MaSSP 2021 - Machine Learning & Data Science
Các hướng tiếp cận trong xử lý dữ liệu mất cân bằng ở bài toán nhận diện gian lận tín dụng.

Phạm Tiến Sơn [1] **Nguyễn Trần Nhật Quốc** [*] **Nguyễn Minh Châu** [*]

Abstract

Khi mà việc sử dụng thẻ tín dụng để thanh toán thay cho tiền ngày càng trở nên phổ biến, số lượng các hành vi lừa đảo, gian lận tín dụng cũng ngày càng tăng lên, gây ra những thiệt hại tài chính cho chủ thẻ và các tổ chức kinh tế quản lý. Những giao dịch gian lận có những đặc điểm phức tạp,thay đổi theo thời gian và càng khó nắm bắt hơn khi số lượng của chúng chỉ chiếm một phần rất nhỏ trong hàng triệu giao dịch tín dụng được thực hiện trong một ngày. Do đó, nhu cầu xây dựng một hệ thống nhận diện gian lận chính xác là ngày càng lớn và số lượng các công trình nghiên cứu liên quan cũng tăng nhanh trong thời gian gần đây. Trong bài báo cáo này sẽ giới thiệu về bài toán nhận diện gian lận tín dụng, các hướng tiếp cận cho vấn đề dữ liệu đầu vào mất cân bằng của bài toán và so sánh hiệu năng của các kĩ thuật được giới thiệu.

1. Học máy trong bài toán gian lận tín dụng

Gian lận thẻ tín dụng là hình thức gian lận sử dụng công nghệ cao để đánh cắp thông tin thẻ tín dụng. Thiệt hại đến nền kinh tế thế giới do các hoạt động gian lận thẻ tín dụng lên đến hàng chục tỉ đô-la.

Với lượng giao dịch khổng lồ ngày nay, khi mà các điều tra viên không còn theo kịp với mức độ biến đổi nhanh chóng về hình thức của các hành vi gian lận, việc sử dụng các mô hình học máy (Machine Learning) trong phát hiện giao dịch gian lận phát triển với tốc độ chóng mặt. Riêng trong năm 2020, đã có 1500 bài báo nghiên cứu về chủ đề này được xuất bản. Tuy nhiên, nhận diện gian lận tín dùng sử dụng các mô hình học máy không phải là một bài toàn dễ dàng.

Một số vấn đề khó khăn khi áp dụng học máy trong nhận diện gian lận tín dụng được đề cập ở phần 1.1 và hai nhóm phương pháp chính nhận diện gian lận tín dụng được nêu ở phần 1.2.

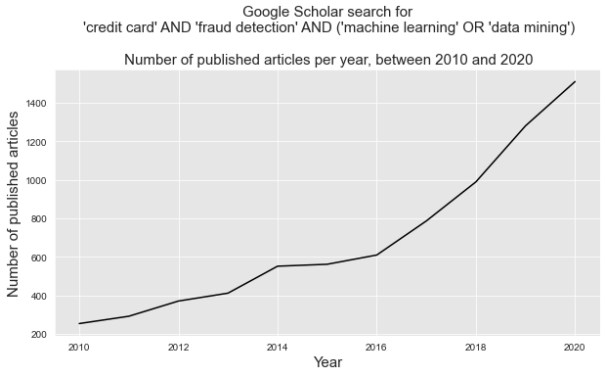

Figure 1. Số lượng bài báo được xuất bản với chủ đề về học máy và hệ thống nhận diện gian lận tín dụng trong khoảng thời gian từ 2010 đến 2020. Nguồn: Google Scholar.

1.1. Các thách thức trong bài toán gian lận tín dụng

Dữ liệu mất cân bằng: Tổng số giao dịch gian lận chỉ chiếm khoảng 1% các giao dịch trên thế giới - dữ liệu mất cân bằng, gây khó khăn cho việc học máy, cần dùng thêm các biện pháp như sampling hoặc loss weighting.

Hình thức hoạt động gian lận thay đổi theo thời gian: Thói quen tiêu dùng của chủ thẻ luôn thay đổi theo thời gian. Kẻ trộm cũng nghĩ ra những cách mới để trộm tiền. Sự thay đổi này yêu cầu các hệ thống nhận diện phải được cập nhật để có thể hoạt động tốt theo thời gian.

Feature Engineering: Các giao dịch tín dụng được biểu diễn dưới dạng các dữ liệu số và dữ liệu kiểu phân loại. Tuy nhiên, hầu hết các mô hình học máy không thể tiếp nhận trực tiếp các dữ liệu phân loại mà phải chuyển sang dữ liệu số.

Tính tuần tự: Các giao dịch của mỗi cá nhân có tình tuần tự với những đặc điểm riêng biệt. Mô hình cần nắm bắt và hiểu những đặc điểm này để nhận diện được sự bất thường.

Đánh giá hiệu năng của mô hình: Những cách đánh giá hiệu năng thông thường như *mean misclassification error* hay *AUC ROC*, đều không phù hợp với bài toán này do tính mất cân bằng trong dữ liệu phân loại và hậu quả của của việc phân loại sai là khác nhau đối với mỗi lớp dữ liệu (Thường

thì bỏ sót một giao dịch gian lận gây thiệt hại lớn hơn nhận diện nhầm). Hiện vẫn chưa có sự thống nhất giữa các nhà nghiên cứu về một metric đánh giá mặc dù đây là một khía cạnh rất quan trọng.

Thiếu dữ liệu công khai: Vì lí do bảo mật, dữ liệu các giao dịch tín dụng không được công khai. Việc thiếu các dữ liệu thực tế công khai ảnh hưởng tới khả năng lặp lại của các nghiên cứu và gây khó khăn trong so sánh các phương pháp khác nhau trong các công trình đôc lập.

1.2. Các phương pháp nhận diện gian lận tín dụng

Các phương pháp nhận diện gian lận tín dụng được xếp vào hai lớp chính: **phân tích hành vi gian lận (miuse detection)** và **phân tích hành vi người dùng (anomaly detection)**.

Lớp phương pháp đầu giải quyết bài toán phân loại với đầu vào là các giao dịch được đánh nhãn là *bình thường* và *gian lận*. Cách tiệp cận này có khả năng trích xuất các đặc điểm của các hành vi gian lận có trong dữ liệu đầu vào và nhận diện các giao dịch gian lận với tỉ lệ nhận nhầm thấp. Tuy nhiên, với sự xuất hiện của các hình thức gian lận mới, hệ thống hoạt động kém với tỉ lệ bỏ sót cao.

Lớp phương pháp thứ hai giải bài toán học không giám sát trên các dữ liệu giao dịch của một người dùng. Một giao dịch được nhận diện là gian lận khi có đặc điểm hành vi khác với những các giao dịch trước của người dùng. Mặc dù có thể bắt được các phương thức gian lận mới, cách tiếp cận này có thể gây ra rắc rối cho người dùng bởi tỉ lệ nhận nhầm cao.

2. Phân loại dữ liệu mất cân bằng

Trong các tập dữ liệu thực tế, chiếm phần đa số là các sự kiện *bình thường* và các sự kiện *đặc biệt* chỉ chiếm phần nhỏ dữ liệu. Điều thường xảy ra trong các bài toán phân loại với dữ liệu mất cân bằng là hậu quả của việc nhận diện sai một sự kiện *đặc biệt* thành *bình thường* lớn hơn nhiều khi nhận diện nhầm theo chiều hướng ngược lại. Tuy nhiên, các thuật toán học máy phổ biến không được thiết kế để dành ưu tiên cho một lớp dữ liệu và trong điều kiện dữ liệu lệch về lớp sự kiện *bình thường*, hiệu năng của chúng giảm đáng kể và khả năng nhận diện lớp thiểu số kém hơn nhiều so với lớp đa số.

Vấn đề khó khăn đầu tiên là phải đi định nghĩa: "thế nào là một mô hình *tốt*?" Với một dữ liệu chỉ gồm 0.1%là các giao dịch gian lận, mô hình có thể cho rằng tất các giao dịch là bình thường mà vẫn đạt được độ chính xác (*accuracy* - tỉ lệ dự đoán đúng trên tổng dữ liệu) lên đến $0.99/1$. Phần tới sẽ giới thiệu hai metrics phổ biến trong bài toán này là diện tích dưới đường cong ROC (AUC ROC) và diện tích dưới đường cong Precision-Recall (Average Precision).

Hai hướng tiếp cận trong bài toán phân loại dữ liệu mất cân bằng là: xử lý sự mất cân bằng của dữ liệu và cái tiến các thuật toán học cho thích hợp với điều kiện mất cân bằng dữ liệu.

Phần 2.1 sẽ giới thiệu hai metrics đánh giá là AUC ROC và Average Precision. Các kĩ thuật xử lý dữ liệu sẽ được nêu ở phần 2.2 và các thuật toán học trong bài toán phân loại dữ liệu mất cân bằng được bàn luận ở phần 2.3.

2.1. Metric đánh giá - AUC ROC và Average Precision

Hai metrics trình bày trong phần này thuộc nhóm *threshold-free metrics*, không phụ thuộc vào một điểm nút t ($0 \leq t \leq 1$ dùng để phân một giao dịch x là gian lận nếu $p(x) > t$ với $p(x)$ được trả về bởi mô hình phân loại). Khác với các metrics phổ biến như Precision, Recall hay F1-Score phụ thuộc vào một giá trị t, AUC ROC và AP đánh giá hiệu năng của mô hình một cách tổng thể trên các giá trị khác nhau của t.

AUC ROC: Diện tích phần phía dưới phía dưới đường cong ROC. AUC ROC càng gần tới 1 thì khả năng mô hình nhận diện chính xác 2 lớp càng cao.

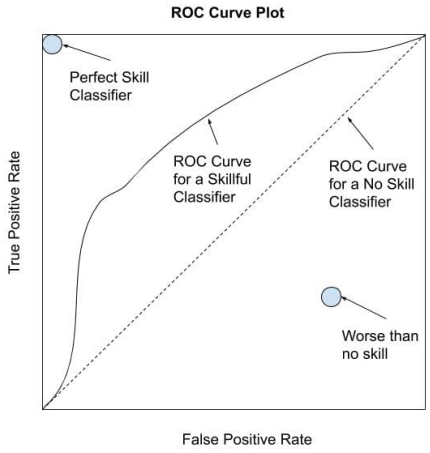

Figure 2. Đường cong ROC.

Đường cong ROC có thể cho thấy hiệu năng của mô hình với các giá trị False Positive Rate (tỉ lệ nhận nhầm một giao dịch là giao dịch gian lận trong bài toán nhận diện gian lận tín dụng). Tuy nhiên đòi hỏi thực tiễn là FPR chỉ có giá trị thấp do những giao dịch được mô hình phân vào lớp gian lận sẽ phải được kiểm tra bởi nguồn nhân lực có giới hạn của công ty. Đó cũng là điểm yếu khi sử dụng AUC ROC cho bài toán này do chỉ có phần diện tích nhỏ phía bên trái (giá trị FPR nhỏ) là có ý nghĩa trong so sánh hiệu năng các mô hình.

Average Precision: Diện tích phần phía dưới đường cong Precision-Recall.

Một điểm khác biệt có thể thấy rõ qua 4 và 3 là một mô hình

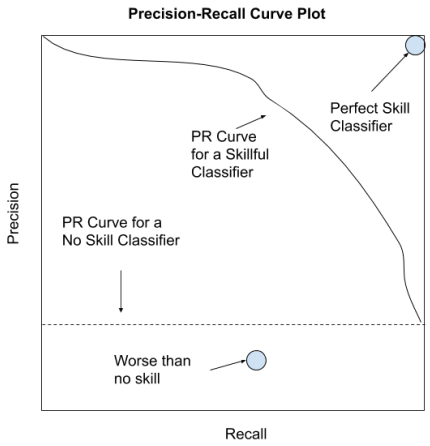

Figure 3. Đường cong Precision-Recall.

phân loại theo kiểu *tung đồng xu* ($p(x) = 0.5$ với mọi giao dịch x) có AUC ROC luôn là 0.5 còn AP sẽ phụ thuộc vào tỉ lệ số lượng hai lớp.

Hơn nữa, dựa vào đường cong PR có thể đánh giá hiệu năng của mô hình với điều kiện FPR thấp.

2.2. Hướng tiếp cận về mặt dữ liệu - Sampling methods

Các kĩ thuật trong phần này được thực hiện trong quá trình xử lý dữ liệu, trước khi đưa vào các mô hình học máy. Mục tiêu của hướng tiếp cận này là tạo ra một tập dữ liệu mới với số lượng phần tử của mỗi lớp cân bằng hơn. Hai kĩ thuật cơ bản để làm giảm sự mất cân bằng là undersampling và oversampling. Sau đó chúng tôi sẽ giới thiệu về kĩ thuật *tạo điểm dữ liệu tổng hợp lớp thiểu số* (SMOTE - Synthetic Minority Oversampling TEchnique). Minh họa cho các kĩ thuật chọn mẫu (sampling methods) ở Figure 4.

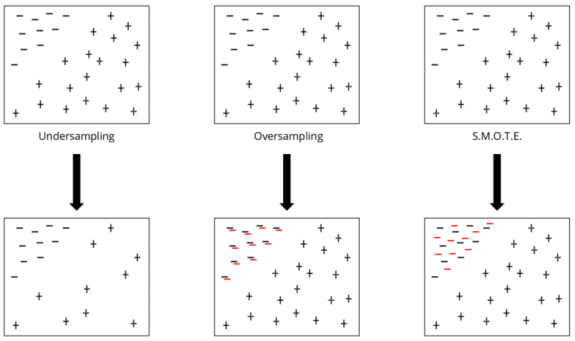

Figure 4. Các kĩ thuật lấy mẫu trong bài toán phân loại dữ liệu mất cân bằng (Dấu + đại diện cho điểm dữ liệu thuộc lớp số nhiều và dấu - đại diện cho lớp số ít. Các dấu có màu đỏ là những điểm dữ liệu được tạo ra bởi oversampling hoặc SMOTE).

Undersampling là tạo ra bộ dữ liệu mới bằng cách lược bớt các phần tử thuộc lớp đa số để dữ liệu trở nên cân bằng hơn với giả định rằng dữ liệu thuộc lớp đa số chứa những thông tin thừa, không có ý nghĩa nhiều cho mô hình phân loại và có thể lược bớt. Vấn đề với undersampling là, trong trường hợp sự chênh lệch giữa số lượng phần tử quá lớn, nhiều điểm dữ liệu thuộc lớp đa số bị lược bỏ và gây ra sự thiếu hụt dữ liệu, mô hình không hoạt động tốt trong thực tế. Tuy vậy, undersampling cũng thường được sử dụng bởi nó làm giảm thời gian huấn luyện mô hình khi giảm bớt dữ liệu đầu vào.

Oversampling làm giảm sự chênh lệch dữ liệu ở các lớp bằng cách tạo các bản sao của điểm dữ liệu lớp thiểu số. Không có thông tin nào bị mất đi khi oversampling nhưng cũng không có thông tin nào được thêm vào, việc các điểm dữ liệu lớp thiểu số có nhiều bản sao trong tập dữ liệu có thể gây ra tình trạng quá khớp (overfitting) ở mô hình phân loại. Overfitting cũng làm tăng thời gian huấn luyện của mô hình lên khi kích cỡ dữ liệu đầu vào tăng lên.

SMOTE là bước cải tiến của kĩ thuật oversampling khi mà dữ liệu mới sẽ được tổng hợp (synthesized) từ các dữ liệu cũ chứ không đơn thuần chỉ là bản sao của những dữ liệu lớp thiểu số sẵn có. SMOTE chọn hai điểm dữ liệu gần giống nhau và tạo ra một điểm dữ liệu mới nằm trên đoạn thẳng nối hai điểm đó. SMOTE hoạt động tốt hơn Over-sampling do cách tạo điểm dữ liệu mới khắc phục được hạn chế làm mô hình bị overfitting nhưng thời gian chạy mô hình cũng tăng lên.

2.3. Hướng tiếp cận về mặt thuật toán - Model-based methods

Được cải tiến để có hiệu năng tốt hơn trong điều kiện dữ liệu mất cân bằng là một số mô hình ensembles như EasyEnsemble, RUSBoost, Balanced Bagging và Balanced Random Forest.

3. Thực nghiệm

3.1. Giới thiệu về dữ liệu

Tập dữ liệu gồm có 284807 dòng, là các giao dịch thẻ tín dụng của khách hàng Châu Âu trong vòng 2 ngày T9/2013.

Tính mất cân bằng của tập dữ liệu: Có tổng cộng 492/284.807 (0.172%) giao dịch là gian lận.

Đa số features đã được biến đổi PCA: Dữ liệu chỉ bao gồm số đã được chuyển đổi theo phương pháp PCA vì lí do bảo mật.

Phép phân tích thành phần chính (PCA) phân tách các chiều không gian - các thành phần chính chứa đựng nhiều thông tin nhất của dữ liệu. Ví dụ của một thành phần chính mà PCA tìm ra có thể là như sau - một tổ hợp tuyến tính của các

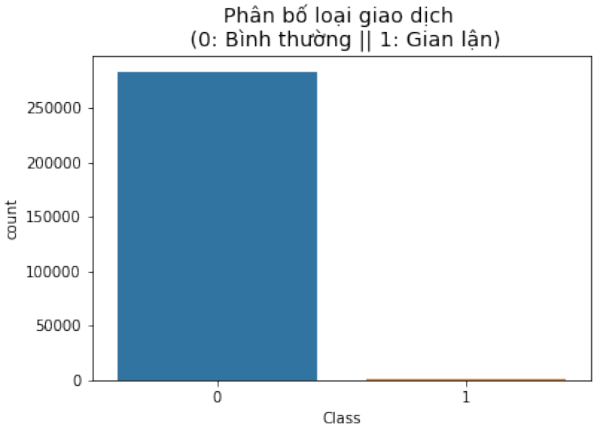

Figure 5. Phân bố target

dữ liệu input:

$$V1 = 4 \cdot TransactionID - \frac{3}{4} \cdot TerminalID + \cdots$$

Mỗi một giao dịch có một giá trị 'V1' và trường 'V1' sẽ lưu giữ một phần độ phân tán của dữ liệu - có ý nghĩa trong khi đưa vào các mô hình học máy. Các trường dữ liệu 'V1' đến 'V27' trong dữ liệu của bài toán này chính là 27 thành phần chính của phép PCA.

Chỉ từ các trường này thì rất khó khôi phục lại được các dữ liệu ban đầu nhưng các mô hình hoàn toàn có thể được huấn luyện tốt trên các dữ liệu được biến đổi này.

Tương quan trường Amount và Time với Target: Trường Amount thể hiển số tiền giao dịch, trường Time thể hiện số giây giữa giao dịch đầu tiên trong tập dữ liệu với mỗi giao dịch.

- Không có sự khác biệt nhiều số tiền giao dịch giữa các giao dịch bình thường và lừa đảo, do hầu hết số tiền trong các giao dịch này đều dưới 2500 €.
- Số lượng các giao dịch bình thường giảm mạnh trong khoảng gần giây thứ 18.000 và giây thứ 100.000, khoảng giảm và phục hồi là 10.000 giây (xấp xỉ 6 tiếng). Xu hướng tăng giảm này lặp lại 2 lần nên có thể đây khoảng giảm là thời gian tối muộn trong hai ngày của dữ liệu và lượng các giao dịch chuyển tiền và mua bán thông thường sẽ giảm vào khoảng thời gian này trong ngày. Đặc biệt, số lượng lớn các giao dịch được gán nhãn lừa đảo diễn ra trong khoảng giây thứ 100.000, điều này phù hợp với các quan sát thực tế khi các tội phạm thường tận dụng thời gian đêm khuyu để khiến nạn nhân mất thời gian nhận ra các hành vi chiếm đoạt.

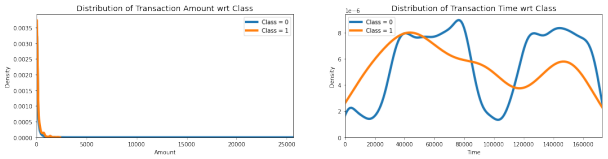

Figure 6. Phân bố các trường Time và Amount với target

3.2. Metrics đánh giá sử dụng

Sử dụng hai metrics được giới thiệu ở 2.1, kí hiệu AUC cho AUC ROC và AP cho Average Precision. Ngoài ra, thời gian huấn luyện (TT) và thời gian đưa ra dự đoán (PT) theo đơn vị giây cũng sẽ được ghi lại.

3.3. Môi trường thử nghiệm

Các thử nghiệm được thực hiện trên Kaggle Notebook. Các thuật toán lấy mẫu và học máy được cung cấp bởi các thư viện scikit-learn và imbalanced-learn.

Chia tập train và test: Tập train là những giao dịch trong khoảng dưới giây thứ 100.000 để mô phỏng việc sử dụng những thông tin giao dịch của ngày t để dự đoán các giao dịch ở ngày $t + 1$. Tỉ lệ kích cỡ tập train - test là $54 - 46$.

Tiền xử lý dữ liệu: Trường Time và Amount được scale lại sử dụng lần lượt công cụ StandardScaler và RobustScaler của thư viện scikit-learn.

Lựa chọn tham số: Các mô hình Logistic Regression, Decision Tree, Random Forest và XGBoost được tối ưu siêu tham số trên số điểm AP trung bình khi thực hiện Stratified K-folds với số folds là 5 sử dụng thư viện Optuna một lần duy nhất khi không sử dụng các kĩ thuật sampling. Bộ siêu tham số này tiếp tục được sử dụng khi có thêm các kĩ thuật sampling để nhìn rõ ảnh hưởng của chúng.

3.4. Các kết quả và nhận xét

Bảng 1 cho kết quả baseline khi không sử dụng các kĩ thuật sampling trên các mô hình phổ biến là Logistic Regression, Decision Tree, Random Forest và XGBoost. Có thể thấy rằng các mô hình hiện tại đều gặp tình trạng quá khớp với tập Train. Hai mô hình Random Forest và XGBoost đang cho kết quả tốt hơn về metric AP, nhưng đây cũng là hai mô hình có thời gian huấn luyện và đưa ra dự đoán (TT và PT) cao hơn.

Kết quả khi sử dụng Random Under Sampling (RUS) trên 4 mô hình trên được cho ở Bảng 2. Trong khi, AP của mô hình Decision Tree giảm mạnh khi sử dụng RUS trên cả hai tập Train và Test, sự cải thiện rõ ràng nhất ở mô hình Random Forest khi cho số điểm AP là 0.790 vượt xa các mô hình khác. Thời gian huấn luyện các mô hình đã giảm mạnh khi dữ liệu đã giảm kích cỡ đi nhiều khi RUS.

	Train			Test		
	AUC	AP	TT	AUC	AP	PT
LR	0.983	0.748	1.881	0.973	0.684	0.078
DT	0.973	0.803	5.120	0.914	0.549	0.045
RF	0.965	0.819	191.886	0.965	0.753	3.455
XGB	1.000	0.989	43.613	0.982	0.772	0.262

Table 1. Kết quả của trên tập Train và Test của bốn mô hình Logistic Regression (LT), Decision Tree (DT), Random Forest (RF) và XGBoost (XGB).

+ RUS	Train			Test		
	AUC	AP	TT	AUC	AP	PT
LR	0.984	0.695	0.138	0.981	0.618	0.056
DT	0.980	0.089	0.145	0.928	0.024	0.036
RF	0.987	0.772	1.256	0.985	0.790	3.754
XGB	1.000	0.902	0.290	0.984	0.701	0.176

Table 2. Kết quả của trên tập Train và Test của bốn mô hình Logistic Regression (LT), Decision Tree (DT), Random Forest (RF) và XGBoost (XGB) khi sử dụng Random Under Sampling.

Bảng 3 cho thấy Random Over Sampling (ROS) cải thiện kết quả trên cả tập Train và tập Test khi so sánh với khi sử dụng RUS. Tuy nhiên thời gian huấn luyện đã tăng lên đáng kể.

+ ROS	Train			Test		
	AUC	AP	TT	AUC	AP	PT
LR	0.987	0.712	5.244	0.970	0.743	0.071
DT	0.990	0.729	6.811	0.891	0.611	0.046
RF	0.987	0.720	259.472	0.981	0.787	3.791
XGB	1.000	0.991	62.981	0.966	0.745	0.168

Table 3. Kết quả của trên tập Train và Test của bốn mô hình Logistic Regression (LT), Decision Tree (DT), Random Forest (RF) và XGBoost (XGB) khi sử dụng Random Over Sampling.

Thời gian huấn luyện các mô hình khi sử dụng SMOTE tăng lên đáng kể nhưng trong quá trình thử nghiệm này chưa cho kết quả thực sự vượt trội so với các kĩ thuật trước. Từ Bảng 4, sự cải thiện rõ nhất là ở mô hình Logistic Regression với AP là 0.754.

Bảng 5 là kết quả trên tập Train và Test của bốn mô hình EasyEnsemble, RUSBoost, BalancedBagging và Balanced Random Forest. Có thể thấy rằng với việc kết hợp kĩ thuật RUS trong bootstrapping, EasyEnsemble và BalancedRandomForest cho hiệu năng tốt với thời gian huấn luyện ít hơn rất nhiều khi sử dụng các kĩ thuật sampling trên dữ liệu rồi đưa vào mô hình hay cho mô hình học trực tiếp dữ liệu mất cân bằng.

+SMOTE	Train			Test		
	AUC	AP	TT	AUC	AP	PT
LR	0.986	0.711	4.927	0.969	0.754	0.067
DT	0.986	0.719	11.909	0.922	0.608	0.051
RF	0.985	0.681	406.313	0.981	0.785	3.973
XGB	1.000	0.981	104.387	0.958	0.744	0.169

Table 4. Kết quả của trên tập Train và Test của bốn mô hình Logistic Regression (LT), Decision Tree (DT), Random Forest (RF) và XGBoost (XGB) khi sử dụng SMOTE.

	Train			Test		
	AUC	AP	TT	AUC	AP	PT
EE	0.999	0.817	3.466	0.970	0.739	13.605
RUB	0.996	0.661	4.940	0.950	0.613	1.391
BB	0.999	0.708	3.090	0.975	0.523	0.356
BRF	1.000	0.984	6.734	0.984	0.774	1.056

Table 5. Kết quả của trên tập Train và Test của bốn mô hình EasyEnsemble, RUSBoost, Balanced Bagging và Balanced Random Forest.

4. Tổng kết

Các kết quả trên cho thấy rằng khi đưa ra những lựa chọn về hướng tiếp cận trong xây dựng mô hình thực tế, ta luôn phải cân nhắc giữa hai yếu tố là khả năng nhận diện và thời gian tính toán. Kết quả trên các mô hình ensembles của thư viện imbalanced-learn đã thể hiện được sự hài hòa giữa hai yếu tố trên khi có thể thấy BalancedRandomForest có thời gian huấn luyện ít đi nhiều khi sử dụng Random Forest trực tiếp, với ROS hay SMOTE và vẫn đạt được kết quả AP tương đương trên tập Test.

Việc sử dụng kết hợp SMOTE và RUS có thể là hướng tiếp cận tiếp theo cho sự cân bằng giữa thời gian tính toán và khả năng phân loại trong các mô hình.

Khuôn khổ bài báo cáo này không đề cập đến các kĩ thuật Feature Engineering trong bài toán nhận diện gian lận tín dụng do những hạn chế về các bộ dữ liệu được tiếp cận. Chúng tôi cũng chưa thể bao gồm phần về cost-sensitive learning - một hướng đi có tính thực tiễn cao khi bao gồm cụ thể thiệt hại khi nhận diện sai một giao dịch theo số tiền giao dịch, và những thành tựu mới nhất khi áp dụng các mạng học sâu vào trong hệ thống nhận diện gian lận tín dụng.

Tài liệu

Machine Learning Group (Université Libre de Bruxelles - ULB). *Fraud Detection Handbook.*

Janio Bachman. *Credit Fraud || Dealing with Imbalanced Datasets*