Bank Customer Churn Prediction

1st Hoàng Ngọc Hồng
21522106
21522106@gm.uit.edu.vn

2nd Nguyễn Thành Huy
21522159
21522159@gm.uit.edu.vn

***Tóm tắt nội dung*—Trong môi trường kinh doanh ngân hàng ngày càng cạnh tranh , giữ chân được khách hàng rất quan trọng luôn được ưu tiên trong các chính sách của ngân hàng. Việc ngân hàng xác định được những khách hàng có ý định rời bỏ dịch vụ giúp ngân hàng có thể kịp thời đưa ra các chính sách nhằm giữ chân được khách hàng.Báo cáo này tập trung xây dựng các mô hình dự đoán thực hiện các việc như phân loại khách hàng có ý định rời bỏ hay không.**

***Index Terms*—Phân tích khách hàng; rời bỏ dịch vụ; machine learning**

I. Introduction

Cuộc chạy đua trong việc giữ chân khách hàng giữa các ngân hàng đang vô cùng căng thẳng, mỗi ngân hàng đều đưa ra những chính sách riêng của họ để có thể thực hiện hóa các chính sách dành cho khách hàng đang có ý định ngưng sử dụng các dịch vụ.Một số nghiên cứu cho thấy vấn đề khách hàng rời bỏ gây tổn thất đáng kể cho ngân hàng. Nghiên cứu của Roberts (2000), Buckinx và Van den Poel (2005), Coussement và Van den Poel (2008) đã chỉ ra rằng chi phí tìm khách hàng mới cao hơn nhiều so với chi phí để giữ chân khách hàng cũ. Cụ thể, chi phí thu hút khách hàng mới gấp 6 lần chi phí giữ chân khách hàng (Athanassopoulos, 2000; Bhattacharya, 1998; Colgate và Danaher,2000; Rasmusson, 1999). Thêm vào đó,chi phí bán hàng cho khách hàng mới nhiều gấp 5 lần so với chi phí bán hàng cho khách hàng cũ (Dixon, 1999; Floyd, 2000; Slater và Narver, 2000). Giữ chân khách hàng trở thành một vấn đề cấp thiết đối với ngân hàng. Vì vậy, dự báo những khách hàng nguy cơ rời bỏ trong tương lai có thể giúp ngân hàng can thiệp kịp thời trong hiện tại để ngăn chặn vấn đề khách hàng rời bỏ.

Tiếp theo một số báo cáo liên quan đến việc trình bày trong Phần II giới thiệu bối cảnh và các công việc liên quan,Phần III là dataset và phân tích dữ liệu, Phần IV là các mô hình đề xuất , Phần V là kết quả , Phần VI là kết luận.

II. Related Work

Các nghiên cứu trước đây đã ứng dụng nhiều phương pháp máy học(Machine Learning)khác nhau để tìm ra các phương pháp tốt nhất cho việc dự báo khả năng khách hàng rời bỏ. Tuy nhiên các nghiên cứu khác nhau cho ra các kết quả khác nhau. Một số phương pháp học máy thể hiện hiệu quả các dự báo nổi bật khi so sánh với các phương pháp khác, bao gồm:SVM(Support Vector Machine), RF(Random Forest)và cây quyết định(Decision Tree)....

Như vậy, có rất nhiều phương pháp đã được áp dụng để dự đoán khách hàng rời bỏ, trong đó một số phương pháp cho kết quả nổi trội như Decision Tree, SVM, RF. Bài viết này sẽ sử dụng các phương pháp đó để dự báo khách hàng rời bỏ, nhằm chọn ra phương pháp dự báo chính xác nhất. Đồng thời nghiên cứu những thuộc tính quan trọng ảnh hưởng đến khả năng rời bỏ của khách hàng.

III. Dataset

3.1 Dataset:

Bộ dữ liệu được sử dụng cho bài toán này có tên Bank Customer Churn Prediction và được cung cấp trên Kaggle.

Bộ dữ liệu bao gồm 10.002 mẫu, mỗi mẫu đại diện cho một khách hàng của ngân hàng.

- Customer ID: ID của khách hàng
- Surname: Họ hoặc tên của khách hàng
- Credit Score: Điểm tín dụng của khách hàng
- Geography: Quốc gia nơi khách hàng cư trú (Pháp, Tây Ban Nha hoặc Đức)
- Gender: Giới tính của khách hàng (Nam hoặc Nữ)
- Age: Độ tuổi của khách hàng.
- Tenure: Số năm khách hàng đã gắn bó với ngân hàng
- Balance: Số dư tài khoản của khách hàng
- NumOfProducts: Số lượng sản phẩm ngân hàng khách hàng sử dụng (ví dụ: tài khoản tiết kiệm, thẻ tín dụng)
- HasCrCard: Khách hàng có thẻ tín dụng hay không (1 = có, 0 = không)
- IsActiveMember: Khách hàng có phải là thành viên tích cực hay không (1 = có, 0 = không)
- EstimatedSalary: Mức lương ước tính của khách hàng
- Exited: Khách hàng có rời đi hay không (1 = có, 0 = không)

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42.0	2	0.00	1	1.0	1.0	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41.0	1	83807.86	1	0.0	1.0	112542.58	0
2	3	15619304	Onio	502	France	Female	42.0	8	159660.80	3	1.0	0.0	113931.57	1
3	4	15701354	Boni	699	France	Female	39.0	1	0.00	2	0.0	0.0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43.0	2	125510.82	1	NaN	1.0	79084.10	0

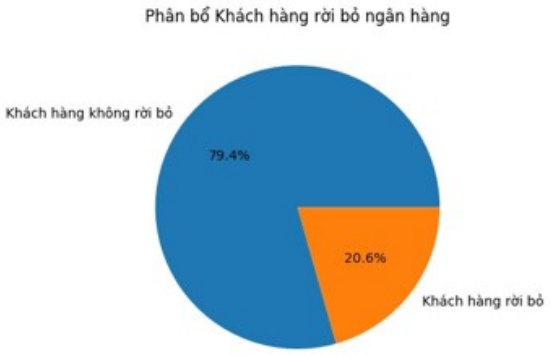
Hình 1: Dataset

Nguồn: **https://www.kaggle.com/datasets/shantanudhakadd/bank-customer-churn-prediction.**

3.2 Phân tích bộ dữ liệu:

3.2.1 Phân bổ khách hàng:

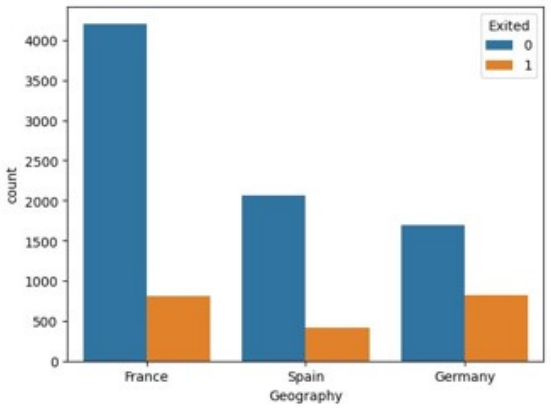
-Khoảng 20% (2038) khách hàng trong bộ dữ liệu đã rời bỏ tài khoản ngân hàng.
- Có một sự chênh lệch khá lớn giữa lượng khách hàng rời bỏ và không rời bỏ (đây cũng là hai nhãn phân loại(target).
=> Sẽ cần phải xử lí sự mất cân bằng này trước khi đưa vào mô hình huấn luyện.



Hình 2: Khách hàng

3.2.2 Phân bổ vị trí địa lý:

- Với biểu đồ phân bổ vị trí địa lý thì Khách hàng đến từ Pháp là nhiều nhất TBN thứ 2 và Đức là thứ 3.
- Tuy nhiên tỉ lệ khách hàng rời bỏ lại chủ yếu đến từ Đức.

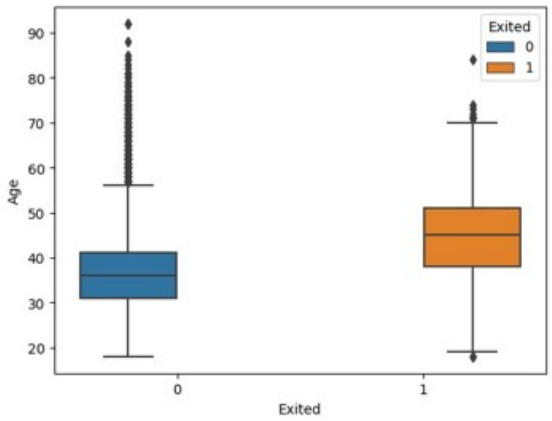


Hình 3: Phân bổ địa lý

3.2.3 Phân bổ về độ tuổi:

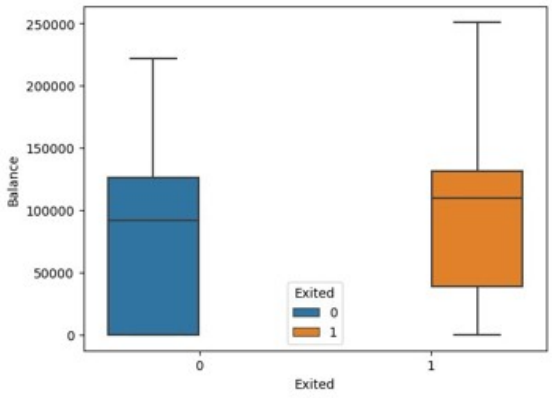
-Độ tuổi ở lại sử dụng dịch vụ của ngân hàng từ 31-41 tuổi.
-Độ tuổi tời bỏ dịch vụ của ngân hàng từ 41-50 tuổi.
=> Đây cũng có thể là một trường dữ liệu quan

trọng trong việc phân loại.



Hình 4: Phân bổ độ tuổi

3.2.4 Phân bổ về số dư tài khoản:
- Có một lượng lớn số dư tài khoản đang rút khỏi ngân hàng.
=> Việc này có thể gây ra những ảnh hưởng trong việc phát triển lâu dài của ngân hàng.



Hình 5: Phân bổ số dư tài khoản

IV. Approach

4.1 Chuẩn hóa dữ liệu:
Để chuẩn hóa dữ liệu của bài toán trên ,chúng ta sử dụng **MinMaxScaler**.

Chuẩn hóa dữ liệu

MinMaxScaler

4.2 Xử lý mất cân bằng:
Như đã đề cập trước đó, bộ dữ liệu **Bank Customer Churn Prediction** có sự mất cân bằng dữ liệu , vì vậy ta cần xử lý sự mất cân bằng.

Sử dụng **SMOTE** để xử lý mất cân bằng.

Xử lý mất cân bằng dữ liệu

SMOTE

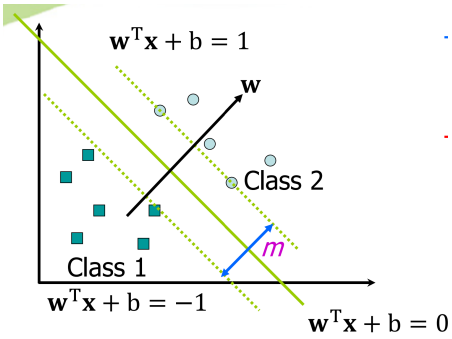
4.3 Phương pháp SVM:
SVM (Support Vector Machine) là một thuật toán học máy có giám sát được sử dụng rất phổ biến ngày nay trong các bài toán phân lớp (classification) hay hồi qui (Regression).

SVM là mô hình xây dựng một siêu phẳng hoặc một tập hợp các siêu phẳng trong một không gian nhiều chiều hoặc vô hạn chiều, có thể được sử dụng cho phân loại, hồi quy, hoặc các nhiệm vụ khác.

Để phân loại tốt nhất thì phải xác định siêu phẳng (Optimal hyperplane) nằm ở càng xa các điểm dữ liệu của tất cả các lớp(Hàm lề) càng tốt, vì nói chung lề càng lớn thì sai số tổng quát hóa của thuật toán phân loại càng bé. Muốn các điểm dữ liệu có thể được chia tách một cách tuyến tính, thì bạn phải cần chọn hai siêu phẳng của lề sao cho không có điểm nào ở giữa chúng và khoảng cách giữa chúng là tối đa.

Trong nhiều trường hợp, không thể phân chia các lớp dữ liệu một cách tuyến tính trong một không gian ban đầu được dùng để mô tả một vấn đề.Vì vậy, nhiều khi cần phải ánh xạ các điểm dữ liệu

trong không gian ban đầu vào một không gian mới nhiều chiều hơn, để việc phân tách chúng trở nên dễ dàng hơn trong không gian mới.



Hình 6: Mô tả thuật toán SVM

Trong đó:

- x là vector
- w là vector pháp tuyến của siêu phẳng
- b là Scaler value

Công thức cho khoảng cách từ một điểm $\mathbf{x}_i$ tới siêu phẳng $\mathbf{w}^T\mathbf{x} + b = 0$ là:

$$r = \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

Margin m của siêu phẳng là khoảng cách giữa các support vector.

4.4 Phương pháp Decision Tree:

Decision Tree là một bộ phân loại có cấu trúc dạng cây

- Nút quyết định: xác định một thử nghiệm trên một thuộc tính duy nhất.
- Nút lá: chỉ ra giá trị của thuộc tính mục tiêu.
- Cạnh: phân chia một thuộc tính.
- Đường đi: một kết hợp của các thử nghiệm để đưa ra quyết định cuối cùng.

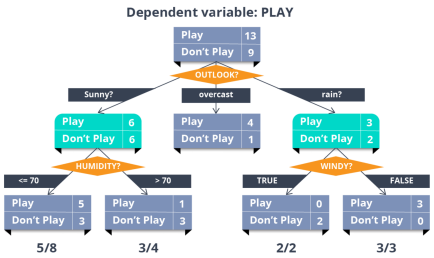
Decision Tree phân loại các trường hợp hoặc ví dụ bằng cách bắt đầu từ gốc của cây và di chuyển qua cây cho đến khi đến một nút lá.

Điểm mạnh của Decision Tree:

- Decision Tree dễ hiểu.
- Decision Tree dễ giải thích. Kết quả của nó là tập các luật.
- Decision Tree có thể xử lý cả các dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại.
- Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết.
- Decision Tree có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn.
- Dễ trực quan hóa.

Điểm yếu của Decision Tree:

- Xác xuất xảy ra quá khớp cao.
- Độ chính xác của việc dự báo thấp so với các thuật toán máy học khác.
- Độ đo Information gain gặp khó khăn với dữ liệu có miền giá trị là dữ liệu phân loại.
- Việc tính toán trở nên phức tạp nếu biến phụ thuộc có nhiều lớp (nhiều nhãn).



Hình 7: Mô tả thuật toán Decision Tree

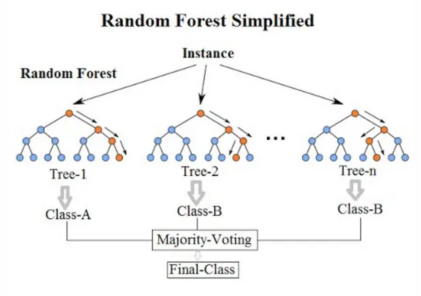
4.5 Phương pháp Random Forest:

Thuật ngữ "Random Forest" lần đầu tiên được đề xuất bởi Ho (1995). Sau đó, Breiman (2001) đã tiếp tục nghiên cứu và mở rộng thuật toán như hiện nay. "Random Forest là một bộ phân loại chứa một số cây quyết định trên các tập con khác nhau của tập dữ liệu đã cho và lấy giá trị trung bình để cải thiện độ chính xác dự đoán của tập dữ liệu đó".

Theo Yeşilkanat (2020), Random Forest là vượt trội so với các phương pháp học máy khác. Random Forest có thể xử lý bài toán hồi quy và phân loại với mức độ chính xác cao. Ngoài ra, nó còn đánh giá được mức độ quan trọng của các thuộc tính đóng góp vào mô hình.

Một nhược điểm của Random Forest là tốn thời gian vì phải xử lý dữ liệu cho từng cây đơn lẻ, đồng

thời cũng cần nhiều tài nguyên để lưu trữ các dữ liệu đó.



Nguồn: Niculescu và Lam (2019)

Hình 8: Mô tả thuật toán Random Forest

4.6 Đánh giá hiệu quả của các phương pháp phân loại:

Trong bài toán phân loại, chỉ số đánh giá Accuracy là tỷ lệ số quan sát được phân loại đúng trên tổng số quan sát. Tuy nhiên, để thấy rõ hơn các quan sát được phânloại đúng sai như thế nào, thường sử dụng các chỉ số chi tiết trong ma trận nhầm lẫn (Confusion matrix).

Trong những bài toán này, lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive (P), lớp còn lại được gọi là Negative (N). Trong bài toán phân loại khách hàng rời bỏ và khách hàng trung thành thì khách hàng rời bỏ chính là Positive, còn khách hàng trung thành là Negative. Từ đó ta định nghĩa True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) tạo thành ma trận nhầm lẫn chưa chuẩn hoá.

Với bài toán phân loại trong đó tập dữ liệu của các lớp có sự chênh lệch lớn, một số phép đo hiệu suất thường được sử dụng là Precision và Recall.

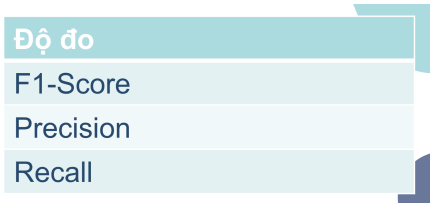
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Ngoài ra, để đo chất lượng của bộ phân lớp dựa vào cả Precision và Recall thường dùng F Score, chính là F1 Score.

$$\text{F}_1 \text{ Score} = \frac{2.Recall.Precision}{Recall + Precision}$$

Hình 9: Công thức



Hình 10: Các độ đo sử dụng

V. Results

5.1 Kết quả:

- Để thực hiện , ta chia ngẫu nhiên dữ liệu thành hai phần, bao gồm tập huấn luyện và tập thử nghiệm với tỉ lệ 80:20.
- Tập huấn luyện được dùng để ước lượng mô hình, còn tập thử nghiệm được dùng để tính các chỉ số đánh giá hiệu quả của mô hình.Các tiêu chí đánh giá hiệu quả phân loại của các mô hình trên tập thử nghiệm gồm có: Precision (PPV),Recall,F-Score (kết quả thể hiện trong bảng).

Remove Outlier	Imbalance	Scaler	Model	Pre_0	Recall_0	F1_0	Pre_1	Recall_1	F1_1	F1_Score
	None	MinMaxScaler	SVM	84	98	91	83	29	43	80.61
	None	MinMaxScaler	Decision Tree	87	83	85	46	54	50	80.6
	None	MinMaxScaler	Random Forest	87	96	91	76	47	58	84.5
	SMOTE	MinMaxScaler	SVM	89	86	88	55	64	59	81.9
	SMOTE	MinMaxScaler	Decision Tree	88	85	86	50	57	53	79.6
	SMOTE	MinMaxScaler	Random Forest	90	89	89	59	62	61	83.3
✓	SMOTE	MinMaxScaler	SVM	90	83	86	51	67	58	80.4
✓	SMOTE	MinMaxScaler	Decision Tree	88	79	83	43	61	50	76.3
✓	SMOTE	MinMaxScaler	Random Forest	90	83	86	50	63	56	82.9

Hình 11: Kết quả mô hình

5.2 Đánh giá kết quả:

- Việc xử lý mất cân bằng dữ liệu (class imbalance) có ảnh hưởng đến kết quả của mô hình. Cụ thể áp dụng SMOTE giúp cải thiện đáng kể F1-score cho nhãn 1 (lớp thiểu số).
- Không xử lý Outlier cho kết quả cao hơn nguyên nhân có thể do việc Outlier không phải là nhiễu mà chứa thông tin giá trị.

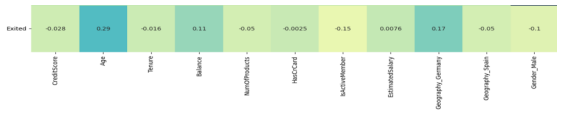
Nhận xét:

Trong 3 mô hình, Random Forest có kết quả tổng

quát tốt hơn so với 2 mô hình còn lại.

5.3 Phân tích lỗi:

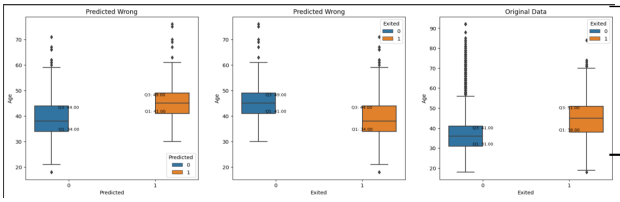
- Trường Age có tương quan mạnh đối với trường dự đoán Exited



- Trong bộ dữ liệu train thì độ tuổi chủ yếu rời bỏ là từ 41-51 tuổi ở lại từ 31-41 tuổi => Độ tuổi 41-51 sẽ là rời bỏ và 31-41 sẽ là không rời bỏ.
- Tuy nhiên trong tập test (của phần dự đoán sai)

- Độ tuổi rời bỏ chủ yếu lại từ 34-44 tuổi và nó dự đoán là nhãn không rời bỏ => Sai.
- Độ tuổi ở lại 41-49 tuổi và nó lại dự đoán là rời bỏ => Sai.

- Kiểu ứng với độ tuổi 41-51 nó sẽ thiên về rời bỏ hơn nên 41-49 nó sẽ dự đoán là rời bỏ.
- Bên cạnh đó thì sự mất cân bằng của 2 Nhãn cũng ảnh hưởng tới việc dự đoán. Vì Smote chỉ tạo ra các điểm dữ liệu ảo dựa trên các điểm dữ liệu đã có nên nó không có tính đa dạng .



VI. Conclusion

- Mô hình đã đạt được hiệu suất tốt hơn trong việc dự đoán sự rời bỏ của khách hàng sau khi thực hiện các bước tiền xử lý và điều chỉnh mô hình.
- Tuy nhiên độ chính xác vẫn chưa cao đối với nhãn tối thiểu.

Hướng cải tiến:

- Thu thập nhiều dữ liệu hơn về nhãn thiểu số.
- Xem lại các đặc trưng quan trọng của mô hình.
- Sử dụng các mô hình phức tạp hơn để huấn luyện.

- Với kết quả như trên, có thể gợi ý cho nhà quản lý ngân hàng một số chính sách giữ chân khách hàng như sau:

- Thứ nhất, áp dụng phương pháp Random Forest để dự báo những khách hàng có khả năng rời bỏ dịch vụ thẻ tín dụng. Sau đó tập trung chăm sóc nhóm khách hàng này để giữ chân họ thay vì chăm sóc toàn bộ khách hàng gây tốn kém và không cần thiết. Đối với nhóm khách hàng này, nhà quản lý nên thay đổi dịch vụ hiện tại hoặc cung cấp thêm cho họ những dịch vụ mới.
- Thứ hai là quảng bá nhiều sản phẩm khác của ngân hàng để những khách hàng này quan tâm và tham gia, từ đó tăng số sản phẩm mà khách hàng nắm giữ. Ngân hàng cũng nên hỗ trợ nhiều dịch vụ ưu đãi cho nhóm khách hàng có nguy cơ rời bỏ này.

VII. Phân công công việc

Tên	Công việc
Nguyễn Thành Huy	Tìm hiểu bài toán, Phân tích dữ liệu, Xây dựng mô hình SVM, Silde, Viết báo cáo, Thuyết trình.
Hoàng Ngọc Hồng	Tìm hiểu bài toán, Phân tích dữ liệu, Xây dựng mô hình Random Forest và Decision Tree, Tinh chỉnh và phân tích lỗi mô hình, Thuyết trình .

Bảng I: Công việc

Tài liệu

[1] https://viblo.asia/p/decision-tree-Do754bbBZM6
[2] https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB
[3] https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz