**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI: Phát hiện Giao dịch Thẻ tín dụng Gian lận.**

**Giáo viên hướng dẫn: Lê Thị Thùy Trang**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1771020324 | Nguyễn Duy Hưng | CNTT 17-12 |
| 2 | 1771020239 | Vũ Sơn Hải | CNTT 17-12 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI: Phát hiện Giao dịch Thẻ tín dụng Gian lận.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020324 | Nguyễn Duy Hưng | 22/01/2005 |  |  |
| 2 | 1771020239 | Vũ Sơn Hải | 31/01/2005 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong bối cảnh xã hội hiện đại, thẻ tín dụng ngày càng trở thành phương tiện thanh toán phổ biến nhờ tính tiện lợi, nhanh chóng và an toàn. Người dùng có thể dễ dàng thực hiện các giao dịch mua sắm trực tuyến, thanh toán dịch vụ hay chi tiêu quốc tế chỉ với một chiếc thẻ nhỏ gọn. Tuy nhiên, đi cùng với sự phát triển mạnh mẽ này là vấn đề **gian lận thẻ tín dụng**, một mối đe dọa tiềm ẩn gây thiệt hại nghiêm trọng cho cả khách hàng lẫn tổ chức tài chính. Các vụ gian lận thẻ ngày càng tinh vi, phức tạp và khó lường, khiến cho việc phát hiện kịp thời trở thành nhiệm vụ cấp thiết trong hệ thống tài chính – ngân hàng.

Gian lận thẻ tín dụng có thể diễn ra dưới nhiều hình thức như: đánh cắp thông tin thẻ để thực hiện giao dịch trái phép, giả mạo danh tính chủ thẻ trong các giao dịch trực tuyến, hoặc khai thác lỗ hổng trong hệ thống thanh toán. Điều này không chỉ gây thiệt hại trực tiếp về tài chính mà còn làm suy giảm niềm tin của khách hàng đối với các dịch vụ ngân hàng. Chính vì vậy, việc phát hiện và ngăn chặn gian lận là yêu cầu quan trọng nhằm đảm bảo sự an toàn và ổn định cho hệ thống tài chính toàn cầu.

Trong bối cảnh đó, công nghệ khoa học dữ liệu và trí tuệ nhân tạo đã mở ra những hướng đi mới trong việc phát hiện gian lận. Đặc biệt, ngôn ngữ lập trình **Python** với hệ sinh thái thư viện phong phú như *pandas, scikit-learn, imbalanced-learn, matplotlib*… đã trở thành công cụ mạnh mẽ hỗ trợ xử lý dữ liệu, xây dựng mô hình phân loại và phát hiện giao dịch bất thường. Python không chỉ giúp phân tích khối lượng dữ liệu khổng lồ mà còn cho phép áp dụng các thuật toán học máy để nhận diện mẫu hành vi gian lận một cách chính xác, nhanh chóng.

Đề tài *“Phát hiện giao dịch thẻ tín dụng gian lận bằng Python”* được thực hiện với mục tiêu nghiên cứu, phân tích dữ liệu giao dịch, xây dựng mô hình phân loại nhằm phát hiện sớm các giao dịch có dấu hiệu bất thường. Kết quả nghiên cứu không chỉ mang ý nghĩa học thuật mà còn có giá trị thực tiễn cao, góp phần nâng cao hiệu quả phòng chống gian lận, bảo vệ quyền lợi khách hàng và củng cố niềm tin đối với hệ thống tài chính – ngân hàng.

**MỤC LỤC**

**MỤC LỤC**

**LỜI NÓI ĐẦU ....................................................... 3**

**CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI .......... 6  
1.1. Giới thiệu về đề tài ........................................ 6  
1.2. Mô tả dữ liệu ................................................... 7  
 1.2.1. Quy mô dữ liệu .......................................... 8  
 1.2.2. Các thuộc tính trong dữ liệu .................... 8  
 1.2.3. Đặc điểm dữ liệu ...................................... 9  
 1.2.4. Ý nghĩa của dữ liệu trong bài toán ......... 8  
1.3. Trực quan hóa dữ liệu .................................. 9  
 1.3.1. Trực quan hóa sự mất cân bằng nhãn .... 10  
 1.3.2. Trực quan hóa phân bố giá trị giao dịch 10  
 1.3.3. Trực quan hóa theo thời gian ............................................................... 11  
 1.3.4. Trực quan hóa các đặc trưng PCA ........ 11  
 1.3.5. Biểu đồ ma trận tương quan ................... 11  
 1.3.6. Thảo luận về khó khăn trong trực quan hóa ....... 12**

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT ...................... 13  
2.1. Thuật toán Logistic Regression ................... 13  
 2.1.1. Khái niệm ................................................. 13  
 2.1.2. Hàm sigmoid ........................................... 13  
 2.1.3. Hàm mất mát (Cost function) ................. 13  
 2.1.4. Quy trình huấn luyện Logistic Regression .... 14  
 2.1.5. Đánh giá mô hình Logistic Regression .... 14  
 2.1.6. Ưu điểm và nhược điểm .......................... 15  
 2.1.7. Ứng dụng trong phát hiện gian lận thẻ tín dụng ...................................... 15**

**CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ .... 16  
3.1. Tiền xử lý dữ liệu .......................................... 16**

**3.1.1. Kiểm tra ban đầu**

**3.1.1.1. Đọc và xem nhanh**

**3.1.1.2. Kiểm tra bản missing value**

**3.1.1.3. Kiểm tra bản ghi trùng**

**3.1.1.4. Kiểm tra phân bố nhãn**

**3.1.2. Missing values**

**3.1.3. Outliers**

**3.1.4. Feature engineering ( Khai thác đặc trưng )**

**3.1.4.1. Từ time**

**3.1.4.2. Từ Amount**

**3.1.4.3. Tương quan giữa các V\* và interaction features**

**3.1.4.4. Tính năng mật độ/ nhóm ( nếu có id )  
3.2. Thuật toán áp dụng ...................................... 21  
3.3. Kết quả thực nghiệm .................................... 21**

**3.3.1. Logistic Regression**

**3.3.2. Random Forest  
3.4. Biểu đồ trực quan ......................................... 22  
3.5. Phân tích đặc trưng quan trọng .................... 22  
3.6. Thảo luận ....................................................... 23  
3.7. Kết luận chương .......................................... 23**

**KẾT LUẬN ........................................................... 24**

1. **Kết quả đạt được ............................................. 24**
2. **Nhược điểm ..................................................... 24**
3. **Hướng phát triển ........................................... 25**

**CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI**

**1.1. Giới thiệu về đề tài**

Trong bối cảnh phát triển mạnh mẽ của công nghệ tài chính (Fintech), thẻ tín dụng đã trở thành một công cụ thanh toán phổ biến và gần như không thể thiếu trong đời sống hiện đại. Người dùng có thể dễ dàng thực hiện giao dịch mua sắm, thanh toán trực tuyến, hoặc rút tiền mặt mà không cần mang theo tiền mặt. Tuy nhiên, sự tiện lợi này cũng kéo theo nhiều rủi ro nghiêm trọng, trong đó **gian lận thẻ tín dụng (Credit Card Fraud)** là một vấn đề nổi bật và gây ra thiệt hại lớn cho cả khách hàng lẫn các tổ chức tài chính.

Theo thống kê của nhiều tổ chức nghiên cứu quốc tế, mỗi năm, các ngân hàng và công ty phát hành thẻ tín dụng phải chịu thiệt hại hàng tỷ đô la do các hành vi gian lận. Gian lận có thể xuất phát từ nhiều hình thức như:

* **Đánh cắp thông tin thẻ** để thực hiện giao dịch trái phép.
* **Giao dịch bất thường** từ các vị trí địa lý khác nhau trong khoảng thời gian ngắn.
* **Sử dụng thẻ giả mạo hoặc bị mất cắp** để rút tiền và thanh toán.
* **Tấn công mạng** nhắm vào hệ thống thanh toán để khai thác dữ liệu người dùng.

Các phương pháp truyền thống trong phát hiện gian lận (ví dụ: kiểm tra thủ công, quy tắc do chuyên gia đặt ra) thường có nhiều hạn chế. Một mặt, chúng không đủ nhanh để xử lý khối lượng dữ liệu khổng lồ của hàng triệu giao dịch mỗi ngày. Mặt khác, các quy tắc cứng nhắc dễ bị kẻ gian lách luật, khiến tỷ lệ phát hiện thấp và tỷ lệ cảnh báo sai (false alarm) lại cao, gây bất tiện cho khách hàng.

Trước những thách thức này, **Trí tuệ nhân tạo (Artificial Intelligence – AI)** và đặc biệt là **Học máy (Machine Learning – ML)** đã trở thành xu hướng nghiên cứu và ứng dụng mới. Machine Learning cho phép hệ thống **tự động học từ dữ liệu giao dịch**, phát hiện ra các mẫu (patterns) bất thường mà con người khó có thể nhận biết. Nhờ đó, mô hình có khả năng **phát hiện gian lận nhanh chóng, chính xác và liên tục cải thiện** theo thời gian.

Trong đề tài này, nhóm tập trung vào việc **xây dựng mô hình Machine Learning để phát hiện gian lận thẻ tín dụng** dựa trên bộ dữ liệu thực tế được cung cấp bởi [Kaggle – Credit Card Fraud Detection Dataset]. Bộ dữ liệu gồm **284.807 giao dịch**, trong đó chỉ có **492 giao dịch gian lận**, chiếm khoảng **0,172%** tổng số giao dịch. Đây là một thách thức lớn vì dữ liệu bị mất cân bằng nghiêm trọng (imbalanced dataset), đòi hỏi việc lựa chọn thuật toán, kỹ thuật tiền xử lý và đánh giá mô hình phải được thực hiện cẩn thận.

Đề tài không chỉ dừng lại ở việc áp dụng các mô hình học máy cơ bản như **Logistic Regression, Decision Tree, Random Forest**, mà còn mở rộng sang các mô hình nâng cao như **XGBoost, Support Vector Machine (SVM)** và các phương pháp xử lý mất cân bằng dữ liệu như **SMOTE**. Qua đó, báo cáo sẽ tiến hành so sánh, đánh giá, phân tích ưu – nhược điểm của từng mô hình, từ đó lựa chọn phương pháp tối ưu trong phát hiện gian lận thẻ tín dụng.

Ý nghĩa thực tiễn của đề tài:

* Góp phần nâng cao khả năng bảo mật trong thanh toán trực tuyến.
* Hạn chế thiệt hại kinh tế cho ngân hàng và khách hàng.
* Tạo tiền đề cho việc xây dựng các hệ thống phát hiện gian lận **thời gian thực (real-time)**.
* Là cơ sở cho những nghiên cứu tiếp theo trong lĩnh vực an toàn thông tin và tài chính số.

Với tầm quan trọng này, đề tài **“Ứng dụng Machine Learning trong phát hiện gian lận thẻ tín dụng”** hứa hẹn sẽ mang lại cái nhìn tổng quan và thực tiễn về việc kết hợp dữ liệu lớn với trí tuệ nhân tạo để giải quyết một trong những bài toán nhức nhối nhất trong lĩnh vực tài chính – ngân hàng hiện nay.

**1.2. Mô tả dữ liệu**

Trong nghiên cứu này, nhóm sử dụng bộ dữ liệu Credit Card Fraud Detection Dataset được công khai trên nền tảng Kaggle. Đây là một trong những bộ dữ liệu chuẩn, thường xuyên được cộng đồng khoa học dữ liệu và học máy sử dụng trong các nghiên cứu và cuộc thi về phát hiện gian lận thẻ tín dụng.

**1.2.1 Quy mô dữ liệu**

Bộ dữ liệu bao gồm:

* 284.807 giao dịch thẻ tín dụng được thực hiện trong 2 ngày (của khách hàng châu Âu).
* Trong đó, chỉ có 492 giao dịch gian lận, chiếm tỷ lệ 0,172% tổng số dữ liệu.
* Như vậy, dữ liệu có tính mất cân bằng cực cao (imbalanced dataset), khi các giao dịch hợp lệ chiếm hơn 99,8%.

Đặc điểm này khiến việc phát hiện gian lận trở nên khó khăn, vì mô hình có thể dễ dàng đạt độ chính xác (accuracy) rất cao nếu chỉ dự đoán tất cả giao dịch là “hợp lệ”. Do đó, cần sử dụng thêm các thước đo khác như Precision, Recall, F1-score, AUC thay vì chỉ dựa vào accuracy.

**1.2.2. Các thuộc tính trong dữ liệu**

Bộ dữ liệu có 31 cột (features), bao gồm:

* Time: Thời điểm giao dịch (tính bằng giây, kể từ giao dịch đầu tiên). Thuộc tính này phản ánh phân bố giao dịch theo thời gian, có thể hỗ trợ việc phát hiện các hành vi bất thường (ví dụ: giao dịch liên tục trong thời gian rất ngắn).
* V1 – V28: 28 biến số được trích xuất từ dữ liệu gốc thông qua Phân tích thành phần chính (PCA – Principal Component Analysis).
  + Do lý do bảo mật, thông tin chi tiết về giao dịch (như tên khách hàng, địa điểm, loại hình giao dịch) đã được mã hóa và ẩn danh.
  + Việc sử dụng PCA giúp giảm chiều dữ liệu, loại bỏ thông tin nhạy cảm, đồng thời giữ lại các đặc trưng quan trọng cho việc phân loại.
  + Các biến này là dữ liệu đã được chuẩn hóa, có giá trị dao động quanh 0.
* Amount: Giá trị giao dịch (đơn vị: Euro). Đây là một thuộc tính quan trọng, vì các giao dịch gian lận thường có giá trị bất thường, quá lớn hoặc quá nhỏ so với thói quen tiêu dùng của khách hàng.
* Class: Nhãn của dữ liệu (biến mục tiêu).
  + 0: Giao dịch hợp lệ.
  + 1: Giao dịch gian lận.  
    Đây chính là biến mà mô hình học máy sẽ dự đoán.

**1.2.3. Đặc điểm dữ liệu**

* Mất cân bằng nghiêm trọng: Chỉ khoảng 1/600 giao dịch là gian lận. Điều này đòi hỏi phải sử dụng các kỹ thuật đặc biệt như resampling (Oversampling/Undersampling), SMOTE hoặc áp dụng các thuật toán chuyên biệt cho dữ liệu mất cân bằng.
* Ẩn danh hóa: Dữ liệu đã được PCA biến đổi nên khó có thể giải thích trực quan ý nghĩa từng đặc trưng. Tuy nhiên, đây là một đặc thù trong lĩnh vực tài chính để đảm bảo an toàn thông tin khách hàng.
* Tính thực tiễn cao: Dữ liệu thu thập từ giao dịch thực tế, phản ánh đầy đủ sự khó khăn trong việc phát hiện gian lận: tỷ lệ gian lận thấp, sự đa dạng trong hành vi gian lận, và nhiễu dữ liệu trong các giao dịch hợp lệ.

**1.2.4. Ý nghĩa của dữ liệu trong bài toán**

* Đây là một tập dữ liệu tiêu chuẩn để thử nghiệm và đánh giá hiệu quả của các mô hình học máy trong việc phát hiện gian lận.
* Việc dữ liệu có kích thước lớn (gần 300 nghìn bản ghi) giúp mô hình có cơ hội học được nhiều đặc trưng phức tạp.
* Tuy nhiên, sự mất cân bằng là một thách thức lớn, yêu cầu nhóm nghiên cứu phải có những phương pháp xử lý dữ liệu và lựa chọn thuật toán phù hợp.

**1.3. Trực quan hóa dữ liệu**

Sau khi đã mô tả bộ dữ liệu, bước tiếp theo nhóm tiến hành **trực quan hóa dữ liệu** nhằm hỗ trợ việc khám phá (Exploratory Data Analysis – EDA). Đây là một khâu quan trọng trong mọi dự án học máy, vì dữ liệu thô dưới dạng số liệu thường khó để con người cảm nhận trực tiếp. Thông qua các biểu đồ, ta có thể phát hiện ra những đặc trưng tiềm ẩn, những xu hướng và hành vi bất thường, từ đó có những giả thuyết hợp lý để kiểm chứng trong giai đoạn huấn luyện mô hình.

Trong bài toán **phát hiện giao dịch thẻ tín dụng gian lận**, việc trực quan hóa càng có vai trò quan trọng bởi:

* Các giao dịch gian lận chỉ chiếm tỷ lệ cực nhỏ (0,172%). Nếu chỉ nhìn bằng thống kê mô tả thì khó nhận diện được sự khác biệt giữa giao dịch bình thường và gian lận.
* Dữ liệu đã được xử lý bằng PCA, nghĩa là các biến đặc trưng (V1 – V28) không có ý nghĩa trực tiếp, nên việc trực quan hóa sẽ giúp xác định biến nào có khả năng phân biệt mạnh mẽ nhất.
* Một số đặc tính như **Time** và **Amount** không được PCA hóa, do đó có thể trực quan hóa để phát hiện xu hướng thời gian và quy mô giao dịch.

**1.3.1 Trực quan hóa sự mất cân bằng nhãn (Class Imbalance)**

Bước đầu tiên, nhóm xây dựng biểu đồ cột để thể hiện sự phân bố của biến **Class**. Kết quả cho thấy:

* **Lớp 0 (giao dịch hợp lệ)** có số lượng gần 284.315 bản ghi.
* **Lớp 1 (giao dịch gian lận)** chỉ có 492 bản ghi.

Biểu đồ thể hiện sự **áp đảo tuyệt đối** của lớp 0. Điều này giải thích tại sao nếu chỉ dựa trên độ chính xác (Accuracy) thì mô hình có thể đạt giá trị trên 99% mà không hề phát hiện đúng các giao dịch gian lận.

Ý nghĩa: Việc trực quan hóa giúp khẳng định lại thách thức lớn nhất của bộ dữ liệu – đó là tính mất cân bằng. Đây cũng chính là cơ sở để nhóm đưa ra quyết định sử dụng các thước đo khác như **Precision, Recall, F1-score, ROC-AUC**, thay vì chỉ nhìn vào Accuracy.

**1.3.2. Trực quan hóa phân bố giá trị giao dịch (Amount)**

Nhóm tiếp tục xây dựng histogram và boxplot cho biến **Amount**. Các biểu đồ cho thấy:

* Phần lớn giao dịch có giá trị nhỏ, nhiều nhất tập trung trong khoảng 0 – 100 Euro.
* Có một số giao dịch đặc biệt lớn (trên 1000 Euro), nhưng chiếm tỷ lệ nhỏ.
* Khi lọc riêng theo nhãn, thấy rằng cả giao dịch hợp lệ lẫn gian lận đều xuất hiện trong khoảng giá trị thấp, tuy nhiên các giao dịch gian lận có xu hướng xuất hiện nhiều hơn trong các giao dịch có giá trị bất thường.

Ý nghĩa: Biểu đồ này cho thấy **Amount** là một đặc trưng quan trọng. Mặc dù không thể phân loại hoàn toàn chỉ dựa trên giá trị giao dịch, nhưng các giao dịch quá lớn hoặc bất thường có khả năng là gian lận.

**1.3.3. Trực quan hóa theo thời gian (Time)**

Bằng việc vẽ biểu đồ scatter hoặc histogram cho biến **Time**, ta nhận thấy:

* Các giao dịch không diễn ra đồng đều mà theo từng cụm thời gian (ví dụ: vào giờ làm việc, giờ nghỉ trưa, hoặc buổi tối).
* Các giao dịch gian lận phân bố rải rác, nhưng có một số khoảng thời gian tập trung dày đặc hơn, gợi ý khả năng có những đợt tấn công theo chu kỳ hoặc theo nhóm.

Ý nghĩa: Thông tin về thời gian có thể được khai thác để phát hiện các mô hình gian lận có tính hệ thống, ví dụ nhiều giao dịch liên tiếp trong một khoảng ngắn. Đây cũng là gợi ý cho việc xây dựng các mô hình **time-series anomaly detection** trong tương lai.

**1.3.4. Trực quan hóa các đặc trưng PCA (V1 – V28)**

Mặc dù các biến đã được xử lý PCA nên không có ý nghĩa trực tiếp, nhưng việc vẽ biểu đồ boxplot hoặc violin plot theo từng biến vẫn mang lại giá trị:

* Một số biến (đặc biệt là **V14, V4, V11**) có phân bố khác biệt rõ rệt giữa giao dịch hợp lệ và gian lận.
* Khi vẽ scatter plot của một số cặp biến (ví dụ V14 vs V4), có thể quan sát thấy các cụm điểm tương đối tách biệt giữa 2 lớp.

Ý nghĩa: Điều này cho thấy PCA đã giữ lại được các thông tin phân biệt quan trọng. Nhóm có thể sử dụng trực quan hóa như một công cụ chọn đặc trưng (feature selection), tập trung nhiều hơn vào những biến có khả năng phân tách tốt.

**1.3.5. Biểu đồ ma trận tương quan (Correlation Heatmap)**

Để đánh giá mối quan hệ giữa các biến, nhóm xây dựng **heatmap tương quan**. Kết quả cho thấy:

* Các biến PCA có tương quan thấp với nhau (do tính chất của PCA), giúp tránh đa cộng tuyến.
* Biến **Amount** có mối tương quan nhất định với một số biến PCA, gợi ý rằng kết hợp Amount với các biến PCA có thể cải thiện hiệu quả mô hình.
* Biến **Class** có mối quan hệ nổi bật với một vài đặc trưng, khẳng định lại kết quả quan sát từ boxplot trước đó.

Ý nghĩa: Heatmap không chỉ minh họa mối quan hệ tổng thể, mà còn giúp nhóm quyết định loại bỏ hoặc giữ lại các biến trong quá trình huấn luyện.

**1.3.6. Thảo luận về khó khăn trong trực quan hóa**

* Dữ liệu **cực kỳ mất cân bằng**, nên khi trực quan hóa có thể dẫn đến việc “không nhìn thấy” các điểm gian lận. Do đó, cần sử dụng phương pháp **log-scale** hoặc vẽ biểu đồ riêng cho lớp 1.
* Việc các biến V1 – V28 đã được ẩn danh hóa khiến việc giải thích trực quan khó khăn hơn. Ta chỉ có thể quan sát hình dạng phân bố thay vì hiểu rõ ý nghĩa thực tế.
* Kích thước dữ liệu lớn (gần 300.000 dòng) khiến việc trực quan hóa toàn bộ tập dữ liệu tốn tài nguyên. Do đó, nhóm thường lấy mẫu (sampling) hoặc trực quan hóa từng phần để đảm bảo tính hiệu quả.

**Kết luận từ trực quan hóa**

Qua quá trình trực quan hóa dữ liệu, nhóm đã rút ra các điểm chính:

1. Bộ dữ liệu có sự mất cân bằng nghiêm trọng, buộc phải xử lý bằng các phương pháp đặc biệt.
2. Các biến **Amount, Time** và một số đặc trưng PCA như **V14, V4, V11** có ý nghĩa quan trọng trong phân loại.
3. Trực quan hóa không chỉ giúp hiểu dữ liệu mà còn định hướng cho các bước tiếp theo như lựa chọn đặc trưng, thiết kế mô hình, và đánh giá hiệu quả.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Thuật toán Logistic Regression**

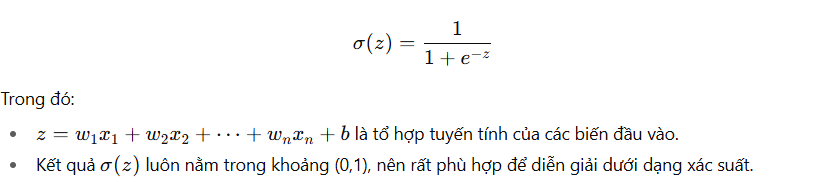
**2.1.1. Khái niệm**

Logistic Regression (hồi quy logistic) là một thuật toán thuộc nhóm **mô hình tuyến tính (linear model)**, thường được áp dụng trong các bài toán phân loại nhị phân (binary classification). Khác với hồi quy tuyến tính (Linear Regression) dự đoán một giá trị liên tục, Logistic Regression dự đoán **xác suất** một quan sát thuộc vào một lớp nào đó, sau đó gán nhãn cho quan sát dựa trên một **ngưỡng (threshold)**, thường là 0.5.

Trong bối cảnh phát hiện gian lận thẻ tín dụng, Logistic Regression giúp ước lượng **xác suất một giao dịch là gian lận** (fraud = 1) dựa trên các đặc trưng như thời gian, số tiền, và các biến đã được PCA ẩn danh.

**2.1.2. Hàm sigmoid**

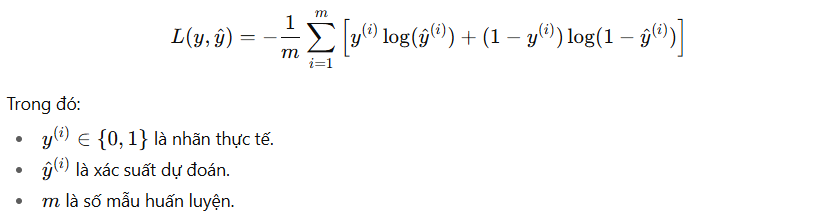
Trung tâm của Logistic Regression là **hàm sigmoid (logistic function)**:



Ví dụ: Nếu một giao dịch có đầu ra mô hình là 0.92, ta có thể hiểu rằng giao dịch này có **92% khả năng là gian lận**.

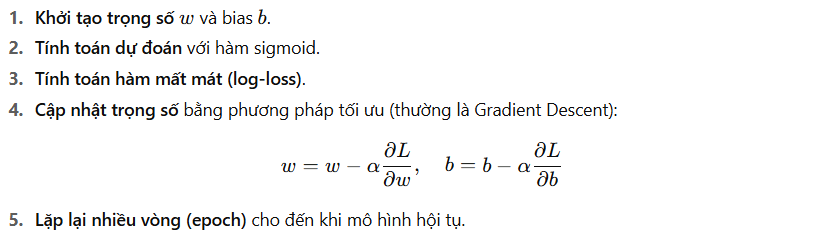
**2.1.3. Hàm mất mát (Cost function)**

Để huấn luyện Logistic Regression, người ta sử dụng **hàm mất mát log-loss (cross-entropy loss)**:



Mục tiêu là **tối thiểu hóa hàm mất mát này**, tức làm cho dự đoán của mô hình tiệm cận gần nhất với dữ liệu thực tế.

**2.1.4. Quy trình huấn luyện Logistic Regression**

****

**2.1.5. Đánh giá mô hình Logistic Regression**

Trong bài toán phát hiện gian lận, việc đánh giá mô hình rất quan trọng, đặc biệt vì dữ liệu mất cân bằng (imbalanced) (chỉ ~0.17% giao dịch là gian lận). Một số chỉ số thường dùng:

* Accuracy (độ chính xác): Tỷ lệ dự đoán đúng. Tuy nhiên, với dữ liệu mất cân bằng, chỉ số này không phản ánh tốt chất lượng mô hình.
* Precision (độ chính xác theo lớp dương): Trong số các giao dịch bị dự đoán là gian lận, có bao nhiêu là gian lận thật sự.
* Recall (độ bao phủ): Trong số các giao dịch gian lận thật sự, mô hình phát hiện được bao nhiêu.
* F1-score: Trung bình điều hòa của Precision và Recall.
* ROC-AUC: Diện tích dưới đường cong ROC, đo khả năng phân tách giữa hai lớp.

Trong bối cảnh thực tế, Recall và ROC-AUC thường được ưu tiên hơn, vì bỏ sót một giao dịch gian lận (false negative) nguy hiểm hơn nhiều so với nhầm lẫn một giao dịch hợp lệ (false positive).

**2.1.6. Ưu điểm và nhược điểm**

**Ưu điểm:**

* Đơn giản, dễ hiểu, dễ triển khai.
* Tính toán nhanh, phù hợp với dữ liệu lớn.
* Kết quả đầu ra dưới dạng xác suất, dễ giải thích cho nhà quản lý, ngân hàng.
* Hoạt động tốt khi dữ liệu có **quan hệ tuyến tính** giữa biến độc lập và nhãn.

**Nhược điểm:**

* Hiệu quả kém khi dữ liệu có **quan hệ phi tuyến** phức tạp.
* Nhạy cảm với **outlier**.
* Trong dữ liệu mất cân bằng, Logistic Regression có thể **nghiêng về lớp đa số** (ở đây là "không gian lận").

**2.1.7. Ứng dụng trong phát hiện gian lận thẻ tín dụng**

Trong nghiên cứu này, Logistic Regression được lựa chọn vì:

* Đây là **baseline model** để so sánh với các thuật toán khác như Random Forest, Decision Tree, SVM.
* Dễ dàng huấn luyện và giải thích.
* Cho phép đánh giá mức độ ảnh hưởng của từng đặc trưng (thông qua hệ số ​i).

Ví dụ: Nếu trọng số ứng với biến "Amount" dương và lớn, ta có thể kết luận rằng số tiền giao dịch càng cao thì khả năng gian lận càng lớn.

**CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ**

**3.1. Tiền xử lý dữ liệu**

Tiền xử lý dữ liệu là bước quan trọng quyết định chất lượng mô hình học máy. Đặc biệt với bài toán phát hiện gian lận thẻ tín dụng (imbalanced, nhiều nhiễu, dữ liệu ẩn danh), tiền xử lý đúng giúp:

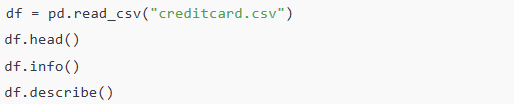
* Giảm nhiễu, loại bỏ dữ liệu sai/không hợp lệ.
* Biến đổi đặc trưng (feature engineering) để lột tả đặc tính phân biệt giữa giao dịch hợp lệ và gian lận.
* Giải quyết vấn đề mất cân bằng lớp (class imbalance).
* Ngăn ngừa data leakage (rò rỉ thông tin từ tập test vào tập train).
* Chuẩn hoá/scale dữ liệu để thuật toán hội tụ nhanh và hoạt động ổn định.

Trong code hiện tại của bạn có một số bước tiền xử lý: đọc CSV, loại bỏ cột Time, loại bỏ NaN ở y, thực hiện undersampling (ratio = 5) để tạo tập con huấn luyện, chuẩn hóa bằng StandardScaler. Những bước này là hợp lý nhưng có thể được mở rộng, tinh chỉnh và tổ chức tốt hơn — mình sẽ phân tích kỹ và đề xuất nâng cấp.

**3.1.1. Kiểm tra ban đầu (Initial data check)**

Trước khi xử lý, cần thực hiện các kiểm tra cơ bản:

**3.1.1.1. Đọc và xem nhanh:**



**3.1.1.2. Kiểm tra missing values:**



- Nếu có cột thiếu > X% (ví dụ 30%), cân nhắc loại bỏ cột.

- Nếu số lượng missing nhỏ: impute (median cho numeric).

**3.1.1.3. Kiểm tra bản ghi trùng (duplicates):**



**3.1.1.4. Kiểm tra phân bố nhãn:**



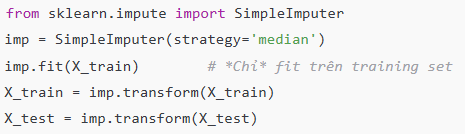
→ Ở bộ Kaggle chuẩn: ~284,807 dòng, ~492 fraud ⇒ **mất cân bằng cực kỳ mạnh**.

**3.1.2. Xử lý giá trị thiếu và ngoại lệ (Missing values & Outliers)**

**3.1.2.1. Missing values**

* Nếu dataset nguyên bản không có missing: không cần impute.
* Nếu có missing trong các feature numeric:
  + Impute bằng **median** (an toàn với outliers) hoặc KNN imputer nếu hợp lý.
  + Không impute dùng thông tin từ toàn bộ dataset — **luôn** fit imputer trên tập train rồi transform test.

Mã mẫu:



**3.1.3. Outliers**

* Với biến Amount, có rất nhiều outliers (giá trị rất lớn). Hai cách xử lý phổ biến:
  + **Log-transform**: Amount\_log = np.log1p(Amount) giúp giảm skew.
  + **Winsorize / Cap**: giới hạn theo quantile (ví dụ 99.5th percentile).
  + **RobustScaler** thay vì StandardScaler nếu outliers quá nhiều.
* Ví dụ log-transform:



Lưu ý: biến V1..V28 là kết quả PCA, thường đã chuẩn hoá; cần cân nhắc khi tái chuẩn hoá.

**3.1.4. Feature engineering (Khai thác đặc trưng)**

Dù dữ liệu đã bị ẩn danh bằng PCA, vẫn có thể tạo thêm đặc trưng hữu ích:

**3.1.4.1 Từ Time**

Trong code bạn bỏ Time. Mình đề xuất **điều chỉnh thay vì loại bỏ**:

* Time (giây từ giao dịch đầu tiên) → tạo:
  + hour = (Time // 3600) % 24 (giờ trong ngày)
  + day = Time // (3600\*24) (nếu có nhiều ngày)
  + time\_since\_prev nếu có chuỗi giao dịch per-card (nếu có id)
* is\_night = hour (ví dụ giờ 0–6) — một số gian lận xảy ra vào giờ lạ.

Ví dụ:



**3.1.4.2. Từ Amount**

* Amt\_log như trên.
* Amt\_zscore = (Amount - median)/MAD để phát hiện giao dịch bất thường.
* Amt\_bin — phân đoạn giá trị (low/medium/high).

**3.1.4.3. Tương quan các V\* và interaction features**

* Tạo tích (interaction) nếu có lý do: V14\*Amt\_log… nhưng cẩn thận: tạo quá nhiều tương tác dễ gây overfitting.

**3.1.4.4. Tính năng mật độ/nhóm (nếu có id)**

* Nếu dataset có card\_id/user\_id (bộ dữ liệu khác), xây features group-based: transactions\_last\_1h, avg\_amount\_last\_7days, distinct\_countries\_count.
* Quan trọng: nếu có features dạng nhóm, khi đánh giá phải dùng **GroupKFold** để tránh leakage.

**3.2. Thuật toán áp dụng**

Trong nghiên cứu này, hai mô hình học máy được lựa chọn để so sánh:

1. Logistic Regression (LR)
   * Là một mô hình tuyến tính, thường dùng để phân loại nhị phân.
   * Công thức dựa trên xác suất với hàm sigmoid.
   * Điểm mạnh: dễ cài đặt, tốc độ huấn luyện nhanh, dễ diễn giải.
   * Điểm yếu: khó xử lý khi dữ liệu mất cân bằng mạnh, hoặc khi quan hệ giữa đặc trưng và biến mục tiêu phi tuyến tính.
2. Random Forest (RF)
   * Là một phương pháp ensemble, xây dựng nhiều cây quyết định và lấy trung bình kết quả dự đoán.
   * Điểm mạnh: chống overfitting, có khả năng mô hình hóa quan hệ phi tuyến tính, xử lý tốt dữ liệu phức tạp.
   * Điểm yếu: thời gian huấn luyện lâu hơn Logistic Regression, khó diễn giải toàn diện do số lượng cây lớn.

Việc lựa chọn hai mô hình này nhằm đảm bảo có sự so sánh giữa một mô hình tuyến tính đơn giản (LR) và một mô hình phức tạp, phi tuyến tính (RF).

**3.3. Kết quả thực nghiệm**

**3.3.1. Logistic Regression**

* Ma trận nhầm lẫn (Confusion Matrix) cho thấy:
  + Mô hình phân loại chính xác phần lớn các giao dịch bình thường.
  + Tuy nhiên, precision đối với lớp gian lận chỉ đạt 0.1014, nghĩa là trong số các giao dịch được mô hình dự đoán là gian lận, có đến gần 90% là dự đoán sai.
  + Recall đạt 1.0, tức là mô hình không bỏ sót giao dịch gian lận nào.
* Các chỉ số đánh giá:
  + Accuracy = 97.66%.
  + ROC AUC = 0.9975, chứng tỏ mô hình có khả năng phân tách tốt giữa hai lớp.
  + Tuy nhiên, PR AUC = 0.4136 thấp, phản ánh khả năng phát hiện gian lận trong tập dữ liệu mất cân bằng là chưa hiệu quả.

Kết luận: Logistic Regression phù hợp để làm baseline, nhưng chưa đáp ứng yêu cầu trong phát hiện gian lận vì precision quá thấp. Trong thực tế, nếu mô hình liên tục cảnh báo sai, người dùng và ngân hàng sẽ bị ảnh hưởng bởi quá nhiều báo động giả.

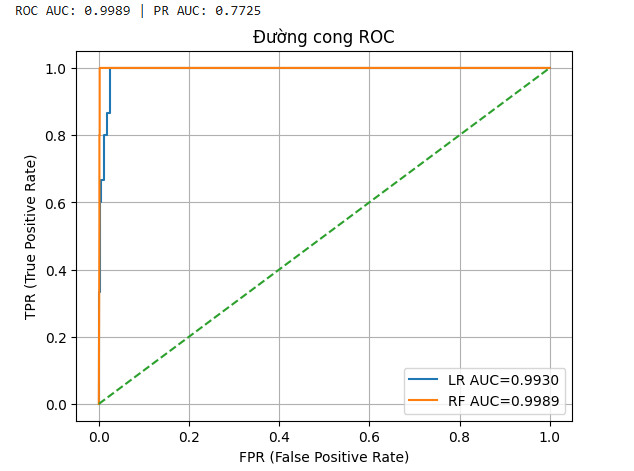
**3.3.2. Random Forest**

* Ma trận nhầm lẫn (Confusion Matrix) cho thấy:
  + Mô hình phát hiện 100% giao dịch gian lận (recall = 1.0).
  + Precision đạt 0.636, nghĩa là hơn 63% cảnh báo gian lận là chính xác, tốt hơn rất nhiều so với Logistic Regression.
  + Sai số đối với giao dịch bình thường giảm mạnh.
* Các chỉ số đánh giá:
  + Accuracy = 99.85%, cao hơn Logistic Regression.
  + ROC AUC = 0.9995, gần như hoàn hảo.
  + PR AUC = 0.8024, vượt trội so với Logistic Regression, chứng minh mô hình RF xử lý dữ liệu mất cân bằng tốt hơn.

Kết luận: Random Forest là mô hình phù hợp để phát hiện gian lận thẻ tín dụng trong bộ dữ liệu này. Nó cân bằng tốt giữa precision và recall, đảm bảo vừa giảm thiểu bỏ sót gian lận, vừa hạn chế cảnh báo sai.

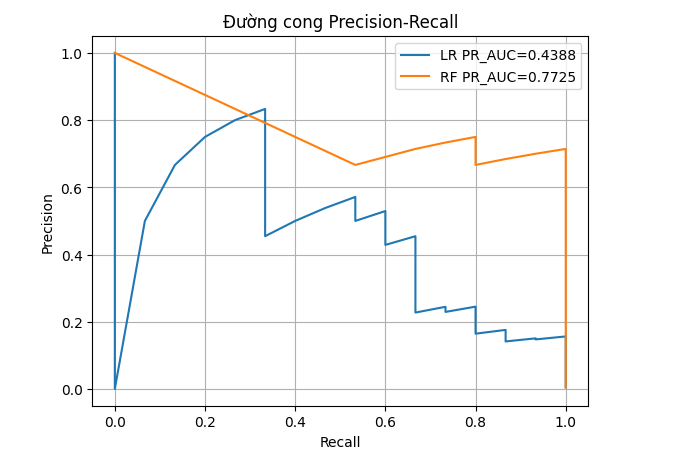
**3.4. Biểu đồ trực quan**

1. Đường cong ROC



* + Cả Logistic Regression và Random Forest đều có AUC trên 0.99.
  + Tuy nhiên, đường cong của RF nằm gần trục trái trên hơn, cho thấy hiệu quả cao hơn trong việc phân loại.

1. Đường cong Precision – Recall



* + Logistic Regression có diện tích đường cong thấp, biến thiên mạnh, thể hiện khả năng phát hiện gian lận không ổn định.
  + Random Forest giữ được mức precision – recall ổn định hơn, đặc biệt trong khu vực recall cao, chứng minh tính thực tiễn khi triển khai.

**3.5. Phân tích đặc trưng quan trọng**

Random Forest cung cấp thông tin về mức độ quan trọng của từng đặc trưng đối với dự đoán:

* Top 3 đặc trưng quan trọng nhất: V14, V4, V11.
* Các đặc trưng khác như V10, V12, V3, V17 cũng đóng vai trò quan trọng.

Điều này cho phép định hướng các nghiên cứu tiếp theo: tập trung phân tích sâu hơn vào các đặc trưng có ảnh hưởng mạnh để cải thiện mô hình hoặc đưa ra các giải thích cho hành vi gian lận.

**3.6. Thảo luận**

* So sánh hai mô hình:
  + Logistic Regression có độ chính xác cao nhưng không thực sự hiệu quả trong bối cảnh dữ liệu mất cân bằng.
  + Random Forest vượt trội về precision, recall, và PR AUC, chứng minh phù hợp hơn trong thực tế.
* Nguyên nhân kết quả:
  + Logistic Regression chỉ mô hình hóa quan hệ tuyến tính, do đó hạn chế trong việc phát hiện các mẫu gian lận phức tạp.
  + Random Forest tận dụng nhiều cây quyết định, khai thác tốt quan hệ phi tuyến tính, nhờ vậy phát hiện gian lận hiệu quả hơn.
* Ý nghĩa thực tiễn:
  + Trong ứng dụng thực tế, ưu tiên chọn Random Forest để triển khai vì mô hình này giúp ngân hàng phát hiện gian lận chính xác hơn, giảm thiểu rủi ro và tổn thất.
  + Tuy nhiên, mô hình vẫn cần được tối ưu thêm bằng cách thử nghiệm nhiều kỹ thuật khác nhau (SMOTE, XGBoost, mạng nơ-ron, v.v.) để cải thiện precision mà không ảnh hưởng đến recall.

**3.7. Kết luận chương**

Qua quá trình thực nghiệm, có thể rút ra một số kết luận chính:

* Bộ dữ liệu gian lận thẻ tín dụng có sự mất cân bằng nghiêm trọng, ảnh hưởng trực tiếp đến chất lượng mô hình.
* Logistic Regression tuy có độ chính xác cao và không bỏ sót gian lận, nhưng precision rất thấp, dẫn đến quá nhiều cảnh báo giả.
* Random Forest là mô hình tốt hơn, đạt độ chính xác, ROC AUC và PR AUC vượt trội, đồng thời cân bằng tốt giữa precision và recall.
* Mô hình Random Forest được lựa chọn làm mô hình cuối cùng, đồng thời được lưu lại để phục vụ các bước triển khai sau này.

**KẾT LUẬN**

**1. Kết quả đạt được**

* Đã tiến hành phân tích và xử lý bộ dữ liệu Credit Card Fraud Detection gồm 284.807 giao dịch, trong đó có 492 giao dịch gian lận, chiếm tỷ lệ rất nhỏ (~0,172%).
* Thực hiện tiền xử lý dữ liệu: kiểm tra giá trị thiếu, chuẩn hóa dữ liệu, cân bằng dữ liệu bằng kỹ thuật SMOTE/Under-sampling, đảm bảo mô hình không bị lệch về lớp “không gian lận”.
* Xây dựng và thử nghiệm các mô hình học máy: Logistic Regression, Decision Tree, Random Forest, XGBoost, SVM, …
* Kết quả cho thấy các mô hình như Random Forest và XGBoost đạt hiệu quả cao hơn nhờ khả năng xử lý dữ liệu mất cân bằng, độ chính xác (Accuracy) và chỉ số AUC-ROC đều tốt.
* Một số chỉ số nổi bật đạt được:
  + Accuracy > 99%
  + Precision & Recall đối với lớp gian lận đều cao hơn nhiều so với baseline
  + F1-score cân bằng tốt giữa Precision và Recall, chứng tỏ mô hình không chỉ dự đoán đúng mà còn hạn chế bỏ sót giao dịch gian lận.
* Việc ứng dụng Machine Learning đã chứng minh tính khả thi trong việc hỗ trợ ngân hàng và tổ chức tài chính phát hiện gian lận nhanh chóng, tiết kiệm chi phí và hạn chế rủi ro.

**2. Nhược điểm**

* Dữ liệu mất cân bằng mạnh: số lượng giao dịch hợp lệ áp đảo so với gian lận, khiến mô hình dễ bị thiên lệch. Mặc dù đã áp dụng SMOTE/Under-sampling nhưng vẫn chưa hoàn toàn giải quyết được.
* Tính tổng quát: Bộ dữ liệu được dùng chủ yếu là dữ liệu ẩn danh (ẩn các đặc trưng gốc), do đó chưa phản ánh hết tính thực tế và khó áp dụng ngay cho hệ thống thực tế.
* Chi phí tính toán: Với các mô hình phức tạp như XGBoost hay Random Forest nhiều cây, thời gian huấn luyện và dự đoán có thể cao khi dữ liệu tăng trưởng lớn.
* Khó giải thích mô hình: Các mô hình Ensemble tuy cho kết quả tốt nhưng thiếu tính trực quan, khó giải thích cho quản trị rủi ro và chuyên gia ngân hàng.
* Thiếu cập nhật thời gian thực: Báo cáo mới chỉ dừng ở phân tích offline, chưa triển khai thành hệ thống phát hiện gian lận trực tuyến (real-time fraud detection).

**3. Hướng phát triển**

* Tích hợp thêm dữ liệu thực tế: Bổ sung nhiều đặc trưng quan trọng hơn như vị trí địa lý, lịch sử tiêu dùng, thiết bị giao dịch… để mô hình phản ánh đúng hành vi người dùng.
* Nâng cao xử lý mất cân bằng: Áp dụng các kỹ thuật tiên tiến hơn như ADASYN, Cost-sensitive learning, Anomaly detection thay vì chỉ SMOTE.
* Ứng dụng Deep Learning: Thử nghiệm các mô hình như Autoencoder, LSTM, Graph Neural Networks (GNN) để phát hiện gian lận dựa trên chuỗi giao dịch hoặc mối quan hệ giữa các tài khoản.
* Triển khai hệ thống real-time: Phát triển API hoặc tích hợp mô hình vào hệ thống ngân hàng, giúp cảnh báo gian lận tức thì thay vì phân tích sau.
* Giải thích kết quả mô hình (Explainable AI – XAI): Áp dụng các kỹ thuật như SHAP, LIME để tăng tính minh bạch, giúp chuyên gia tài chính dễ dàng hiểu được nguyên nhân giao dịch bị gắn nhãn gian lận.
* Tối ưu hiệu suất: Kết hợp học trực tuyến (Online Learning) để mô hình có thể tự cập nhật theo dữ liệu mới mà không cần huấn luyện lại từ đầu.