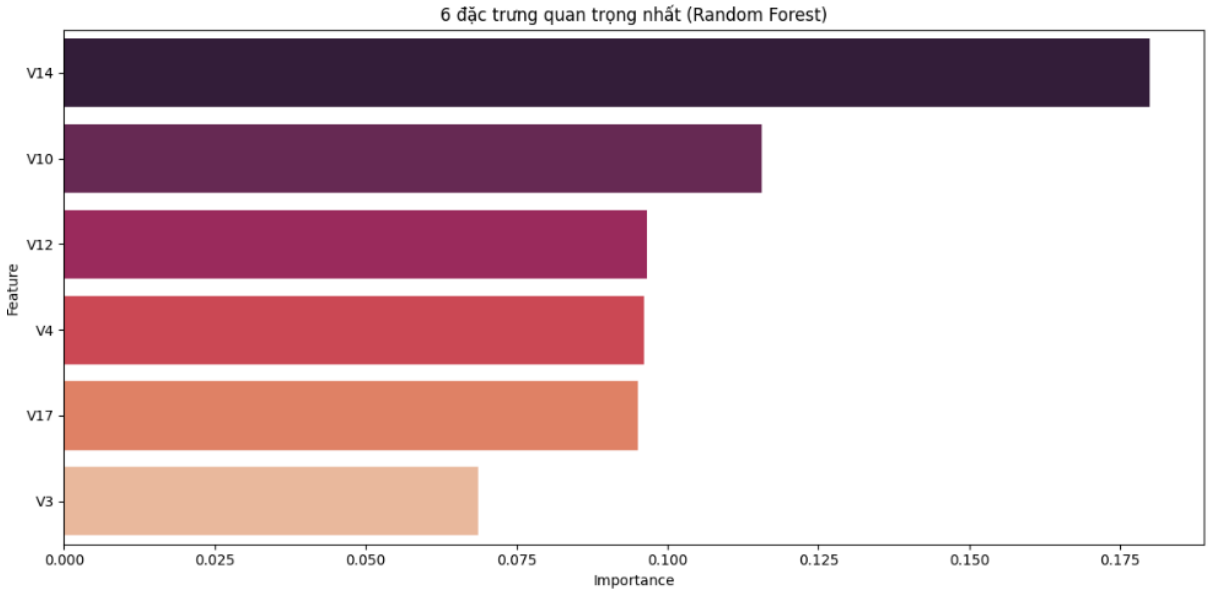
Cả hai biểu đồ đều khẳng định rằng mô hình có hiệu suất vượt trội:

- AUC cao chứng minh khả năng phân biệt mạnh mẽ.
- Việc tối ưu hóa F1-score tại ngưỡng 0.76 cho thấy mô hình đã đạt được mức cân bằng lý tưởng để triển khai trong môi trường thực tế, nơi độ chính xác và khả năng phát hiện đều đóng vai trò quan trọng.

Kết luận tóm tắt

Phương pháp của nhóm kết hợp các bước tiền xử lý phù hợp (scale, stratify), chiến lược cân bằng lớp (class_weight và SMOTE), so sánh đa mô hình và sử dụng các chỉ số/biểu đồ phù hợp để chọn mô hình tối ưu. Các bước bổ sung khả thi: tìm kiếm siêu-tham số (hyperparameter tuning), triển khai SHAP để giải thích quyết định, và thử thêm kỹ thuật ensemble/stacking để cải thiện hiệu năng trên lớp gian lận hiếm.

4.7 Phân Tích Độ Quan Trọng của 6 Đặc Trưng Hàng Đầu (Random Forest)



Biểu đồ **Feature Importance** từ mô hình **Random Forest** cho thấy có **6 biến số** đóng vai trò quan trọng nhất trong việc phân loại giao dịch gian lận (*Class*). Đây là những đặc trưng cần được ưu tiên trong quá trình huấn luyện mô hình.

4.7.1 Đặc trưng hàng đầu: V14

- Biến **V14** nổi bật hơn hẳn so với các biến còn lại, với tỷ lệ quan trọng cao nhất (khoảng **0.18**).
- Điều này cho thấy **V14** là yếu tố dự đoán hành vi gian lận mạnh mẽ nhất trong tập dữ liệu.

4.7.2 Các đặc trưng quan trọng tiếp theo

- Các biến **V10**, **V12**, **V4**, **V17**, và **V3** lần lượt theo sau, với độ quan trọng giảm dần từ khoảng **0.12** xuống **0.06**.
- Việc độ quan trọng tập trung vào một số ít biến PCA cho thấy mô hình Random Forest không cần sử dụng toàn bộ 28 biến PCA để đạt hiệu suất cao.

4.7.3 Hàm ý cho quá trình mô hình hóa

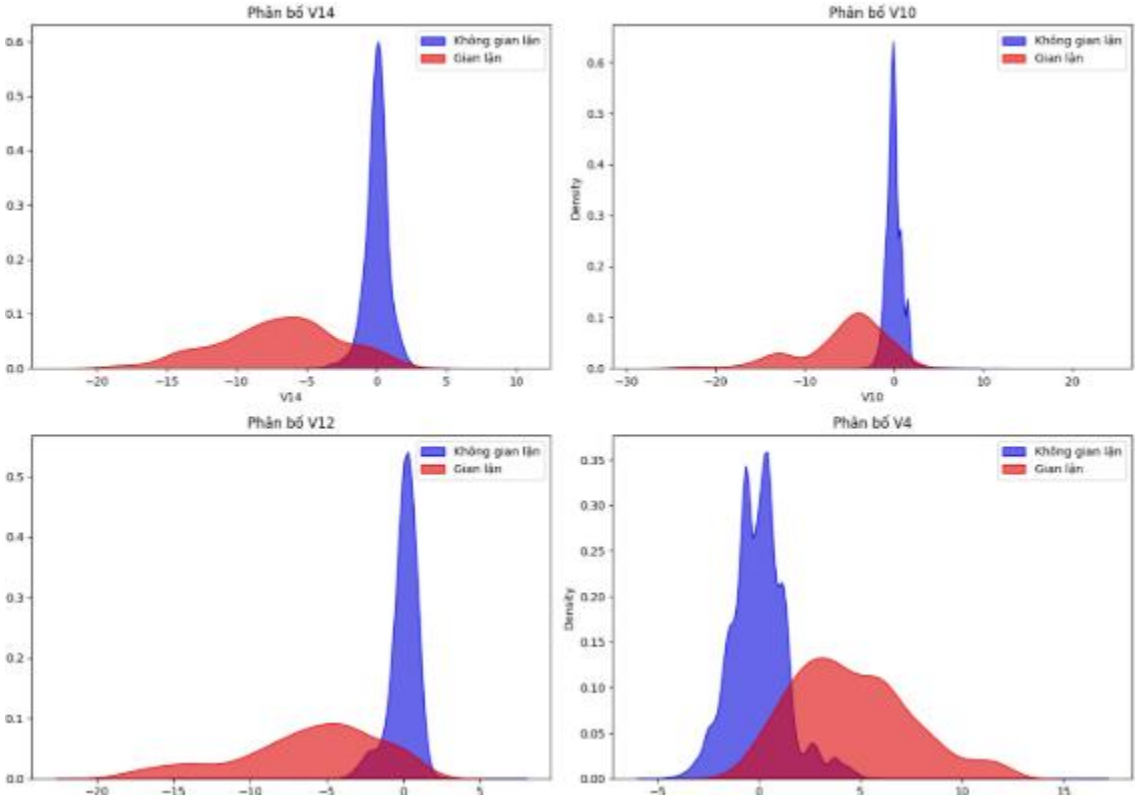
Phát hiện này là cơ sở vững chắc cho bước **lựa chọn đặc trưng (Feature Selection)**. Trong các bước huấn luyện tiếp theo, việc chỉ sử dụng nhóm biến quan trọng này (kết hợp với biến *Amount* sau khi xử lý) có thể mang lại nhiều lợi ích:

- **Giảm độ phức tạp của mô hình**
- **Tăng tốc độ huấn luyện**
- **Giảm nhiễu từ các biến ít quan trọng**, từ đó cải thiện hiệu suất dự đoán tổng thể

Tóm lại

Phân tích độ quan trọng của đặc trưng giúp xác định rõ các biến **V14**, **V10**, **V12**, **V4**, **V17**, và **V3** là những yếu tố cốt lõi trong việc phát hiện gian lận. Việc tập trung vào các biến này sẽ giúp tối ưu hóa quá trình xây dựng mô hình một cách hiệu quả và chính xác hơn.

4.8 Phân Tích Sự Khác Biệt Phân Phối Giữa Giao Dịch Gian Lận và Không Gian Lận



Phân tích biểu đồ **KDE (Kernel Density Estimation)** cho **4 đặc trưng quan trọng nhất** gồm **V14, V10, V12 và V4** cho thấy các biến này có khả năng phân biệt rõ rệt giữa giao dịch gian lận và không gian lận.

4.8.1 Khả năng phân biệt rõ rệt: V14, V10, V12

- **V14, V10 và V12** thể hiện sự tách biệt rõ giữa hai nhóm phân phối:
    - **Giao dịch không gian lận (màu xanh)**: Phân phối tập trung chặt quanh mức **0**, tạo thành một đỉnh nhọn.
    - **Giao dịch gian lận (màu đỏ)**: Phân phối rộng hơn, lệch về phía **giá trị âm** (đối với V14 và V10) hoặc lệch về **cả hai phía** (đối với V12).
- **Ý nghĩa**: Khi giá trị của các biến này càng lệch xa khỏi mức 0, đặc biệt về phía âm, thì khả năng giao dịch đó là gian lận càng cao. Đây là đặc điểm lý tưởng để các thuật toán phân loại học được ngưỡng quyết định hiệu quả.

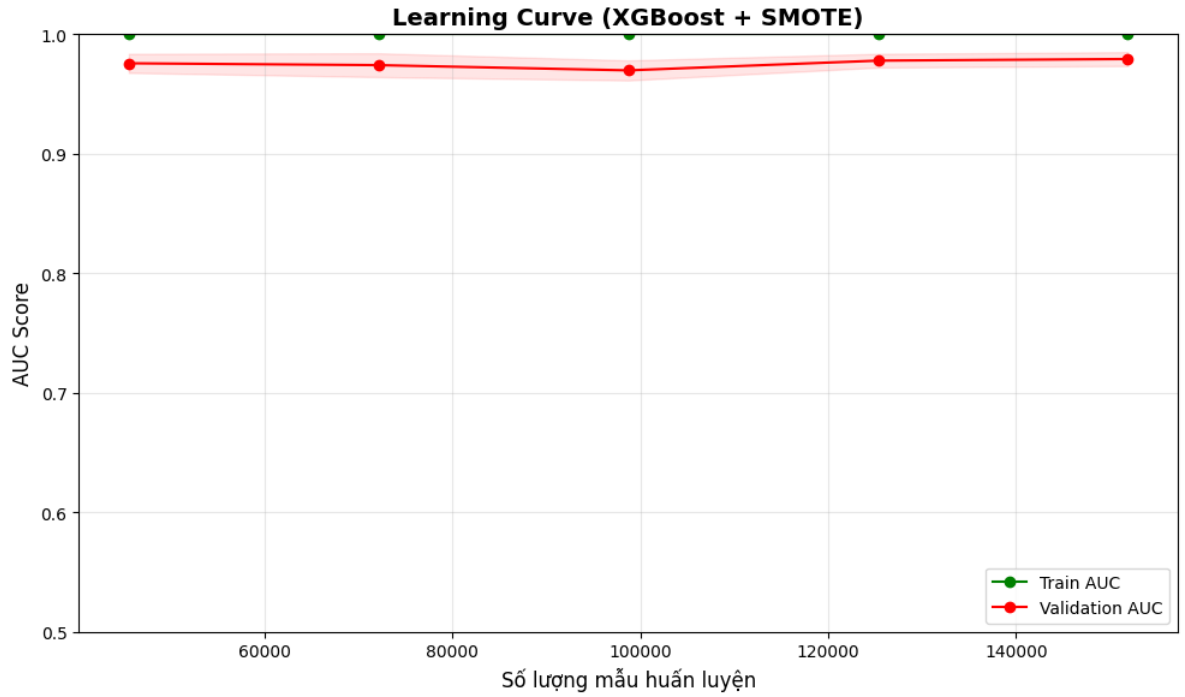
4.8.2 Khả năng phân biệt trung bình: V4

- Biến **V4** cũng cho thấy sự khác biệt giữa hai nhóm, nhưng mức độ tách biệt không rõ ràng bằng các biến trên:
    - Phân phối của nhóm gian lận (đỏ) có sự **chồng lấn** nhiều hơn với nhóm không gian lận (xanh).
    - Tuy nhiên, nhóm gian lận vẫn có **phân phối rộng hơn** và **đỉnh phân phối khác biệt**, cho thấy V4 vẫn là một đặc trưng có giá trị trong việc dự đoán.

Tóm lại

Phân tích này cung cấp bằng chứng trực quan về tầm quan trọng của các biến PCA trong việc phát hiện gian lận. Đặc biệt, các biến **V14, V10 và V12** có khả năng phân biệt rất rõ ràng, giúp mô hình học máy dễ dàng xác định các ngưỡng phân loại. Đây là cơ sở quan trọng để lựa chọn đặc trưng và tối ưu hóa hiệu suất mô hình.

4.9 Phân Tích Đường Cong Học Tập (Learning Curve) của Mô Hình XGBoost



Biểu đồ đường cong học tập thể hiện hiệu suất của mô hình **XGBoost** (đo bằng **AUC Score**) khi kích thước tập huấn luyện tăng dần. Mô hình đã được huấn luyện sau khi áp dụng kỹ thuật **SMOTE** để xử lý mất cân bằng dữ liệu.

4.9.1 Hiệu suất đào tạo và khả năng khái quát hóa

- **Train AUC (đường màu xanh lá)**: Đạt mức gần **1.0** ngay cả khi số lượng mẫu huấn luyện còn nhỏ (khoảng **50.000**) và duy trì ổn định ở mức rất cao khi số lượng mẫu tăng lên.
- **Validation AUC (đường màu đỏ)**: Ổn định trong khoảng **0.96 – 0.97**, và dần hội tụ về gần giá trị của Train AUC khi kích thước tập huấn luyện tăng. Điều này cho thấy mô hình học tốt và không bị quá khớp (*overfitting*).

4.9.2 Đánh giá độ phù hợp và kích thước dữ liệu

- **Sự hội tụ cao**: Cả hai đường cong (Train và Validation) đều đạt AUC rất cao và gần nhau, cho thấy mô hình không bị *underfitting* (không học đủ từ dữ liệu).
- **Khả năng khái quát hóa tốt**: Đường Validation AUC ổn định và cao chứng minh rằng mô hình có thể dự đoán tốt trên dữ liệu mới chưa từng thấy.
- **Không cần thêm dữ liệu huấn luyện**: Khi kích thước tập huấn luyện vượt mốc khoảng **100.000**, Validation AUC không còn cải thiện đáng kể. Điều này cho thấy mô hình đã khai thác tối đa thông tin từ dữ liệu hiện tại.

Tóm lại

Đường cong học tập của mô hình **XGBoost** cho thấy hiệu suất lý tưởng:

- Mô hình học tốt từ dữ liệu,
- Có khả năng khái quát hóa cao,
- Không cần thêm dữ liệu huấn luyện để cải thiện hiệu suất.

Phân tích này củng cố quyết định sử dụng các mô hình **Boosting** như XGBoost trong bài toán phát hiện gian lận.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong bài toán phát hiện gian lận thẻ tín dụng, nhóm đã tiến hành toàn bộ quy trình từ tiền xử lý, trực quan hóa dữ liệu đến huấn luyện, đánh giá và so sánh nhiều mô hình học máy khác nhau.

Kết quả cho thấy các mô hình như XGBoost, CatBoost và LightGBM đạt hiệu suất cao nhất với điểm AUC và F1 vượt trội so với các thuật toán truyền thống như Logistic Regression hay Decision Tree.

Quy trình tiền xử lý đóng vai trò quan trọng, đặc biệt là chuẩn hóa dữ liệu và xử lý mất cân bằng bằng SMOTE, giúp mô hình học tốt hơn trên lớp gian lận hiếm gặp.

Việc tìm ngưỡng phân loại tối ưu cũng cải thiện đáng kể độ cân bằng giữa Precision và Recall, giúp hệ thống giảm thiểu bỏ sót các giao dịch gian lận mà vẫn hạn chế báo động sai.

Tổng thể, mô hình đạt được độ chính xác cao, khả năng tổng quát tốt và chứng minh được tính hiệu quả của việc áp dụng học máy vào lĩnh vực phát hiện gian lận tài chính.

5.2 Hướng phát triển

Mặc dù kết quả đạt được khả quan nhưng bài toán còn nhiều tiềm năng để cải thiện.

Một số hướng phát triển có thể triển khai trong tương lai:

Tối ưu siêu tham số: Sử dụng các phương pháp như Grid Search, Random Search hoặc Bayesian Optimization để tìm bộ tham số tối ưu cho mô hình mạnh nhất (ví dụ: XGBoost, CatBoost).

Kết hợp mô hình (Ensemble / Stacking): Tích hợp nhiều mô hình tốt để khai thác điểm mạnh của từng loại, giúp tăng độ ổn định và giảm sai số tổng thể.

Ứng dụng học sâu (Deep Learning): Thử nghiệm các mô hình mạng nơ-ron như Autoencoder, LSTM hoặc mạng Fully Connected để tự động trích xuất đặc trưng và phát hiện bất thường.

Mở rộng bộ dữ liệu và đặc trưng: Tích hợp thêm các thuộc tính khác như vị trí, thiết bị, lịch sử khách hàng hoặc hành vi giao dịch để mô hình nắm bắt được ngữ cảnh tốt hơn.

VI. Tài liệu tham khảo

1. Kaggle – Credit Card Fraud Detection Dataset. https://www.kaggle.com/mlg-ulb/creditcardfraud
2. Scikit-learn Documentation. https://scikit-learn.org/stable/
3. Imbalanced-learn Documentation. https://imbalanced-learn.org/
4. XGBoost, LightGBM, CatBoost Official Docs.
5. Towards Data Science – Handling Imbalanced Data for Fraud Detection.