

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC UEH - TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH
CHUYÊN NGÀNH KHOA HỌC DỮ LIỆU



Khóa luận tốt nghiệp

PHÁT HIỆN GIAN LẬN TRONG GIAO DỊCH TÀI
CHÍNH BẰNG MÔ HÌNH HỌC MÁY

Họ tên sinh viên:	Đặng Châu Kỳ
Mã sinh viên:	31211027647
Lớp:	DS002
Khóa:	K47
Họ tên giáo viên hướng dẫn:	TS.Bùi Thanh Hiếu

Niên khóa: 2021 - 2024

Tp Hồ Chí Minh, ngày 20 tháng 10 năm 2024.

Lời cảm ơn

Tôi xin chân thành cảm ơn thầy/cô đã dành thời gian quý báu để đọc và đánh giá luận văn tốt nghiệp của tôi về việc phát hiện gian lận trong giao dịch tài chính bằng mô hình học máy. Qua quá trình nghiên cứu và viết luận văn này, tôi đã có cơ hội hiểu sâu hơn về những kỹ thuật tiên tiến trong lĩnh vực khoa học dữ liệu và cách chúng có thể được ứng dụng để giải quyết các vấn đề thực tiễn trong ngành tài chính.

Đặc biệt, tôi muốn gửi lời cảm ơn sâu sắc đến thầy TS.Bùi Thanh Hiếu đã cung cấp những kiến thức nền tảng và hướng dẫn tận tình trong suốt quá trình học tập và thực hiện luận văn. Những kiến thức và kinh nghiệm mà thầy/cô chia sẻ đã giúp tôi có cái nhìn toàn diện và sâu sắc hơn về chủ đề này. Sự định hướng và phản hồi của thầy Bùi Thanh Hiếu đã giúp tôi cải thiện và hoàn thiện luận văn một cách tốt nhất.

Cuối cùng, tôi mong rằng luận văn này sẽ mang lại những thông tin hữu ích và góp phần vào việc nâng cao nhận thức và ứng dụng học máy trong việc phát hiện gian lận tài chính.

Xin chân thành cảm ơn!

Trân trọng,



Đặng Châu Kỳ

Tóm tắt

Gian lận liên quan đến thanh toán là một khía cạnh quan trọng của các cơ quan chống tội phạm mạng và nghiên cứu gần đây đã chỉ ra rằng các kỹ thuật học máy có thể được áp dụng thành công để phát hiện các giao dịch gian lận trong một khối lượng lớn dữ liệu thanh toán. Các kỹ thuật này có khả năng phát hiện các giao dịch gian lận mà các kiểm toán viên con người có thể không phát hiện được, và cũng thực hiện điều này theo thời gian thực.

Trong dự án này, tôi áp dụng nhiều kỹ thuật học máy có giám sát vào vấn đề phát hiện gian lận bằng cách sử dụng dữ liệu giao dịch thanh toán giả lập có sẵn công khai. Tôi nhằm mục đích chứng minh cách các kỹ thuật học máy có giám sát có thể được sử dụng để phân loại dữ liệu với sự măt cân bằng lớp cao với độ chính xác cao.

Tôi chứng minh rằng phân tích khám phá có thể được sử dụng để phân tách các giao dịch gian lận và không gian lận. Tôi cũng chứng minh rằng đối với một tập dữ liệu được phân tách tốt, các thuật toán dựa trên cây như Random Forest hoạt động tốt hơn nhiều so với Logistic Regression. Điều này nhấn mạnh khả năng của các thuật toán dựa trên cây trong việc xử lý dữ liệu phức tạp và măt cân bằng lớp trong bài toán phát hiện gian lận tài chính.

Mục lục

Lời cảm ơn.....	2
Tóm tắt.....	3
Mục lục.....	4
Danh mục hình ảnh.....	8
Danh mục bảng.....	10
Chương 1: Giới thiệu đề tài.....	10
1.1. Lý do chọn đề tài.....	10
1.2. Mục tiêu của đề tài.....	11
1.3. Đối tượng và phạm vi nghiên cứu.....	12
1.4. Phương pháp nghiên cứu.....	13
1.5. Hạn chế của nghiên cứu.....	14
1.6. Bố cục của luận văn.....	15
Chương 2: Cơ sở lý thuyết.....	16
2.1. Định nghĩa về các giao dịch tài chính.....	16
2.1.1. Giao dịch tài chính là gì?.....	16
2.1.2. Các loại giao dịch tài chính.....	16
2.1.3. Vai trò của giao dịch tài chính.....	17
2.2. Gian lận trong giao dịch tài chính.....	18
2.2.1. Khái niệm.....	18
2.2.2. Các hình thức gian lận trong giao dịch tài chính.....	19
2.2.4. Ảnh hưởng của gian lận giao dịch tài chính đến các bên liên quan.....	21
2.2.5. Các phương pháp phát hiện gian lận tại thị trường Việt Nam.....	23
2.3. Phương pháp SMOTE giải quyết sự mất cân bằng dữ liệu.....	24
2.3.1. Giới Thiệu về SMOTE.....	24
2.3.2. Nguyên Lý Hoạt Động của SMOTE.....	24
2.3.3. Lợi ích của SMOTE.....	25
2.3.4. Hạn chế của SMOTE.....	25
2.4. Các mô hình máy học.....	26
2.4.1. Mô hình Logistic Regression.....	26
2.4.1.1. Khái niệm.....	26
2.4.1.2. Hàm Logistic.....	26
2.4.1.4. Tối ưu hoá.....	27
2.4.1.5. Ưu điểm và hạn chế.....	27
2.4.1.6. Ứng dụng.....	28
2.4.2. Mô hình Random Forest.....	29
2.4.2.1. Khái niệm.....	29
2.4.2.2. Cấu trúc của Random Forest.....	29
2.4.2.2.1. Cây Quyết Định (Decision Trees).....	29

2.4.2.2. Random Forest.....	29
2.4.2.3. Thuật toán.....	30
2.4.2.4. Ưu điểm và hạn chế.....	30
2.4.2.5. Ứng dụng.....	31
2.4.4. Mô hình XGBoost.....	32
2.4.4.1. Khái niệm.....	32
2.4.4.2. Cấu Trúc và Nguyên Lý Hoạt Động.....	32
2.4.4.2.1. Gradient Boosting.....	32
2.4.4.2.2. Cải Tiết của XGBoost.....	32
2.4.4.3. Thuật Toán XGBoost.....	33
2.4.4.4. Ưu Điểm và Hạn Chế.....	33
2.4.4.5. Ứng dụng.....	34
2.5. Tổng quan tài liệu.....	34
Chương 3: Phương pháp nghiên cứu.....	36
3.1. Phương pháp.....	36
3.2. Công cụ sử dụng.....	37
3.3. Nguồn dữ liệu.....	37
3.4. Tổng quan dự án.....	38
Chương 4: Xây dựng mô hình học máy.....	41
4.1. Phân tích dữ liệu.....	41
4.2. Phân tích chi tiết.....	42
4.2.1. Làm sạch dữ liệu.....	42
4.2.1.1. Mô tả dữ liệu.....	42
4.2.1.2. Chuyển đổi kiểu dữ liệu.....	43
4.2.1.3. Thông kê mô tả.....	44
4.2.1.4. Kiểm tra giá trị bị thiếu.....	44
4.2.2. Phân tích khám phá.....	45
4.2.2.1. Kiểm tra cột Amount.....	45
4.2.2.2. Sự mất cân bằng của các lớp dữ liệu.....	46
4.2.2.3. Các loại giao dịch.....	47
a. Tần suất các loại giao dịch.....	47
b. Giao dịch gian lận theo loại giao dịch.....	48
c. Phần trăm loại giao dịch là gian lận.....	50
d. Tạo dữ liệu mới.....	51
4.2.2.4. Kiểm tra tính hợp lệ của dữ liệu.....	52
4.2.2.4.1. Số tiền giao dịch âm hoặc bằng không.....	52
4.2.2.4.2. Số dư của người khởi tạo và số dư của người nhận.....	54
4.2.2.4.3. Phân tích giao dịch gian lận.....	55
4.2.3. Mô hình dự đoán để phát hiện gian lận.....	62
4.2.3.1. Tạo tập dữ liệu cho mô hình hóa.....	62

4.2.3.1.1. Tạo các biến giả định.....	62
4.2.3.1.2. Chia dữ liệu.....	63
4.2.3.1.3. Giải quyết sự mất cân bằng dữ liệu.....	64
4.2.3.1.4. Chia dữ liệu trên tập dữ liệu cân bằng.....	67
4.2.3.2. Các mô hình phân loại để phát hiện gian lận.....	67
4.2.3.2.1 Logistic Regression Model.....	67
4.2.3.2.2. Random Forest Model.....	70
4.2.3.2.3. XGBoost Model.....	73
4.2.3.3. Lựa chọn phù hợp.....	76
4.2.3.3.1. Tiêu chí lựa chọn mô hình.....	77
4.2.3.3.2. Đánh giá từng mô hình.....	77
4.2.3.3.3. Lựa chọn mô hình phù hợp nhất.....	79
4.2.4. Tóm tắt Phân tích.....	81
Chương 5: Kết luận.....	83
5.1. Kết luận.....	83
5.2. Đề xuất và hướng phát triển.....	83
TÀI LIỆU THAM KHẢO.....	85

Danh mục hình ảnh

Hình 1: Phương pháp luận của dự án.....	12
Hình 2: Ảnh chụp nhanh của tập dữ liệu thô.....	36
Hình 3: Cấu trúc phân tích.....	39
Hình 4: Kiểu dữ liệu ban đầu của các cột.....	41
Hình 5: [Đoạn mã] Chuyển đổi kiểu dữ liệu.....	42
Hình 6: Tóm tắt thống kê các biến số.....	42
Hình 7: Tóm tắt thống kê của các biến phân loại.....	42
Hình 8: [Đoạn mã] Kiểm tra giá trị bị thiếu.....	43
Hình 9: Biểu đồ boxplot của biến Amount.....	43
Hình 10: [Đoạn mã] Tỷ lệ % giao dịch gian lận.....	44
Hình 11: [Đoạn mã] Tỷ lệ % giao dịch bình thường.....	44
Hình 12: Minh họa sự mất cân bằng dữ liệu.....	45
Hình 13: Số lượng giao dịch theo từng loại.....	46
Hình 14: Minh họa số lượng của từng loại giao dịch.....	46
Hình 15: Minh họa số lượng giao dịch gian lận và không gian lận theo loại giao dịch.....	47
Hình 16: Số lượng giao dịch gian lận theo từng loại giao dịch.....	48
Hình 17: Minh họa số lượng giao dịch gian lận theo từng loại giao dịch.....	48
Hình 18: [Đoạn mã] Chỉ giữ lại các giao dịch CASH-OUT và TRANSFER.....	49
Hình 19: Thông tin của bộ dữ liệu mới vừa được lọc.....	49
Hình 20: Tóm tắt thống kê của các biến phân loại trong bộ dữ liệu mới.....	50
Hình 21: [Đoạn mã] Số tiền giao dịch âm hoặc bằng không.....	50
Hình 22: [Đoạn mã] Xóa các giao dịch có số tiền bằng 0.....	50
Hình 23: Minh họa số tiền giao dịch của giao dịch gian lận và không gian lận....	51
Hình 24: Số tiền giao dịch trung bình và trung vị theo loại.....	52
Hình 25: [Đoạn mã] Kiểm tra số dư bằng không.....	52
Hình 26: [Đoạn mã] Xác định tính năng cân bằng.....	53
Hình 27:[Đoạn mã] Kiểm tra số dư không chính xác.....	53
Hình 28: Giao dịch gian lận và không gian lận được tính theo bước thời gian.....	54
Hình 29: Minh họa bước thời gian theo số giao dịch gian lận và không gian lận..	55
Hình 30: Số tiền giao dịch của các giao dịch gian lận và không gian lận.....	56
Hình 31: [Đoạn mã] So sánh các giao dịch gian lận và không gian lận khi số dư ban đầu của người khởi tạo là 0.....	57
Hình 32: [Đoạn mã] Xác định tính năng cân bằng không chính xác.....	57
Hình 33: Cột origBalance_inacc và destBalance_inacc vừa được khởi tạo.....	57
Hình 34: Minh họa số dư của người khởi tạo không chính xác về các giao dịch gian lận và không gian lận.....	58
Hình 35: Minh họa sự không chính xác trong số dư tài khoản người nhận của các giao dịch gian lận và không gian lận.....	58

Hình 36: Phân biệt giữa giao dịch gian lận và không gian lận.....	60
Hình 37: các cột trong tập dữ liệu.....	60
Hình 38: [Đoạn mã] Loại bỏ những cột không cần thiết.....	60
Hình 39: Đoạn mã] Mã hóa biến phân loại 'type'	61
Hình 40: Danh sách những cột mới.....	61
Hình 41: [Đoạn mã] Tạo tập dữ liệu huấn luyện và thử nghiệm.....	61
Hình 43: [Đoạn mã] Chuẩn hóa dữ liệu.....	62
Hình 43: [Đoạn mã] Cân bằng dữ liệu bằng SMOTE.....	63
Hình 44: Kết quả của bộ dữ liệu trước và sau khi áp dụng SMOTE.....	63
Hình 45: [Đoạn mã] Tạo dataframe mới với bộ dữ liệu cân bằng.....	64
Hình 46: Minh họa các lớp trong bộ dữ liệu sau khi SMOTE.....	64
Hình 47: [Đoạn mã] Tạo tập dữ liệu huấn luyện và thử nghiệm trên bộ dữ liệu cân bằng.....	65
Hình 48: Khởi tạo mô hình Logistic Regression.....	66
Hình 49: Logistic Regression - Classification report.....	66
Hình 50: Logistic Regression - Confusion Matrix.....	67
Hình 51: Khởi tạo mô hình Random Forest.....	68
Hình 52: Random Forest - Classification Report.....	69
Hình 53: Random Forest - Confusion Matrix.....	70
Hình 54: [Đoạn mã] Khởi tạo mô hình XGBoost.....	71
Hình 55: XGBoost Classifier - Classification Report.....	72
Hình 56: XGBoost Classifier - Confusion Matrix.....	73
Hình 57: Minh họa cây quyết định trong mô hình XGBoost.....	74
Hình 58: Chi tiết mô hình Random Forest.....	78
Hình 59: Xếp hạng đặc điểm mô hình Random Forest.....	78
Hình 60: Tầm quan trọng của đặc điểm mô hình Random Forest.....	78
Hình 61: Đường cong ROC của Mô hình Random Forest.....	79

Danh mục bảng

Bảng 1: Tần suất sử dụng các kỹ thuật học máy trong vấn đề phát hiện gian lận..	34
Bảng 2: Mô tả chi tiết dự án.....	35
Bảng 3: Các biến trong tập dữ liệu.....	41
Bảng 4: So sánh kết quả của ba mô hình.....	76

Chương 1: Giới thiệu đề tài

1.1. Lý do chọn đề tài

Gian lận trong các giao dịch tài chính là một vấn đề nghiêm trọng, gây ra thiệt hại lớn cho các tổ chức tài chính và khách hàng. Với sự phát triển nhanh chóng của công nghệ và Internet, các hình thức gian lận ngày càng trở nên tinh vi, phức tạp và khó phát hiện hơn. Các cuộc tấn công này không chỉ gây thiệt hại về tài chính mà còn làm giảm niềm tin của khách hàng vào hệ thống tài chính. Việc phát hiện và ngăn chặn gian lận kịp thời không chỉ bảo vệ tài sản của khách hàng và tổ chức tài chính mà còn đảm bảo sự ổn định của hệ thống tài chính, duy trì niềm tin của công chúng.

Bên cạnh đó, các phương pháp phát hiện gian lận truyền thống thường dựa trên các quy tắc và luật lệ do con người thiết lập. Mặc dù những phương pháp này có thể hiệu quả trong một số trường hợp, chúng thường gặp phải nhiều hạn chế. Các quy tắc cố định khó có thể điều chỉnh và mở rộng để phù hợp với các hình thức gian lận mới, điều này dẫn đến việc hệ thống phát hiện gian lận trở nên kém hiệu quả. Hơn nữa, việc dựa vào kiểm tra thủ công tốn kém thời gian và nguồn lực, dẫn đến khả năng phản ứng chậm trước các mối đe dọa mới. Các phương pháp truyền thống cũng thường gặp phải tỷ lệ báo động giả cao, gây lãng phí tài nguyên và thời gian của các nhân viên kiểm tra.

Việc áp dụng mô hình học máy để phát hiện gian lận trong các giao dịch tài chính không chỉ có ý nghĩa lý thuyết mà còn mang lại những giá trị thực tiễn to lớn. Đầu tiên, việc áp dụng các mô hình này giúp bảo vệ tài sản của khách hàng và tổ chức tài chính, giảm thiểu thiệt hại tài chính và bảo vệ danh tiếng của các tổ chức. Thứ hai, các mô hình học máy nâng cao niềm tin của khách hàng khi họ biết rằng các giao dịch của mình được bảo vệ bởi các hệ thống phát hiện gian lận tiên tiến. Cuối cùng, các mô hình này giúp giảm chi phí cho các hoạt động kiểm tra và xác minh giao dịch, từ đó tăng cường hiệu quả hoạt động của tổ chức tài chính.

Với những lý do trên, đề tài "Phát Hiện Gian Lận Trong Các Giao Dịch Tài Chính Bằng Mô Hình Học Máy" không chỉ mang lại giá trị lý thuyết và thực tiễn cao mà còn là một hướng nghiên cứu đầy tiềm năng, góp phần vào sự phát triển của ngành tài chính và công nghệ thông tin. Việc áp dụng học máy vào phát hiện gian lận không

chỉ giúp ngăn chặn các hành vi gian lận hiệu quả mà còn mở ra những hướng đi mới trong việc bảo vệ tài chính và dữ liệu của khách hàng, từ đó xây dựng một hệ thống tài chính an toàn và bền vững hơn.

1.2. Mục tiêu của đề tài

- Mục tiêu chung:
 - Xây dựng một mô hình phát hiện gian lận trong giao dịch tài chính nhằm áp dụng các kiến thức học máy vào việc giải quyết vấn đề thực tế, đồng thời nâng cao hiểu biết và kỹ năng trong lĩnh vực học máy và phân tích dữ liệu.
 - Chứng minh tiềm năng của việc khai thác dữ liệu trong ngành tài chính, tìm ra các giá trị tiềm ẩn trong dữ liệu có thể giúp cải thiện hiệu suất và độ chính xác của các hệ thống phát hiện gian lận hiện có.
- Mục tiêu cụ thể:
 - Nghiên cứu tài liệu về phát hiện gian lận tài chính và hiểu các khía cạnh khác nhau của vấn đề.
 - Sử dụng các kỹ năng về xử lý và khai thác dữ liệu, tìm ra các insight đặc trưng trong các giao dịch tài chính.
 - Giải quyết vấn đề phát hiện gian lận tài chính trên một tập dữ liệu mẫu có sẵn công khai bằng cách sử dụng các kỹ thuật học máy có giám sát.
 - So sánh các kỹ thuật phân loại khác nhau để hiểu kỹ thuật nào phù hợp nhất cho ứng dụng này.

1.3. Đối tượng và phạm vi nghiên cứu

- Đối Tượng Nghiên Cứu:
 - Các Giao Dịch Tài Chính: Bao gồm các giao dịch trực tuyến, giao dịch thẻ tín dụng, giao dịch ngân hàng điện tử, và các hình thức thanh toán trực tuyến khác.
 - Dữ Liệu Giao Dịch: Tập hợp các dữ liệu liên quan đến giao dịch tài chính, bao gồm thông tin về tài khoản, số tiền giao dịch, thời gian, địa điểm, và các thông tin liên quan khác.
 - Các Thuật Toán Học Máy: Các thuật toán học máy có giám sát phổ biến như Random Forest, Logistic Regression và XGBoost.

- Phạm vi nghiên cứu

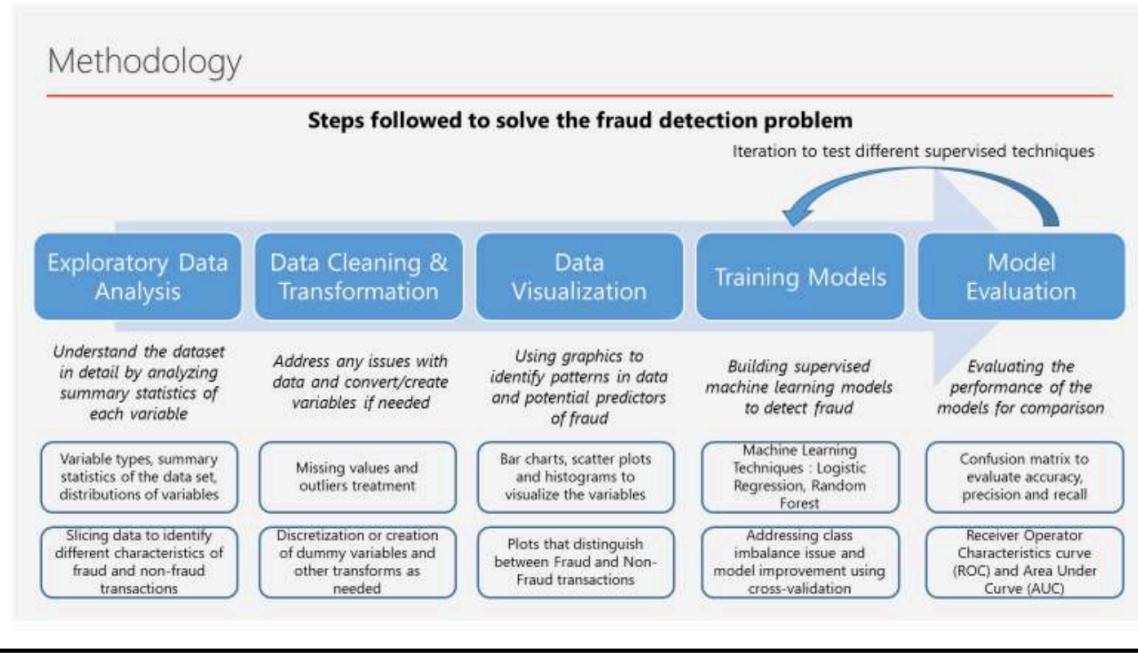
- Nghiên cứu lý thuyết: Về các giao dịch tài chính, gian lận trong tài chính và các thuật toán học máy có thể áp dụng để phát hiện gian lận.
- Nghiên cứu thực tiễn: Về việc áp dụng các thuật toán học máy vào việc phát hiện gian lận trong dữ liệu giao dịch tài chính.
- Nghiên cứu giới hạn trên tập dữ liệu thực tế: Tập dữ liệu về các giao dịch tài chính được công khai trên các nền tảng như Kaggle hoặc các nguồn dữ liệu mở khác. Tập dữ liệu này được chọn vì độ tin cậy cao và đầy đủ các thông tin về các yếu tố liên quan đến giao dịch gian lận và không gian lận. Lý do sử dụng một tập dữ liệu có sẵn là để tránh các vấn đề về quyền riêng tư và bảo mật thông tin cá nhân trong các dữ liệu nội bộ của doanh nghiệp tài chính.

1.4. Phương pháp nghiên cứu

Phương pháp học máy điển hình đã được áp dụng trong dự án này. Tập dữ liệu được xác định có biến lớp được gán nhãn, được sử dụng làm biến dự đoán trong các mô hình học máy.

- Thông qua phân tích khám phá, tôi đã phân tích chi tiết tập dữ liệu và xác định các yếu tố dự đoán có thể của gian lận.
- Thông qua các kỹ thuật trực quan hóa khác nhau, tôi đã quan sát sự phân tách giữa các giao dịch gian lận và không gian lận.
- Để giải quyết vấn đề phát hiện gian lận, tôi đã thử nghiệm với hai kỹ thuật học máy có giám sát – Logistic Regression, Random Forest, XGBoost Classification.
- Ngoài ra, tôi cũng đã thử tăng mẫu để giải quyết sự mất cân bằng lớp trong tập dữ liệu.
- Các mô hình đã được phát triển với kiểm tra chéo để tránh hiện tượng quá khớp và đạt được tính đồng nhất về hiệu suất.
- Các chỉ số hiệu suất, như Ma trận nhầm lẫn (Confusion Matrix) và Diện tích dưới đường cong (AUC), đã được sử dụng để so sánh hiệu suất của các mô hình.

Phân tích này được thực hiện bằng Python thông qua Jupyter notebook. Các thư viện và phương pháp tích hợp sẵn đã được sử dụng để chạy các mô hình học máy. Khi cần thiết, các hàm đã được định nghĩa để đơn giản hóa các phân tích hoặc trực quan hóa cụ thể.



Hình 1: Phương pháp luận của dự án. [1]

1.5. Hạn chế của nghiên cứu

Trong nghiên cứu này, tôi đã đánh giá hiệu quả của việc sử dụng các kỹ thuật học máy có giám sát cụ thể để giải quyết vấn đề phát hiện gian lận trong các giao dịch tài chính. Những hạn chế của các phương pháp áp dụng trong nghiên cứu này như sau:

- Tôi đã sử dụng một tập dữ liệu đã được gán nhãn trước để huấn luyện các thuật toán. Tuy nhiên, thường thì rất khó để tìm dữ liệu đã được gán nhãn, vì vậy việc áp dụng các kỹ thuật học máy có giám sát có thể không khả thi. Trong những trường hợp như vậy, chúng ta nên đánh giá các kỹ thuật không giám sát, điều này nằm ngoài phạm vi của nghiên cứu này.
- Nghiên cứu này xem xét dữ liệu giao dịch kỹ thuật số bao gồm số tiền giao dịch, số dư của người nhận và người gửi, cũng như thời gian giao dịch. Những biến này giúp phát hiện gian lận có thể không áp dụng cho các loại giao dịch tài chính khác, chẳng hạn như gian lận thẻ tín dụng.

- Tôi đã đánh giá hai thuật toán học máy – Logistic Regression, Random Forest và XGBoost. Mặc dù kết quả của nghiên cứu sử dụng các thuật toán này là tốt, nhưng cần thiết phải đánh giá các kỹ thuật khác để xác định thuật toán nào hoạt động tốt nhất cho ứng dụng này.
- Do kích thước dữ liệu lớn, tôi bị hạn chế bởi khả năng tính toán để khám phá các kỹ thuật khác nhau như tìm kiếm lưới (grid search) để tinh chỉnh tham số. Những kỹ thuật này có thể giúp cải thiện kết quả của nghiên cứu này hơn nữa.

1.6. Bố cục của luận văn

Luận văn được tổ chức thành 5 chương, trong đó:

- Chương 1: Trình bày về lý do chọn đề tài, mục tiêu, các đối tượng và phạm vi nghiên cứu của luận văn, nêu rõ phương pháp nghiên cứu, đóng góp và bối cảnh của luận văn.
- Chương 2: Tổng quan về tình hình thị trường tài chính, nghiên cứu liên quan và các phương pháp giải quyết gian lận truyền thông tại Việt Nam, cùng cơ sở lý thuyết về các phương pháp xử lý dữ liệu và thuật toán học máy.
- Chương 3: Trình bày các phương pháp, công cụ và bộ dữ liệu được sử dụng trong bài luận. Từ đó, mô tả quy trình tổng quan của dự án.
- Chương 4: Xây dựng mô hình phát hiện gian lận thông qua việc thu thập, tiền xử lý dữ liệu và áp dụng các thuật toán học máy. Sau đó, đánh giá hiệu quả của các mô hình học máy, phân tích so sánh và chọn ra mô hình phù hợp nhất.
- Chương 5: Kết luận và hướng phát triển, đánh giá kết quả nghiên cứu so với mục tiêu và đề xuất kiến nghị cho doanh nghiệp.

Chương 2: Cơ sở lý thuyết

2.1. Định nghĩa về các giao dịch tài chính

2.1.1. Giao dịch tài chính là gì?

Giao dịch tài chính là hoạt động trao đổi giá trị không thể thiếu trong đời sống kinh tế. Đó là sự đồng thuận giữa các bên nhằm trao đổi hàng hóa, dịch vụ hoặc tài sản, từ tiền mặt, vàng, bạc đến các loại chứng khoán, bất động sản. Mỗi giao dịch đều tạo ra sự luân chuyển dòng tiền, thúc đẩy sản xuất và tiêu dùng. Giao dịch mua bán là hình thức phổ biến nhất, diễn ra hàng ngày trong cuộc sống. Bên cạnh đó, còn có nhiều loại giao dịch khác như giao dịch ngoại hối, giao dịch chứng khoán, góp phần tạo nên sự đa dạng và phức tạp của thị trường tài chính. Nhìn chung, giao dịch tài chính đóng vai trò quan trọng trong việc phân bổ nguồn lực, thúc đẩy tăng trưởng kinh tế và nâng cao chất lượng cuộc sống. [2]

2.1.2. Các loại giao dịch tài chính

- **Giao dịch tiền mặt**

Giao dịch tiền mặt là những giao dịch được thực hiện bằng tiền tệ vật chất như tiền xu, tiền giấy hoặc thẻ ghi nợ. Trong loại giao dịch này, tiền được chuyển trực tiếp từ người mua sang người bán để đổi lấy hàng hóa hoặc dịch vụ. Đây là hình thức giao dịch truyền thống và phổ biến nhất, đặc biệt trong các giao dịch nhỏ lẻ hàng ngày. Khác với giao dịch tín dụng, giao dịch tiền mặt không có sự trì hoãn trong việc thanh toán. Điều này có nghĩa là người mua phải có sẵn tiền để thực hiện giao dịch, và người bán nhận được tiền ngay lập tức sau khi giao dịch hoàn tất. [3]

- **Giao dịch tín dụng**

Giao dịch tín dụng cho phép người mua thanh toán vào một thời điểm sau khi hàng hóa hoặc dịch vụ đã được cung cấp. Trong giao dịch này, người bán nhận được một tài sản (khoản tiền thanh toán trong tương lai), trong khi người mua có một khoản nợ phải trả. Thẻ tín dụng là một ví dụ điển hình của giao dịch tín dụng. Ngân hàng cấp cho khách hàng một hạn mức tín dụng để chi tiêu, và khoản tiền này phải được trả lại vào một ngày cố định hàng tháng. Các khoản vay và vay thế chấp cũng là hình thức của giao dịch tín dụng. Trong các khoản vay thông thường, người vay nhận được một

khoản tiền và phải trả lại cùng với lãi suất trong một thời gian nhất định. Vay thế chấp thường liên quan đến các khoản tiền lớn và có tài sản đảm bảo, như bất động sản. Nếu người vay không thể trả nợ đúng hạn, tổ chức cho vay có quyền thu giữ và thanh lý tài sản thế chấp. [4]

● Giao dịch nội bộ và giao dịch bên ngoài

Giao dịch bên ngoài là những giao dịch liên quan đến nhiều hơn một bên. Ví dụ, một công ty mua hàng từ một nhà cung ứng khác là một giao dịch bên ngoài. Tất cả các giao dịch tiền mặt và tín dụng đều là giao dịch bên ngoài vì chúng ảnh hưởng đến tình hình tài chính của nhiều cá nhân hoặc tổ chức. Trái ngược với giao dịch bên ngoài, giao dịch nội bộ chỉ ảnh hưởng đến một doanh nghiệp duy nhất. Chẳng hạn, việc trao đổi hàng hóa giữa các đơn vị trong cùng một công ty là giao dịch nội bộ, vì nó không làm thay đổi tình hình tài chính chung của toàn công ty. [5]

2.1.3. Vai trò của giao dịch tài chính

● Phân bổ vốn hiệu quả

Một trong những vai trò quan trọng nhất của giao dịch tài chính là phân bổ vốn hiệu quả trong nền kinh tế. Thông qua các thị trường tài chính, vốn từ những người tiết kiệm (nhà đầu tư) được chuyển đến những người cần vốn (doanh nghiệp). Nhờ đó, nguồn lực xã hội được sử dụng một cách tối ưu, thúc đẩy sản xuất kinh doanh và tạo ra nhiều sản phẩm, dịch vụ mới. Giao dịch tài chính đảm bảo rằng vốn được phân bổ đến những dự án có khả năng sinh lời cao nhất, góp phần tăng trưởng kinh tế. [6]

● Hình thành và điều chỉnh giá cả

Giá cả của các tài sản, hàng hóa, dịch vụ được hình thành và điều chỉnh liên tục trên thị trường thông qua giao dịch. Cơ chế cung và cầu là yếu tố quyết định giá cả. Khi nhu cầu về một sản phẩm tăng lên, giá cả có xu hướng tăng và ngược lại. Sự biến động của giá cả phản ánh thông tin về tình hình cung cầu, lạm phát, lãi suất và các yếu tố kinh tế khác, giúp các nhà đầu tư đưa ra quyết định đầu tư hợp lý. [7]

● Quản lý rủi ro

Giao dịch tài chính cung cấp các công cụ hữu hiệu để quản lý rủi ro. Các sản phẩm phái sinh như hợp đồng tương lai, quyền chọn cho phép các nhà đầu tư bảo vệ giá trị tài sản của mình trước những biến động bất ngờ của thị trường. Bằng cách đa

dạng hóa danh mục đầu tư, nhà đầu tư có thể phân tán rủi ro và giảm thiểu thiệt hại.

[8]

- **Thúc đẩy tăng trưởng kinh tế**

Giao dịch tài chính là động lực quan trọng thúc đẩy tăng trưởng kinh tế. Bằng cách huy động vốn cho các doanh nghiệp, đặc biệt là các doanh nghiệp nhỏ và vừa, giao dịch tài chính giúp tạo ra việc làm, tăng thu nhập và nâng cao chất lượng cuộc sống. Ngoài ra, giao dịch tài chính còn khuyến khích đổi mới sáng tạo, thúc đẩy sự phát triển của các ngành công nghiệp mới. [9]

- **Cải thiện hiệu quả hoạt động của nền kinh tế**

Giao dịch tài chính góp phần cải thiện hiệu quả hoạt động của nền kinh tế bằng cách tăng tính thanh khoản của các tài sản. Khi các tài sản dễ dàng mua bán, nó sẽ tạo điều kiện cho các doanh nghiệp huy động vốn nhanh chóng, linh hoạt và các nhà đầu tư có thể dễ dàng chuyển đổi danh mục đầu tư. [10]

2.2. Gian lận trong giao dịch tài chính

2.2.1. Khái niệm

Gian lận trong giao dịch tài chính là mọi hành vi cố ý sử dụng thủ đoạn, phương thức trái pháp luật để trực lợi cá nhân hoặc tổ chức trong quá trình thực hiện các giao dịch tài chính. Những hành vi này thường nhằm mục đích chiếm đoạt tài sản của người khác, gây thiệt hại cho các bên tham gia giao dịch và làm mất niềm tin vào thị trường. [11]

Gian lận trong giao dịch tài chính không chỉ đơn thuần là một hành vi vi phạm pháp luật mà còn là một vấn nạn xã hội, gây ra những hậu quả nghiêm trọng. Khi thông tin về một vụ gian lận lớn bị phanh phui, niềm tin của nhà đầu tư vào thị trường sẽ bị lung lay nghiêm trọng. Điều này dẫn đến việc rút vốn hàng loạt, gây ra sự sụt giảm mạnh của thị trường chứng khoán và làm ảnh hưởng đến hoạt động sản xuất kinh doanh của các doanh nghiệp. Hơn nữa, gian lận tài chính còn tạo ra một môi trường cạnh tranh không lành mạnh, trong đó các doanh nghiệp chân chính bị thiệt thòi, làm giảm hiệu quả của nền kinh tế. Để hạn chế tình trạng này, cần có sự phối hợp chặt chẽ giữa các cơ quan quản lý, các tổ chức tài chính và cộng đồng xã hội trong

việc xây dựng và hoàn thiện khung pháp lý, tăng cường giám sát, nâng cao nhận thức và ứng dụng công nghệ để phát hiện và xử lý các hành vi gian lận.

2.2.2. Các hình thức gian lận trong giao dịch tài chính

Gian lận trong giao dịch tài chính xuất hiện dưới nhiều hình thức, từ gian lận chuyển khoản, thanh toán qua thẻ tín dụng, cho đến các giao dịch quốc tế và thanh toán điện tử. Những hành vi này không chỉ ảnh hưởng đến cá nhân, tổ chức mà còn gây ra những thiệt hại nghiêm trọng cho toàn bộ hệ thống tài chính.

- Gian lận chuyển khoản (Wire Transfer Fraud)

- Gian lận chuyển khoản là một trong những hình thức phổ biến nhất trong giao dịch tài chính. Một phương thức điển hình là chuyển khoản giả mạo, khi tội phạm xâm nhập vào tài khoản ngân hàng của nạn nhân để chuyển tiền trái phép [12]. Hình thức này thường xảy ra khi tội phạm lợi dụng các lỗ hổng bảo mật hoặc đánh cắp thông tin đăng nhập của người dùng. Tấn công bằng cách giả mạo email (email spoofing) cũng là một hình thức phổ biến. Kẻ gian giả mạo các email từ lãnh đạo công ty hoặc đối tác kinh doanh yêu cầu chuyển khoản một khoản tiền lớn đến tài khoản lừa đảo. Phương thức này, còn được gọi là CEO fraud, đã gây ra tổn thất hàng triệu USD cho các doanh nghiệp mỗi năm.
- Gian lận chuyển khoản cũng bao gồm việc lừa đảo qua hệ thống chuyển tiền quốc tế, đặc biệt là qua các hệ thống như SWIFT. Một ví dụ nổi bật là vụ tấn công vào ngân hàng trung ương Bangladesh, khi các hacker đã đánh cắp hơn 80 triệu USD bằng cách lợi dụng hệ thống SWIFT [13]. Điều này cho thấy ngay cả các hệ thống chuyển tiền quốc tế an toàn nhất cũng có thể bị khai thác nếu không có các biện pháp bảo mật chặt chẽ.

- Gian lận thanh toán qua thẻ tín dụng và thẻ ghi nợ (Card Fraud)

- Gian lận thẻ tín dụng là một trong những hình thức phổ biến nhất và ảnh hưởng trực tiếp đến hàng triệu người tiêu dùng. Kẻ gian có thể đánh cắp thông tin thẻ của người khác để thực hiện các giao dịch bất hợp pháp, đặc biệt là qua các giao dịch trực tuyến.

Phương thức làm giả thẻ cũng phổ biến, khi tội phạm sử dụng thiết bị skimming để sao chép thông tin thẻ từ máy ATM hoặc các điểm thanh toán POS [14].

- Bên cạnh đó, tội phạm cũng sử dụng phương thức gian lận thanh toán trực tuyến, trong đó kẻ gian sử dụng các thông tin thẻ bị đánh cắp để thực hiện mua sắm trên các nền tảng thương mại điện tử. Khi không có sự xác thực mạnh mẽ, giao dịch này dễ dàng được thực hiện mà không cần đến sự có mặt của chủ thẻ thực sự.

- Gian lận thanh toán quốc tế (International Payment Fraud):

- Giao dịch tài chính quốc tế mở ra nhiều cơ hội cho các doanh nghiệp, nhưng đồng thời cũng làm tăng rủi ro về gian lận. Gian lận trong chuyển tiền quốc tế thường bao gồm việc sử dụng các tài khoản ngân hàng không chính thức hoặc hệ thống chuyển tiền phi pháp để rửa tiền. Một số hệ thống chuyển tiền truyền thống như hawala có thể bị lợi dụng để chuyển tiền qua biên giới mà không qua hệ thống ngân hàng chính thức, giúp che giấu nguồn gốc tiền bất hợp pháp [15].
- Các tội phạm cũng có thể lợi dụng giao dịch ngoại hối (Forex fraud) để thao túng tỷ giá và lừa đảo các nhà đầu tư nhỏ lẻ. Một số sàn giao dịch ngoại hối giả mạo tạo ra các giao dịch giả để chiếm đoạt tiền của nhà đầu tư, hoặc cung cấp thông tin sai lệch để thao túng thị trường [16].

- Gian lận thông qua nền tảng thanh toán điện tử (Online Payment Fraud):

- Sự phát triển của các nền tảng thanh toán điện tử và ví điện tử như PayPal, Venmo, hoặc Momo đã mang lại nhiều tiện ích cho người dùng, nhưng cũng đi kèm với nhiều rủi ro gian lận. Một trong những hình thức phổ biến là **chiếm đoạt tài khoản** thông qua việc đánh cắp thông tin đăng nhập của người dùng. Khi kiểm soát được tài khoản, tội phạm có thể thực hiện các giao dịch trái phép, chuyển tiền sang các tài khoản khác hoặc mua sắm trực tuyến [17].

- Một phương thức khác là gian lận qua SIM Swap [18], khi kẻ gian lừa nhà cung cấp dịch vụ viễn thông để chiếm quyền kiểm soát số điện thoại của nạn nhân. Sau đó, chúng sử dụng số điện thoại này để vượt qua bước xác thực hai yếu tố (2FA) và truy cập vào các tài khoản tài chính của nạn nhân. Đây là một hình thức ngày càng tinh vi, gây thiệt hại lớn cho người dùng không am hiểu về bảo mật.

- Gian lận liên quan đến chuyển tiền mặt (Money Mule Fraud):

- Một trong những phương thức tội phạm tài chính phức tạp là việc sử dụng người chuyển tiền (money mule). Đây là các cá nhân hoặc tổ chức nhận tiền từ tài khoản bị đánh cắp và sau đó chuyển tiền này cho kẻ gian, thường qua các kênh khó theo dõi như chuyển tiền quốc tế hoặc tiền mặt. Trong một số trường hợp, người chuyển tiền có thể không ý thức được rằng mình đang tham gia vào một vụ gian lận, nhưng họ vẫn có thể bị truy tố vì đã tiếp tay cho hành vi rửa tiền.

2.2.4. Ảnh hưởng của gian lận trong giao dịch tài chính đến các bên liên quan

Gian lận giao dịch tài chính không chỉ là hành vi vi phạm pháp luật, mà còn để lại những hậu quả nghiêm trọng đối với nhiều bên liên quan (stakeholders) trong một hệ sinh thái kinh tế. Các bên liên quan này bao gồm doanh nghiệp, nhà đầu tư, khách hàng, nhân viên, cơ quan quản lý và thậm chí toàn bộ xã hội. Gian lận tài chính không chỉ phá hoại uy tín của một doanh nghiệp cụ thể mà còn gây bất ổn đối với niềm tin vào thị trường tài chính, làm lung lay cơ sở của nền kinh tế.

- Đối với doanh nghiệp:

- Khi một doanh nghiệp bị phát hiện dính líu đến gian lận tài chính, hậu quả đầu tiên và dễ thấy nhất là sự tổn thất về tài chính. Các doanh nghiệp thường mất đi một phần lớn tài sản hoặc chịu chi phí khắc phục không lồ để bù đắp thiệt hại. Hơn nữa, gian lận có thể đẩy công ty vào tình trạng phá sản, đặc biệt nếu không có đủ nguồn lực hoặc kế hoạch dự phòng để đối phó. Chẳng hạn, vụ phá sản của Enron vào năm 2001 là minh chứng điển hình cho

việc gian lận tài chính có thể làm sụp đổ cả một tập đoàn khổng lồ.

- Không chỉ dừng lại ở khía cạnh tài chính, uy tín của doanh nghiệp bị ảnh hưởng nghiêm trọng sau mỗi vụ gian lận. Uy tín là tài sản vô hình nhưng có giá trị cao, khi bị suy giảm, nó không chỉ làm mất lòng tin của khách hàng mà còn ảnh hưởng đến mối quan hệ với các đối tác kinh doanh và nhà đầu tư.

- Đối với các nhà đầu tư và cổ đông:

- Một trong những nhóm chịu ảnh hưởng nặng nề nhất của gian lận tài chính là các nhà đầu tư và cổ đông. Khi xảy ra gian lận, giá trị cổ phiếu của doanh nghiệp bị sụt giảm nghiêm trọng, đôi khi dẫn đến việc mất trắng đối với các nhà đầu tư nhỏ lẻ. Các vụ gian lận như của công ty Wirecard vào năm 2020 đã gây thiệt hại hàng tỷ USD cho cổ đông, khi toàn bộ giá trị của công ty gần như bốc hơi sau khi vụ việc bị phanh phui [19].
- Hơn nữa, khi một công ty có hành vi gian lận, niềm tin của các nhà đầu tư vào công ty cũng bị suy giảm. Họ trở nên do dự trong việc đầu tư vào các công ty tương tự, điều này có thể làm giảm mức độ đầu tư vào thị trường chung, gây ra những hệ lụy lớn hơn cho nền kinh tế. Việc mất niềm tin vào sự minh bạch và quản trị doanh nghiệp có thể khiến nhà đầu tư rút vốn và đổ dồn vào những tài sản ít rủi ro hơn, gây ra sự biến động thị trường.

- Đối với khách hàng:

- Khách hàng cũng không nằm ngoài tầm ảnh hưởng của các vụ gian lận tài chính. Trong một số trường hợp, khách hàng có thể mất đi tài sản hoặc dịch vụ mà họ đã trả tiền, đặc biệt trong các vụ gian lận liên quan đến ngân hàng hoặc các công ty tài chính. Ví dụ, vụ lừa đảo Madoff đã gây ra thiệt hại trực tiếp cho hàng ngàn nhà đầu tư cá nhân, trong đó có cả các khách hàng nhỏ lẻ đã đặt niềm tin vào hệ thống tài chính [20].
- Ngay cả khi khách hàng không bị thiệt hại trực tiếp về tài chính, niềm tin của họ vào công ty cung cấp dịch vụ cũng bị ảnh hưởng.

Một khi niềm tin đó mất đi, khách hàng có thể chọn cách rời bỏ công ty để tìm đến các đối thủ cạnh tranh, điều này làm cho công ty gian lận mất thị phần và gặp khó khăn trong việc phục hồi.

- Đối với người lao động:

- Những người lao động trong doanh nghiệp cũng phải gánh chịu hậu quả từ các vụ gian lận tài chính. Khi công ty mất mát tài chính lớn, các biện pháp cắt giảm chi phí thường được đưa ra, bao gồm việc sa thải nhân viên hoặc đóng cửa các bộ phận hoạt động không hiệu quả. Điều này làm cho hàng ngàn người lao động mất việc làm, gây ra ảnh hưởng sâu rộng đến đời sống và gia đình họ.
- Bên cạnh đó, tinh thần làm việc của nhân viên trong công ty cũng bị suy giảm khi họ phải làm việc trong một môi trường bất ổn và không chắc chắn. Sự bất an và thiếu tin tưởng vào ban lãnh đạo có thể khiến năng suất lao động giảm sút, góp phần vào sự suy thoái tổng thể của công ty.

2.2.5. Các phương pháp phát hiện gian lận tại thị trường Việt Nam

Gian lận trong giao dịch tài chính là một vấn đề nghiêm trọng, gây ra tổn thất lớn cho cả các tổ chức tài chính và khách hàng. Tại Việt Nam, với sự phát triển nhanh chóng của thị trường tài chính, việc phát hiện và ngăn chặn gian lận càng trở nên cấp thiết. Các phương pháp phát hiện gian lận trong giao dịch tài chính tại thị trường Việt Nam có thể kể đến như sau:

- Triển khai hệ thống phát hiện gian lận thời gian thực là một trong những cách hiệu quả nhất để ngăn chặn gian lận. Hệ thống này giám sát các giao dịch khi chúng diễn ra, sử dụng các thuật toán phân tích dữ liệu và học máy để phát hiện các mẫu giao dịch bất thường. Khi phát hiện gian lận, hệ thống sẽ cảnh báo kịp thời để các biện pháp ngăn chặn được thực hiện ngay lập tức.
- Học máy và trí tuệ nhân tạo đang ngày càng được áp dụng rộng rãi trong việc phát hiện gian lận tài chính. Các mô hình học máy có thể được huấn luyện trên các bộ dữ liệu lớn để nhận diện các mẫu giao dịch gian lận. Các thuật toán như

mạng nơ-ron sâu, cây quyết định, và hồi quy logistic đều có thể được sử dụng để phát hiện gian lận với độ chính xác cao.

- Triển khai phương thức bảo mật xác thực hai yếu tố cho các giao dịch trực tuyến và truy cập tài khoản là một biện pháp hiệu quả để ngăn chặn gian lận. Xác thực hai yếu tố yêu cầu người dùng cung cấp một yếu tố xác minh bổ sung, như mã xác minh được gửi tới thiết bị di động, ngoài tên đăng nhập và mật khẩu. Điều này giúp bảo vệ tài khoản người dùng khỏi các cuộc tấn công từ kẻ gian.

2.3. Phương pháp SMOTE giải quyết sự mất cân bằng dữ liệu

Sự mất cân bằng dữ liệu là một vấn đề phổ biến trong lĩnh vực học máy và khai thác dữ liệu, xảy ra khi một hoặc nhiều lớp trong tập dữ liệu có số lượng mẫu nhỏ hơn nhiều so với các lớp khác. Điều này dẫn đến việc các mô hình học máy có xu hướng bị thiên vị và kém hiệu quả khi dự đoán các lớp thiểu số. Để giải quyết vấn đề này, nhiều kỹ thuật đã được phát triển, trong đó nổi bật nhất là SMOTE (Synthetic Minority Over-sampling Technique). SMOTE là một phương pháp oversampling, giúp tạo ra các mẫu tổng hợp từ lớp thiểu số để cân bằng tỷ lệ giữa các lớp.

2.3.1. Giới Thiệu về SMOTE

SMOTE là một kỹ thuật oversampling tiên tiến được giới thiệu bởi Chawla và cộng sự vào năm 2002 [21]. Khác với các phương pháp oversampling truyền thống, chỉ đơn giản là sao chép lại các mẫu từ lớp thiểu số, SMOTE tạo ra các mẫu tổng hợp mới bằng cách nội suy giữa các mẫu hiện có. Điều này giúp giảm nguy cơ overfitting và cải thiện khả năng tổng quát hóa của mô hình.

2.3.2. Nguyên Lý Hoạt Động của SMOTE

SMOTE hoạt động dựa trên nguyên lý nội suy giữa các mẫu thiểu số hiện có để tạo ra các mẫu mới. Các bước chính của SMOTE bao gồm:

1. Chọn mẫu thiểu số ngẫu nhiên: Từ lớp thiểu số, chọn ngẫu nhiên một mẫu dữ liệu.
2. Tìm k hàng xóm gần nhất (k-nearest neighbors): Sử dụng khoảng cách Euclidean để tìm k hàng xóm gần nhất của mẫu đã chọn từ lớp thiểu số.

3. Tạo mẫu tổng hợp: Chọn ngẫu nhiên một trong k hàng xóm gần nhất. Tạo một mẫu tổng hợp mới bằng cách nội suy giữa mẫu đã chọn và hàng xóm này. Quá trình nội suy này được thực hiện theo công thức:

$$x_{\text{new}} = x_i + \lambda \cdot (x_{\text{neighbor}} - x_i)$$

- Trong đó:
 - x_{new} là mẫu tổng hợp mới.
 - x_i là mẫu thiểu số đã chọn ban đầu.
 - x_{neighbor} là một trong k hàng xóm gần nhất của x_i .
 - λ là một giá trị ngẫu nhiên trong khoảng [0,1].

Quá trình này được lặp lại cho đến khi số lượng mẫu tổng hợp mới đạt đến mức mong muốn để cân bằng dữ liệu.

2.3.3. Lợi ích của SMOTE

SMOTE mang lại nhiều lợi ích quan trọng trong việc xử lý dữ liệu mất cân bằng:

- Giảm thiên vị: Bằng cách tạo ra các mẫu tổng hợp từ lớp thiểu số, SMOTE giúp mô hình học máy giảm thiểu sự thiên vị đối với các lớp đa số.
- Cải thiện khả năng tổng quát hóa: Khác với phương pháp oversampling truyền thống, SMOTE tạo ra các mẫu mới từ dữ liệu hiện có, giúp cải thiện khả năng tổng quát hóa của mô hình.
- Giảm nguy cơ overfitting: Do các mẫu tổng hợp không phải là bản sao trực tiếp của các mẫu thiểu số hiện có, nguy cơ overfitting giảm đáng kể.

2.3.4. Hạn chế của SMOTE

Mặc dù SMOTE có nhiều lợi ích, nhưng cũng có một số hạn chế:

- Giới hạn trong không gian đặc trưng: SMOTE tạo ra các mẫu tổng hợp dựa trên không gian đặc trưng hiện có. Nếu không gian đặc trưng không

đủ đại diện, các mẫu tổng hợp có thể không phản ánh chính xác phân phối thực của dữ liệu.

- Không xử lý được dữ liệu ngoại lai: SMOTE có thể tạo ra các mẫu tổng hợp gần các dữ liệu ngoại lai, làm giảm hiệu quả của mô hình.
- Yêu cầu tính toán cao: Với các tập dữ liệu lớn và phức tạp, quá trình tính toán khoảng cách và tạo mẫu tổng hợp có thể tốn nhiều thời gian và tài nguyên.

2.4. Các mô hình máy học

2.4.1. Mô hình Logistic Regression

2.4.1.1. Khái niệm

Hồi quy logistic là một dạng hồi quy được thiết kế để dự đoán một biến phụ thuộc nhị phân, tức là biến có hai giá trị (thường là 0 và 1, hoặc "có" và "không"). Mục tiêu của mô hình là xác định xác suất một sự kiện xảy ra dựa trên các biến độc lập (hoặc đặc trưng). Khác với hồi quy tuyến tính, hồi quy logistic sử dụng một hàm phi tuyến tính, được gọi là hàm logistic, để chuyển đổi kết quả dự đoán từ khoảng vô hạn ($-\infty$ đến ∞) về phạm vi xác suất giữa 0 và 1. [22]

2.4.1.2. Hàm Logistic

Hàm logistic, hay còn gọi là hàm sigmoid, là một hàm toán học được sử dụng để mô tả xác suất. Hàm sigmoid có dạng:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

- Trong đó:
 - z là tổ hợp tuyến tính của các biến đầu vào
$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$
 - $\sigma(z)$ là xác suất của lớp dương tính (label 1).
- Hàm sigmoid nén đầu ra của tổ hợp tuyến tính z vào khoảng (0, 1), biến nó thành xác suất.

2.4.1.3. Hàm mục tiêu

Trong Logistic Regression, chúng ta tối ưu hàm mất mát dựa trên Cross-Entropy Loss:

$$L(\beta) = - [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

- Trong đó:

- y là nhãn thực tế (0 hoặc 1).
- \hat{y} là xác suất dự đoán của mô hình $\hat{y} = \sigma(z)$

2.4.1.4. Tối ưu hoá

Để tìm các tham số β , ta sử dụng phương pháp Gradient Descent. Gradient của hàm mất mát đối với các tham số β được tính bằng:

$$\frac{\partial L}{\partial \beta_j} = \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$$

- Với m là số lượng mẫu trong tập huấn luyện.

2.4.1.5. Ưu điểm và hạn chế

- Ưu điểm:
 - Đơn giản và hiệu quả: Logistic Regression là một mô hình dễ hiểu và có thể triển khai nhanh chóng.
 - Dễ giải thích: Các hệ số β cho biết mức độ ảnh hưởng của từng biến đầu vào đến xác suất của lớp.
 - Khả năng xử lý đa biến: Có thể mở rộng để xử lý nhiều biến đầu vào.
- Nhược điểm:
 - Tuyến tính: Logistic Regression giả định mối quan hệ tuyến tính giữa các biến đầu vào và log-odds của đầu ra, có thể không phù hợp với các dữ liệu phức tạp.

- Dễ bị ảnh hưởng bởi đa cộng tuyến: Sự tương quan cao giữa các biến đầu vào có thể gây ra bất ổn trong mô hình.

2.4.1.6. Ứng dụng

Logistic Regression được sử dụng rộng rãi trong các lĩnh vực như:

- Phát hiện gian lận: Phát hiện gian lận là một lĩnh vực quan trọng trong các ngành công nghiệp như ngân hàng, bảo hiểm và thương mại điện tử. Gian lận có thể gây tổn thất tài chính nghiêm trọng, do đó, việc phát hiện và ngăn chặn gian lận là rất cần thiết.
 - Ví dụ thực tiễn: Ngân hàng sử dụng Logistic Regression để phân tích các giao dịch thẻ tín dụng nhằm phát hiện các hành vi bất thường. Khi một giao dịch được xác định có khả năng gian lận, hệ thống sẽ cảnh báo và ngăn chặn giao dịch đó để giảm thiểu tổn thất.
- Phân tích y tế: Trong y tế, Logistic Regression thường được sử dụng để phân loại bệnh nhân dựa trên nguy cơ mắc bệnh hoặc dự đoán kết quả điều trị.
 - Dự đoán nguy cơ mắc bệnh: Logistic Regression có thể được sử dụng để dự đoán nguy cơ bệnh tim mạch dựa trên các yếu tố như tuổi tác, giới tính, huyết áp và mức cholesterol.
 - Phân loại kết quả điều trị: Mô hình này cũng có thể dự đoán liệu một bệnh nhân có khả năng hồi phục sau một quá trình điều trị hay không dựa trên các yếu tố lâm sàng.
 - Ví dụ thực tiễn: Một bệnh viện sử dụng Logistic Regression để dự đoán nguy cơ bệnh tim của bệnh nhân. Mô hình giúp bác sĩ xác định những bệnh nhân có nguy cơ cao và áp dụng các biện pháp phòng ngừa thích hợp.
- Tài chính: Trong lĩnh vực tài chính, Logistic Regression được sử dụng để phân tích rủi ro tín dụng và xác định khả năng vỡ nợ của khách hàng.
 - Phân tích rủi ro tín dụng: Dựa trên các yếu tố như thu nhập, lịch sử tín dụng và số lần thanh toán trễ.

- Dự đoán vỡ nợ: Xác định khả năng một khách hàng có thể không thanh toán đúng hạn các khoản vay.
- Ví dụ thực tiễn: Một công ty tài chính sử dụng Logistic Regression để dự đoán khả năng vỡ nợ của khách hàng. Dựa trên các dự đoán này, công ty có thể điều chỉnh hạn mức tín dụng và lãi suất cho phù hợp.

2.4.2. Mô hình Random Forest

2.4.2.1. Khái niệm

Random Forest là một thuật toán học máy mạnh mẽ, thường được sử dụng cho các bài toán phân loại và hồi quy. Được giới thiệu bởi Leo Breiman vào năm 2001 [23], Random Forest là một kỹ thuật ensemble learning, kết hợp nhiều cây quyết định (decision trees) để cải thiện hiệu suất và giảm thiểu overfitting.

2.4.2.2. Cấu trúc của Random Forest

2.4.2.2.1. Cây Quyết Định (Decision Trees)

Mô hình cơ bản của Random Forest là cây quyết định. Một cây quyết định dựa trên việc phân chia dữ liệu đầu vào thành các nhánh dựa trên các điều kiện quyết định tại mỗi node. Mỗi nhánh dẫn đến các nhánh con hoặc lá (leaf) cho đến khi đạt đến một điều kiện dừng nào đó (chẳng hạn như số lượng mẫu trong node quá nhỏ hoặc đạt đến độ sâu tối đa của cây).

2.4.2.2.2. Random Forest

Random Forest xây dựng trên ý tưởng của cây quyết định nhưng tạo ra một tập hợp (forest) các cây quyết định độc lập bằng cách:

1. **Bagging (Bootstrap Aggregating):** Chọn mẫu ngẫu nhiên có lặp lại từ tập dữ liệu huấn luyện để tạo ra các tập con. Mỗi cây trong rừng được huấn luyện trên một tập con này.

2. **Random Feature Selection:** Tại mỗi node, thay vì xem xét tất cả các đặc trưng để chọn ra đặc trưng tốt nhất cho việc phân chia, Random Forest chỉ xem xét một tập con ngẫu nhiên của các đặc trưng.

2.4.2.3. Thuật toán

Quy trình xây dựng một mô hình Random Forest [24] gồm các bước chính sau:

1. **Tạo tập huấn luyện bootstrap:** Chọn ngẫu nhiên có lặp lại n_{nn} mẫu từ tập dữ liệu huấn luyện gốc để tạo ra các tập con.
2. **Xây dựng cây quyết định:** Đối với mỗi tập con:
 - Tạo một cây quyết định mà không cần cắt tỉa (pruning).
 - Tại mỗi node, chọn ngẫu nhiên m_{mm} đặc trưng từ p_{pp} đặc trưng ban đầu và chọn đặc trưng tốt nhất trong số đó để phân chia.
3. **Lặp lại:** Lặp lại quá trình trên cho k lần để tạo ra k cây quyết định.
4. **Kết hợp kết quả:** Đối với bài toán phân loại, lấy kết quả đầu ra của từng cây và áp dụng phương pháp đa số phiếu bầu (majority voting) để quyết định lớp dự đoán cuối cùng.

2.4.2.4. Ưu điểm và hạn chế

- **Ưu điểm:**
 - **Hiệu suất cao:** Random Forest thường cho kết quả chính xác và hiệu suất cao hơn so với các mô hình đơn lẻ như cây quyết định.
 - **Giảm thiểu overfitting:** Do sử dụng nhiều cây và kỹ thuật bagging, Random Forest ít bị overfitting hơn so với các cây quyết định đơn lẻ.
 - **Xử lý được dữ liệu thiếu:** Random Forest có khả năng xử lý dữ liệu bị thiếu một cách hiệu quả.

- **Tính quan trọng của đặc trưng:** Random Forest cung cấp thông tin về tầm quan trọng của các đặc trưng, giúp hiểu rõ hơn về dữ liệu.
- Hạn chế:
 - **Phức tạp và tốn kém tính toán:** Việc huấn luyện nhiều cây quyết định và kết hợp kết quả của chúng có thể tốn nhiều thời gian và tài nguyên tính toán.
 - **Khó giải thích:** Mặc dù các cây quyết định riêng lẻ dễ hiểu, nhưng việc kết hợp nhiều cây trong Random Forest làm cho mô hình tổng thể trở nên phức tạp và khó giải thích hơn.

2.4.2.5. Ứng dụng

Random Forest được sử dụng rộng rãi trong nhiều lĩnh vực:

- Phát hiện gian lận: Ngân hàng sử dụng Random Forest để phân tích các giao dịch thẻ tín dụng nhằm phát hiện các hành vi bất thường. Khi một giao dịch được xác định có khả năng gian lận, hệ thống sẽ cảnh báo và ngăn chặn giao dịch đó để giảm thiểu tổn thất. Random Forest giúp giảm thiểu tỷ lệ bỏ sót các giao dịch gian lận và tăng cường độ chính xác trong việc phát hiện các hành vi bất thường.
- Phân tích y tế: Một bệnh viện sử dụng Random Forest để dự đoán nguy cơ bệnh tim của bệnh nhân. Mô hình giúp bác sĩ xác định những bệnh nhân có nguy cơ cao và áp dụng các biện pháp phòng ngừa thích hợp. Random Forest cũng có thể được sử dụng để phân loại các loại ung thư dựa trên dữ liệu di truyền và các yếu tố nguy cơ khác.
- Tài chính: Một công ty tài chính sử dụng Random Forest để dự đoán khả năng vỡ nợ của khách hàng. Dựa trên các dự đoán này, công ty có thể điều chỉnh hạn mức tín dụng và lãi suất cho phù hợp. Random Forest giúp cải thiện khả năng phân loại các khách hàng có rủi ro cao và tối ưu hóa các quyết định cho vay.
- Tiếp thị: Một công ty bán lẻ sử dụng Random Forest để phân loại khách hàng thành các nhóm khác nhau dựa trên lịch sử mua sắm và phản hồi từ các chiến dịch tiếp thị. Điều này giúp công ty điều chỉnh các chiến lược

tiếp thị và quảng cáo để tối ưu hóa doanh thu. Random Forest có thể giúp xác định những khách hàng tiềm năng cao và tăng cường hiệu quả của các chiến dịch tiếp thị.

2.4.4. Mô hình XGBoost

2.4.4.1. Khái niệm

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy mạnh mẽ và hiệu quả, đặc biệt nổi tiếng trong các cuộc thi học máy và thực tế triển khai công nghiệp. Được phát triển bởi Tianqi Chen và Carlos Guestrin vào năm 2016 [25], XGBoost là một phiên bản nâng cao của thuật toán Gradient Boosting, với nhiều cải tiến về tốc độ và hiệu quả.

2.4.4.2. Cấu Trúc và Nguyên Lý Hoạt Động

2.4.4.2.1. Gradient Boosting

Gradient Boosting là một phương pháp học ensemble, kết hợp nhiều mô hình yếu (weak learners) để tạo ra một mô hình mạnh hơn. Mô hình yếu phổ biến nhất là cây quyết định (decision trees). Quá trình Gradient Boosting bao gồm:

1. Khởi tạo mô hình cơ sở: Bắt đầu với một mô hình cơ sở (thường là một cây quyết định đơn giản).
2. Tính toán lỗi: Tính toán lỗi của mô hình hiện tại.
3. Huấn luyện mô hình mới: Huấn luyện một mô hình mới để dự đoán phần dư (residuals) của mô hình hiện tại.
4. Cập nhật mô hình: Cập nhật mô hình hiện tại bằng cách cộng mô hình mới với một trọng số.
5. Lặp lại: Lặp lại quá trình trên cho đến khi đạt được số lượng mô hình nhất định hoặc lỗi giảm xuống mức chấp nhận được.

2.4.4.2.2. Cải Tiết của XGBoost

XGBoost cải tiến trên Gradient Boosting truyền thống với một số điểm chính:

1. Regularization: XGBoost thêm vào các thuật toán regularization (L1 và L2) để giảm overfitting.
2. Handling Missing Data: XGBoost có khả năng xử lý dữ liệu thiếu mà không cần tiền xử lý đặc biệt.
3. Parallel Processing: XGBoost hỗ trợ xử lý song song, giúp tăng tốc độ huấn luyện.
4. Tree Pruning: Sử dụng thuật toán "max depth" và "pruning" để xây dựng cây quyết định hiệu quả hơn.
5. Shrinkage and Column Subsampling: Sử dụng kỹ thuật shrinkage (giảm trọng số của mỗi cây) và column subsampling (chọn ngẫu nhiên các đặc trưng) để giảm overfitting.

2.4.4.3. Thuật Toán XGBoost

Quy trình xây dựng một mô hình XGBoost gồm các bước chính sau:

1. Khởi tạo mô hình cơ sở: Thường bắt đầu với một giá trị dự đoán cố định (ví dụ: trung bình của nhãn đầu ra).
2. Tính toán phần dư: Tính toán phần dư của mỗi mẫu dữ liệu.
3. Huấn luyện cây quyết định: Huấn luyện một cây quyết định để dự đoán phần dư.
4. Cập nhật mô hình: Cập nhật mô hình hiện tại bằng cách cộng cây mới với một trọng số.
5. Lặp lại: Lặp lại quá trình trên cho đến khi đạt được số lượng cây nhất định hoặc lỗi giảm xuống mức chấp nhận được.

2.4.4.4. Ưu Điểm và Hạn Chế

- **Ưu Điểm**
 - Hiệu suất cao: XGBoost thường cho kết quả chính xác cao và là sự lựa chọn hàng đầu trong nhiều cuộc thi học máy.
 - Xử lý dữ liệu thiếu: XGBoost có khả năng xử lý dữ liệu thiếu mà không cần tiền xử lý đặc biệt.
 - Tối ưu hóa tốc độ: Hỗ trợ xử lý song song và các kỹ thuật tối ưu hóa giúp tăng tốc độ huấn luyện.

- Giảm overfitting: Các thuật toán regularization và kỹ thuật column subsampling giúp giảm thiểu overfitting.
- Hạn chế:
 - Phức tạp và tốn kém tính toán: Việc huấn luyện và tối ưu hóa mô hình XGBoost có thể tốn nhiều thời gian và tài nguyên tính toán.
 - Khó giải thích: Mặc dù các cây quyết định riêng lẻ dễ hiểu, nhưng việc kết hợp nhiều cây trong XGBoost làm cho mô hình tổng thể trở nên phức tạp và khó giải thích hơn.

2.4.4.5. Ứng dụng

XGBoost được sử dụng rộng rãi trong nhiều lĩnh vực:

- Phát hiện gian lận: XGBoost được sử dụng để phát hiện gian lận trong các giao dịch tài chính, chẳng hạn như giao dịch thẻ tín dụng. Các mô hình XGBoost có khả năng phân loại chính xác các giao dịch gian lận dựa trên các đặc trưng của giao dịch.
- Y tế: Dự đoán bệnh, phân loại các loại bệnh và xác định các yếu tố nguy cơ. Ví dụ, XGBoost có thể được sử dụng để dự đoán nguy cơ mắc bệnh tim mạch.
- Tài chính: Phân tích rủi ro tín dụng và xác định khả năng vỡ nợ của khách hàng. XGBoost giúp cải thiện khả năng phân loại các khách hàng có rủi ro cao và tối ưu hóa các quyết định cho vay.
- Tiếp thị: Phân loại khách hàng và dự đoán hành vi mua sắm. XGBoost giúp xác định những khách hàng tiềm năng cao và tăng cường hiệu quả của các chiến dịch tiếp thị.

2.5. Tổng quan tài liệu

Có nhiều tài liệu về phát hiện gian lận tài chính do tầm quan trọng cao của nó trong việc giảm tội phạm mạng cũng như từ góc độ kinh doanh. Một số nhà nghiên cứu cũng đã thực hiện các tổng quan tài liệu về các bài viết được xuất bản trong những năm 2000 và 2010.

Để phát hiện gian lận tài chính, các nhà nghiên cứu thường sử dụng các kỹ thuật phát hiện ngoại lệ (Jayakumar & Thomas, 2013) [26] với các tập dữ liệu có sự mất cân bằng cao. Cũng có thể xảy ra nhiều loại gian lận tài chính khác nhau. Một bài viết đề xuất bốn loại gian lận tài chính – gian lận báo cáo tài chính, gian lận giao dịch, gian lận bảo hiểm và gian lận tín dụng (Jans et al., 2011) [27]. Trong dự án này, trọng tâm là gian lận giao dịch cụ thể, đặc biệt là trong thanh toán di động.

Nhiều kỹ thuật khác nhau đã được thử nghiệm để phát hiện gian lận tài chính:

- Phua et al., (2004) đã sử dụng Mạng nơ-ron, Naïve Bayes và Cây quyết định để phát hiện gian lận bảo hiểm ô tô [28].
- Ravisankar et al., (2011) phát hiện gian lận báo cáo tài chính ở các công ty Trung Quốc; một bài viết khác đã sử dụng SVM, Lập trình di truyền, Hồi quy Logistic và Mạng nơ-ron [29].
- Phân cụm dựa trên mật độ (Dharwa & Patel, 2011) [30] và Cây quyết định nhạy cảm với chi phí (Sahin et al., 2013) đã được sử dụng cho gian lận thẻ tín dụng [31].
- Sorournejad et al., (2016) thảo luận về cả hai cách tiếp cận dựa trên học máy có giám sát và không giám sát, liên quan đến ANN (Mạng nơ-ron nhân tạo), SVM, HMM (Mô hình Hidden Markov), và phân cụm [32].
- Wedge et al., (2018) giải quyết vấn đề dữ liệu không cân bằng dẫn đến số lượng cao các kết quả dương tính giả, và một số tài liệu đề xuất các kỹ thuật để giảm thiểu vấn đề này [33].

Tuy nhiên, có rất ít tài liệu về việc phát hiện các giao dịch gian lận trong thanh toán di động, có lẽ do những tiến bộ công nghệ tương đối gần đây. Albashrawi, (2016) [34] trình bày một tổng quan hệ thống về các phương pháp được sử dụng nhiều nhất trong phát hiện gian lận tài chính.

Năm kỹ thuật hàng đầu được trình bày trong bảng dưới đây:

Kỹ thuật	Tần suất sử dụng
Logistic Regression	13% (17 bài báo)

Neural Networks	11% (15 bài báo)
Decision Trees	11% (15 bài báo)
Support Vector Machines	9% (12 bài báo)
Naïve Bayes	6% (8 bài báo)

Bảng 1: Tần suất sử dụng các kỹ thuật học máy trong các vấn đề phát hiện gian lận

Chương 3: Phương pháp nghiên cứu

3.1. Phương pháp

Phương pháp này được xem như các sản phẩm đầu ra của dự án. Nó mô tả kết quả của từng giai đoạn đã được thử nghiệm và thực hiện so sánh giữa chúng để xác định kỹ thuật nào là tốt nhất để giải quyết vấn đề phát hiện gian lận.

Mỗi giai đoạn của dự án đều có một sản phẩm đầu ra mô tả các phát hiện trong giai đoạn đó. Các sản phẩm đầu ra này được sử dụng trong dự án cuối cùng được giải thích dưới đây:

Các giai đoạn	Mô tả
Hiểu tập dữ liệu	<ul style="list-style-type: none">○ Báo cáo tóm tắt về tập dữ liệu và từng biến mà nó chứa, kèm theo các trực quan hóa cần thiết.
Phân tích dữ liệu khám phá	<ul style="list-style-type: none">○ Báo cáo về các phân tích đã thực hiện và những phát hiện quan trọng với mô tả đầy đủ về các phân đoạn dữ liệu đã xem xét.○ Giải thuyết về sự phân tách giữa các giao dịch gian lận và không gian lận.○ Các trực quan hóa và biểu đồ cho thấy sự khác biệt giữa các giao dịch gian lận và không gian lận.○ Mã Python của các phân tích đã thực hiện.
Khởi tạo mô hình	<ul style="list-style-type: none">○ Báo cáo về kết quả của các kỹ thuật khác nhau đã được thử nghiệm, các vòng lặp đã được thực hiện, các phép biến đổi dữ liệu và cách tiếp cận lập mô hình chi tiết.○ Mã Python được sử dụng để xây dựng các mô hình học máy
Báo cáo dự án	<ul style="list-style-type: none">○ Báo cáo cuối cùng tóm tắt công việc đã thực

	hiện trong suốt quá trình dự án, nêu bật các phát hiện chính, so sánh các mô hình khác nhau và xác định mô hình tốt nhất cho việc phát hiện gian lận tài chính.
--	---

Bảng 2: Mô tả chi tiết dự án

3.2. Công cụ sử dụng

Dự án này hoàn toàn được thực hiện bằng Python và phân tích đã được ghi lại trong sổ ghi chép Jupyter. Các thư viện python tiêu chuẩn đã được sử dụng để tiến hành các phân tích khác nhau. Những thư viện này được mô tả bên dưới:

- **sklearn** – được sử dụng cho các mô hình học máy
- **seaborn** – được sử dụng để tạo ra các biểu đồ và trực quan hóa
- **pandas** – được sử dụng để đọc và chuyển đổi dữ liệu

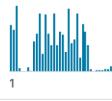
3.3. Nguồn dữ liệu

Do tính chất riêng tư của dữ liệu tài chính, có rất ít tập dữ liệu công khai có sẵn để sử dụng cho phân tích. Trong dự án này, một tập dữ liệu tổng hợp, có sẵn công khai trên Kaggle, được tạo ra bằng cách sử dụng một trình giả lập có tên là PaySim. Tập dữ liệu này được tạo ra bằng cách sử dụng các chỉ số tổng hợp từ tập dữ liệu riêng tư của một công ty dịch vụ tài chính di động đa quốc gia, sau đó đã được thêm các mục dữ liệu gian lận (TESTIMON @ NTNU, n.d.)

Tập dữ liệu chứa 11 cột thông tin cho gần 6 triệu hàng dữ liệu. Các cột chính có sẵn là:

- Loại giao dịch
- Số tiền giao dịch
- ID khách hàng và ID người nhận
- Số dư cũ và mới của khách hàng và người nhận
- Thời gian của giao dịch
- Liệu giao dịch có phải là gian lận hay không

Trong hình dưới đây, một bức ảnh chụp nhanh các dòng đầu tiên của tập dữ liệu được trình bày:

# step	type	# amount	# nameOrig	# oldbalanceOrg	# newbalanceOrig	# nameDest
	CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER	amount of the transaction in local currency	customer who started the transaction	initial balance before the transaction	customer's balance after the transaction.	recipient ID of the transaction.
1	CASH_OUT PAYMENT Other (1973625)	35% 34% 31%	 743	0 92.4m	6353307 unique values	0 59.6m
1	PAYMENT	9839.64	C1231006815	178136.0	160296.36	M1979787155
1	PAYMENT	1864.28	C1666544295	21249.8	19384.72	M2044282225
1	TRANSFER	181.0	C1305486145	181.0	0.0	C553264065
1	CASH_OUT	181.0	C840083671	181.0	0.0	C38997810
1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703
1	PAYMENT	7817.71	C90045638	53868.0	46042.29	M573487274
1	PAYMENT	7107.77	C154988899	183195.0	176087.23	M408869119
1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333
1	PAYMENT	4024.36	C1265012928	2671.0	0.0	M1176932104
1	DEBIT	5337.77	C712410124	41728.0	36382.23	C195600860

Hình 2: Ảnh chụp nhanh của tập dữ liệu thô

3.4. Tổng quan dự án

Để xây dựng mô hình phát hiện gian lận trong giao dịch gian lận, tôi sử dụng bộ dữ liệu công khai liên quan đến các giao dịch tài chính được mô phỏng tiền di động PaySim, được chia sẻ bởi nhà khoa học dữ liệu EDGAR LOPEZ-ROJAS vào năm 2016. Tập dữ liệu Synthetic Financial Datasets For Fraud Detection, được đăng tải công khai trên Kaggle:

<https://www.kaggle.com/datasets/ealaxi/paysim1/data>

Tập dữ liệu tổng hợp được tạo ra bằng cách sử dụng trình mô phỏng có tên là PaySim như một cách tiếp cận cho vấn đề như vậy. PaySim sử dụng dữ liệu tổng hợp từ bộ dữ liệu riêng tư để tạo ra một bộ dữ liệu tổng hợp giống với hoạt động bình thường của các giao dịch và tiêm hành vi độc hại để sau này đánh giá hiệu suất của các phương pháp phát hiện gian lận.

PaySim mô phỏng các giao dịch tiền di động dựa trên một mẫu giao dịch thực được trích xuất từ nhật ký tài chính một tháng từ dịch vụ tiền di động được triển khai ở một quốc gia châu Phi. Nhật ký gốc được cung cấp bởi một công ty đa quốc gia, nhà cung cấp dịch vụ tài chính di động hiện đang hoạt động tại hơn 14 quốc gia trên toàn thế giới.

Bộ dữ liệu tổng hợp này được thu nhỏ 1/4 bộ dữ liệu gốc và nó được tạo riêng cho Kaggle.

Quy trình mô hình tổng quan gồm các bước:

- Bước 1: Tìm kiếm dữ liệu:

- Từ ý tưởng ban đầu là xây dựng mô hình phát hiện gian lận với điểm nổi trội là nhận diện các giao dịch bất thường, cùng với việc áp dụng mô hình máy học không giám sát để tìm ra các giao dịch có dấu hiệu gian lận trong lĩnh vực tài chính. Bộ dữ liệu được thu thập bao gồm các thông tin về số tiền giao dịch, tần suất giao dịch, loại giao dịch, thông tin tài khoản gửi và nhận, lịch sử giao dịch và các thông tin giao dịch khác.
- Bộ dữ liệu sau đó được xử lý thô và chứa đựng hơn 6,362,604 quan sát để phục vụ cho việc chạy các mô hình học máy.

- Bước 2: Xử lý dữ liệu tập huấn luyện train và huấn luyện mô hình

- Sử dụng ngôn ngữ Python và công cụ Google Colab để thực hiện tiền xử lý dữ liệu.
- Kiểm tra tổng quan các dữ liệu đầu vào: Tính số lượng quan sát và mô tả dữ liệu, xây dựng mô hình, xem qua các kiểu dữ liệu của từng thuộc tính và giá trị của thuộc tính nào bị thiếu.
- Xác định các nhóm thuộc tính:
 - Dạng dữ liệu định lượng
 - Dạng dữ liệu định tính trong bộ dữ liệu.
- Trực quan hóa các dữ liệu định tính và định lượng: Đánh giá bộ dữ liệu.
- Loại bỏ các biến ít gây ảnh hưởng tới mô hình: Điều này giúp giảm bớt độ phức tạp và tăng hiệu suất mô hình.
- Đánh giá và làm sạch dữ liệu: Xử lý các dữ liệu thiếu và các dữ liệu không phù hợp.
- Xử lý dữ liệu mất cân bằng: Sử dụng phương pháp SMOTE (Synthetic Minority Over-sampling Technique) để cân bằng dữ liệu. Phương pháp này giúp tăng số lượng mẫu của lớp thiểu số bằng cách tạo ra các mẫu mới dựa trên các mẫu hiện có.

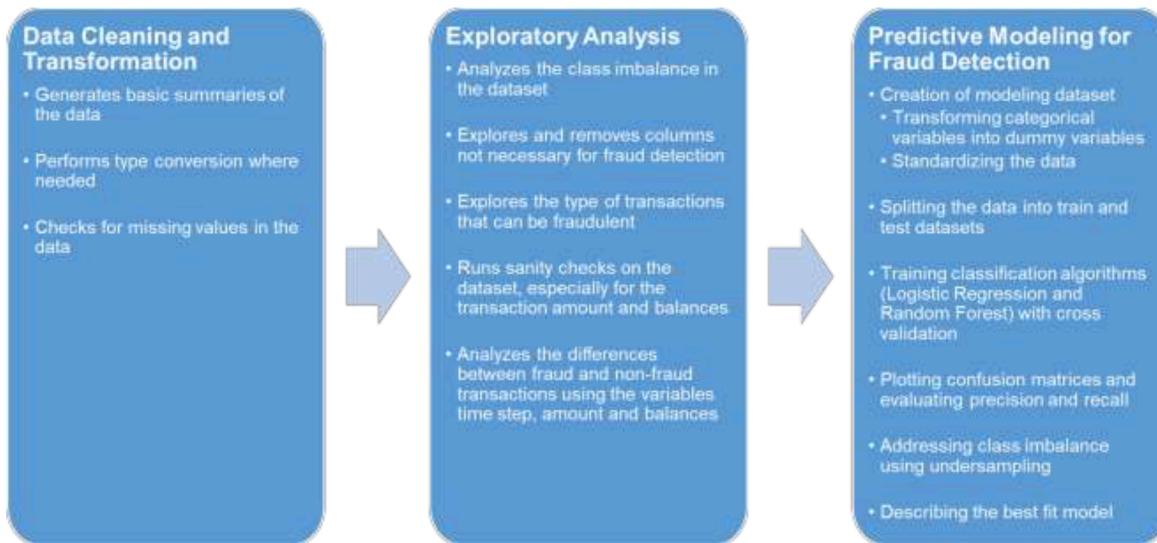
- Tiến hành huấn luyện các mô hình học máy: Logistic Regression, Random Forest, XGBoost trên bộ dữ liệu sau khi đã xử lý ở các bước trên, các mô hình này sẽ được sử dụng để phát hiện các giao dịch bất thường và có khả năng gian lận.
 - Bước 3: Đánh giá và lựa chọn mô hình học máy phù hợp
- Đánh giá kết quả của các mô hình đã được huấn luyện: Dựa trên bộ dữ liệu kiểm thử (test data), đánh giá hiệu suất của các mô hình đã được huấn luyện trong bước trước.
- Lựa chọn thuật toán phù hợp nhất: Sử dụng các tiêu chí đánh giá như độ chính xác (accuracy), độ nhạy (recall) và F1-score để so sánh các mô hình. Từ đó, lựa chọn thuật toán có hiệu suất tốt nhất để áp dụng cho việc dự đoán trên dữ liệu thực tế.

Chương 4: Xây dựng mô hình học máy

4.1. Phân tích dữ liệu

Phần này mô tả từng bước của quá trình phân tích được thực hiện một cách chi tiết. Tất cả các phân tích được tài liệu hóa dưới định dạng Jupyter notebook, và mã được trình bày cùng với các đầu ra.

Phân tích được chia thành ba phần chính. Các phần này được mô tả trong sơ đồ dưới đây.



Hình 3: Cấu trúc phân tích [16]

4.2. Phân tích chi tiết

Các trang tiếp theo trình bày quy trình từng bước được thực hiện theo cấu trúc phân tích đã đề cập. Các đoạn mã và đồ họa liên quan được bao gồm dựa trên ngôn ngữ lập trình Python.

4.2.1. Làm sạch dữ liệu

Phần này mô tả việc khám phá dữ liệu được thực hiện để hiểu dữ liệu và sự khác biệt giữa các giao dịch gian lận và không gian lận.

4.2.1.1. Mô tả dữ liệu

Dữ liệu được sử dụng cho phân tích này là một tập dữ liệu giao dịch kỹ thuật số được tạo ra một cách tổng hợp bằng cách sử dụng một trình giả lập có tên là PaySim. PaySim mô phỏng các giao dịch tiền di động dựa trên một mẫu các giao dịch thực được trích xuất từ một tháng nhật ký tài chính của một dịch vụ tiền di động được triển khai tại một quốc gia ở châu Phi. Nó tổng hợp dữ liệu ẩn danh từ tập dữ liệu riêng để tạo ra một tập dữ liệu tổng hợp và sau đó tiêm vào các giao dịch gian lận.

Tập dữ liệu có hơn 6 triệu giao dịch và 11 biến. Có một biến được đặt tên là ‘isFraud’ cho biết tình trạng gian lận thực tế của giao dịch. Đây là biến lớp cho phân tích của tôi.

Các cột trong tập dữ liệu được mô tả như sau:

Tên biến	Mô tả
step	Ánh xạ một đơn vị thời gian trong thế giới thực. 1 bước tương đương với 1 giờ thời gian.
type	Chỉ định loại giao dịch: CASH-IN, CASH-OUT, DEBIT, PAYMENT hoặc TRANSFER.
amount	Số tiền giao dịch bằng tiền tệ địa phương.
nameOrig	Định danh của khách hàng đã bắt đầu giao dịch.
oldbalanceOrg	Số dư ban đầu của người gửi trước giao dịch.
newbalanceOrg	Số dư của người gửi sau giao dịch.
nameDest	Định danh của người nhận đã nhận giao dịch.
oldbalanceDest	Số dư ban đầu của người nhận trước giao dịch.
newbalanceDest	Số dư của người nhận sau giao dịch.
isFraud	Chỉ ra liệu giao dịch có thực sự gian lận hay không. Giá trị 1 chỉ ra gian lận và 0 chỉ ra không gian lận.

Bảng 3: Các biến trong tập dữ liệu

4.2.1.2. Chuyển đổi kiểu dữ liệu

Vì tất cả các cột trong dữ liệu đều cần có kiểu dữ liệu phù hợp để phân tích, nên sẽ kiểm tra xem có cần chuyển đổi kiểu dữ liệu hay không. Sau đây là các kiểu dữ liệu ban đầu của các cột được python đọc:

	step	int64
	type	object
	amount	float64
	nameOrig	object
	oldbalanceOrg	float64
	newbalanceOrig	float64
	nameDest	object
	oldbalanceDest	float64
	newbalanceDest	float64
	isFraud	int64
	isFlaggedFraud	int64
	dtype:	object

Hình 4: Kiểu dữ liệu ban đầu của các cột

Biến isFraud được đọc là số nguyên. Vì đây là biến lớp, chúng ta chuyển đổi nó thành kiểu đối tượng. Đoạn code python sau được sử dụng để thực hiện chuyển đổi này:

```
# Convert class variables type to object
df['isFraud'] = df['isFraud'].astype('object')

✓ 0.0s
```

Hình 5: [Đoạn mã] Chuyển đổi kiểu dữ liệu

4.2.1.3. Thống kê mô tả

Trước khi tiến hành phân tích, tôi trình bày số liệu thống kê tóm tắt của các biến. Trong trường hợp biến số, tôi đánh giá giá trị trung bình, độ lệch chuẩn và phạm vi giá trị ở các phần trăm khác nhau. Trong trường hợp biến phân loại, tôi chỉ đánh giá số lượng danh mục duy nhất, danh mục thường gặp nhất và tần suất của danh mục đó.

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08

Hình 6: Tóm tắt thống kê các biến số

Trong trường hợp biến phân loại, chúng ta chỉ đánh giá số lượng danh mục duy nhất, danh mục có tần suất xuất hiện nhiều nhất và tần suất của danh mục đó.

	type	nameOrig	nameDest	isFraud
count	6362620	6362620	6362620	6362620
unique	5	6353307	2722362	2
top	CASH_OUT	C1902386530	C1286084959	0
freq	2237500	3	113	6354407

Hình 7: Tóm tắt thống kê của các biến phân loại

4.2.1.4. Kiểm tra giá trị bị thiếu

Trong giai đoạn này, tôi kiểm tra xem có giá trị nào bị thiếu trong tập dữ liệu không. Đoạn mã và kết quả đầu ra sau đây cho biết tổng số giá trị bị thiếu trong tất cả các cột.

```
missing_values(df)
```

✓ 0.6s

	index	Missing Values	% of Total Values	Data_type
0	step	0	0.0	int64
1	type	0	0.0	object
2	amount	0	0.0	float64
3	nameOrig	0	0.0	object
4	oldbalanceOrg	0	0.0	float64
5	newbalanceOrig	0	0.0	float64
6	nameDest	0	0.0	object
7	oldbalanceDest	0	0.0	float64
8	newbalanceDest	0	0.0	float64
9	isFraud	0	0.0	int64
10	isFlaggedFraud	0	0.0	int64

Hình 8: [Đoạn mã] Kiểm tra giá trị bị thiếu

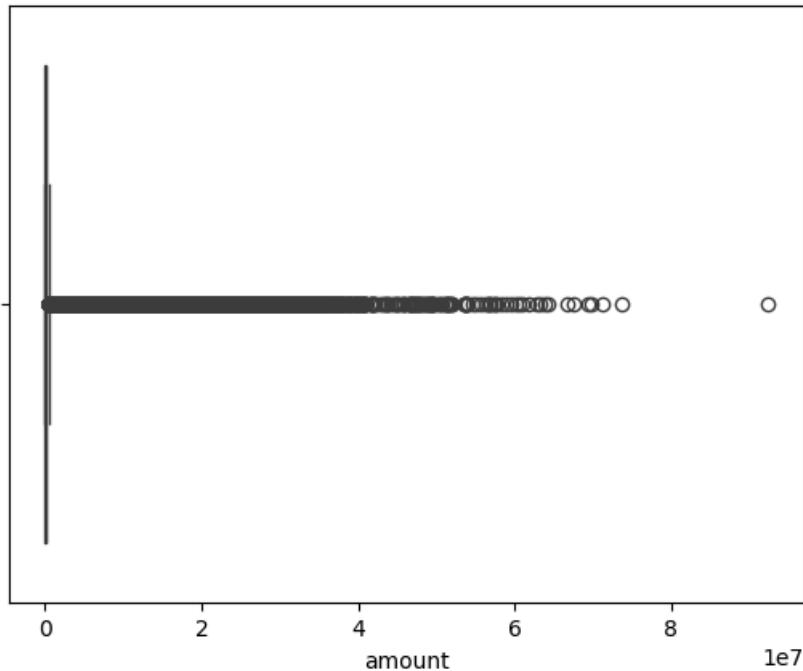
- Bộ dữ liệu không có dữ liệu bị thiếu.

4.2.2. Phân tích khám phá

4.2.2.1. Kiểm tra cột Amount

Kiểm tra giá trị ngoại lệ (outliers) của cột amount:

Hình 9: Biểu đồ boxplot của biến Amount



- Cột amount có các giá trị ngoại lệ.

4.2.2.2. Sự mất cân bằng của các lớp dữ liệu

Trong phân tích thăm dò này, tôi đánh giá sự mất cân bằng lớp trong tập dữ liệu. Sự mất cân bằng lớp được định nghĩa là phần trăm của tổng số giao dịch được trình bày trong cột isFraud.

Tần suất **phần trăm** đầu ra cho biến lớp isFraud được hiển thị bên dưới:

```
Total_transactions = len(df)
normal = len(df[df.isFraud == 0])
fraudulent = len(df[df.isFraud == 1])
fraud_percentage = round(fraudulent/Total_transactions*100, 2)
print('Tổng số giao dịch là {}'.format(Total_transactions))
print('Số giao dịch bình thường là {}'.format(normal))
print('Số giao dịch gian lận là {}'.format(fraudulent))
print('Tỷ lệ phần trăm giao dịch gian lận là {}'.format(fraud_percentage))
```

✓ 0.2s
Tổng số giao dịch là 6362620
Số giao dịch bình thường là 6354407
Số giao dịch gian lận là 8213
Tỷ lệ phần trăm giao dịch gian lận là 0.13

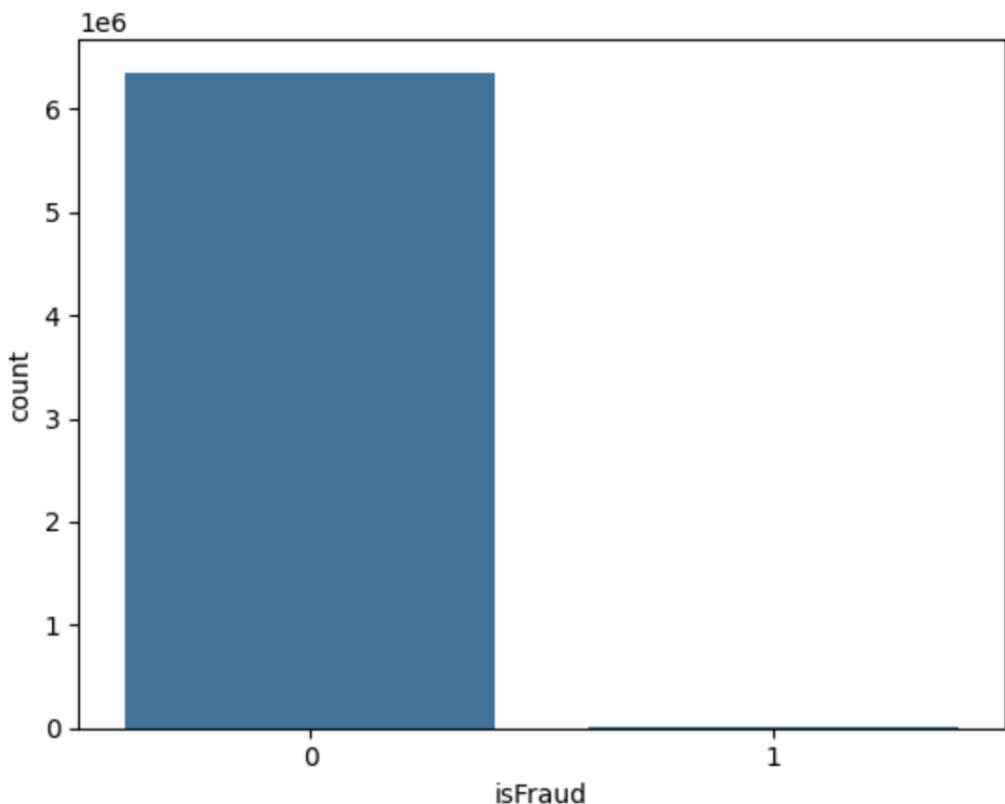
Hình 10: [Đoạn mã] Tỷ lệ % giao dịch gian lận

```
normal_percentage = round(normal/Total_transactions*100, 2)
print('Tỷ lệ giao dịch bình thường là {}'.format(normal_percentage))

✓ 0.0s
```

Tỷ lệ giao dịch bình thường là 99.87

Hình 11: [Đoạn mã] Tỷ lệ % giao dịch bình thường



Hình 12: Minh họa sự mất cân bằng dữ liệu

Như chúng ta có thể thấy từ hình trên, có một sự khác biệt rất lớn giữa các giao dịch.

Chỉ có 0,13% (8.213) giao dịch trong tập dữ liệu là gian lận, cho thấy sự mất cân bằng cấp cao trong tập dữ liệu. Điều này quan trọng vì nếu chúng ta xây dựng một mô hình học máy trên dữ liệu bị lệch cao này, các giao dịch không gian lận sẽ ảnh hưởng đến quá trình đào tạo mô hình gần như hoàn toàn, do đó ảnh hưởng đến kết quả.

4.2.2.3. Các loại giao dịch

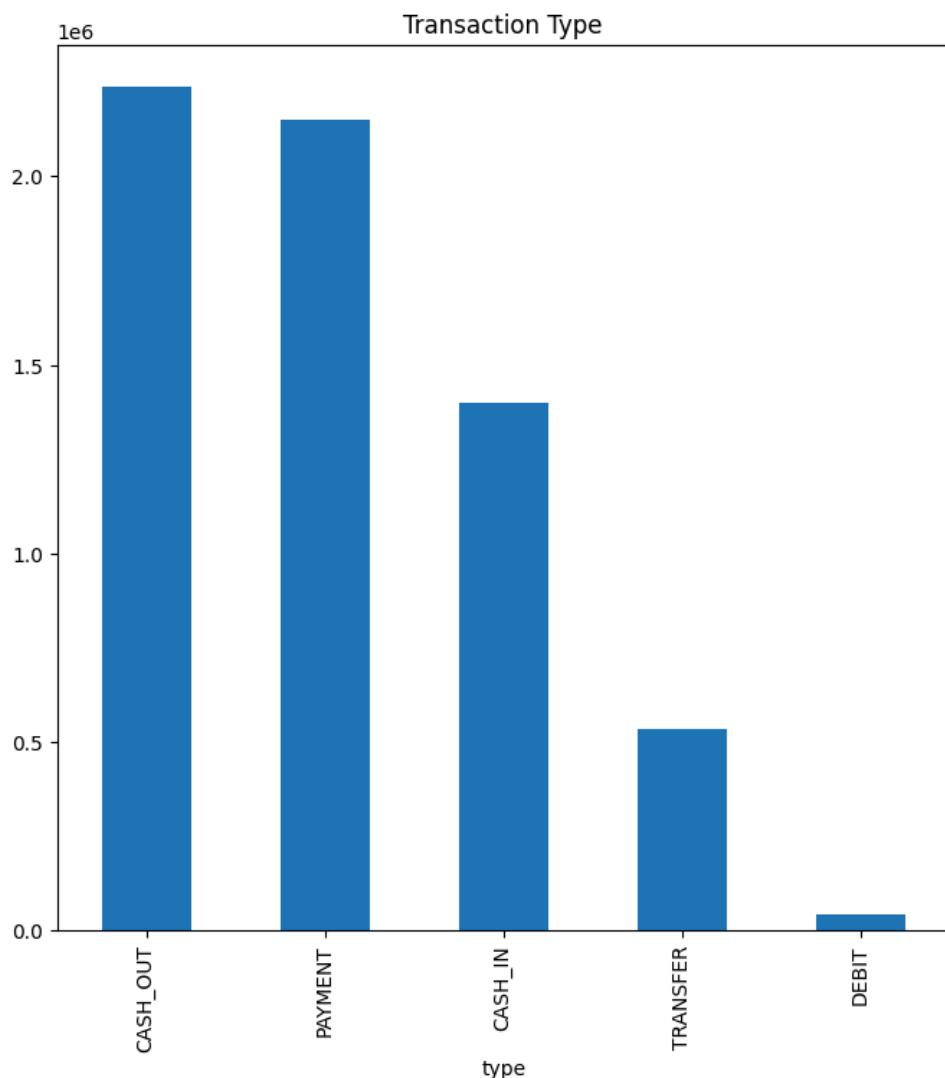
a. Tần suất các loại giao dịch

Trong phần này, tôi sẽ khám phá tập dữ liệu bằng cách kiểm tra biến 'loại'. Tôi trình bày các 'loại' giao dịch khác nhau là gì và loại nào trong số các loại này có thể là gian lận.

Biểu đồ sau đây cho thấy tần suất của các loại giao dịch khác nhau:

```
type
CASH_OUT    2237500
PAYMENT     2151495
CASH_IN     1399284
TRANSFER    532909
DEBIT        41432
Name: count, dtype: int64
```

Hình 13: Số lượng giao dịch theo từng loại



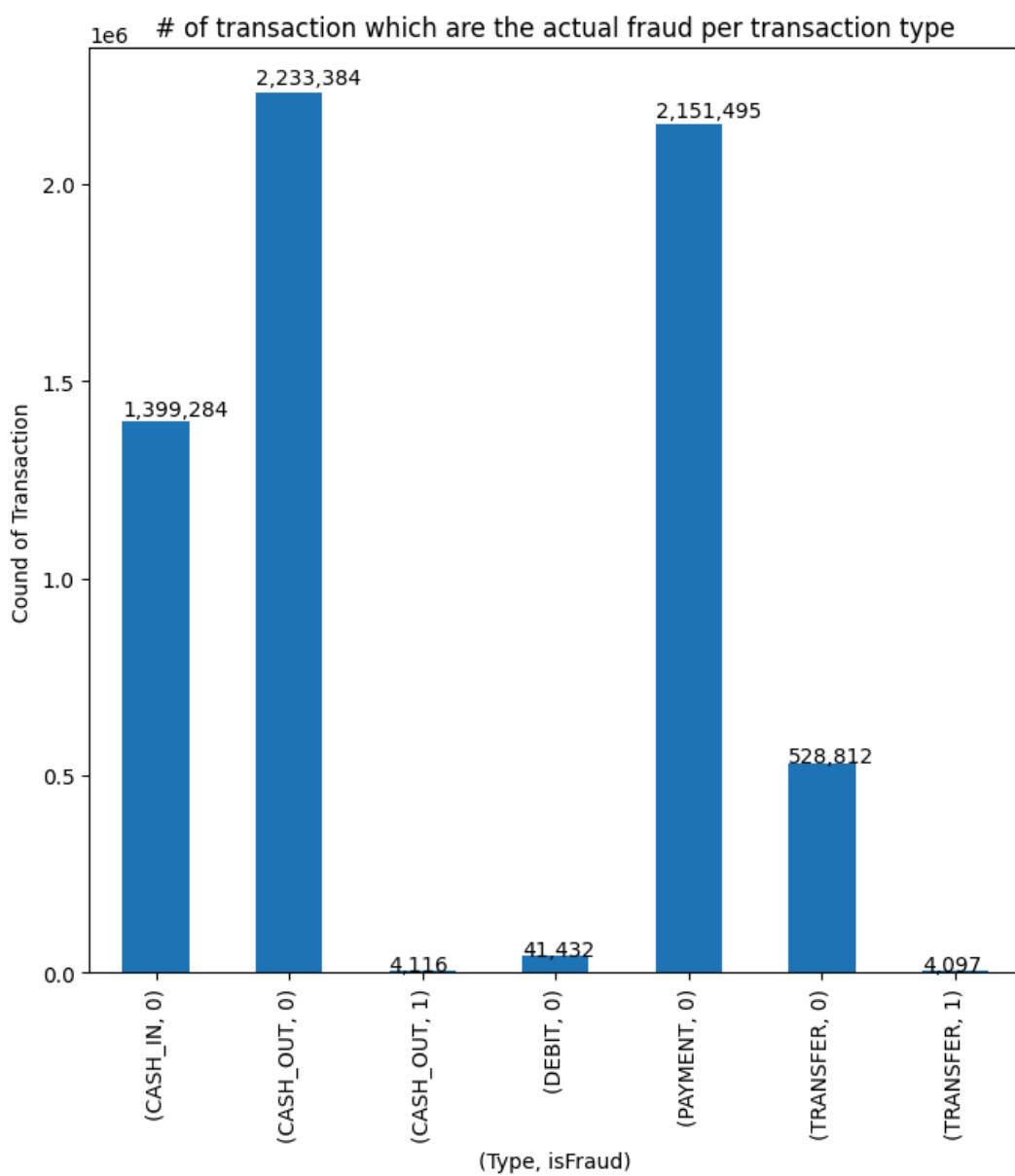
Hình 14: Minh họa số lượng của từng loại giao dịch

Các loại giao dịch thường gặp nhất là CASH OUT và PAYMENT.

b. Giao dịch gian lận theo loại giao dịch

Có 2 cờ mà tôi thấy nổi bật và rất thú vị để xem xét: cột isFraud và isFlaggedFraud. Từ từ điển dữ liệu, isFraud là chỉ số cho biết các giao dịch thực sự là gian lận, trong khi isFlaggedFraud là những gì hệ thống ngăn chặn giao dịch do một số ngưỡng được kích hoạt.

Hãy nhanh chóng kiểm tra loại giao dịch nào đang bị gắn cờ và là gian lận:



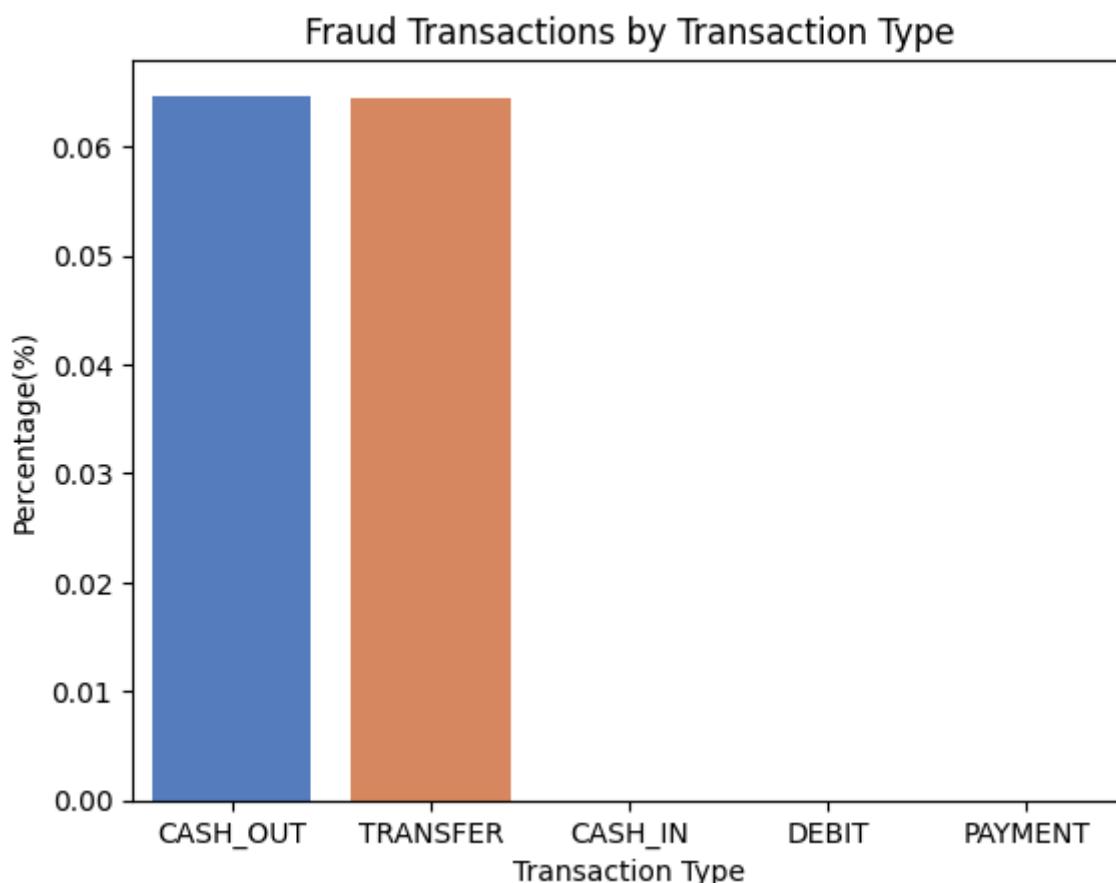
Hình 15: Minh họa số lượng giao dịch gian lận và không gian lận theo loại giao dịch

- Chỉ có các giao dịch CASH-OUT và TRANSFER mới có thể là gian lận.

c. Phần trăm loại giao dịch là gian lận

```
type
CASH_OUT    4116
TRANSFER    4097
CASH_IN      0
DEBIT        0
PAYMENT      0
Name: isFraud, dtype: object
```

Hình 16: Số lượng giao dịch gian lận theo từng loại giao dịch



Hình 17: Minh họa số lượng giao dịch gian lận theo từng loại giao dịch

Do chỉ có các giao dịch CASH-OUT và TRANSFER mới có thể là gian lận. Vì vậy, chỉ giữ lại hai loại giao dịch này trong tập dữ liệu của tôi là hợp lý.

Và từ hình trên, các giao dịch gian lận được chia thành tỷ lệ phần trăm bằng nhau.

Do đó, khả năng giao dịch gian lận là CASH_OUT hoặc TRANSFER gần như bằng nhau.

d. Tạo dữ liệu mới (new_df) chỉ có giao dịch CASH-OUT và TRANSFER

Vì chỉ có các giao dịch CASH-OUT và TRANSFER mới có thể là gian lận, chúng ta hãy giảm kích thước của tập dữ liệu bằng cách chỉ giữ lại các loại giao dịch này và xóa PAYMENT, CASH-IN và DEBIT.

```
new_df = df.loc[df['type'].isin(['CASH_OUT', 'TRANSFER']),:]
print('Dữ liệu mới có', len(new_df), 'giao dịch.')
```

✓ 0.3s
Dữ liệu mới có 2770409 giao dịch.

Hình 18: [Đoạn mã] Chỉ giữ lại các giao dịch CASH-OUT và TRANSFER

Do đó, tôi đã giảm được lượng dữ liệu từ hơn 6 triệu giao dịch xuống còn khoảng 2,8 triệu giao dịch.

Thông tin và tóm tắt thống kê các biến phân loại của new_df

Hình 19: Thông tin của bộ dữ liệu mới vừa được lọc

```
new_df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 2770409 entries, 2 to 6362619
Data columns (total 11 columns):
 #   Column           Dtype  
 --- 
 0   step             int64  
 1   type             object  
 2   amount            float64
 3   nameOrig          object  
 4   oldbalanceOrg     float64
 5   newbalanceOrig    float64
 6   nameDest          object  
 7   oldbalanceDest    float64
 8   newbalanceDest    float64
 9   isFraud           object  
 10  isFlaggedFraud   int64  
dtypes: float64(5), int64(2), object(4)
memory usage: 253.6+ MB
```

	type	nameOrig	nameDest	isFraud
count	2770409	2770409	2770409	2770409
unique	2	2768630	509565	2
top	CASH_OUT	C724452879	C1286084959	0
freq	2237500	3	75	2762196

Hình 20: Tóm tắt thống kê của các biến phân loại trong bộ dữ liệu mới

4.2.2.4. Kiểm tra tính hợp lệ của dữ liệu

4.2.2.4.1. Số tiền giao dịch âm hoặc bằng không

Đầu tiên, chúng ta kiểm tra xem cột "amount" có luôn dương hay không. Hai đoạn mã sau đây chia nhỏ thành số giao dịch có số tiền âm và số giao dịch có số tiền bằng 0.

```
#Check that there are no negative amounts
print("Số lượng giao dịch có số tiền giao dịch là số âm: " + str(sum(new_df['amount'] < 0)))

#Check instances where transacted amount is 0
print("Số lượng giao dịch có số tiền giao dịch là 0: " + str(sum(new_df['amount'] == 0)))
✓ 0.1s

Số lượng giao dịch có số tiền giao dịch là số âm: 0
Số lượng giao dịch có số tiền giao dịch là 0: 16
```

Hình 21: [Đoạn mã] Số tiền giao dịch âm hoặc bằng không

Chỉ có một số ít trường hợp mà số tiền giao dịch là 0. Ta quan sát bằng cách khám phá dữ liệu của các giao dịch này rằng tất cả chúng đều là giao dịch gian lận. Vì vậy, ta có thể cho rằng nếu số tiền giao dịch là 0, giao dịch là gian lận.

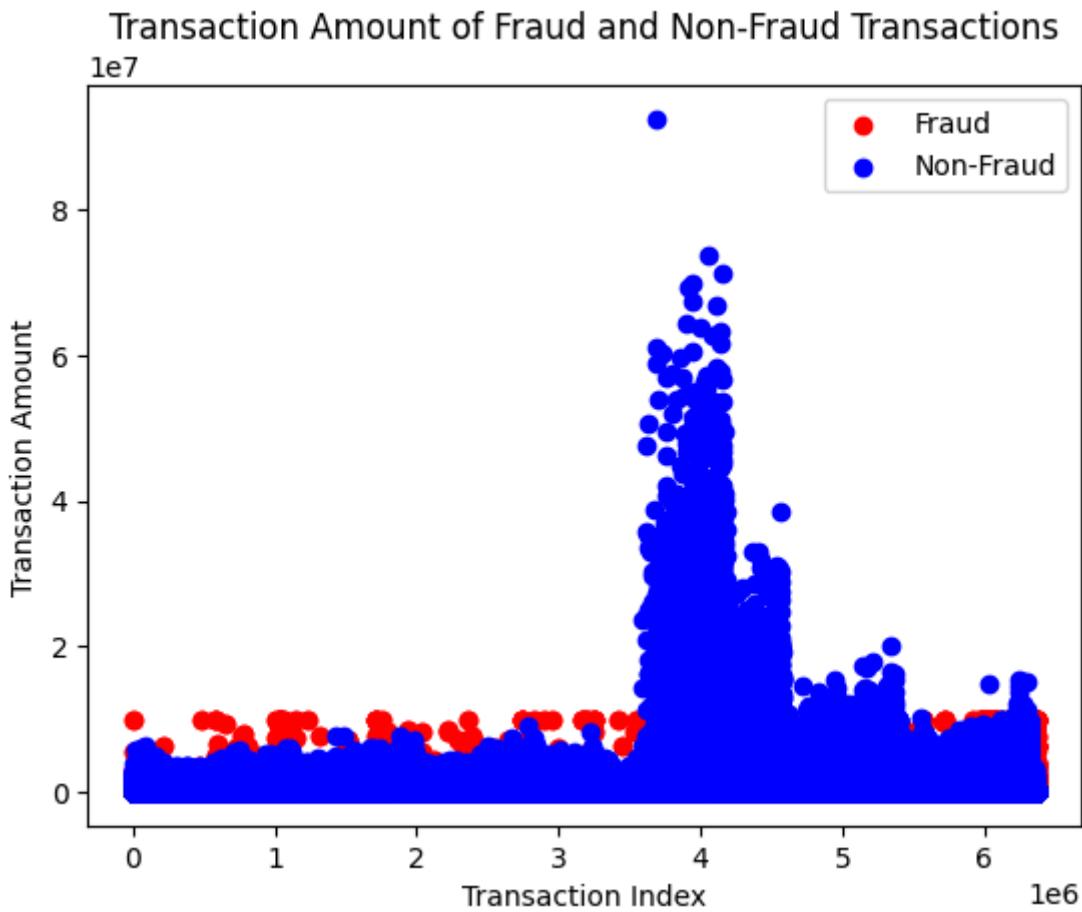
Ta sẽ loại bỏ các giao dịch này khỏi dữ liệu và đưa điều kiện này vào khi đưa ra dự đoán cuối cùng.

```
# Remove 0 amount values
new_df = new_df[new_df['amount'] > 0,:]

✓ 0.1s
```

Hình 22: [Đoạn mã] Xóa các giao dịch có số tiền bằng 0

Số tiền giao dịch của các giao dịch gian lận và không gian lận:



Hình 23: Minh họa số tiền giao dịch của các giao dịch gian lận và không gian lận

Dựa trên biểu đồ phân tán của số tiền giao dịch cho các giao dịch gian lận và không gian lận, dường như không có sự khác biệt rõ ràng giữa hai loại này. Mặc dù có thể có sự khác biệt nhỏ trong phân phối, nhưng nó không đủ đáng kể để đưa ra kết luận.

Tiếp theo, ta sẽ tìm số tiền giao dịch trung bình và trung vị theo loại:

Số tiền giao dịch trung bình theo loại:
 type
 CASH_OUT 176275.224862
 TRANSFER 910647.009645
 Name: amount, dtype: float64

Số tiền giao dịch trung vị theo loại:
 type
 CASH_OUT 147073.795
 TRANSFER 486308.390
 Name: amount, dtype: float64

Hình 24: số tiền giao dịch trung bình và trung vị theo loại

Từ số tiền giao dịch trung bình và trung vị theo loại, chúng ta có thể quan sát thấy rằng số tiền giao dịch trung bình cho các giao dịch chuyển khoản cao hơn nhiều so với các giao dịch rút tiền. Điều này có thể chỉ ra rằng các hoạt động gian lận cũng có thể liên quan đến các khoản giao dịch lớn hơn. Ngoài ra, chúng ta có thể quan sát thấy rằng số tiền giao dịch trung vị cho các giao dịch rút tiền thấp hơn so với các giao dịch chuyển khoản.

4.2.2.4.2. Số dư của người khởi tạo và số dư của người nhận

Trong phần này, ta sẽ kiểm tra xem có bất kỳ sự mơ hồ nào trong số dư của người khởi tạo hoặc số dư của người nhận không. Đầu ra sau đây xác định các trường hợp mà số dư ban đầu của người khởi tạo hoặc số dư cuối cùng của người nhận là 0.

```
new_df_count = len(new_df)
orig_initial_balance = len(new_df[new_df.oldbalanceOrg == 0])
per_orig_initial_balance = str(round((orig_initial_balance/new_df_count)*100, 2))
print(f"Tỷ lệ phần trăm các giao dịch mà số dư ban đầu của bên người khởi tạo là 0: {per_orig_initial_balance}%")

dest_final_balance = len(new_df[new_df.newbalanceDest == 0])
per_dest_final_balance = str(round(dest_final_balance/new_df_count*100, 2))
print(f"Tỷ lệ phần trăm các giao dịch mà số dư cuối cùng của bên người nhận là 0: {per_dest_final_balance}%")
```

Tỷ lệ phần trăm các giao dịch mà số dư ban đầu của bên người khởi tạo là 0: 47.23%
Tỷ lệ phần trăm các giao dịch mà số dư cuối cùng của bên người nhận là 0: 0.6%

Hình 25: [Đoạn mã] Kiểm tra số dư bằng không

Do đó, trong gần một nửa số giao dịch, số dư ban đầu của người khởi tạo được ghi là 0. Tuy nhiên, trong ít hơn 1% trường hợp, số dư cuối cùng của người nhận được ghi là 0.

Lý tưởng nhất là số dư cuối cùng của người nhận phải bằng số dư ban đầu của người nhận cộng với số tiền giao dịch. Tương tự như vậy, số dư cuối cùng của người khởi tạo phải bằng số dư ban đầu của người khởi tạo trừ đi số tiền giao dịch.

Sau đó, kiểm tra các điều kiện này để xem các biến số dư cũ và số dư mới có được ghi lại chính xác cho cả người khởi tạo và người nhận hay không.

```

new_df['dest_final_balance'] = new_df['oldbalanceDest'] + new_df['amount']

new_df['orig_final_balance'] = new_df['oldbalanceOrg'] - new_df['amount']

✓ 0.0s

```

Hình 26: [Đoạn mã] Xác định tính năng cân bằng

```

c1 = len(new_df[new_df.newbalanceDest != new_df.dest_final_balance])
print("Giao dịch mà số dư đích không được ghi lại chính xác: " + str(round(c1/new_df_count*100, 2)) + "%")

c2 = len(new_df[new_df.oldbalanceOrig != new_df.orig_final_balance])
print("Các giao dịch mà số dư của người khởi tạo không được ghi lại chính xác: " + str(round(c2/new_df_count*100, 2)) + "%")

✓ 0.4s

```

Giao dịch mà số dư đích không được ghi lại chính xác: 42.09%
 Các giao dịch mà số dư của người khởi tạo không được ghi lại chính xác: 93.72%

Hình 27:[Đoạn mã] Kiểm tra số dư không chính xác

Do đó, trong hầu hết các giao dịch, số dư cuối cùng của người khởi tạo không được ghi lại chính xác và trong gần một nửa số trường hợp, số dư cuối cùng của người nhận không được ghi lại chính xác.

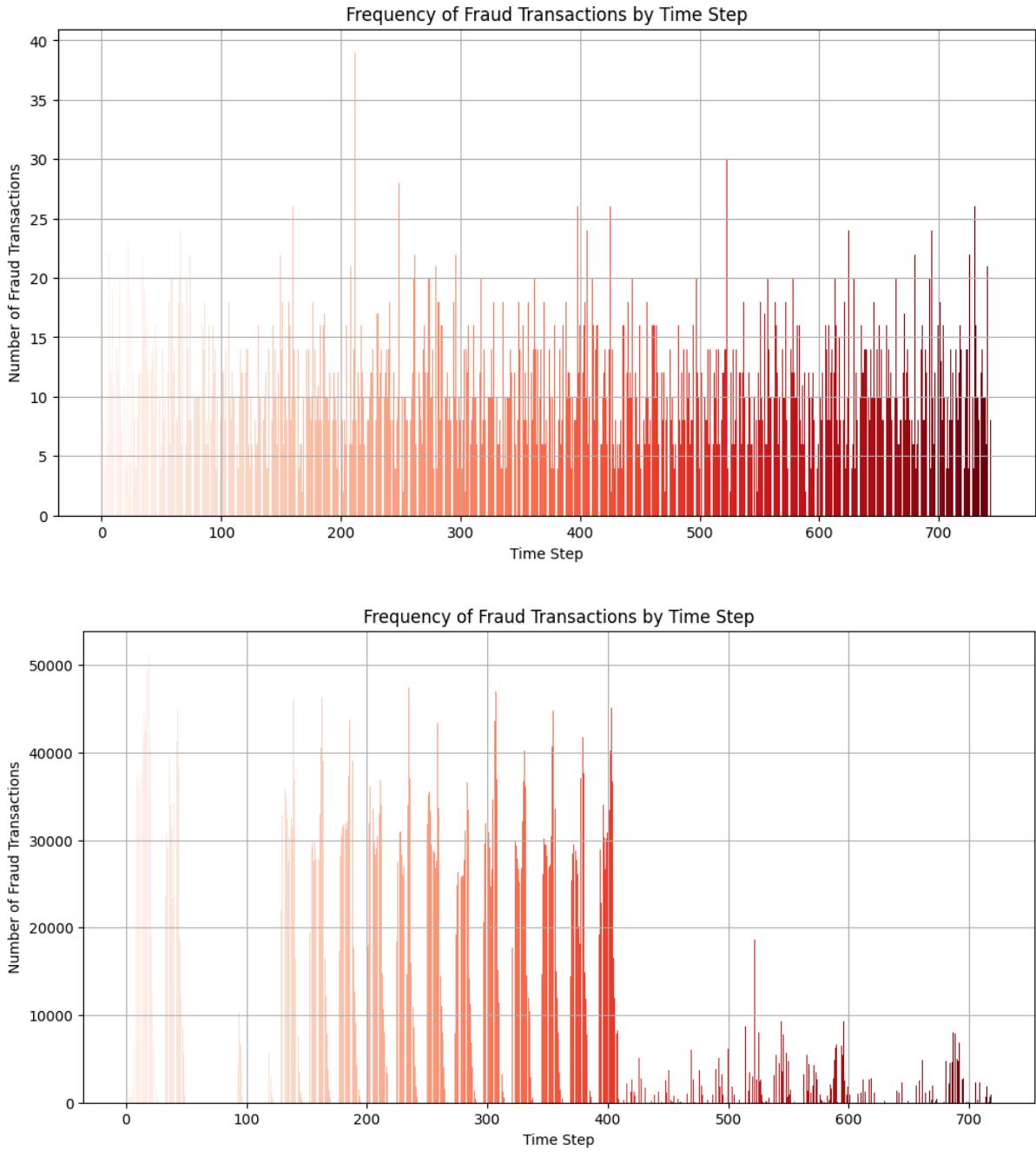
Có thể sẽ rất thú vị khi xem liệu bất kỳ sự khác biệt nào được xác định ở trên có khác nhau giữa các giao dịch gian lận và các giao dịch không gian lận hay không. Điều này sẽ được thực hiện trong các phần tiếp theo.

4.2.2.4.3. Phân tích giao dịch gian lận

Trong phần này, một phân tích thăm dò bổ sung được thực hiện để xác định xem có biến nào có thể dự đoán được gian lận hay không.

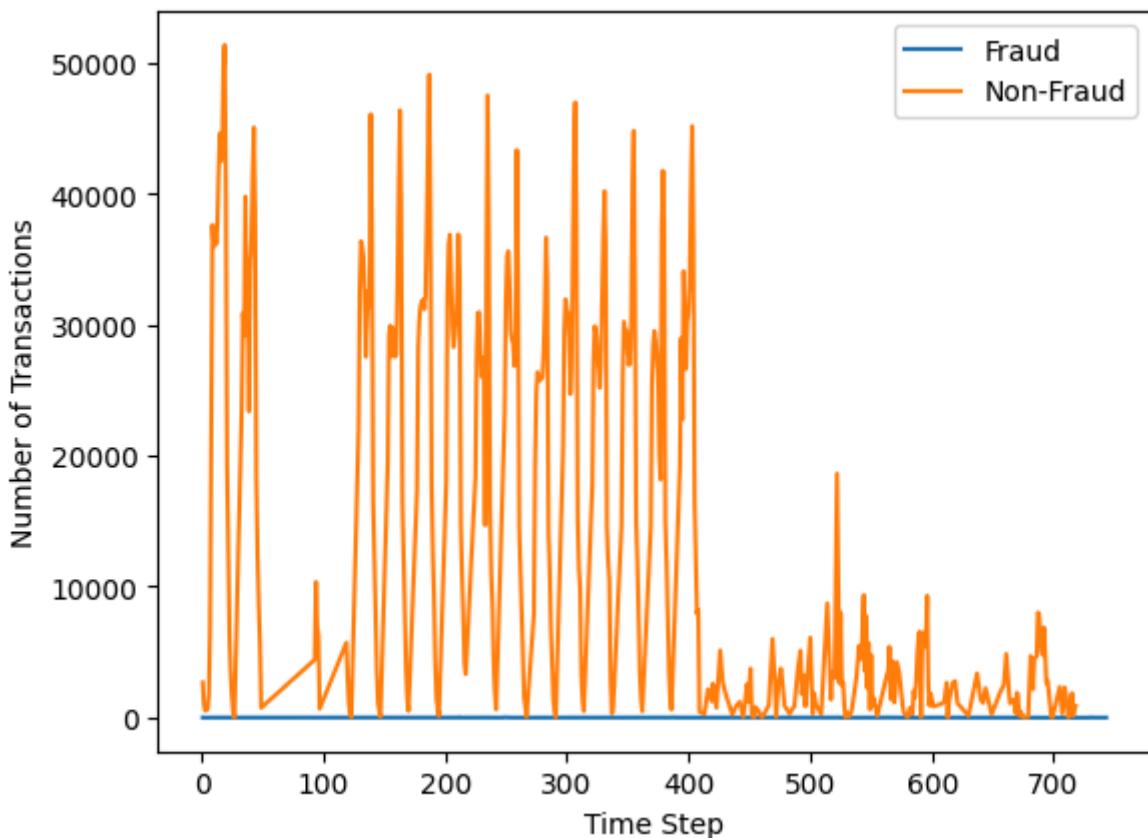
Bước thời gian (step):

Tôi bắt đầu bằng cách phân tích biến bước thời gian. Số lượng giao dịch trong mỗi bước thời gian theo trạng thái gian lận được đo lường để xác định xem có bất kỳ bước thời gian cụ thể nào mà giao dịch gian lận phổ biến hơn những bước khác hay không. Từ mô tả dữ liệu, tôi biết rằng mỗi bước thời gian là một giờ.



Hình 28: Giao dịch gian lận và không gian lận được tính theo bước thời gian

Từ Hình 28 cho thấy các giao dịch gian lận gần như được phân bổ đồng đều qua các bước thời gian, trong khi các giao dịch không gian lận tập trung nhiều hơn vào các bước thời gian cụ thể. Đây có thể là một điểm khác biệt giữa hai loại và có thể giúp đào tạo các mô hình phân loại.

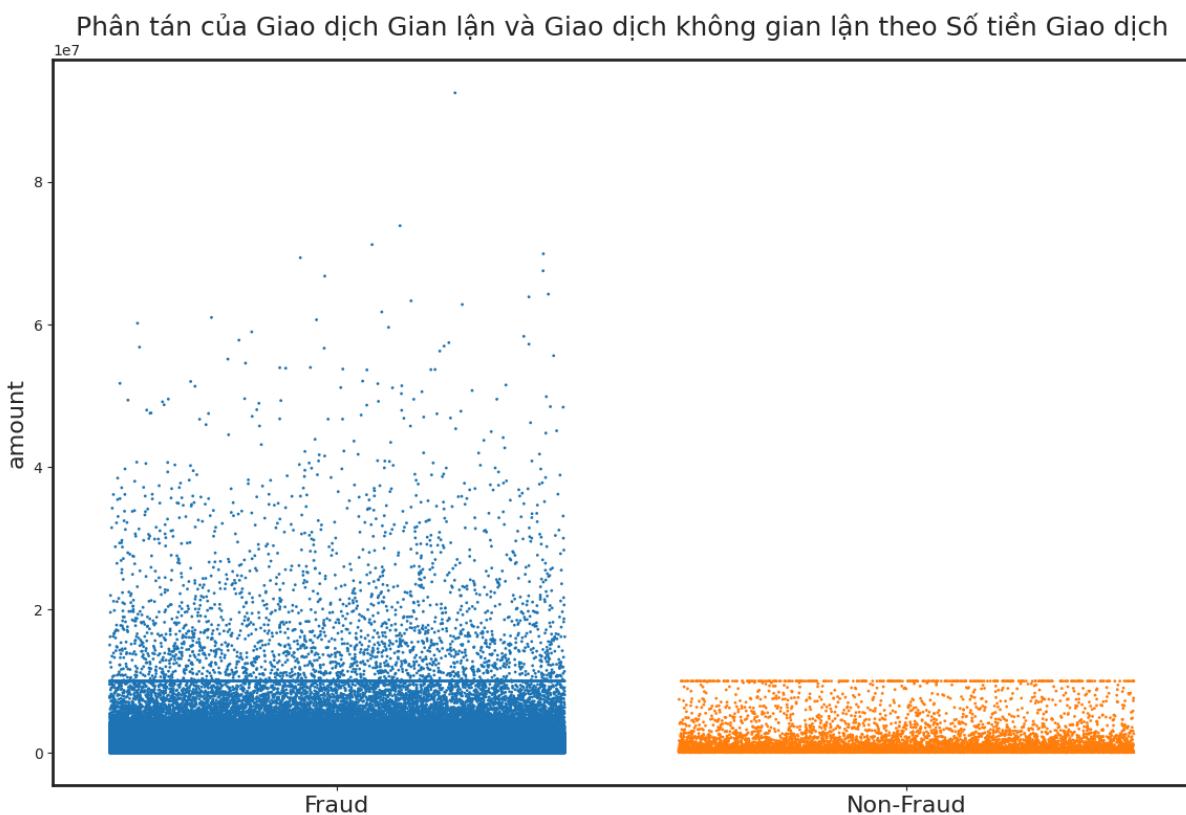


Hình 29: Minh họa bước thời gian theo số giao dịch (gian lận và không gian lận)

Phân phối tần suất của các giao dịch gian lận đều hơn qua các khoảng thời gian, trong khi các giao dịch không gian lận có khả năng xảy ra cao hơn vào những thời điểm cụ thể.

Số tiền giao dịch (Amount):

Tôi sẽ kiểm tra xem có sự khác biệt nào giữa các giao dịch gian lận và không gian lận về mặt số tiền giao dịch hay không.



Hình 30: Số tiền giao dịch của các giao dịch gian lận và không gian lận

Phân phối số tiền giao dịch cho thấy số tiền có thể hơi cao hơn đối với các giao dịch không gian lận, nhưng không thể kết luận được sự khác biệt giữa gian lận và không gian lận về mặt số tiền giao dịch.

Số dư (Balances):

Trong phần trước về Kiểm tra sự hợp lệ, tôi nhận thấy có sự không chính xác trong cách ghi lại biến số dư cho cả người khởi tạo và người nhận. Tôi cũng quan sát thấy rằng trong gần một nửa số trường hợp, số dư ban đầu của người khởi tạo được ghi là 0.

Trong mã bên dưới, tôi so sánh tỷ lệ phần trăm các trường hợp mà số dư ban đầu của người khởi tạo là 0.

```

fraud_trans = len(new_df[new_df.isFraud == 1])
c3 = len(new_df[(new_df.oldbalanceOrg == 0) & (new_df.isFraud == 1)])
print("% giao dịch gian lận có số dư ban đầu của người khởi tạo là 0: " + str(round(c3/fraud_trans*100, 2)) + "%")

gen_trans = len(new_df[new_df.isFraud == 0])
c4 = len(new_df[(new_df.oldbalanceOrg == 0) & (new_df.isFraud == 0)])
print("% giao dịch hợp lệ có số dư ban đầu của người khởi tạo là 0: " + str(round(c4/gen_trans*100, 2)) + "%")
✓ 0.2s

% giao dịch gian lận có số dư ban đầu của người khởi tạo là 0: 0.3%
% giao dịch hợp lệ có số dư ban đầu của người khởi tạo là 0: 47.37%

```

Hình 31: [Đoạn mã] So sánh các giao dịch gian lận và không gian lận khi số dư ban đầu của người khởi tạo là 0

Trong các giao dịch gian lận, số dư ban đầu của người khởi tạo chỉ là 0 0,3% thời gian so với 47% trong trường hợp giao dịch không gian lận. Đây có thể là một điểm khác biệt tiềm ẩn giữa hai loại.

Hãy kiểm tra độ không chính xác trong biến số dư và so sánh giữa gian lận và không gian lận. Độ không chính xác được định nghĩa là sự khác biệt giữa số dư phải tính đến số tiền giao dịch và số tiền được ghi nhận là số dư.

Tôi tính toán độ không chính xác của số dư cho cả bên khởi tạo và bên nhận như sau:

```
new_df['origBalance_inacc'] = (new_df['oldbalanceOrg'] - new_df['amount']) - new_df['newbalanceOrig']
new_df['destBalance_inacc'] = (new_df['oldbalanceDest'] + new_df['amount']) - new_df['newbalanceDest']
✓ 0.0s
```

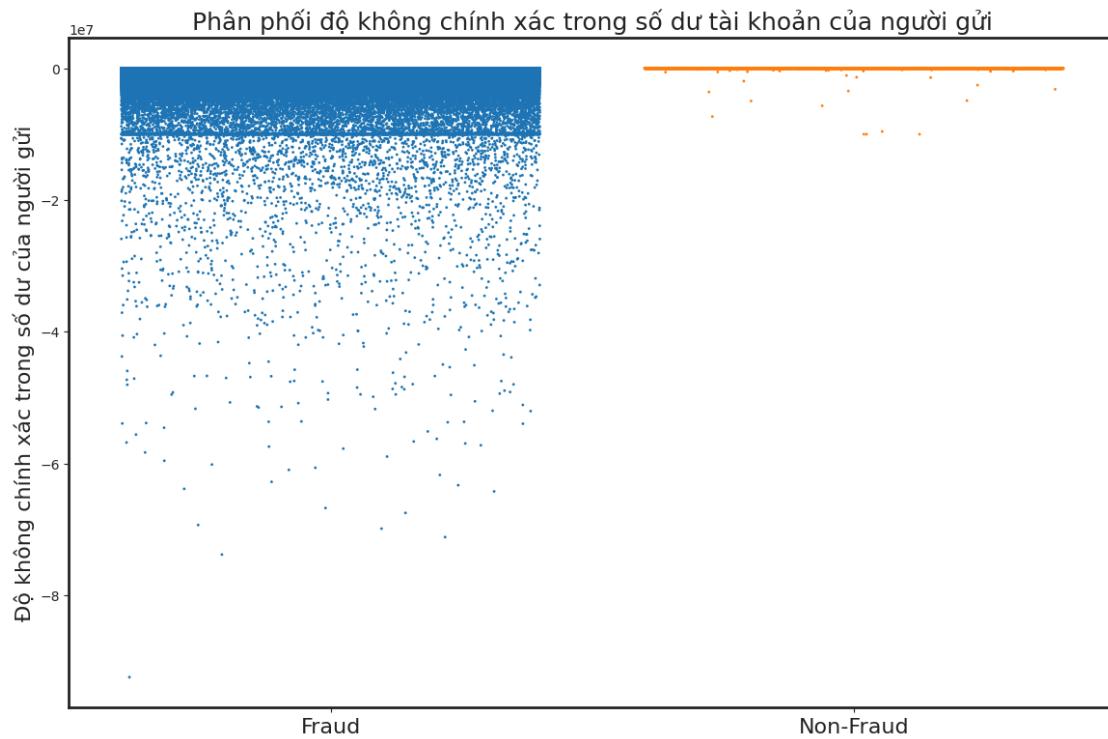
Hình 32: [Đoạn mã] Xác định tính năng cân bằng không chính xác

origBalance_inacc	destBalance_inacc
0.00	181.0
0.00	21363.0
-213808.94	182703.5
-214605.30	237735.3
-300850.89	-2401220.0

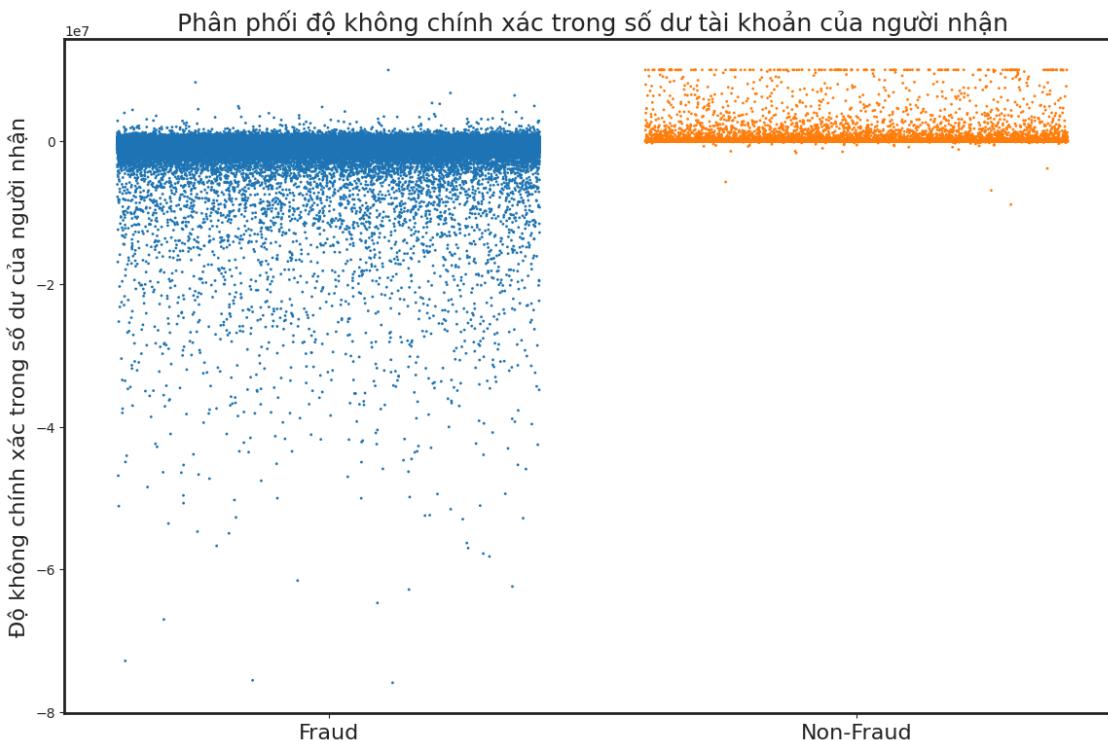
Hình 33: Cột origBalance_inacc và destBalance_inacc vừa được khởi tạo

Trong các hình sau, tôi mô tả sự phân bổ tính năng không chính xác về số dư của số dư người gửi và người nhận đối với các giao dịch gian lận và không gian lận như sau:

Hình 34: Minh họa số dư của người khởi tạo không chính xác về các giao dịch gian lận và không gian lận



Hình 35: Minh họa sự không chính xác trong số dư tài khoản người nhận của các giao dịch gian lận và không gian lận



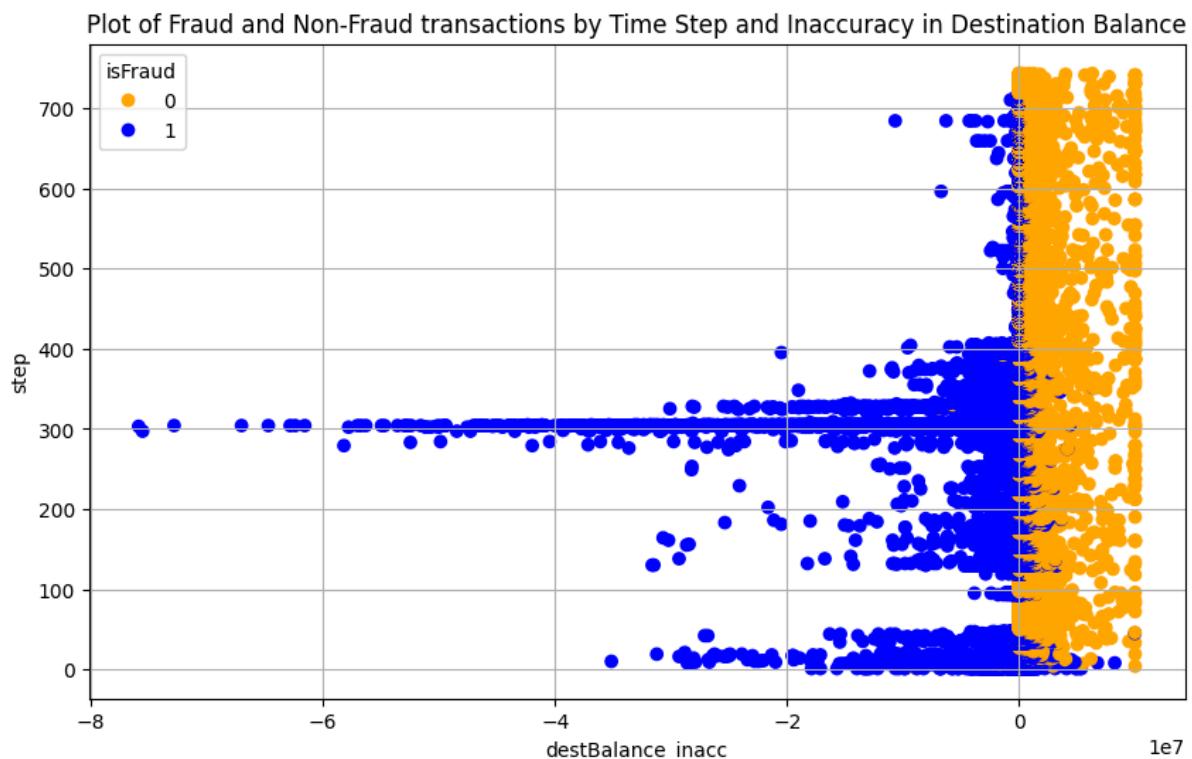
Có sự khác biệt giữa gian lận và không gian lận trong các biện pháp không chính xác mà tôi đã phân tích ở trên. Đặc biệt, có vẻ như sự không chính xác trong số

dư đích hầu như luôn là tiêu cực đối với các giao dịch không gian lận, trong khi nó hầu như luôn là tích cực đối với các giao dịch gian lận. Điều này cũng có thể là những yếu tố dự báo gian lận tiềm tàng.

Nhìn chung, tôi đã xác định một số chiêu mà qua đó có thể phân biệt được các giao dịch gian lận với các giao dịch không gian lận. Các chiêu này như sau:

- **Bước thời gian (time step)** - các giao dịch gian lận có khả năng xảy ra nhau ở mọi bước thời gian, nhưng các giao dịch thực sự đạt đỉnh ở các bước thời gian cụ thể
- **Số dư (Balances)** - số dư ban đầu của bên khởi tạo có nhiều khả năng là 0 trong trường hợp giao dịch thực sự hơn là giao dịch gian lận
- **Số dư không chính xác (inaccuracies in balance)** - số dư đích không chính xác có khả năng là số âm trong trường hợp giao dịch thực sự nhưng là số dương trong trường hợp giao dịch gian lận.

Biểu đồ phân tán dưới đây cho thấy sự phân biệt rõ ràng giữa các giao dịch gian lận và không gian lận dọc theo các bước thời gian và độ không chính xác của số dư người nhận.



Hình 36: Phân biệt giữa giao dịch gian lận và không gian lận

4.2.3. Mô hình dự đoán để phát hiện gian lận

Trong các phần trước, tôi đã xác định các khía cạnh giúp phát hiện các giao dịch gian lận. Dựa trên những kết quả này, tôi xây dựng các mô hình phân loại có giám sát.

4.2.3.1. Tạo tập dữ liệu cho mô hình hóa

Trong phần này, tôi chọn các biến cần thiết cho mô hình ML, mã hóa các biến phân loại thành số và chuẩn hóa dữ liệu.

Hãy xem lại các cột trong tập dữ liệu:

```
new_df.columns  
✓ 0.0s  
  
Index(['step', 'type', 'amount', 'nameOrig', 'oldbalanceOrg', 'newbalanceOrig',  
       'nameDest', 'oldbalanceDest', 'newbalanceDest', 'isFraud',  
       'isFlaggedFraud', 'origBalance_inacc', 'destBalance_inacc',  
       'dest_final_balance', 'orig_final_balance'],  
      dtype='object')
```

Hình 37: các cột trong tập dữ liệu

Tên (hoặc ID) của người khởi tạo và người nhận không cần thiết cho việc phân loại. Vì vậy, tôi loại bỏ chúng cùng với một số biến không cần thiết cho mô hình.

```
new_df = new_df.drop(['nameOrig', 'nameDest', 'dest_final_balance', 'orig_final_balance',  
                     'isFlaggedFraud', 'origBalance_inacc', 'destBalance_inacc'], axis=1)  
✓ 0.0s
```

Hình 38: [Đoạn mã] Loại bỏ những cột không cần thiết

4.2.3.1.1. Tạo các biến giả định

Tôi có một biến phân loại trong tập dữ liệu – loại giao dịch. Biến này cần được mã hóa thành các biến nhị phân, và các biến giả cần được tạo ra.

Đoạn mã dưới đây được sử dụng để thực hiện việc này:

```
# Tạo biến giả thông qua mã hóa one hot cho cột 'type'  
new_df = pd.get_dummies(new_df, columns=['type'], prefix=['type'])  
✓ 0.1s
```

Hình 39: Đoạn mã] Mã hóa biến phân loại 'type'

```
new_df.columns
✓ 0.0s

Index(['step', 'amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest',
       'newbalanceDest', 'isFraud', 'type_CASH_OUT', 'type_TRANSFER'],
      dtype='object')
```

Hình 40: Danh sách những cột mới

- Điều này tạo ra hai biến giả nhị phân – type_CASH_OUT và type_TRANSFER.

4.2.3.1.2. Chia dữ liệu

Tôi chia tập dữ liệu đã chuẩn hóa thành các tập dữ liệu huấn luyện và kiểm tra. Tôi quyết định sử dụng 70% dữ liệu gốc cho việc huấn luyện và 30% còn lại cho kiểm tra.

Đoạn mã dưới đây được sử dụng để tạo các tập dữ liệu huấn luyện và kiểm tra:

```
# Biến độc lập (ước lượng)
X = new_df.drop("isFraud", axis = 1)

# Biến phụ thuộc (nhãn)
y = new_df["isFraud"]

✓ 0.0s

# Split your data to train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, shuffle=True, stratify=y, random_state = 42)
✓ 0.4s
```

Hình 41: [Đoạn mã] Tạo tập dữ liệu huấn luyện và thử nghiệm

4.2.3.1.2. Chuẩn hóa dữ liệu

Trong quá trình biến đổi này, tôi chuyển đổi tất cả các cột trong dữ liệu để có cùng một khoảng giá trị. Việc này được thực hiện thông qua tính năng chuẩn hóa sẵn có trong Python. Đoạn mã dưới đây được sử dụng để thực hiện quá trình biến đổi này:

```

# Scaling data
scaler = StandardScaler()

X_train_scale = scaler.fit_transform(X_train)
X_test_scale = scaler.transform(X_test)

X_train = pd.DataFrame(X_train_scale, columns = X_train.columns)
X_test = pd.DataFrame(X_test_scale, columns = X_test.columns)

✓ 0.5s

```

Hình 43: [Đoạn mã] Chuẩn hóa dữ liệu

Phương pháp được sử dụng để chuẩn hóa dữ liệu là "Standard Scaling" và nó được sử dụng để chuẩn hóa dữ liệu bằng cách trừ đi giá trị trung bình và chia cho độ lệch chuẩn, sao cho dữ liệu đã biến đổi có giá trị trung bình là 0 và độ lệch chuẩn là 1. Điều này rất quan trọng khi làm việc với các thuật toán giả định dữ liệu phân phối chuẩn hoặc các thuật toán nhạy cảm với quy mô của các đặc trưng đầu vào. Chuẩn hóa dữ liệu cũng có thể giúp ngăn chặn việc một đặc trưng nào đó chiếm ưu thế trong mô hình và có thể cải thiện hiệu suất tổng thể của nhiều thuật toán máy học.

4.2.3.1.3. Giải quyết sự mất cân bằng dữ liệu

Trong quá trình xử lý dữ liệu mất cân bằng lớp, có nhiều phương pháp khác nhau có thể được áp dụng để cân bằng lại số lượng mẫu giữa các lớp. Một trong những phương pháp hiệu quả và được sử dụng rộng rãi là SMOTE (Synthetic Minority Over-sampling Technique). SMOTE không chỉ đơn giản là lặp lại các quan sát của lớp thiểu số mà còn tạo ra các mẫu tổng hợp mới, giúp tăng sự đa dạng và thông tin trong dữ liệu.

SMOTE là một kỹ thuật oversampling, trong đó các điểm dữ liệu mới của lớp thiểu số được tạo ra bằng cách tổng hợp các điểm dữ liệu hiện có. Thay vì lặp lại các mẫu thiểu số nhiều lần, SMOTE chọn các điểm dữ liệu ngẫu nhiên trong lớp thiểu số và tạo ra các điểm mới dọc theo đường thẳng nối các điểm này với các hàng xóm gần nhất của chúng. Bằng cách này, các điểm dữ liệu tổng hợp mới có đặc điểm tương tự nhưng không trùng lặp hoàn toàn với các điểm dữ liệu ban đầu.

SMOTE tạo ra các mẫu tổng hợp mới cho lớp gian lận, giúp cân bằng số lượng mẫu giữa các lớp. Đoạn mã sau đây được sử dụng để thực hiện điều này:

```

# Khởi tạo kỹ thuật smote
smote = SMOTE()

y_train = y_train.astype(int)

# Chuyển đổi y_train thành một mảng NumPy
y_train_array = np.array(y_train)

# Khởi tạo thuật toán SMOTE
smote = SMOTE(random_state=42)

# Lấy mẫu lại dữ liệu đào tạo bằng SMOTE
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train_array)

```

✓ 0.0s

Hình 43: [Đoạn mã] Cân bằng dữ liệu bằng SMOTE

Kết quả trước và sau khi áp dụng smote:

```

# In các giá trị trước và sau SMOTE
print("Trước SMOTE:", Counter(y_train))
print("Sau SMOTE:", Counter(y_train_smote))

✓ 0.5s

```

Trước SMOTE: Counter({0: 1933537, 1: 5738})
 Sau SMOTE: Counter({0: 1933537, 1: 1933537})

Hình 44: Kết quả của bộ dữ liệu trước và sau khi áp dụng SMOTE

- Sau khi áp dụng SMOTE, số lượng giao dịch gian lận đã được tăng cường một cách đáng kể để cân bằng với số lượng giao dịch không gian lận. Cụ thể, số lượng mẫu của lớp gian lận đã tăng từ 5,738 lên 1,933,537, tương đương với số lượng mẫu của lớp không gian lận.
- Điều này giúp tập dữ liệu trở nên cân bằng hơn, từ đó cải thiện khả năng học tập và dự đoán của mô hình. Các mô hình học máy khi được huấn luyện trên tập dữ liệu cân bằng sẽ có cơ hội học hỏi và nhận diện các đặc trưng của cả hai lớp một cách công bằng hơn, dẫn đến hiệu suất tổng thể tốt hơn.

Tạo một Dataframe mới với bộ dữ liệu cân bằng:

```

# Tạo một DataFrame mới với dữ liệu cân bằng
balanced_df = pd.DataFrame(X_train_smote, columns=columns)

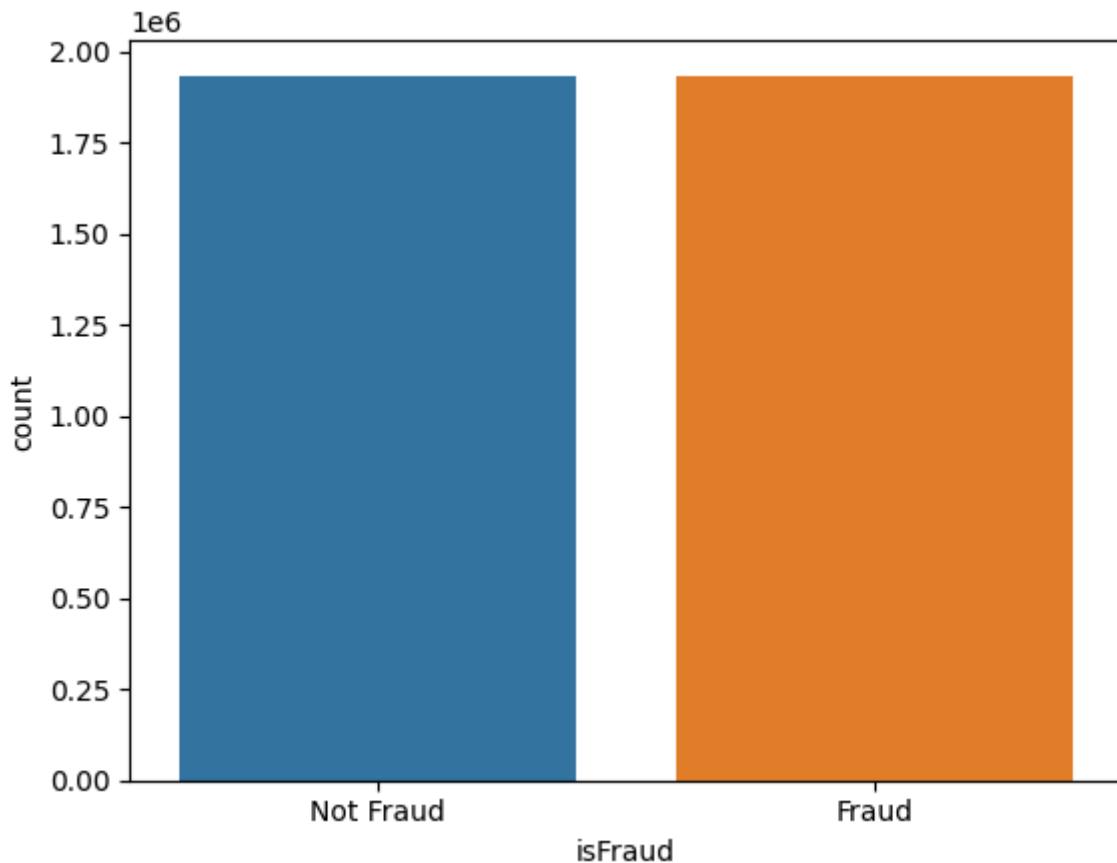
# Lưu giá trị mặc định vào y_train_smote
balanced_df['isFraud'] = y_train_smote

✓ 0.1s

```

Hình 45: [Đoạn mã] Tạo dataframe mới với bộ dữ liệu cân bằng

Hiển thị phân phối lớp sau khi SMOTE:



Hình 46: Minh họa các lớp trong bộ dữ liệu sau khi SMOTE

- Biểu đồ này minh họa rõ ràng rằng cả hai lớp giờ đây đều có số lượng mẫu ngang nhau. Việc này rất quan trọng trong các bài toán phân loại, đặc biệt là trong phát hiện gian lận, vì nó giúp các thuật toán học máy không bị thiên lệch về một lớp nào đó.

4.2.3.1.4. Chia dữ liệu trên tập dữ liệu cân bằng

Sau khi áp dụng kỹ thuật SMOTE, chúng tôi chuẩn bị dữ liệu để huấn luyện và kiểm thử mô hình bằng cách chia dữ liệu thành các biến độc lập và biến phụ thuộc, sau đó chia thành các tập huấn luyện và kiểm thử.

```
# independent variable (estimator)
X = balanced_df.drop("isFraud", axis = 1)

# dependent variable (label)
y = balanced_df["isFraud"]

# Split your data to train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, shuffle=True, stratify=y, random_state = 42)
✓ 0.0s
```

Hình 47: [Đoạn mã] Tạo tập dữ liệu huấn luyện và thử nghiệm trên bộ dữ liệu cân bằng

4.2.3.2. Các mô hình phân loại để phát hiện gian lận

Tôi định nghĩa hai mô hình để thực hiện phân loại: Hồi quy Logistic, Random Forest và XG Boost.

Trong các vấn đề phát hiện gian lận, việc xác định chính xác các giao dịch gian lận là quan trọng hơn việc phân loại sai các giao dịch hợp pháp thành gian lận. Ngoài ra, chúng ta cũng có thể sử dụng diện tích dưới đường cong (AUC) của đường cong ROC. Tuy nhiên, điều này sẽ không đủ để nắm bắt chính xác nếu mô hình đang xác định hầu hết các giao dịch gian lận. Do đó, tôi sử dụng điều này như một sự xác nhận của hiệu suất mô hình.

4.2.3.2.1 Logistic Regression Model

- Huấn luyện Mô hình Hồi quy Logistic

Trong giai đoạn huấn luyện mô hình, ta sử dụng lớp LogisticRegression của thư viện scikit-learn để xây dựng một mô hình hồi quy logistic. Cụ thể, đoạn mã dưới đây khởi tạo và huấn luyện mô hình với dữ liệu huấn luyện (X_{train} và y_{train}):

```

lr = LogisticRegression()

lr.fit(X_train, y_train)
✓ 3.1s

```

LogisticRegression

Hình 48: Khởi tạo mô hình Logistic Regression

- Dự đoán và Đánh giá Mô hình

Sau khi huấn luyện, mô hình được sử dụng để dự đoán trên tập dữ liệu kiểm tra (X_{test}). Các dự đoán này sau đó được so sánh với các nhãn thực sự (y_{test}) để đánh giá hiệu suất của mô hình.

----- Logistic Regression Model -----

Classification Report:				
	precision	recall	f1-score	support
0	0.95	0.94	0.95	580062
1	0.94	0.95	0.95	580061
accuracy			0.95	1160123
macro avg	0.95	0.95	0.95	1160123
weighted avg	0.95	0.95	0.95	1160123

Accuracy of logistic regression classifier on test set: 0.9486
AUC Score: 0.9891170430270803

Hình 49: Logistic Regression - Classification report

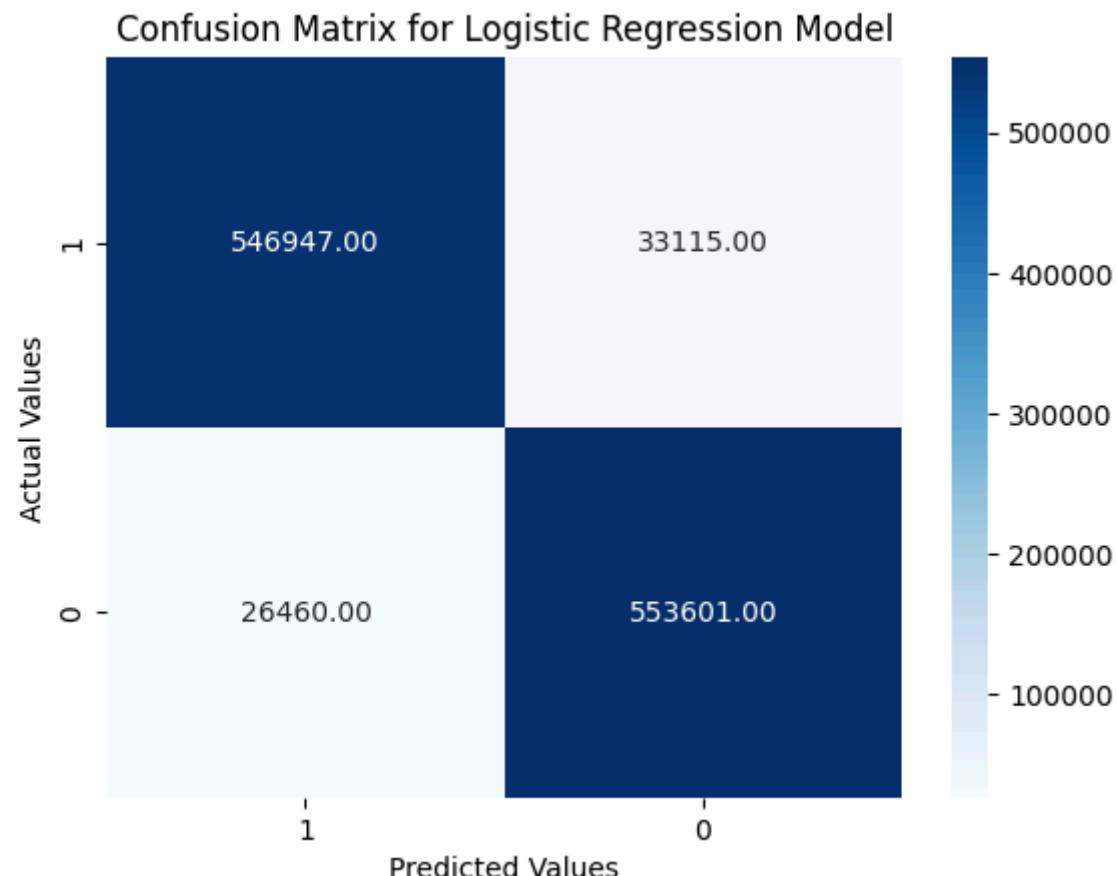
Kết quả đánh giá bao gồm:

- Classification Report: Báo cáo phân loại cung cấp các chỉ số precision, recall, f1-score và support cho từng lớp. Trong kết quả, cả hai lớp 0 và 1 đều có các chỉ số rất cao, với precision và recall xấp xỉ 0.95.
- Accuracy: Độ chính xác của mô hình trên tập kiểm tra là 0.9486, tức là mô hình dự đoán đúng khoảng 94.86% các trường hợp.
- AUC Score: AUC (Area Under Curve) là 0.9891, cho thấy mô hình có khả năng phân biệt giữa hai lớp rất tốt.

- Ma trận Nhầm lẫn

Ma trận nhầm lẫn trong hình minh họa kết quả phân loại của mô hình Hồi Quy Logistic (Logistic Regression) trên một tập dữ liệu lớn. Nó hiển thị các giá trị dự đoán so với các giá trị thực tế cho bài toán phát hiện gian lận, trong đó:

- **Lớp 1** đại diện cho giao dịch gian lận.
- **Lớp 0** đại diện cho giao dịch hợp pháp.



Hình 50: Logistic Regression - Confusion Matrix

Các thành phần chính của ma trận này bao gồm:

- **True Positives (TP)** = **546,947**: Đây là số giao dịch gian lận mà mô hình dự đoán chính xác là gian lận. Mô hình đã hoạt động tốt trong việc phát hiện được phần lớn các giao dịch gian lận.
- **False Negatives (FN)** = **33,115**: Số giao dịch gian lận bị dự đoán nhầm là hợp pháp. Đây là những trường hợp quan trọng mà mô hình không thể phát hiện ra, vì bỏ sót các giao dịch gian lận có thể gây thiệt hại lớn.
- **False Positives (FP)** = **26,460**: Đây là số giao dịch hợp pháp bị mô hình dự đoán sai thành gian lận. Số lượng này không quá lớn, nhưng trong

thực tế, số cảnh báo giả cao có thể gây phiền toái cho khách hàng và dẫn đến chi phí kiểm tra giao dịch không cần thiết.

- **True Negatives (TN) = 553,601:** Đây là số giao dịch hợp pháp được dự đoán đúng là hợp pháp, mô hình đã phân loại chính xác rất nhiều giao dịch hợp pháp, đảm bảo sự chính xác tổng thể.

4.2.3.2.2. Random Forest Model

- Khởi tạo và huấn luyện mô hình

Khởi tạo một đối tượng của lớp RandomForestClassifier và tiến hành huấn luyện mô hình với tập dữ liệu huấn luyện (X_train và y_train):

```
#creating Instance of Random Forest
rf_clf= RandomForestClassifier()

rf_clf.fit(X_train, y_train)
✓ 6m 35.0s
```

▼ RandomForestClassifier ⓘ ?
RandomForestClassifier()

Hình 51: Khởi tạo mô hình Random Forest

- Dự đoán và đánh giá hiệu suất mô hình

Sau khi huấn luyện, mô hình được sử dụng để dự đoán trên tập dữ liệu kiểm tra (X_test). Các dự đoán này sau đó được so sánh với các nhãn thực sự (y_test) để đánh giá hiệu suất của mô hình.

----- Random Forest Model -----

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	580062
1	1.00	1.00	1.00	580061
accuracy			1.00	1160123
macro avg	1.00	1.00	1.00	1160123
weighted avg	1.00	1.00	1.00	1160123

Accuracy of Random Forest classifier on test set: 0.9990
AUC Score: 0.9999902636344505

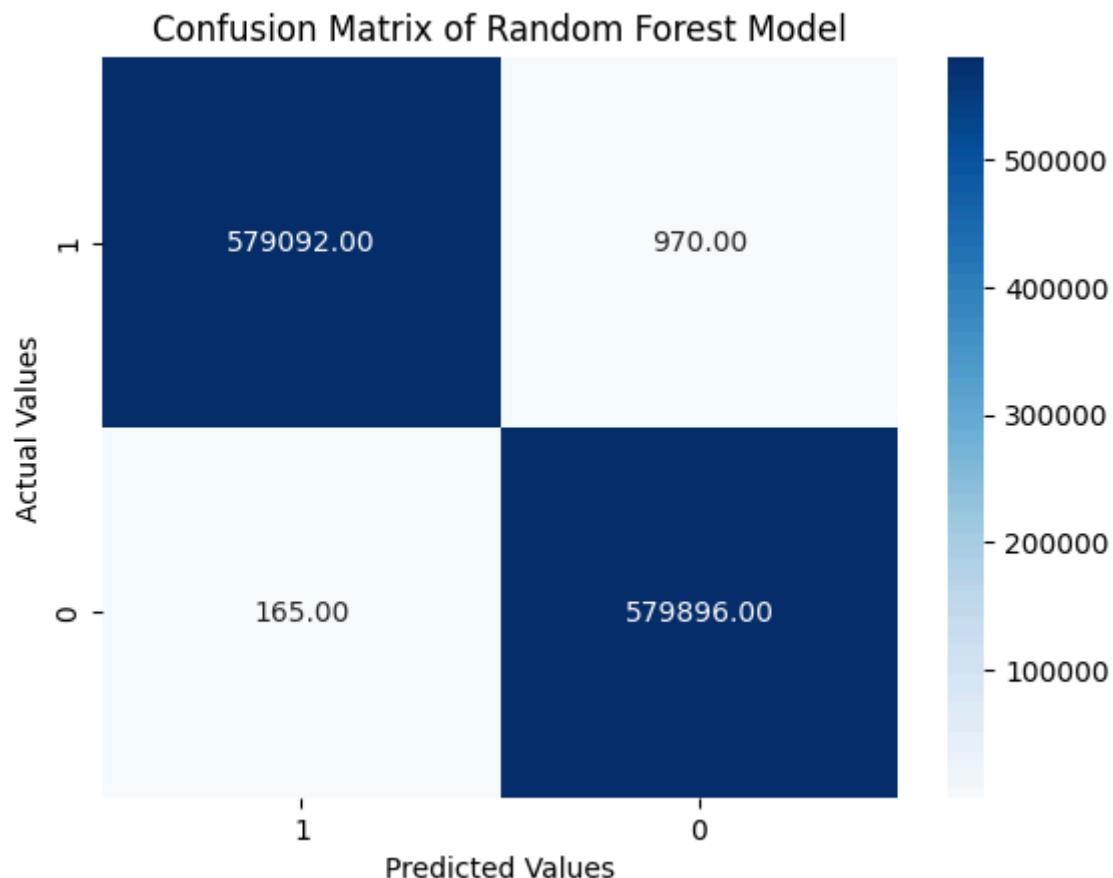
Hình 52: Random Forest - Classification Report

- Kết quả đánh giá:

- Trong bối cảnh phát hiện gian lận, các chỉ số precision, recall và f1-score là rất quan trọng vì chúng phản ánh khả năng phát hiện đúng các giao dịch gian lận mà không gây ra quá nhiều cảnh báo giả.
 - Precision:** Với giá trị **1.00** cho cả hai lớp (giao dịch hợp pháp và gian lận), điều này có nghĩa là mô hình đã phân loại chính xác tất cả các giao dịch được dự đoán là gian lận, không có trường hợp cảnh báo giả nào (dự đoán nhầm một giao dịch hợp pháp là gian lận). Trong bài toán phát hiện gian lận, điều này đặc biệt quan trọng vì cảnh báo giả có thể gây phiền toái cho người dùng và tăng chi phí xử lý.
 - Recall:** Chỉ số **1.00** cho recall ở cả lớp gian lận và hợp pháp cho thấy mô hình không bỏ sót bất kỳ giao dịch gian lận nào. Trong bối cảnh phát hiện gian lận, việc phát hiện đầy đủ các trường hợp gian lận là rất quan trọng vì bỏ sót một giao dịch gian lận có thể gây thiệt hại lớn về tài chính.
 - F1-Score:** F1-score là chỉ số kết hợp giữa precision và recall, đặc biệt hữu ích trong những bài toán mất cân bằng dữ liệu như phát hiện gian lận. Với f1-score **1.00**, mô hình đã duy trì sự cân bằng hoàn hảo giữa việc phát hiện chính xác các giao dịch gian lận và

giảm thiểu số lượng cảnh báo giả, một yếu tố then chốt trong các hệ thống phát hiện gian lận.

- Mô hình đạt độ chính xác tổng thể (accuracy) là **0.9990 (~1.00)**, tức là đã phân loại gần đúng tất cả **1,160,123** giao dịch trong tập kiểm thử, bao gồm cả giao dịch hợp pháp và gian lận. Độ chính xác này cho thấy mô hình không mắc phải sai sót nào trong việc dự đoán trên dữ liệu kiểm thử, cho thấy hiệu suất cao.
 - Chỉ số **AUC (Area Under the Curve)** đạt giá trị **0.999988**, gần như hoàn hảo. AUC đo lường khả năng của mô hình phân biệt giữa các giao dịch hợp pháp và gian lận. AUC càng gần 1.00, mô hình càng tốt trong việc phân loại. Với giá trị cao gần mức tuyệt đối, mô hình có khả năng phân biệt rất chính xác giữa các giao dịch hợp pháp và gian lận mà không có bất kỳ sự nhầm lẫn nào.
- Ma trận nhầm lẫn



Hình 53: Random Forest - Confusion Matrix

Các thành phần chính của ma trận này bao gồm:

- **True Positives (TP) = 579,092** Đây là số giao dịch gian lận mà mô hình dự đoán chính xác là gian lận. Mô hình đã hoạt động tốt trong việc phát hiện được phần lớn các giao dịch gian lận.
- **False Negatives (FN) = 970**: Số giao dịch gian lận bị dự đoán nhầm là hợp pháp. Đây là những trường hợp quan trọng mà mô hình không thể phát hiện ra, vì bỏ sót các giao dịch gian lận có thể gây thiệt hại lớn.
- **False Positives (FP) = 165**: Đây là số giao dịch hợp pháp bị mô hình dự đoán sai thành gian lận. Số lượng này không quá lớn, nhưng trong thực tế, số cảnh báo giả cao có thể gây phiền toái cho khách hàng và dẫn đến chi phí kiểm tra giao dịch không cần thiết.
- **True Negatives (TN) = 579,896**: Đây là số giao dịch hợp pháp được dự đoán đúng là hợp pháp, mô hình đã phân loại chính xác rất nhiều giao dịch hợp pháp, đảm bảo sự chính xác tổng thể.

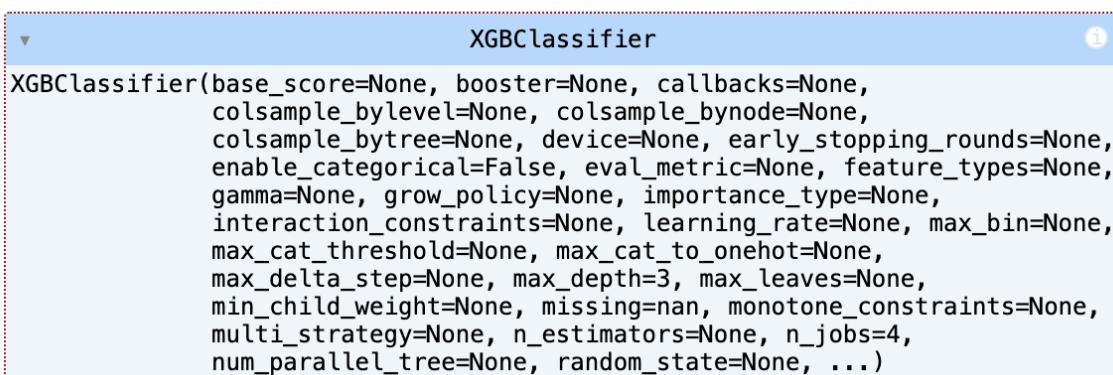
4.2.3.2.3. XGBoost Model

- Khởi tạo và huấn luyện mô hình

Khởi tạo một đối tượng của lớp XGBClassifier và tiến hành huấn luyện mô hình với tập dữ liệu huấn luyện (X_train và y_train):

```
weights = (y == 0).sum() / (1.0 * (y == 1).sum())
clf = XGBClassifier(max_depth = 3, scale_pos_weight = weights, n_jobs = 4)
clf.fit(X_train, y_train)
```

✓ 2.2s



```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=3, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=4,
              num_parallel_tree=None, random_state=None, ...)
```

Hình 54: [Đoạn mã] Khởi tạo mô hình XGBoost

- Dự đoán và đánh giá hiệu suất mô hình

Sau khi huấn luyện, mô hình được sử dụng để dự đoán trên tập dữ liệu kiểm tra (X_{test}). Các dự đoán này sau đó được so sánh với các nhãn thực sự (y_{test}) để đánh giá hiệu suất của mô hình.

----- XGBoost Model -----

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	580062
1	0.99	1.00	0.99	580061
accuracy			0.99	1160123
macro avg	0.99	0.99	0.99	1160123
weighted avg	0.99	0.99	0.99	1160123

Accuracy of XGBoost classifier on test set: 0.9926
AUC Score: 0.9995272119990608

Hình 55: XGBoost Classifier - Classification Report

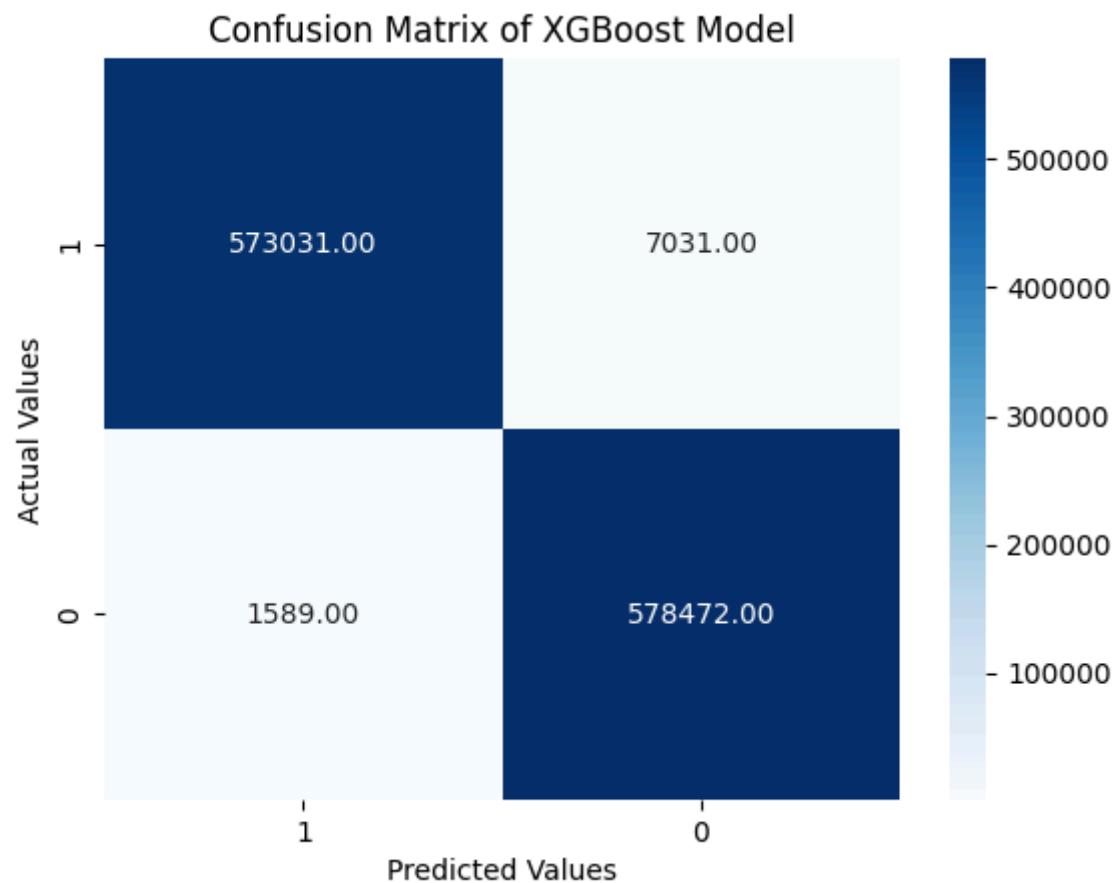
Kết quả đánh giá:

- **Precision (Độ Chính Xác):** phản ánh tỷ lệ các giao dịch được mô hình dự đoán đúng trong tổng số các dự đoán. Mô hình có precision rất cao, gần như tuyệt đối đối với lớp giao dịch hợp pháp (1.00) và tương đối tốt cho lớp gian lận (0.99). Điều này cho thấy mô hình có độ chính xác cao, nhưng vẫn có một lượng nhỏ giao dịch hợp pháp bị phân loại nhầm là gian lận.
- **Recall:** phản ánh khả năng mô hình phát hiện đúng các giao dịch thật sự thuộc về từng lớp. Lớp gian lận đạt tỷ lệ phát hiện hoàn hảo (1.00), tức không bỏ sót bất kỳ giao dịch gian lận nào. Tuy nhiên, lớp giao dịch hợp pháp có recall là 0.99, cho thấy có một số lượng nhỏ giao dịch hợp pháp bị bỏ qua và bị nhầm là gian lận.
- **F1-Score:** là chỉ số kết hợp giữa precision và recall, phản ánh sự cân bằng giữa hai yếu tố này. Với F1-score đạt 0.99 cho cả hai lớp, mô hình

đạt hiệu quả cao trong việc cân bằng giữa phát hiện gian lận và hạn chế nhầm lẫn với các giao dịch hợp pháp.

- **Độ Chính Xác Tổng Thể (Accuracy):** 99,26%, cho thấy mô hình có khả năng phân loại chính xác 99,26% các giao dịch. Đây là một con số rất ấn tượng, chứng tỏ mô hình hoạt động hiệu quả và ổn định trên tập dữ liệu lớn.
- **AUC Score (0.9995):** đo lường khả năng của mô hình trong việc phân biệt giữa hai lớp. Với AUC đạt gần mức hoàn hảo (0.9995), mô hình XGBoost có khả năng rất tốt trong việc phân loại chính xác giữa giao dịch hợp pháp và gian lận. Điều này thể hiện rằng mô hình có thể đưa ra quyết định phân loại một cách mạnh mẽ ngay cả trong những tình huống khó phân biệt.

- Ma trận nhầm lẫn:



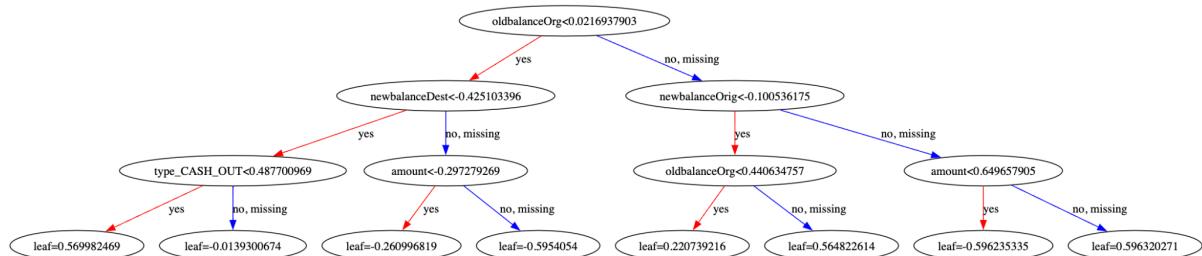
Hình 56: XGBoost Classifier - Confusion Matrix

Các thành phần chính của ma trận này bao gồm:

- **True Positives (TP) = 573,031** Đây là số giao dịch gian lận mà mô hình dự đoán chính xác là gian lận. Mô hình đã hoạt động tốt trong việc phát hiện được phần lớn các giao dịch gian lận.
- **False Negatives (FN) = 7,031**: Số giao dịch gian lận bị dự đoán nhầm là hợp pháp. Đây là những trường hợp quan trọng mà mô hình không thể phát hiện ra, vì bỏ sót các giao dịch gian lận có thể gây thiệt hại lớn.
- **False Positives (FP) = 1,589**: Đây là số giao dịch hợp pháp bị mô hình dự đoán sai thành gian lận. Số lượng này không quá lớn, nhưng trong thực tế, số cảnh báo giả cao có thể gây phiền toái cho khách hàng và dẫn đến chi phí kiểm tra giao dịch không cần thiết.
- **True Negatives (TN) = 578,472**: Đây là số giao dịch hợp pháp được dự đoán đúng là hợp pháp, mô hình đã phân loại chính xác rất nhiều giao dịch hợp pháp, đảm bảo sự chính xác tổng thể.

- Trực quan hóa mô hình XGBoost:

Nút gốc trong cây quyết định được hiển thị bên dưới thực sự là tính năng *oldbalanceOrg*, như mong đợi từ tầm quan trọng cao của nó đối với mô hình.



Hình 57: Minh họa cây quyết định trong mô hình XGBoost

4.2.3.3. Lựa chọn phù hợp

Phát hiện gian lận trong các giao dịch tài chính là một bài toán phân loại nhị phân, nơi yêu cầu độ chính xác cao để giảm thiểu tổn thất do gian lận gây ra. Việc chọn mô hình học máy phù hợp đóng vai trò quyết định trong việc đạt được mục tiêu này. Ba mô hình được so sánh trong bài toán này gồm: Random Forest, Logistic Regression, và XGBoost. Để đưa ra quyết định, ta sẽ đánh giá các mô hình dựa trên các chỉ số như Precision, Recall, F1-Score, và AUC (Area Under the Curve).

4.2.3.3.1. Tiêu chí lựa chọn mô hình

Trong bối cảnh phát hiện gian lận, hai chỉ số đặc biệt quan trọng là Precision và Recall:

- Precision (Độ chính xác): Được sử dụng để đo lường tỷ lệ giao dịch bị đánh dấu là gian lận mà thực sự là gian lận. Mô hình với Precision cao sẽ giúp tránh việc đánh dấu sai những giao dịch hợp pháp là gian lận.
- Recall (Độ nhạy): Đo lường tỷ lệ giao dịch gian lận bị phát hiện, tức là số lượng giao dịch gian lận thực sự được mô hình phát hiện. Recall cao giúp giảm thiểu trường hợp bỏ sót gian lận.
- F1-Score: Là chỉ số tổng hợp cân bằng giữa Precision và Recall. Trong các bài toán như phát hiện gian lận, F1-Score rất quan trọng để tối ưu hóa cả hai.
- AUC: AUC là một thước đo tổng thể cho khả năng phân biệt giữa các lớp (gian lận và không gian lận). AUC càng gần 1, mô hình càng hiệu quả.

4.2.3.3.2. Đánh giá từng mô hình

Dựa trên các kết quả đã cung cấp, ta lần lượt đánh giá từng mô hình theo các tiêu chí trên.

- Random Forest
 - Precision và Recall: Cả hai đều đạt giá trị hoàn hảo là 1.00 cho cả hai lớp (gian lận và không gian lận). Điều này có nghĩa là mô hình không bỏ sót giao dịch gian lận và không đánh dấu sai các giao dịch hợp pháp.
 - F1-Score: 1.00, đồng nghĩa với việc mô hình không chỉ tối ưu về độ chính xác mà còn cân bằng tốt giữa Precision và Recall.
 - AUC: 0.9999, gần như hoàn hảo. Mô hình này có khả năng phân loại chính xác gần tuyệt đối.
- Logistic Regression

- **Precision:** Đối với lớp 0 (không gian lận), Precision đạt 0.95, và đối với lớp 1 (gian lận), Precision đạt 0.94. Điều này cho thấy mô hình Logistic Regression bỏ sót một số giao dịch gian lận và cũng đánh dấu sai một số giao dịch hợp pháp.
- **Recall:** 0.95 cho lớp 1 (gian lận), nghĩa là mô hình này bỏ sót một lượng nhỏ các giao dịch gian lận.
- **F1-Score:** 0.95, thể hiện một sự cân bằng tương đối giữa Precision và Recall nhưng không cao bằng Random Forest.
- **AUC:** 0.9486, thấp hơn đáng kể so với Random Forest, cho thấy khả năng phân loại của Logistic Regression không vượt trội trong bài toán này.

- XGBoost

- **Precision:** Đối với lớp 0 (không gian lận), Precision đạt 1.00, và đối với lớp 1 (gian lận) là 0.99, thể hiện khả năng đánh dấu chính xác gần như hoàn hảo.
- **Recall:** 0.99 cho lớp 0 và 1.00 cho lớp 1, nghĩa là mô hình rất hiếm khi bỏ sót giao dịch gian lận.
- **F1-Score:** 0.99, rất cao và gần như hoàn hảo. Mô hình XGBoost đạt được sự cân bằng tốt giữa Precision và Recall.
- **AUC:** 0.9926, rất gần với Random Forest nhưng thấp hơn một chút.

Bảng 4: So sánh kết quả của ba mô hình

Model	Accuracy	Precision (class 1)	Recall (class 1)	AUC
Logistic Regression	0.95	0.94	0.95	0.9891
Random Forest	0.9990	1.00	1.00	0.9999
XGBoost	0.9926	0.99	1.00	0.9995

4.2.3.3. Lựa chọn mô hình phù hợp nhất

Dựa trên các chỉ số đã phân tích, có thể kết luận rằng Random Forest là mô hình tốt nhất cho bài toán phát hiện gian lận trong giao dịch tài chính. Mô hình này đạt được độ chính xác cao nhất (Precision và Recall đều đạt 1.00), F1-Score hoàn hảo (1.00), và AUC gần như tuyệt đối (0.9999). Điều này cho thấy mô hình có khả năng phân loại chính xác tuyệt đối trên tập dữ liệu kiểm thử, không bỏ sót giao dịch gian lận và cũng không đánh dấu sai các giao dịch hợp pháp.

Trong khi XGBoost cũng là một lựa chọn mạnh mẽ với kết quả gần giống Random Forest, thì Logistic Regression không phù hợp bằng trong bối cảnh yêu cầu độ chính xác cao của bài toán phát hiện gian lận, do AUC và các chỉ số Precision, Recall thấp hơn.

Do đó, Random Forest là mô hình phù hợp nhất để triển khai trong bài toán này, mang lại độ tin cậy cao trong việc phát hiện gian lận và giúp tối ưu hóa hoạt động giao dịch tài chính.

4.2.3.4. Chi tiết mô hình phù hợp

Mô hình Random Forest đã cho kết quả tốt nhất ở trên. Các tham số của mô hình này như sau:

```
RandomForestClassifier Model

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'sqrt',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

Hình 58: Chi tiết mô hình Random Forest

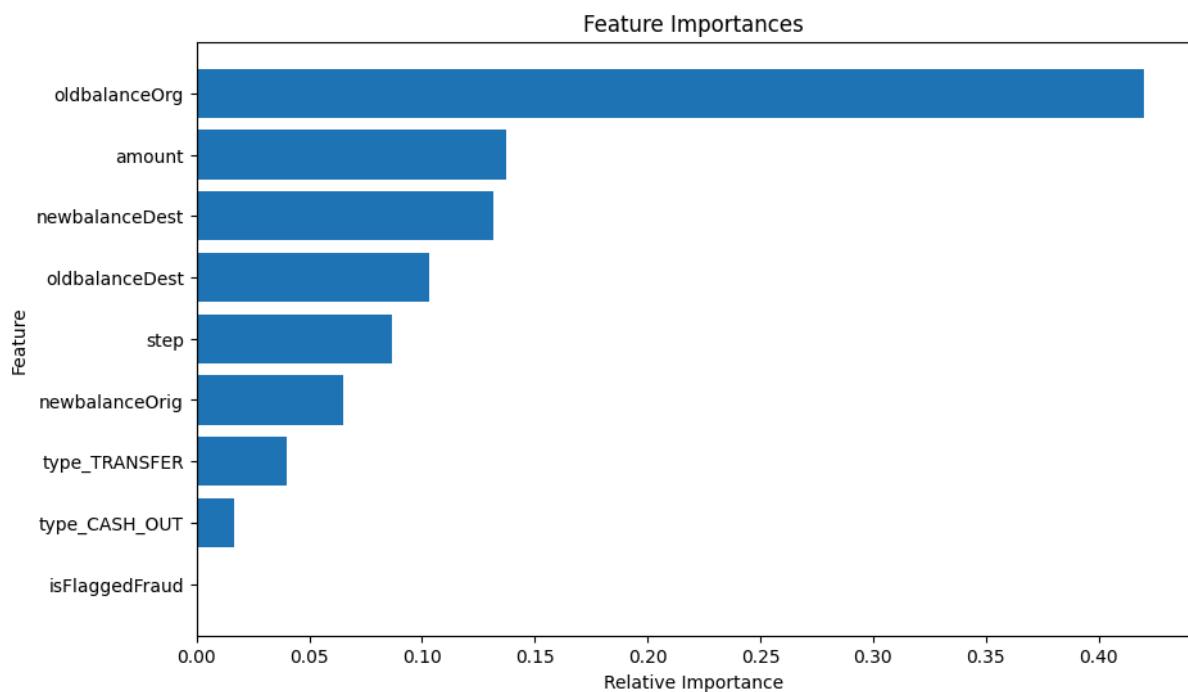
Mô hình sử dụng 10 cây trong rừng (n_estimators) và có độ sâu tối đa vô hạn. Kết quả cross-validation tích cực loại bỏ khả năng overfitting.

Trong hình dưới đây, tôi trình bày tầm quan trọng tương đối của các đặc trưng trong mô hình random forest. Biểu đồ sau cho thấy các biến nào đóng góp nhiều hơn trong việc dự đoán gian lận.

Feature ranking:

1. feature oldbalanceOrg (0.4200012760365622)
2. feature amount (0.1375423534209876)
3. feature newbalanceDest (0.13142560166663572)
4. feature oldbalanceDest (0.1030216946616578)
5. feature step (0.08642142619490382)
6. feature newbalanceOrig (0.06478378332040031)
7. feature type_TRANSFER (0.03993216657671022)
8. feature type_CASH_OUT (0.01685612967736593)
9. feature isFlaggedFraud (1.5568444776606693e-05)

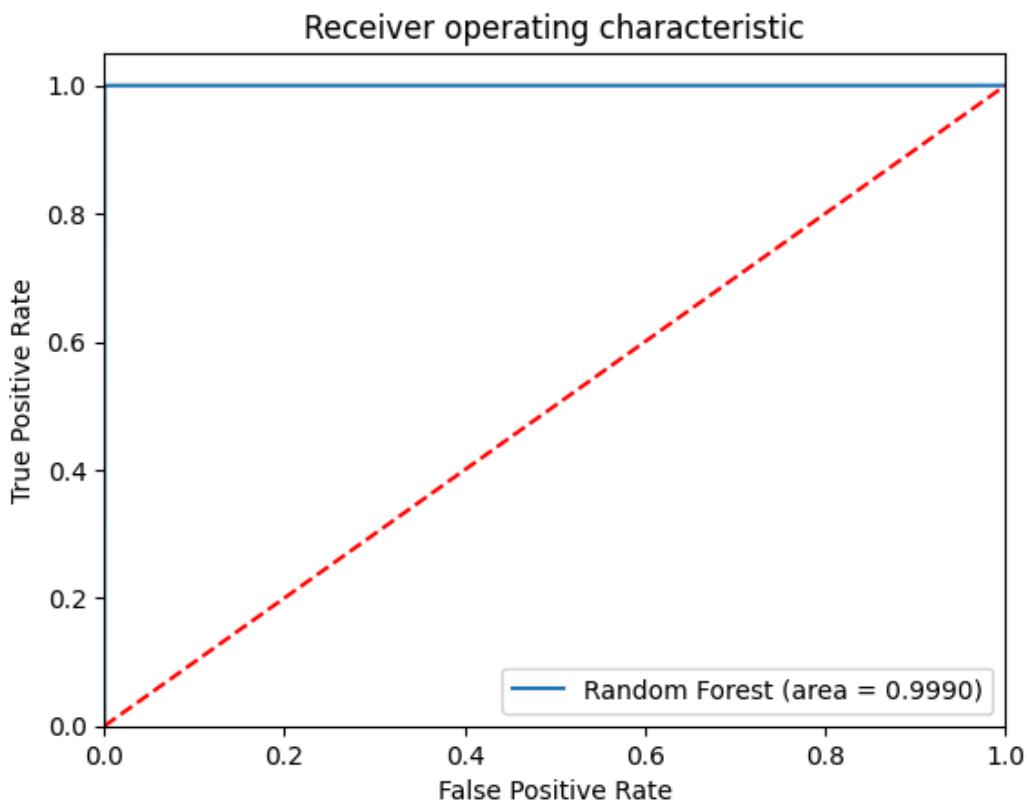
Hình 59: Xếp hạng đặc điểm mô hình Random Forest



Hình 60: Tầm quan trọng của đặc điểm mô hình Random Forest

Do đó, đặc trưng số dư ban đầu của người gửi (“oldbalanceOrg”) đóng vai trò quan trọng trong việc đưa ra dự đoán so với tất cả các biến khác.

Đường cong Receiver-Operator Characteristics (ROC) và tính toán Diện Tích Dưới Đường Cong (AUC) cho mô hình này được mô tả trong hình dưới đây.



Hình 61: Đường cong ROC của Mô hình Random Forest

4.2.4. Tóm tắt Phân tích

Phát hiện gian lận trong giao dịch tài chính là một bài toán quan trọng, giúp các tổ chức tài chính bảo vệ hệ thống và khách hàng khỏi các rủi ro tiềm ẩn. Trong dự án này, tôi đã thực hiện quá trình phân tích dữ liệu giao dịch tài chính, bao gồm làm sạch dữ liệu, phân tích khám phá dữ liệu (EDA) và phát triển mô hình dự đoán sử dụng các thuật toán học máy như Logistic Regression, Random Forest và XGBoost.

Làm sạch dữ liệu là bước khởi đầu để đảm bảo tính nhất quán và chất lượng của dữ liệu, trong đó tôi đã kiểm tra các giá trị bị thiếu, chuyển đổi kiểu dữ liệu và xử lý sự mất cân bằng giữa các lớp. Quá trình phân tích khám phá tập trung vào việc hiểu sâu hơn về các biến số như loại giao dịch, số tiền, và thời gian, đồng thời sử dụng trực quan hóa dữ liệu để rút ra những điểm khác biệt giữa các giao dịch hợp lệ và gian lận.

Đối với mô hình dự đoán, tôi đã thử nghiệm ba mô hình: Logistic Regression, Random Forest và XGBoost. Kết quả cho thấy Random Forest hoạt động tốt nhất, đạt độ chính xác và hồi phục gần như hoàn hảo, trong khi XGBoost cũng cho hiệu suất cao nhưng vẫn kém hơn chút ít. Logistic Regression dù được cải thiện bằng cách giảm mẫu vẫn không đạt hiệu suất mong đợi.

Tóm lại, Random Forest là mô hình hiệu quả nhất cho việc phát hiện gian lận trong dữ liệu giao dịch tài chính này, với khả năng phân loại xuất sắc và không bị quá khóp.

Chương 5: Kết luận

5.1. Kết luận

Qua quá trình thử nghiệm với ba thuật toán học máy, bao gồm Logistic Regression, Random Forest, và XGBoost, mô hình Random Forest và XGBoost đã cho thấy hiệu suất vượt trội trong việc phát hiện gian lận so với Hồi quy Logistic. Cả hai mô hình dựa trên cây, đặc biệt là Random Forest và XGBoost, đều chứng minh được khả năng xử lý tốt dữ liệu giao dịch phức tạp, nơi các lớp được phân biệt rõ ràng, và đạt độ chính xác cùng các chỉ số đánh giá rất cao. Trong đó, XGBoost cũng là một mô hình mạnh mẽ, nhưng kết quả cho thấy Random Forest có hiệu suất nhỉnh hơn một chút trong bài toán này.

Kết quả này không chỉ làm nổi bật vai trò của các thuật toán dựa trên cây mà còn nhấn mạnh tầm quan trọng của việc thực hiện phân tích khám phá dữ liệu (EDA) một cách kỹ lưỡng. Thông qua phân tích này, tôi đã xác định được những đặc trưng quan trọng, giúp phân biệt các lớp giao dịch một cách hiệu quả hơn so với khi chỉ dựa vào dữ liệu thô. Điều này đã góp phần cải thiện đáng kể hiệu suất của các mô hình, đặc biệt là trong các bài toán phân loại nhị phân như phát hiện gian lận.

XGBoost, mặc dù không đạt độ chính xác tuyệt đối như Random Forest, nhưng vẫn là một lựa chọn mạnh mẽ với khả năng xử lý nhanh, hiệu quả và đặc biệt phù hợp với những bài toán có dữ liệu lớn, phức tạp. Trong một số trường hợp nhất định, XGBoost có thể được tối ưu hóa tốt hơn và mang lại kết quả vượt trội trong các kịch bản sản xuất thực tế.

Kết quả cuối cùng khẳng định rằng việc thử nghiệm và lựa chọn giữa các mô hình khác nhau là bước cần thiết để tối ưu hóa cho bài toán cụ thể, và việc nắm bắt các đặc trưng của dữ liệu trước khi áp dụng các thuật toán học máy là yếu tố quyết định sự thành công của dự án phát hiện gian lận tài chính.

5.2. Đề xuất và hướng phát triển

Thông qua dự án này, tôi đã chứng minh rằng có thể xác định các giao dịch gian lận trong dữ liệu giao dịch tài chính với độ chính xác rất cao. Tôi đưa ra những khuyến nghị sau từ dự án này:

- Việc phát hiện gian lận trong dữ liệu giao dịch, nơi có thông tin về số tiền giao dịch và số dư của người nhận và người gửi, có thể được thực hiện tốt nhất bằng cách sử dụng các thuật toán dựa trên cây như Random Forest.
- Sử dụng biểu đồ phân tán và biểu đồ rải để hình dung sự phân tách giữa các giao dịch gian lận và không gian lận là rất cần thiết để chọn các đặc trưng phù hợp.
- Để giải quyết sự mất cân bằng lớp lớn thường thấy trong các vấn đề phát hiện gian lận, có thể sử dụng các kỹ thuật lấy mẫu như giảm mẫu, tăng mẫu và SMOTE. Tuy nhiên, có những hạn chế về yêu cầu tính toán với những phương pháp này, đặc biệt là khi xử lý các tập dữ liệu lớn.
- Để đo lường hiệu suất của các hệ thống phát hiện gian lận, chúng ta cần cẩn thận khi chọn tiêu chí đo lường phù hợp. Tham số độ hồi phục là một tiêu chí tốt vì nó phản ánh liệu có số lượng giao dịch gian lận lớn được phân loại chính xác hay không. Chúng ta không nên chỉ dựa vào độ chính xác vì nó có thể gây hiểu lầm.

TÀI LIỆU THAM KHẢO

- [1] M. A. Marri and A. AlAli, “Financial Fraud Detection using Machine Learning Techniques”.
- [2] Chen, J., & Perez, Y. (n.d.). *Transaction: Definition, Accounting, and Examples*. Investopedia. Retrieved October 14, 2024, from
<https://www.investopedia.com/terms/t/transaction.asp>
- [3] *What Is Money? Definition, History, Types, and Creation*. (n.d.). Investopedia. Retrieved October 14, 2024, from
<https://www.investopedia.com/insights/what-is-money/>
- [4] *What is credit?* (n.d.). CIBC. Retrieved October 14, 2024, from
<https://www.cibc.com/en/personal-banking/loans-and-lines-of-credit/articles-resources/what-is-credit.html>
- [5] *Accounting Transactions - Overview, Types, Double-Entry Recording*. (n.d.). Corporate Finance Institute. Retrieved October 14, 2024, from
<https://corporatefinanceinstitute.com/resources/accounting/accounting-transactions/>
- [6] A. Durnev, K. Li, R. Morck, and B. Yeung, “Capital Markets and Capital Allocation: Implications for Economies of Transition”.
- [7] M. J. Brennan and A. Subrahmanyam, “Investment analysis and price formation in securities markets,” *Journal of Financial Economics*, vol. 38, no. 3, pp. 361–381, Jul. 1995, doi: [10.1016/0304-405X\(94\)00811-E](https://doi.org/10.1016/0304-405X(94)00811-E).
- [8] V. Olkhov, “Financial Variables, Market Transactions, and Expectations as Functions of Risk,” *IJFS*, vol. 7, no. 4, p. 66, Nov. 2019, doi: [10.3390/ijfs7040066](https://doi.org/10.3390/ijfs7040066).
- [9] Y. Chen, E. K. Kumara, and V. Sivakumar, “RETRACTED ARTICLE: Investigation of finance industry on risk awareness model and digital economic growth,” *Ann Oper Res*, vol. 326, no. S1, pp. 15–15, Jul. 2023, doi: [10.1007/s10479-021-04287-7](https://doi.org/10.1007/s10479-021-04287-7).

- [10] E. Henry, E. A. Gordon, B. J. Reed, and T. J. Louwers, “The Role of Related Party Transactions in Fraudulent Financial Reporting,” *SSRN Journal*, 2007, doi: [10.2139/ssrn.993532](https://doi.org/10.2139/ssrn.993532).
- [11] *Gian lận kinh doanh và thanh toán là gì cũng như cách thức để giữ an toàn.* (2023, June 30). Payoneer. Retrieved October 14, 2024, from <https://www.payoneer.com/vi/resources/fraud/>
- [12] R. Chang *et al.*, “Scalable and Interactive Visual Analysis of Financial Wire Transactions for Fraud Detection,” *Information Visualization*, vol. 7, no. 1, pp. 63–76, Mar. 2008, doi: [10.1057/palgrave.ivs.9500172](https://doi.org/10.1057/palgrave.ivs.9500172).
- [13] Quadir, S. (2016, March 11). *Malware suspected in Bangladesh bank heist: officials.* Reuters. Retrieved October 15, 2024, from <https://www.reuters.com/article/technology/malware-suspected-in-bangladesh-bank-heist-officials-idUSKCN0WD1EV/>
- [14] L. Delamaire, H. Abdou, and J. Pointon, “Credit card fraud and detection techniques: a review,” *Banks and Bank Systems*, vol. 4, no. 2, 2009.
- [15] S. J. Leacock, “Fraud in the International Transaction: Enjoining Payment of Letters of Credit in International Transactions”.
- [16] Agbaje. W. Henry, “Effect Of Forensic Accounting Services On Fraud Reduction In The Nigerian Banking Industry,” *ASSRJ*, vol. 4, no. 12, Jun. 2017, doi: [10.14738/assrj.412.3342](https://doi.org/10.14738/assrj.412.3342).
- [17] P. Vanini, S. Rossi, E. Zvizdic, and T. Domenig, “Online payment fraud: from anomaly detection to risk management,” *Financ Innov*, vol. 9, no. 1, p. 66, Mar. 2023, doi: [10.1186/s40854-023-00470-w](https://doi.org/10.1186/s40854-023-00470-w).
- [18] YMT College of Management, Navi Mumbai, Maharashtra, India, S. M. Awale, and Dr. P. Gupta, “Awareness of Sim Swap Attack,” *IJTSRD*, vol. Volume-3, no. Issue-4, pp. 995–997, Jun. 2019, doi: [10.31142/ijtsrd23982](https://doi.org/10.31142/ijtsrd23982).
- [19] *Ngôi sao Wirecard sụp đổ: Bê bối tài chính chấn động nước Đức khiến hàng tỷ USD của nhà đầu tư "không cánh mà bay".* (2022, May 30). CafeF. Retrieved October 15, 2024, from

<https://cafef.vn/ngoi-sao-wirecard-sup-do-be-boi-tai-chinh-chan-dong-nuoc-duc-khien-hang-ty-usd-cua-nha-dau-tu-khong-canhang-ma-bay-20220530144255148.chn>

[20] *Vụ lừa gạt giới đầu tư của Madoff – Wikipedia tiếng Việt.* (n.d.). Wikipedia.

Retrieved October 15, 2024, from

https://vi.wikipedia.org/wiki/V%E1%BB%A5_l%E1%BB%ABa_g%E1%BA%A1t_g%C4%91%E1%BA%A7u_t%C6%B0_c%E1%BB%A7a_Madoff

[21] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary,” *jair*, vol. 61, pp. 863–905, Apr. 2018, doi: [10.1613/jair.1.11192](https://doi.org/10.1613/jair.1.11192).

[22] D. Speelman, “Logistic regression: A confirmatory technique for comparisons in corpus linguistics,” in *Human Cognitive Processing*, vol. 43, D. Glynn and J. A. Robinson, Eds., Amsterdam: John Benjamins Publishing Company, 2014, pp. 487–533. doi: [10.1075/hcp.43.18spe](https://doi.org/10.1075/hcp.43.18spe).

[23] L. Breiman, A. Cutler, A. Liaw, and M. Wiener, “randomForest: Breiman and Cutlers Random Forests for Classification and Regression.” p. 4.7–1.2, Apr. 01, 2002. doi: [10.32614/CRAN.package.randomForest](https://CRAN.R-project.org/package=randomForest).

[24] G. Biau and E. Scornet, “A Random Forest Guided Tour,” Nov. 18, 2015, *arXiv: arXiv:1511.05741*. Accessed: Oct. 14, 2024. [Online]. Available: <http://arxiv.org/abs/1511.05741>.

[25] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).

[26] G. S. D. S. Jayakumar and B. J. Thomas, “A New Procedure of Clustering Based on Multivariate Outlier Detection,” *Journal of Data Science*, vol. 11, no. 1, pp. 69–84, Jul. 2021, doi: [10.6339/JDS.2013.11\(1\).1091](https://doi.org/10.6339/JDS.2013.11(1).1091).

[27] M. Jans, J. M. Van Der Werf, N. Lybaert, and K. Vanhoof, “A business process mining application for internal transaction fraud mitigation,” *Expert Systems with*

Applications, vol. 38, no. 10, pp. 13351–13359, Sep. 2011, doi: [10.1016/j.eswa.2011.04.159](https://doi.org/10.1016/j.eswa.2011.04.159).

[28] Phua, C., Alahakoon, D., & Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, 6(1), 50-59.

[29] P. Ravisankar, V. Ravi, G. Raghava Rao, and I. Bose, “Detection of financial statement fraud and feature selection using data mining techniques,” *Decision Support Systems*, vol. 50, no. 2, pp. 491–500, Jan. 2011, doi: [10.1016/j.dss.2010.11.006](https://doi.org/10.1016/j.dss.2010.11.006).

[30] J. N. Dharwa and A. R. Patel, “A Data Mining with Hybrid Approach Based Transaction Risk Score Generation Model (TRSGM) for Fraud Detection of Online Financial Transaction,” *IJCA*, vol. 16, no. 1, pp. 18–25, Feb. 2011, doi: [10.5120/1977-2651](https://doi.org/10.5120/1977-2651).

[31] Y. Sahin, S. Bulkan, and E. Duman, “A cost-sensitive decision tree approach for fraud detection,” *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, Nov. 2013, doi: [10.1016/j.eswa.2013.05.021](https://doi.org/10.1016/j.eswa.2013.05.021).

[32] Z. Zojaji, R. E. Atani, and A. H. Monadjemi, “A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective”.

[33] R. Wedge, J. M. Kanter, K. Veeramachaneni, S. M. Rubio, and S. I. Perez, “Solving the False Positives Problem in Fraud Prediction Using Automated Feature Engineering,” in *Machine Learning and Knowledge Discovery in Databases*, vol. 11053, U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, and N. Hurley, Eds., in Lecture Notes in Computer Science, vol. 11053. , Cham: Springer International Publishing, 2019, pp. 372–388. doi: [10.1007/978-3-030-10997-4_23](https://doi.org/10.1007/978-3-030-10997-4_23).

[34] A. Mousa, “Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015,” *Journal of Data Science*, vol. 14, no. 3, pp. 553–570, Aug. 2022, doi: [10.6339/JDS.201607_14\(3\).0010](https://doi.org/10.6339/JDS.201607_14(3).0010).

[35] TESTIMON @ NTNU. (n.d.). *Synthetic Financial Datasets For Fraud Detection*. Kaggle. <https://www.kaggle.com/datasets/ealaxi/paysim1/data>

[36] Structure of the analysis of fraud detection:

<https://images.app.goo.gl/gbLGZEvcT7fbPeBWA>

