

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP
XÂY DỰNG ỨNG DỤNG PHÁT HIỆN GIAN
LẬN THẺ TÍN DỤNG BẰNG KỸ THUẬT
HỌC MÁY

SINH VIÊN THỰC HIỆN : PHÙNG HẢI TRUNG
MÃ SINH VIÊN : 1451020246
KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



PHÙNG HẢI TRUNG
XÂY DỰNG ỨNG DỤNG PHÁT HIỆN GIAN
LẬN THẺ TÍN DỤNG BẰNG KỸ THUẬT
HỌC MÁY

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN: Th.S LÊ TRUNG HIẾU
TG. NGUYỄN THÁI KHÁNH

HÀ NỘI - 2024

NHẬN XÉT

LỜI CAM ĐOAN

Em Phùng Hải Trung, dưới đây xin ký tên xác nhận rằng toàn bộ nội dung báo cáo và mã nguồn trong xây dựng đồ án tốt nghiệp đề tài: “Xây dựng ứng dụng phát hiện gian lận thẻ tín dụng bằng kỹ thuật học máy” là kết quả nghiên cứu và làm việc độc lập của em.

Em cũng xin cam kết rằng không có sự sao chép hay làm dụng tài nguyên từ nguồn nào khác mà không được chỉ rõ. Mọi tài liệu tham khảo của bên thứ ba đều được em trích dẫn đầy đủ.

Trong quá trình xây dựng bài toán, em đã tự tin với sự hiểu biết và kỹ năng của mình. Khi thử sức với những điều mới trong học tập thì mọi quyết định và lựa chọn thiết kế đều dựa trên kiến thức và kinh nghiệm của em trong lĩnh vực này.

Hy vọng rằng chương trình này sẽ mang lại giá trị và tiện ích cho người sử dụng và là một bước tiến quan trọng trong sự phát triển cá nhân và chuyên môn của em.

Ngày 24, tháng 5 năm 2024

Sinh viên ký tên

LỜI MỞ ĐẦU

Trong một thế giới đang liên tục thay đổi, các thời đại, sự phát triển cũng không ngừng thay đổi. Thời đại 4.0 đã đánh dấu một bước đột phá về công nghệ và chuyển đổi số. Chúng ta đang ở thế kỷ XXI và đang tiến vào một thời kỳ mới: thời đại 5.0.

Trong thời đại như thế, việc giao dịch tiền mặt ngày càng ít dần, và giao dịch không dùng tiền mặt đang ngày càng phát triển. Nó giúp con người giao dịch thuận tiện và nhanh chóng hơn. Từ đó, thẻ tín dụng ngày càng phổ biến vì sự tiện lợi của nó trong giao dịch không dùng tiền mặt. Nên là nó trở thành miếng bánh cho kẻ xấu lợi dụng nó cho các mục đích giao dịch phi pháp và chiếm đoạt tài sản. Vì vấn đề như thế, em đã triển khai dự án: “Xây dựng ứng dụng phát hiện gian lận thẻ tín dụng bằng kỹ thuật học máy” cũng như là đề án tốt nghiệp của mình.

Đề án này không chỉ là kết hợp của những kiến thức mới, kỹ năng và kỹ thuật đã rèn luyện trong suốt 4 năm học tại Đại học Đại Nam. Đây cũng là mong muốn và nỗ lực để tạo ra được giá trị thực tế cũng như thể hiện tầm nhìn với tương lai.

Trong bản báo cáo đề án này, em sẽ trình bày quá trình nghiên cứu, phân tích, xây dựng, đánh giá, cải thiện bài toán cũng như là những thách thức trong quá trình phát triển. Đồng thời, em cũng chia sẻ về những dạng bài toán, công cụ, công nghệ sử dụng trong đề án và những kiến thức về phát triển.

LỜI CẢM ƠN

Sau thời gian học tập tại trường, sinh viên được hệ thống lại toàn bộ lý thuyết chuyên ngành và được tham gia kiến tập một số khâu nghiệp vụ cơ bản của các kiến thức cơ bản của các kiến thức đã được học. Được sự cho phép của khoa Công nghệ thông tin trường Đại học Đại Nam, được sự quan tâm, chỉ đạo, hướng dẫn của thầy cô, em đã thực hiện đồ án tốt nghiệp của mình. Khoảng thời gian làm đồ án này tuy khá là ít nhưng cũng đủ để em học hỏi, có được những trải nghiệm và những yêu cầu, kiến thức trong quá trình làm.

Vì bài đồ án được thực hiện trong thời gian ngắn nên không thể tránh được việc thiếu kiến thức chuyên môn và những sai sót. Đồng thời, bài đồ án là kết quả của việc tổng hợp, tìm hiểu, nghiên cứu từ việc xây dựng một bài toán học máy, những rút ra từ quá trình và kinh nghiệm học tập của em. Em mong có những ý kiến đóng góp của thầy, cô để bài báo cáo và bản thân em được hoàn thiện hơn.

Qua bài báo cáo này, em xin chân thành cảm ơn thầy Lê Trung Hiếu và thầy trợ giảng Nguyễn Thái Khánh – giảng viên khoa công nghệ thông tin đã giúp đỡ và chỉ dẫn tận tình, tạo điều kiện để em hoàn thiện tốt học phần đồ án tốt nghiệp của mình.

Em xin chân thành cảm ơn!

MỤC LỤC

NHẬN XÉT	i
LỜI CAM ĐOAN	ii
LỜI MỞ ĐẦU	iii
LỜI CẢM ƠN.....	iv
MỤC LỤC	v
MỤC LỤC HÌNH ẢNH	viii
DANH MỤC BẢNG	ix
DANH MỤC KÝ HIỆU VÀ TỪ NGỮ VIẾT TẮT	x
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 Giới thiệu đề tài	1
1.2 Lý do chọn đề tài	1
1.3 Mục tiêu của đề tài	2
1.4 Phương pháp tiếp cận	2
1.5 Phạm Vi Nghiên Cứu	3
1.6 Kết cấu đồ án	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CHUNG	5
2.1 Giới thiệu các thuật toán học máy và công cụ sử dụng.....	5
2.1.1 Các thuật toán học máy phân loại phân lớp	5
2.1.2 Thư viện Pandas	21
2.1.3 Google Colab và thư viện scikit-learn.....	21
2.2 Vấn đề phát hiện gian lận thẻ tín dụng trong thời đại hiện nay.....	25
2.2.1 Về vấn đề gian lận thẻ tín dụng	25
2.2.2 Tầm quan trọng của phát hiện gian lận	26
2.2.3 Các phương pháp truyền thống phát hiện gian lận.....	27

2.2.4 Ưu điểm và hạn chế của phương pháp truyền thống	28
2.2.5 Sự ra đời của việc ứng dụng học máy trong phát hiện gian lận thẻ tín dụng ...	29
2.3 Kết luận chương	30
CHƯƠNG 3: THIẾT KẾ, CẢI THIẾN VÀ ĐÁNH GIÁ KẾT QUẢ.....	31
3.1 Mô hình chung của bài toán	31
3.2 Thu thập dữ liệu.....	33
3.2.1 Giới thiệu chung về các tập dữ liệu sử dụng	33
3.2.2 Các tập dữ liệu công bố	33
3.2.3 Tổng kết bộ dữ liệu.....	34
3.3 Tiền xử lý dữ liệu	34
3.3.1 Phân đoạn dữ liệu	34
3.3.2 Kiểm tra và làm sạch dữ liệu	35
3.3.3 Kết quả dữ liệu sau làm sạch	35
3.4 Huấn luyện mô hình	36
3.4.1 Chuẩn hóa dữ liệu.....	36
3.4.2 Chia dữ liệu thành tập huấn luyện và tập kiểm tra	37
3.4.3 Mô hình phân lớp sử dụng.....	39
3.5 Triển khai mô hình và huấn luyện.....	43
3.5.1 Môi trường cài đặt	43
3.5.2 Phương pháp xây dựng bài toán	44
3.5.3 Quy trình huấn luyện mô hình bài toán.....	44
3.6 Phương pháp đánh giá kết quả	46
3.6.1 Cách đo độ đánh giá	46
3.6.2 Phương pháp đánh giá	47
3.7 Kết quả thực nghiệm.....	48
3.7.1 Kết quả thực nghiệm trên thuật toán KNN.....	48

3.7.2 Kết quả thực nghiệm trên thuật toán SVM.....	51
3.7.3 Kết quả thực nghiệm trên thuật toán Random Forest.....	53
3.7.4 Kết quả đánh giá chung	55
3.8 Kết luận chương	55
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	56
TÀI LIỆU THAM KHẢO	58
PHỤ LỤC	59

MỤC LỤC HÌNH ẢNH

Hình 2. 1 KNN	5
Hình 2. 2 Random Forests	9
Hình 2. 3 nguyên tắc hoạt động của random forest	10
Hình 2. 4 Hyper Lane	13
Hình 2. 5 đường hyper plane	14
Hình 2. 6 Scenario-3	15
Hình 2. 7 Serinato 4.....	16
Hình 2. 8 Serinato 4-2	16
Hình 2. 9 Dữ liệu biến đổi	17
Hình 2. 10 Hyper lan biến đổi	18
Hình 3. 1 Quy trình phát hiện	31
Hình 3. 2 Kiểm tra dữ liệu trống	35
Hình 3. 3 Kết quả sau kiểm tra	35
Hình 3. 4 Dữ liệu class 0	36
Hình 3. 5 Dữ liệu class 1	37
Hình 3. 6 Trang chủ colab	38
Hình 3. 7 Mở sổ tay mới.....	38
Hình 3. 8 Khai báo dataset.....	38
Hình 3. 9 Tách các biến và nhãn data.....	39
Hình 3. 10 Tỷ lệ mẫu.....	39
Hình 3. 11 Chia data.....	39
Hình 3. 12 Kích thước sau khi chia	39
Hình 3. 13 Mô tả KNN	40
Hình 3. 14 Siêu phẳng SVM.....	41
Hình 3. 15 Vector hỗ trợ SVM	42
Hình 3. 16 Trang chủ colab	43
Hình 3. 17 Kết nối gg driver.....	43
Hình 3. 18 Ma trận nhầm lẫn KNN	50
Hình 3. 19 Ma trận nhầm lẫn SVM	52
Hình 3. 20 Ma trận nhầm lẫn random forest.....	54

DANH MỤC BẢNG

Bảng 3. 1 Mô tả các trường	33
Bảng 3. 2 Đánh giá mô hình trên KNN	48
Bảng 3. 3 Đánh giá mô hình trên SVM	51
Bảng 3. 4 Đánh giá mô hình random forest.....	53

DANH MỤC KÝ HIỆU VÀ TỪ NGỮ VIẾT TẮT

STT	Ký hiệu viết tắt	Từ viết tắt đầy đủ
1	Euclid	Euclidean
2	FN	False Negatives
3	FP	False Positives
4	KNN	K-nearest neighbors
5	ML	Machine Learning
6	SVM	Support Vector Machine
7	TN	True Negatives
8	TP	True Positives

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu đề tài

Trong thời đại công nghệ 4.0 hiện nay, việc giao dịch bằng thẻ tín dụng ngày càng phổ biến, vì độ tiện lợi mà nó mang lại. Chính vì sự tiện lợi đó, nên điều đối tượng xấu dùng nó để thực hiện các giao dịch gian lận thẻ tín dụng, gây hoang mang cho những chủ sở hữu thẻ, gây ra mất lòng tin, ảnh hưởng đến danh tiếng trong xã hội. Ngoài ra, còn ảnh hưởng đến kinh tế của quốc gia. Và gian lận ngày nay cũng ngày càng tinh vi do những đối tượng xấu thường hoạt động theo những băng nhóm khép kín, thủ đoạn ngày càng tinh vi hơn. Nhưng nhờ vào phân tích dữ liệu thời gian thực và các kỹ thuật học máy tiên tiến, việc xác định các giao dịch gian lận một cách nhanh chóng và chính xác, giúp giảm thiểu thiệt hại khách hàng và doanh nghiệp.

Đề tài: “xây dựng ứng dụng phát hiện gian lận thẻ tín dụng bằng kỹ thuật học máy” nhằm đáp ứng nhu cầu và vấn đề này. Đề tài hướng đến việc xây dựng và đánh giá các phương pháp phát hiện được gian lận thẻ tín dụng khác nhau và tìm ra những vấn đề và cải thiện, để cho ra kết quả chính xác và hiệu quả nhất.

Đặc biệt, đề tài hướng tới việc phát hiện được gian lận thẻ tín dụng ở mức chính xác cao nhất. Bằng cách này, em tìm ra được một phương pháp học máy có thể cho kết quả phát hiện gian lận ở mức tốt nhất.

1.2 Lý do chọn đề tài

Gian lận thẻ tín dụng là một vấn đề dần trở nên đáng lo ngại, và trong tương lai, với sự phát triển như vũ bão của công nghệ hiện nay, và xu hướng thanh toán không dùng tiền mặt và mua sắm online ngày một tăng ở thế hệ người dùng đi theo thời đại công nghệ số hiện nay. Theo báo vietnamnet: công thông tin cảnh báo lừa đảo của Bộ thông tin và truyền thông cho biết: “17.400 phản ánh lừa đảo trực tuyến hướng đến người dùng Internet Việt Nam, gây thiệt hại hơn 300 tỷ đồng vào năm 2023” [12]. Chính vì thiệt hại lớn cho phía ngân hàng, tổ chức tài chính, tín dụng. Nên cần có giải pháp hiệu quả để phát hiện và ngăn chặn gian lận.

Học máy (machine learning) là một lĩnh vực công nghệ hiện đại đang phát triển lớn mạnh, tiềm năng mang lại của nó quan trọng trong xử lý vấn đề phức tạp. Vì học máy có

thể phát hiện và ngăn chặn sớm hành vi gian lận. Vì góp phần giải quyết vấn đề gian lận thẻ tín dụng – là vấn đề nan giải trong xã hội công nghệ hiện nay, và mang lại hiệu quả đột phá trong phát hiện gian lận thẻ tín dụng.

Phát hiện và ngăn chặn gian lận thẻ tín dụng không chỉ mang lại lợi ích kinh tế mà còn có ý nghĩa xã hội quan trọng. Bằng cách giảm thiểu các rủi ro và thiệt hại tài chính, các giải pháp học máy giúp bảo vệ quyền lợi của khách hàng và củng cố lòng tin vào hệ thống tài chính. Ngoài ra, còn thể hiện sự nỗ lực của các tổ chức tài chính trong việc đảm bảo an ninh và minh bạch trong hoạt động kinh doanh.

Đề tài này không chỉ mang tính nghiên cứu mà còn có khả năng ứng dụng thực tiễn cao. Các mô hình học máy được phát triển từ đề tài có thể được triển khai trực tiếp vào hệ thống quản lý giao dịch của các ngân hàng và công ty phát hành thẻ. Điều này giúp cải thiện hiệu suất phát hiện gian lận một cách tự động và liên tục, giảm thiểu sự can thiệp thủ công và tối ưu hóa quy trình quản lý rủi ro. Từ đó, đề tài này không chỉ có giá trị học thuật mà còn có tiềm năng ứng dụng thực tiễn lớn, mang lại lợi ích thiết thực cho ngành tài chính.

Với những lý do trên, việc chọn đề tài này có thể mang lại nhiều lợi ích cho xã hội, giúp giải quyết được vấn đề nan giải của các ngân hàng và tổ chức tài chính hiện nay, và giúp người dùng yên tâm với việc sử dụng thẻ tín dụng.

1.3 Mục tiêu của đề tài

Phân tích dữ liệu giao dịch thẻ tín dụng để tìm hiểu các đặc điểm của giao dịch gian lận.

Áp dụng và so sánh hiệu quả của các thuật toán học máy trong việc phát hiện gian lận.

Xây dựng mô hình dự báo có khả năng nhận diện giao dịch gian lận với độ chính xác cao.

1.4 Phương pháp tiếp cận

Thu thập dữ liệu: Sử dụng tập dữ liệu từ các nguồn uy tín như Kaggle hoặc các tổ chức tài chính có công khai dữ liệu giao dịch.

Xử lý dữ liệu: Bao gồm việc làm sạch dữ liệu, xử lý các giá trị thiếu, và chuẩn hóa dữ liệu để phù hợp với các mô hình học máy.

Lựa chọn đặc trưng: Sử dụng các kỹ thuật như PCA (Principal Component Analysis) hoặc Feature Importance từ mô hình học máy để xác định các đặc trưng quan trọng.

Xây dựng mô hình: Triển khai các thuật toán học máy và điều chỉnh tham số (hyperparameter tuning) để tối ưu hóa hiệu suất mô hình.

Đánh giá mô hình: Áp dụng k-fold cross-validation để đánh giá mô hình một cách khách quan và lựa chọn mô hình tốt nhất dựa trên các chỉ số đánh giá.

1.5 Phạm Vi Nghiên Cứu

Thu thập và xử lý dữ liệu: Sử dụng các tập dữ liệu công khai về giao dịch thẻ tín dụng, xử lý các giá trị thiếu và chuẩn hóa dữ liệu.

Lựa chọn đặc trưng (feature selection): Chọn ra các đặc trưng quan trọng giúp mô hình học máy nhận diện được giao dịch gian lận.

Xây dựng và huấn luyện mô hình: Sử dụng các thuật toán học máy như KNN, SVM, Random Forest.

Đánh giá mô hình: Sử dụng các chỉ số như độ chính xác (accuracy), độ nhạy (recall), và F1-score để đánh giá hiệu quả của các mô hình.

1.6 Kết cấu đồ án

Chương 1: Tổng quan đề tài: Chương này sẽ chủ yếu tập trung vào các vấn đề liên quan chung về đề tài như giới thiệu, lý do chọn đề tài, mục tiêu nghiên cứu, phương pháp nghiên cứu.

Chương 2: Cơ sở lý thuyết chung: Chương này sẽ giới thiệu cụ thể và chi tiết các công nghệ và thuật toán học máy cụ thể như KNN, SVM, Random Forest; các công nghệ sử dụng trong bài như Google colab, các thư viện quan trọng trong lập trình học máy như Pandas, scikit – learn. Sau đó là giới thiệu về các vấn đề liên quan đến gian lận thẻ tín dụng như: vấn đề, tầm quan trọng, các phương pháp phát hiện gian lận thẻ tín dụng, ưu điểm và hạn chế của các phương pháp truyền thống, và sự ra đời của học máy trong phát hiện gian lận thẻ tín dụng.

Chương 3: Thiết kế, cải thiện và đánh giá kết quả: Chương này đi sâu vào các vấn đề liên quan đến việc thiết kế bài toán như: thu thập dữ liệu, tiền xử lý dữ liệu, chuẩn hóa dữ liệu, chia dữ liệu. Và việc thao tác để xây dựng các bài toán học máy phát hiện gian lận

thể tín dụng. Cuối cùng, cho biết kết quả và dùng các phương pháp để đánh giá xem mô hình bài toán học máy nào cho ra kết quả tốt nhất trong phát hiện gian lận thẻ tín dụng.

Kết quả đạt được: Tóm tắt và đánh giá kết quả đồ án: bao gồm những thành công cũng như là khó khăn và thách thức đã gặp phải trong quá trình thực hiện đồ án, và hướng phát triển của đề tài.

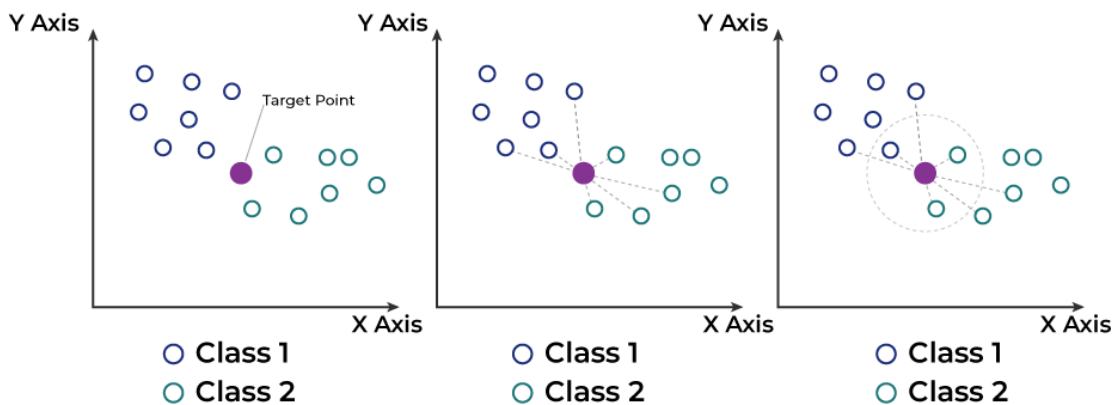
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CHUNG

Trong chương này, em giới thiệu về các mô hình học máy phân lớp sử dụng, và các lý thuyết liên quan đến học máy. Như đã giới thiệu trong đề án, chương 1 sẽ có 2 phần: phần 1 giới thiệu liên quan đến python, học máy, các thư viện sử dụng và mô hình phân lớp liên quan đến bài toán. Phần 2 liên quan đến những gì về thể tín dụng, vấn đề gian lận thẻ tín dụng, các phương pháp phát hiện gian lận thẻ tín dụng và ứng dụng học máy trong gian lận thẻ tín dụng hiện nay.

2.1 Giới thiệu các thuật toán học máy và công cụ sử dụng

2.1.1 Các thuật toán học máy phân loại phân lớp

2.1.1.1 K-nearest neighbor(KNN)



Hình 2. 1 KNN

(nguồn: <https://www.geeksforgeeks.org/k-nearest-neighbours/>) [2]

2.1.1.1.1 Khái niệm chung

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới (hay kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training

set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label. Chi tiết sẽ được nêu trong phần tiếp theo.

Trong bài toán Regression, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp $K=1$), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K -lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.

Khoảng cách trong không gian vector

Trong không gian một chiều, khoảng cách giữa hai điểm là trị tuyệt đối giữa hiệu giá trị của hai điểm đó. Trong không gian nhiều chiều, khoảng cách giữa hai điểm có thể được định nghĩa bằng nhiều hàm số khác nhau, trong đó độ dài đường thẳng nối hai điểm chỉ là một trường hợp đặc biệt trong đó. Nhiều thông tin bổ ích (cho Machine Learning) có thể được tìm thấy tại Norms (chuẩn) của vector trong tab Math. [6]

2.1.1.1.2 Các bước trong KNN

- Đo khoảng cách đến điểm dữ liệu gần đó, thường dùng Euclid.
- Xác định K là hàng xóm gần nó xem cái nào phổ biến nhất, K do người thiết kế chọn.
- Xác định xem điểm dữ liệu nào phổ biến nhất hoặc trung bình điểm dữ liệu quanh nó để đưa ra kết luận.

Ví dụ, có các bức thư spam biết là thư spam là thư không có tiêu đề, thư có nội dung có chữ quảng cáo, bình luận, ... ; KNN sẽ tìm các bức thư gần nhất với các nội dung tương tự. Nếu có 5 bức thư khác nhau, có 4 bức thư không có tiêu đề, KNN sẽ đưa ra dự đoán là bức thư có spam vì nó dựa trên hàng xóm gần nhất.

2.1.1.1.3 Đo khoảng cách Euclidean trong KNN:

Trong thuật toán KNN (K -nearest neighbors), khoảng cách Euclidean thường được sử dụng để đo độ tương đồng hoặc khoảng cách giữa các điểm dữ liệu. Công thức để tính

khoảng cách Euclidean giữa hai điểm trong không gian nhiều chiều (ví dụ: trong không gian hai chiều, ba chiều hoặc nhiều hơn) là:

Công thức tổng quát:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

Trong đó, p_i và q_i là các thành phần của các điểm dữ liệu p và q tương ứng trong không gian n chiều.

Lặp lại cho tất cả các điểm dữ liệu trong tập dữ liệu: Bạn tính khoảng cách Euclidean từ điểm thử nghiệm tới tất cả các điểm dữ liệu trong tập dữ liệu.

Chọn k điểm gần nhất: Sau khi tính toán khoảng cách Euclidean, bạn chọn k điểm gần nhất với điểm thử nghiệm.

Quyết định dự đoán hoặc phân loại: Dựa trên nhãn của k điểm gần nhất, bạn quyết định dự đoán hoặc phân loại cho điểm thử nghiệm. Ví dụ, trong phân loại, bạn có thể sử dụng đa số phiếu để quyết định lớp của điểm thử nghiệm

2.1.1.1.4 Ưu nhược điểm của KNN

* Ưu điểm:

Đơn giản và dễ hiểu: Thuật toán KNN có ý tưởng cơ bản và dễ dàng triển khai. Việc tính toán và dự đoán không phức tạp, giúp người dùng dễ dàng nắm bắt và áp dụng.

Không có giả định: KNN không yêu cầu bất kỳ giả định nào về phân phối dữ liệu. Điều này giúp thuật toán linh hoạt và có thể áp dụng cho nhiều bài toán khác nhau, đặc biệt là những bài toán có dữ liệu phi tuyến tính.

Hiệu quả với nhiều lớp phân loại: KNN hoạt động tốt trong các bài toán phân loại với nhiều lớp. Thuật toán có thể phân biệt nhiều nhóm dữ liệu khác nhau một cách hiệu quả.

Xử lý tốt nhiễu dữ liệu: KNN có khả năng chống nhiễu dữ liệu tương đối tốt. Thuật toán ít bị ảnh hưởng bởi các điểm dữ liệu ngoại lệ hoặc bị nhiễu.

Dễ dàng điều chỉnh: KNN cho phép người dùng điều chỉnh tham số " k " (số lượng hàng xóm gần nhất) để tối ưu hóa hiệu suất cho từng bài toán cụ thể.

* Nhược điểm:

Tính toán tốn kém: Khi tập dữ liệu lớn, việc tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm trong tập dữ liệu có thể tốn nhiều thời gian, ảnh hưởng đến hiệu suất của thuật toán.

Nhạy cảm với dữ liệu nhiễu: Khi giá trị của "k" nhỏ, KNN có thể nhạy cảm với dữ liệu nhiễu. Một vài điểm dữ liệu ngoại lệ có thể ảnh hưởng đáng kể đến kết quả dự đoán.

Kích thước dữ liệu ảnh hưởng: Hiệu suất của KNN phụ thuộc vào kích thước tập dữ liệu. Khi tập dữ liệu lớn, thuật toán có thể trở nên chậm và tốn kém về mặt tính toán.

Lựa chọn tham số "k" khó khăn: Việc lựa chọn giá trị "k" phù hợp cho KNN có thể ảnh hưởng đáng kể đến hiệu suất. Việc lựa chọn này thường đòi hỏi kinh nghiệm và thử nghiệm thực tế.

Khó giải thích kết quả: KNN là thuật toán học máy "hộp đen", nghĩa là nó không cung cấp thông tin chi tiết về cách đưa ra dự đoán. Việc giải thích kết quả của KNN có thể khó khăn cho người dùng.

2.1.1.2 Random Forest

Random Forests là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu.

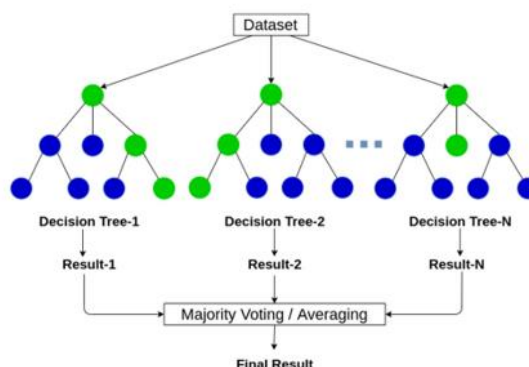
- Nguyên tắc hoạt động:

Ví dụ, như có một cuộc tranh luận xem là con gà có trước hay quả trứng có trước. Trong quá trình tranh luận đó sẽ có một nhóm người cho rằng con gà có trước, một nhóm cho rằng quả trứng có trước. Đầu tiên, nhóm người cho rằng gà có trước sẽ đưa ra các bằng chứng khoa học về con gà có trước như là gen di truyền, các loài chim cổ đại,... Sau đó nhóm cho rằng quả trứng có trước sẽ đưa ra một số bằng chứng về triết học, tiến hóa,... để cho rằng trứng có trước thì mới có gà.

Sau đó, nhóm người đó chia thành 2 nhóm là: một bên cho là gà có trước một bên cho là trứng có trước. Cuối cùng, để biết bên nào thắng thì do bình chọn của 2 bên xem bên nào có tỷ lệ bình chọn cao hơn sẽ thắng. Nếu bằng nhau, 2 bên sẽ hòa.

Về mặt kỹ thuật, nó là một phương pháp tổng hợp (dựa trên cách tiếp cận phân chia và chinh phục) của các cây quyết định được tạo ra trên một tập dữ liệu được chia ngẫu nhiên. Bộ sưu tập phân loại cây quyết định này còn được gọi là rừng. Cây quyết định riêng lẻ được tạo ra bằng cách sử dụng chỉ báo chọn thuộc tính như tăng thông tin, tỷ lệ tăng và chỉ số Gini cho từng thuộc tính. Mỗi cây phụ thuộc vào một mẫu ngẫu nhiên độc lập. Trong bài toán phân loại, mỗi phiếu bầu chọn và lớp phổ biến nhất được chọn là kết quả cuối cùng. Trong trường hợp hồi quy, mức trung bình của tất cả các kết quả đầu ra của cây được coi là kết quả cuối cùng. Nó đơn giản và mạnh mẽ hơn so với các thuật toán phân loại phi tuyến tính khác.

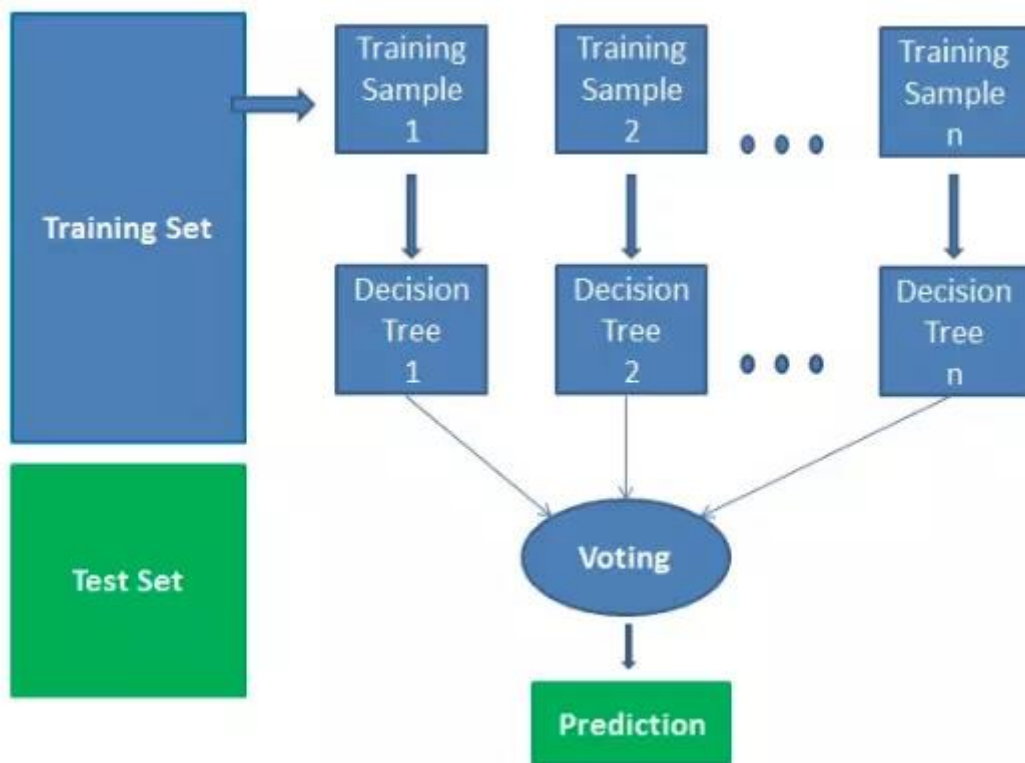
Random Forest



Hình 2. 2 Random Forests

(nguồn: random forest algorithm explained) [1]

2.1.1.2.1 Thuật toán hoạt động như thế nào?



Hình 2. 3 nguyên tắc hoạt động của random forest

(nguồn: Phân lớp bằng Random Forests trong Python0)

Nó hoạt động theo bốn bước:

Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.

Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây.

Hãy bỏ phiếu cho mỗi kết quả dự đoán.

Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng. [10]

2.1.1.2.2 Công thức tổng quát

- Chọn mẫu ngẫu nhiên: Với mỗi cây quyết định trong rừng, một mẫu con dữ liệu được lấy ngẫu nhiên từ tập dữ liệu huấn luyện. Điều này giúp đa dạng hóa các cây quyết định trong rừng.
- Xây dựng cây quyết định: Mỗi cây quyết định được xây dựng bằng cách chọn thuộc tính tốt nhất từ một tập hợp ngẫu nhiên của các thuộc tính và sử dụng thuộc tính đó để phân

chia tập dữ liệu thành các nhóm. Quá trình này được lặp lại đến khi cây quyết định đạt được một tiêu chí dừng.

- **Tính toán dự đoán:** Sau khi cây quyết định được xây dựng, mỗi mẫu dữ liệu trong tập kiểm tra được đưa qua mỗi cây quyết định và được dự đoán một nhãn. Trong trường hợp phân loại, phần đông phiếu bầu của tất cả các cây được sử dụng làm dự đoán cuối cùng. Trong trường hợp dự đoán, trung bình của các dự đoán từ tất cả các cây được sử dụng.
- **Tính toán độ chính xác:** Độ chính xác của Random Forest được tính bằng cách so sánh dự đoán của mô hình với nhãn thực tế trong tập kiểm tra hoặc tập dữ liệu không nhìn thấy.
- **Tinh chỉnh tham số (nếu cần):** Các tham số của mô hình như số cây, độ sâu của cây, và số lượng mẫu con có thể được tinh chỉnh bằng cách sử dụng kỹ thuật tinh chỉnh siêu tham số như Grid Search hoặc Random Search để cải thiện hiệu suất của mô hình.

2.1.1.2.3 Các tính năng quan trọng

Random forests cũng cung cấp một chỉ số lựa chọn tính năng tốt. Scikit-learn cung cấp thêm một biến với mô hình, cho thấy tầm quan trọng hoặc đóng góp tương đối của từng tính năng trong dự đoán. Nó tự động tính toán điểm liên quan của từng tính năng trong giai đoạn đào tạo. Sau đó, nó cân đối mức độ liên quan xuống sao cho tổng của tất cả các điểm là 1.

Điểm số này sẽ giúp bạn chọn các tính năng quan trọng nhất và thả các tính năng quan trọng nhất để xây dựng mô hình.

Random forests sử dụng tầm quan trọng của gini hoặc giảm tạp chất trung bình (MDI) để tính toán tầm quan trọng của từng tính năng. Gini tầm quan trọng còn được gọi là tổng giảm trong tạp chất nút. Đây là mức độ phù hợp hoặc độ chính xác của mô hình giảm khi bạn thả biến. Độ lớn càng lớn thì biến số càng có ý nghĩa. Ở đây, giảm trung bình là một tham số quan trọng cho việc lựa chọn biến. Chỉ số Gini có thể mô tả sức mạnh giải thích tổng thể của các biến. Random Forests và cây quyết định Random Forests là một tập hợp của nhiều cây quyết định. Cây quyết định sâu có thể bị ảnh hưởng quá mức, nhưng Random forests ngăn cản việc lấp đầy bằng cách tạo cây trên các tập con ngẫu nhiên. Cây

quyết định nhanh hơn tính toán. Random forests khó giải thích, trong khi cây quyết định có thể diễn giải dễ dàng và có thể chuyển đổi thành quy tắc. [6]

2.1.1.2.4 Ưu nhược điểm

* Ưu điểm

Độ chính xác cao:

Random Forests thường có độ chính xác cao nhờ việc kết hợp nhiều cây quyết định (decision trees) và giảm thiểu hiện tượng overfitting so với các cây quyết định đơn lẻ.

Khả năng xử lý dữ liệu đa dạng:

Random Forests có thể xử lý cả dữ liệu phân loại và dữ liệu liên tục, cũng như làm việc tốt với dữ liệu thiếu hụt.

Giảm thiểu overfitting: Bằng cách kết hợp nhiều cây quyết định với nhau, Random Forests giảm thiểu nguy cơ overfitting, vì nó dựa trên kết quả của nhiều cây thay vì chỉ một cây đơn lẻ.

Tự động ước lượng tầm quan trọng của biến: Random Forests cung cấp thông tin về tầm quan trọng của từng biến đầu vào trong việc dự đoán kết quả, giúp ích trong việc hiểu rõ hơn về dữ liệu và mô hình.

Khả năng mở rộng tốt: Random Forests có thể mở rộng tốt cho cả dữ liệu lớn và có thể song song hóa quá trình huấn luyện và dự đoán.

Khả năng chống nhiễu tốt: Do sự tổng hợp của nhiều cây, Random Forests có khả năng chống nhiễu cao hơn, giúp mô hình ổn định hơn khi dữ liệu đầu vào có nhiễu.

* Nhược điểm:

Thời gian và tài nguyên tính toán: Việc xây dựng nhiều cây quyết định đòi hỏi thời gian và tài nguyên tính toán lớn, đặc biệt khi số lượng cây và dữ liệu lớn.

Giải thích kết quả: Random Forests là một mô hình “hộp đen” (black-box), khó giải thích rõ ràng từng quyết định như với cây quyết định đơn lẻ.

Tối ưu hóa mô hình: Việc điều chỉnh các tham số của Random Forests (như số lượng cây, độ sâu của cây,...) có thể phức tạp và tốn thời gian.

Bộ nhớ: Random Forests có thể tiêu tốn nhiều bộ nhớ vì lưu trữ thông tin của nhiều cây quyết định.

Không tối ưu cho dữ liệu cao chiều: Khi dữ liệu có rất nhiều biến đầu vào (high-dimensional data), Random Forests có thể không hiệu quả như một số thuật toán khác như SVM hoặc các kỹ thuật giảm chiều (dimensionality reduction techniques).

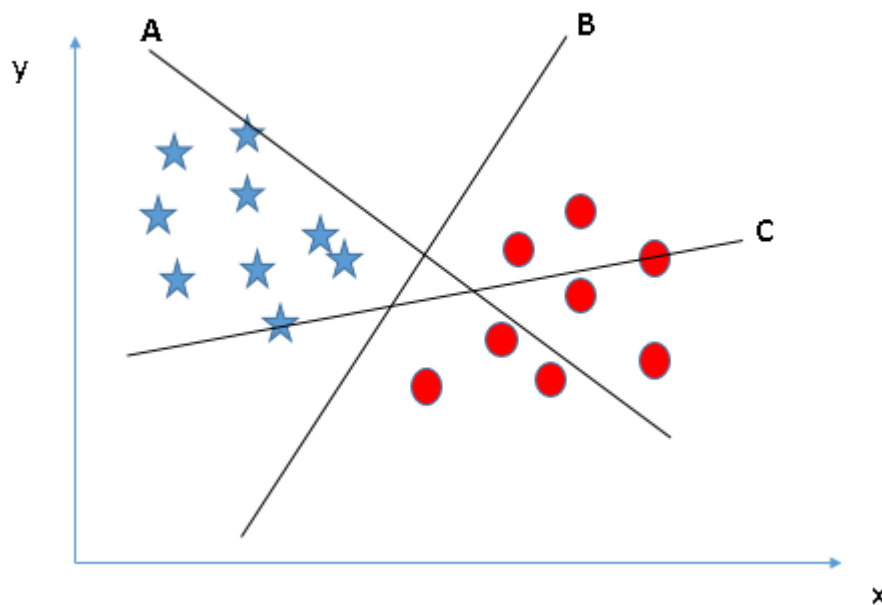
2.1.1.3 SVM

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc dự đoán. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt. [8]

2.1.1.3.1 SVM làm việc như thế nào?

Identify the right hyper-plane (Scenario-1):

Ở đây, có 3 đường hyper-plane (A, B and C). Bây giờ đường nào là hyper-plane đúng cho nhóm ngôi sao và hình tròn.



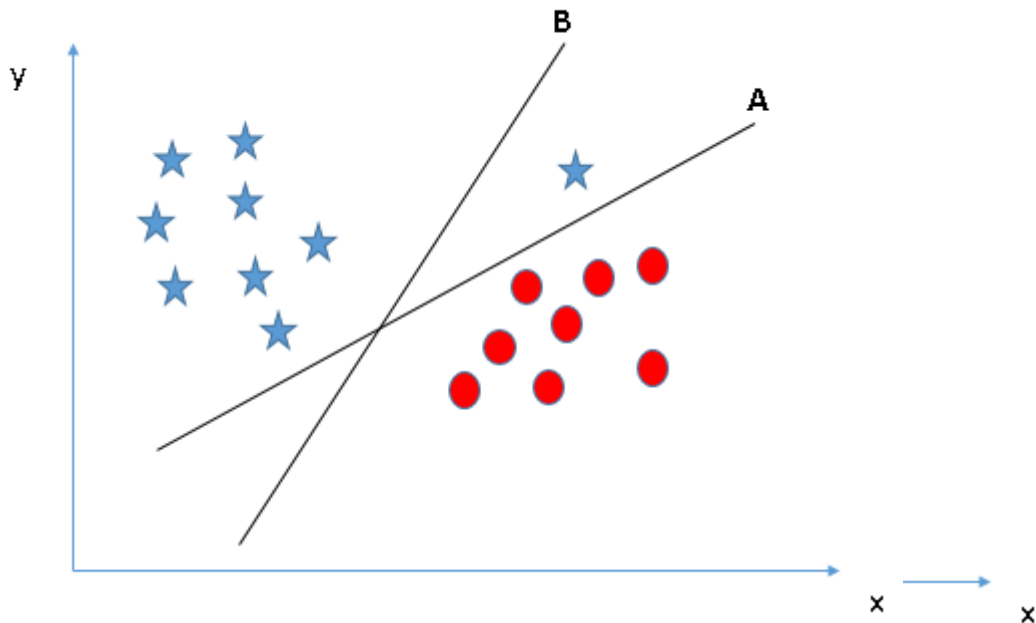
Hình 2. 4 Hyper Lane

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Quy tắc số một để chọn 1 hyper-plane, chọn một hyper-plane để phân chia hai lớp tốt nhất. Trong ví dụ này chính là đường B.

Identify the right hyper-plane (Scenario-2):

Ở đây chúng ta cũng có 3 đường hyper-plane (A,B và C), theo quy tắc số 1, chúng đều thỏa mãn.

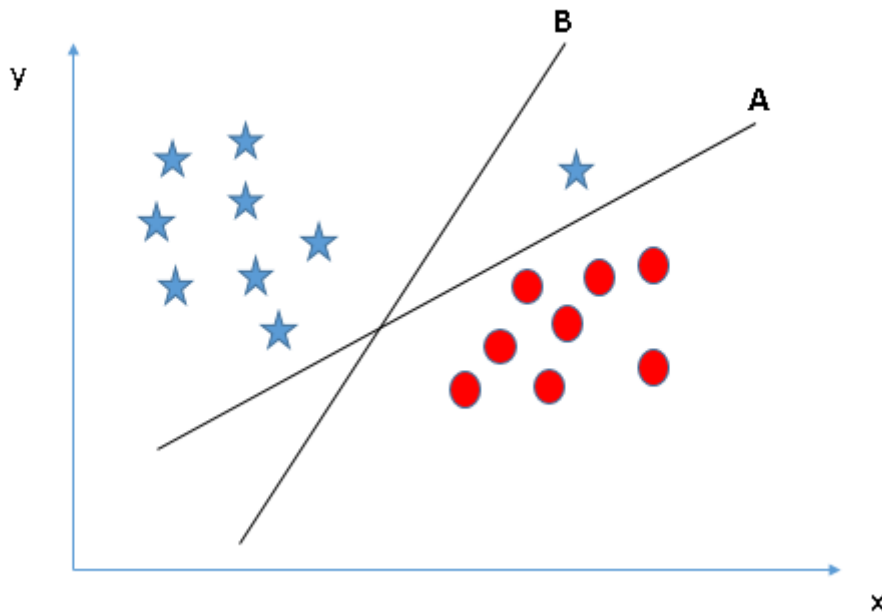


Hình 2. 5 đường hyper plane

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Quy tắc thứ hai chính là xác định khoảng cách lớn nhất từ điều gần nhất của một lớp nào đó đến đường hyper-plane. Khoảng cách này được gọi là "Margin", Hãy nhìn hình bên dưới, trong đây có thể nhìn thấy khoảng cách margin lớn nhất đây là đường C. Cần nhớ nếu chọn làm hyper-plane có margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu.

Identify the right hyper-plane (Scenario-3):



Hình 2. 6 Scenario-3

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Sử dụng các nguyên tắc đã nêu trên để chọn ra hyper-plane cho trường hợp sau:

Có thể có một vài người sẽ chọn đường B bởi vì nó có margin cao hơn đường A, nhưng đây sẽ không đúng bởi vì nguyên tắc đầu tiên sẽ là nguyên tắc số 1, chúng ta cần chọn hyper-plane để phân chia các lớp thành riêng biệt. Vì vậy đường A mới là lựa chọn chính xác.

Can we classify two classes (Scenario-4)?

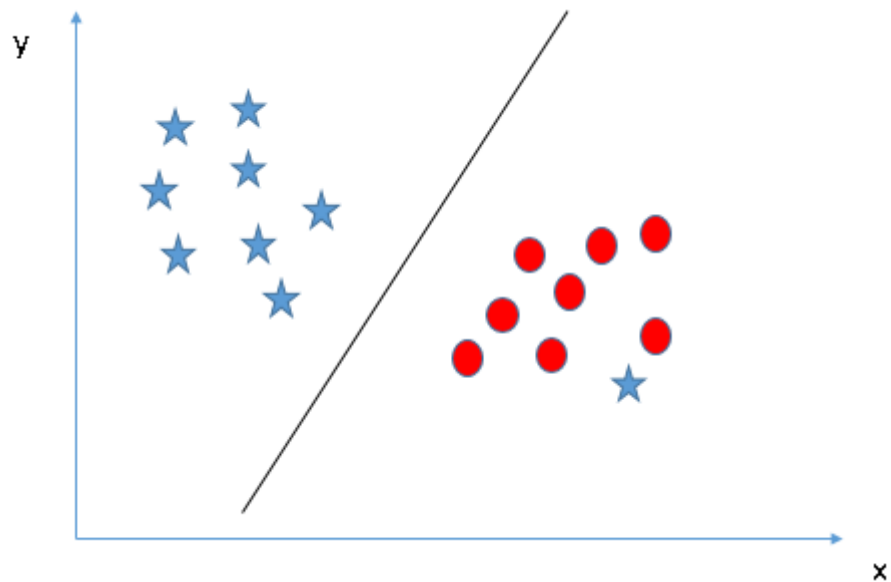
Tiếp theo hãy xem hình bên dưới, không thể chia thành hai lớp riêng biệt với 1 đường thẳng, để tạo 1 phần chỉ có các ngôi sao và một vùng chỉ chứa các điểm tròn.



Hình 2. 7 Serinato 4

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Ở đây sẽ chấp nhận, một ngôi sao ở bên ngoài cuối được xem như một ngôi sao phía ngoài hơn, SVM có tính năng cho phép bỏ qua các ngoại lệ và tìm ra hyper-plane có biên giới tối đa. Do đó có thể nói, SVM có khả năng mạnh trong việc chấp nhận ngoại lệ.

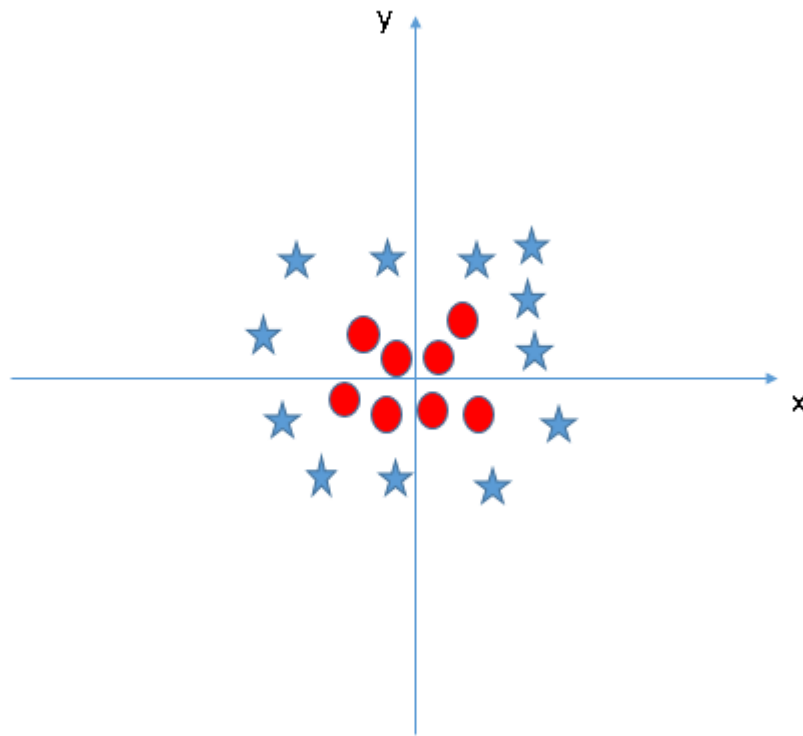


Hình 2. 8 Serinato 4-2

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Find the hyper-plane to segregate to classes (Scenario-5)

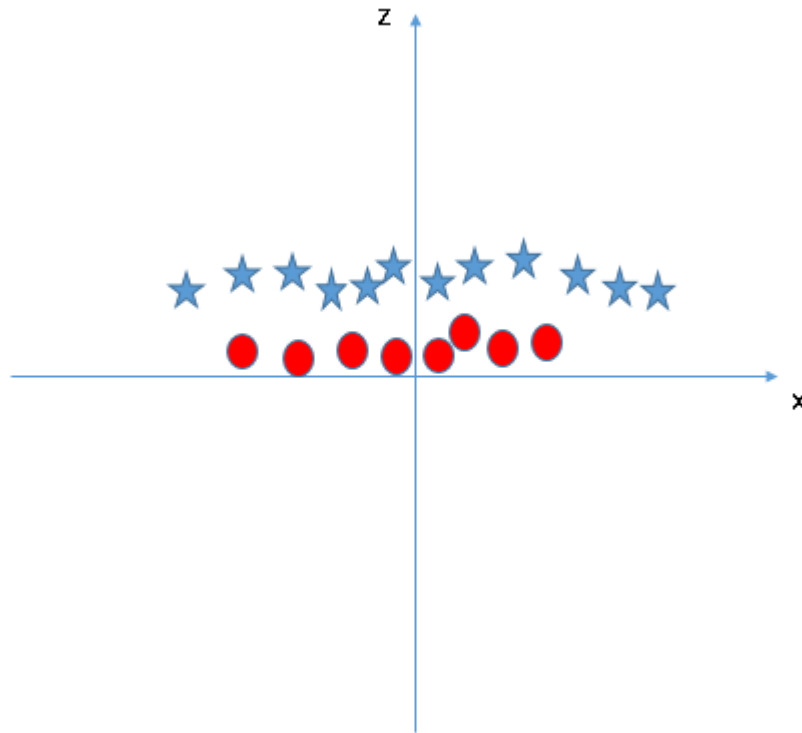
Trong trường hợp dưới đây, không thể tìm ra 1 đường hyper-plane tương đối để chia các lớp, vậy làm thế nào để SVM phân tách dữ liệu thành hai lớp riêng biệt? Cho đến bây giờ chúng ta chỉ nhìn vào các đường tuyến tính hyper-plane.



Hình 2. 9 Dữ liệu biến đổi

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

SVM có thể giải quyết vấn đề này, Khá đơn giản, nó sẽ được giải quyết bằng việc thêm một tính năng, Ở đây chúng ta sẽ thêm tính năng $z = x^2 + y^2$. Bây giờ dữ liệu sẽ được biến đổi theo trục x và z như sau



Hình 2. 10 Hyper lan biến đổi

(nguồn: Giới thiệu về Support Vector Machine (SVM)) [8]

Trong sơ đồ trên, các điểm cần xem xét là:

- Tất cả dữ liệu trên trục z sẽ là số dương vì nó là tổng bình phương x và y
 - Trên biểu đồ các điểm tròn đỏ xuất hiện gần trục x và y hơn vì thế z sẽ nhỏ hơn
- => nằm gần trục x hơn trong đồ thị (z, x) Trong SVM, rất dễ dàng để có một siêu phẳng tuyến tính (linear hyper-plane) để chia thành hai lớp, Nhưng một câu hỏi sẽ nảy sinh đây là, chúng ta có cần phải thêm một tính năng phân chia này bằng tay hay không. Không, bởi vì SVM có một kỹ thuật được gọi là kernel trick (kỹ thuật hạt nhân), đây là tính năng có không gian đầu vào có chiều sâu thẳm và biến đổi nó thành không gian có chiều cao hơn, tức là nó không phân chia các vấn đề thành các vấn đề riêng biệt, các tính năng này được gọi là kernel. Nói một cách đơn giản nó thực hiện một số biến đổi dữ liệu phức tạp, sau đó tìm ra quá trình tách dữ liệu dựa trên các nhãn hoặc đầu ra mà chúng ta đã xác định trước.

Margin trong SVM:

Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Trong ví dụ quả táo quả lê đặt trên mặt bán, margin chính là khoảng cách giữa cây que và hai quả táo và lê gần nó nhất. Điều quan trọng ở đây đó là phương pháp

SVM luôn cố gắng cực đại hóa margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả táo và lê. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào. [8]

2.1.1.3.2 Hàm Loss

Dùng để đo lường sự mất mát của mô hình dự đoán với dữ liệu huấn luyện.

Được chia thành 2 phần: phần đầu đo lường sự chính xác dự đoán, phần hai đo lường sự phức tạp mô hình

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(w \cdot x_i + b)) + \lambda ||w||^2$$

$L(w, b)$ Đây là hàm loss tổng quát, đo lường mức độ mất mát của mô hình trong quá trình dự đoán. Mục tiêu là giảm thiểu hàm loss này thông qua quá trình huấn luyện.

$\frac{1}{n}$: Là trung bình sai số, chia tổng sai số cho số lượng mẫu n để tính trung bình.

$\sum_{i=1}^n$: Tính tổng sai số qua tất cả các mẫu từ 1 đến n .

$\max(0, 1 - y_i(w \cdot x_i + b))$ Được gọi là Hinge Loss, đo lường khoảng cách giữa dự đoán của mô hình và nhãn thực tế. Nếu dự đoán đúng, hàm loss sẽ là 0; nếu không, hàm loss sẽ tăng tỷ lệ với khoảng cách giữa dự đoán và giá trị thực tế.

0,1 hệ số điều chỉnh mức độ nghiêm ngặt

w là các hệ số của siêu phẳng.

x là các biến đầu vào.

b là hệ số độ dời (bias).

w là vector trọng số của siêu phẳng.

y là nhãn mục tiêu của biến số đầu vào

$||w||^2$: Đây là phần regularization, giúp kiểm soát độ lớn của trọng số w trong mô hình. Bằng cách giảm thiểu độ lớn của w , ta có thể ngăn chặn overfitting và làm cho mô hình tổng quát hơn.

$||w||^2$: là norm bậc hai của vector trọng số w .

λ : Là tham số điều chỉnh mức độ của regularization được áp dụng. Khi λ càng lớn, mức độ regularization càng mạnh mẽ, và ngược lại. Điều này ảnh hưởng đến sự cân nhắc giữa việc cải thiện hiệu suất trên dữ liệu huấn luyện và việc tránh overfitting trên dữ liệu mới.

Regularization (Chính quy hóa) là một kỹ thuật quan trọng trong học máy được sử dụng để giảm thiểu hiện tượng overfitting.

Hệ số regularization này kiểm soát mức độ phức tạp của mô hình. Giá trị hệ số regularization càng lớn, mô hình càng đơn giản và ít bị overfitting hơn.

2.1.1.3.3 Ưu nhược điểm của SVM

Ưu điểm:

Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn

Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định

Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

Nhược điểm:

Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi

Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào. Tuy nhiên hiệu quả của việc phân lớp có thể được xác định dựa vào khái niệm margin từ điểm dữ liệu mới đến siêu phẳng phân lớp mà chúng ta đã bàn luận ở trên. [11]

2.1.2 Thư viện Pandas

2.1.2.1 Thư viện Pandas là gì?

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi. [9]

2.1.2.2 Vì sao lại cần sử dụng Pandas?

Vì pandas phù hợp với việc xử lý và chuẩn hóa dữ liệu:

- Dữ liệu dạng bảng với các cột được nhập không đồng nhất, như trong bảng SQL hoặc bảng tính Excel.
- Dữ liệu chuỗi thời gian theo thứ tự và không có thứ tự (không nhất thiết phải có tần số cố định).
- Dữ liệu ma trận tùy ý (được nhập đồng nhất hoặc không đồng nhất) với nhãn hàng và cột. Bất kỳ hình thức khác của các bộ dữ liệu quan sát / thống kê.
- Dữ liệu thực sự không cần phải được dán nhãn vào cấu trúc dữ liệu pandas. Pandas được xây dựng dựa trên NumPy.
- Hai cấu trúc dữ liệu chính của pandas là Series (1 chiều) và DataFrame (2 chiều) xử lý được phần lớn các trường hợp điển hình trong tài chính, thống kê, khoa học xã hội và nhiều lĩnh vực kỹ thuật. [3]

2.1.3 Google Colab và thư viện scikit-learn

2.1.3.1 Google Colab

Google Colab là một dịch vụ miễn phí từ Google cho phép bạn viết và chia sẻ mã Python. Người dùng có thể lưu trữ và chạy Notebook trên đám mây thông qua trình duyệt mà không cần kích hoạt cấu hình bộ đệm phức tạp cho máy tính cá nhân.

Nền tảng cũng cung cấp GPU và TPU để huấn luyện mô hình học và thực hiện các yêu cầu tính toán lớn. Điều này có tác dụng mở rộng sức mạnh xử lý trong điện toán và giảm thời gian chờ đợi khi thực hiện các nhiệm vụ tính toán nặng nề.

Nguồn gốc hình thành Google Colab từ đâu?

Google Colab được phát triển bởi nhóm Google Research nhằm hỗ trợ quá trình nghiên cứu và phát triển trong lĩnh vực học máy tính và khoa học dữ liệu. Công nghệ xuất phát từ nền tảng Jupyter Notebook phổ biến. Sau đó, Colab được kết hợp thêm khả năng tính toán trên tiện ích đám mây vô cùng mạnh mẽ của Google.

Tính ứng dụng vượt trội của Google Colab

Những lợi ích dưới đây làm cho Google Colab trở thành một công cụ hữu ích. Đặc biệt trong các lĩnh vực nghiên cứu, phát triển và triển khai các dự án liên quan đến học máy và khoa học dữ liệu.

Các tiện ích miễn phí

Google Colab cung cấp dịch vụ sử dụng tài nguyên tính toán đám mây miễn phí. Trong đó bao gồm cả GPU và TPU giúp tăng tốc độ huấn luyện mô hình máy học và thực thi tính toán nhanh chóng. Người dùng không cần phải lo lắng về cấu hình phức tạp hoặc chi phí tài nguyên khi ứng dụng Colab.

Bạn có thể truy cập và làm việc trên Colab từ bất kỳ thiết bị nào có kết nối internet mà không cần thực hiện các bước cài đặt phức tạp hoặc cập nhật phần mềm. Công nghệ hiện đại mang đến nhiều điều kiện thuận lợi cho việc lập trình và nghiên cứu.

Môi trường lập trình linh hoạt

Google Colab cung cấp môi trường lập trình linh hoạt với nhiều ưu điểm. Bạn có thể truy cập và làm việc trên Google Colab từ bất kỳ thiết bị nào với kết nối mạng. Hàng loạt tiện ích cho thấy tính năng lập trình linh hoạt trên công cụ như sau:

Sử dụng GPU và TPU: Colab cung cấp khả năng sử dụng GPU và TPU miễn phí, giúp tăng tốc độ tính toán đặc biệt cho việc huấn luyện mô hình máy học và xử lý dữ liệu lớn.

Tích hợp công cụ mạnh mẽ: Colab tích hợp với các công nghệ và thư viện phổ biến như tensorflow, pytorch, CUDA. Mục đích để hỗ trợ các tác vụ học máy và tính toán khoa học dữ liệu.

Lưu trữ và chia sẻ dữ liệu dễ dàng: Bạn có thể lưu trữ dữ liệu và Notebooks trực tiếp trên Google Drive. Đồng thời chia sẻ chúng với đồng nghiệp hoặc cộng tác viên một cách thuận tiện.

Tích hợp với Google Drive

Google Colab được tích hợp một cách thuận tiện với Google Drive. Điều này nghĩa là bạn có thể truy cập và làm việc trên các tệp tài liệu trong Google Drive ngay từ Google Colab và ngược lại. Việc tích hợp này mang lại nhiều lợi ích như:

Lưu trữ dữ liệu: Bạn có thể lưu trữ các tệp tài liệu được sử dụng trong quá trình lập trình trên Google Drive và truy cập chúng một cách dễ dàng từ Google Colab.

Chia sẻ và cộng Tác: Google Drive cho phép chia sẻ tệp tài liệu với người khác, điều này cũng áp dụng cho các tệp được sử dụng trong Google Colab.

Tích hợp dự án: Công cụ tích hợp với Google Drive cũng làm cho việc lưu trữ và quản lý các dự án liên quan đến học máy và khoa học dữ liệu trở nên thuận tiện hơn.

Hỗ trợ công nghệ đa dạng

Google Colab hỗ trợ đa dạng tiện ích được ứng dụng trong lĩnh vực học máy và khoa học dữ liệu, bao gồm những điểm sau:

Python và Jupyter Notebooks: Google Colab được thiết kế để hoạt động tốt với Python và hỗ trợ các tệp Notebooks trong định dạng Jupiter. Từ đó giúp việc phân tích dữ liệu và lập trình trở nên thuận tiện hơn.

CUDA và Tensorflow: Colab hỗ trợ CUDA, một nền tảng tính toán song song của NVIDIA, cho phép sử dụng GPU để gia tăng hiệu suất huấn luyện mô hình máy học. Ngoài ra, Colab tích hợp sẵn với tensorflow, một thư viện phổ biến trong lĩnh vực học máy.

Thư viện máy tính và khoa học dữ liệu: Ngoài tensorflow, Google Colab cũng hỗ trợ nhiều thư viện máy tính và khoa học dữ liệu khác. Điển hình như pytorch, scikit-learn, pandas, matplotlib, seaborn và nhiều thư viện khác. Điều này giúp người dùng phát triển và thực thi các tác vụ tính toán một cách linh hoạt.

Tích hợp với công cụ nổi tiếng: Colab cung cấp tích hợp sẵn với các dịch vụ và công nghệ phổ biến như Google bigquery, Google Cloud Storage và Google Sheets. Công nghệ mở rộng khả năng tương tác và tích hợp dữ liệu từ nhiều nguồn khác nhau.

Google Colab hỗ trợ cộng tác và chia sẻ một cách thuận tiện. Tính năng này tạo điều kiện thuận lợi cho việc cộng tác và chia sẻ kiến thức, dữ liệu và kết quả làm việc với người khác trong cộng đồng khoa học dữ liệu. Dưới đây là một số tính năng về cộng tác và chia sẻ trên nền tảng này:

- Chia sẻ Notebooks: Bạn có thể chia sẻ các Notebooks trực tiếp từ Google Colab, cho phép người khác xem và chỉnh sửa. Điều này rất hữu ích khi cần thảo luận hoặc cộng tác trong quá trình phát triển dự án hoặc học tập.
- Hợp tác cùng lúc: Google Colab cho phép nhiều người dùng cùng truy cập và chỉnh sửa một Notebook cùng lúc, tạo điều kiện cho môi trường làm việc cộng tác và hợp tác trong dự án.
- Kết nối qua Link chia sẻ: Bạn có thể tạo liên kết chia sẻ từ Google Colab để mời người khác tham gia hoặc xem Notebooks của bạn một cách nhanh chóng và dễ dàng. [4]

2.1.3.2 Thư viện scikit-learn

Dự án scikit-learn bắt đầu với tên scikits.learn, một dự án Google Summer of Code của nhà khoa học dữ liệu người Pháp David Cournapeau. Tên của dự án bắt nguồn từ ý tưởng rằng nó là "SciKit" (Bộ công cụ SciPy), một tiện ích mở rộng của bên thứ ba được phát triển và phân phối riêng cho SciPy. Cơ sở mã ban đầu sau đó được các nhà phát triển khác viết lại. Năm 2010, các cộng tác viên Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort và Vincent Michel, từ Viện Nghiên cứu Khoa học Máy tính và Tự động hóa Pháp ở Saclay, Pháp, đã lãnh đạo dự án và phát hành phiên bản công khai đầu tiên của thư viện vào ngày 1 tháng 2, 2010. Vào tháng 11 năm 2012, scikit-learn cũng như scikit-image được mô tả là hai trong số các thư viện scikits “được duy trì tốt và phổ biến”. Vào năm 2019, người ta đã lưu ý rằng scikit-learn là một trong những thư viện máy học phổ biến nhất trên GitHub. [2]

Scikit-learn là thư viện học máy có nguồn mở và toàn diện nhất trong Python. Vì học máy thường là một thành phần của một ứng dụng tổng quát hơn (chẳng hạn như dịch vụ Web), nên nó được trang bị bằng ngôn ngữ lập trình giống như các phần khác của ứng dụng để tích hợp liền mạch. Hưởng lợi từ khả năng ứng dụng rộng rãi của Python, Scikit-learn trở thành một thư viện ngày càng phổ biến cho các ứng dụng liên quan đến học máy. [5]

Ngoài sự hỗ trợ mạnh mẽ từ hệ sinh thái Python, bản thân Scikit-learn còn có nhiều tính năng khiến nó nổi bật trong số các phần mềm học máy. Đầu tiên là phạm vi bao quát toàn diện của nó về các phương pháp học máy. Quy trình đánh giá của cộng đồng được áp dụng để xác định và quyết định xem nên đưa phương pháp học máy nào vào gói Scikit-learn. Cơ chế như vậy đảm bảo sự cân bằng giữa phạm vi bao phủ rộng rãi và tính chọn lọc của các phương pháp học máy có trong gói. Thứ hai là việc triển khai thuật toán của các phương pháp học máy trong Scikit-learn được tối ưu hóa để mang lại hiệu quả tính toán. [5]

Mặc dù Python là ngôn ngữ lập trình diễn giải nhưng hầu hết các phương pháp học máy trong Scikit-learn đều dựa trên các thư viện nhị phân được biên dịch ban đầu được lập trình bằng Fortran, C hoặc Cpp. Việc triển khai dựa trên nhị phân này cải thiện đáng kể hiệu quả tính toán. Thứ ba là Scikit-learn có cộng đồng hỗ trợ mạnh mẽ về tài liệu, theo dõi lỗi và đảm bảo chất lượng. Cộng đồng Scikit-learn duy trì tài liệu chung, quy trình theo dõi/sửa lỗi thống nhất trên GitHub và quy trình đảm bảo chất lượng nghiêm ngặt dựa trên một thỏa thuận toàn cộng đồng. Cuối cùng nhưng không kém phần quan trọng, Scikit-learn áp dụng quy ước dữ liệu đầu vào/đầu ra thống nhất và có quy trình điều chỉnh mô hình cố định (như chúng tôi sẽ trình bày trong phần tiếp theo), khiến việc chuyển đổi từ phương pháp này sang phương pháp khác gần như dễ dàng. [5]

2.2 Vấn đề phát hiện gian lận thẻ tín dụng trong thời đại hiện nay

2.2.1 Về vấn đề gian lận thẻ tín dụng

Hiện nay, giao dịch bằng thẻ tín dụng trở nên phổ biến và ngày càng phổ biến trên khắp thế giới, đang dần thay thế giao dịch truyền thống bằng tiền mặt và các hình thức chuyển khoản, ví điện tử,... vì những lý do sau:

- Tiện lợi: không cần phải cầm theo tiền mặt, đếm xem đã đủ tiền chưa, và cũng không phải lấy điện thoại để quét mã hoặc nhập mã banking trên ứng dụng ngân hàng hoặc ví điện tử. Chỉ cần ở những nơi đó có máy POS (viết tắt của chữ Point of Sale) là có thể mang theo thẻ tín dụng để qua một lượt quét là thanh toán được.
- Không cần mang theo tiền mặt: vì chỉ cần có tiền trong thẻ được gửi trong tài khoản ngân hàng là hoàn toàn dùng thanh toán được mọi dịch vụ như điện, nước, thực phẩm, viện phí...

- Hưởng nhiều ưu đãi: vì trong thời đại 4.0, bắt đầu xu thế tiêu dùng thanh toán không dùng tiền mặt, nên các siêu thị, trung tâm thương mại, ngân hàng,... đang khuyến khích người dân thanh toán không dùng tiền mặt vì thẻ tín dụng trở nên phổ biến. Nên những siêu thị, trung tâm thương mại,... đã mở các gói ưu đãi khi thanh toán bằng thẻ tín dụng để kích thích việc thanh toán không dùng tiền mặt.
- Cho phép một hạn mức vay tín dụng: khi đăng ký mở thẻ tín dụng để sử dụng, ngân hàng sẽ cấp cho một mức hạn mức vay tín dụng nhất định, để có thể chi tiêu theo kiểu “tiêu trước trả sau”.
- Sử dụng để cứu cánh trong những trường hợp cần tiền gấp: vì thẻ tín dụng có thể chi tiêu trước trả tiền sau, lên những lúc cần tiền gấp cho là “phao cứu sinh” vì rút hạn mức tín dụng dễ dàng, không mất thủ tục chờ vay, không cần cầm cố tài sản.
- Trải nghiệm các dịch vụ: một số khách hàng cao cấp của các ngân hàng có thể mở được các thẻ tín dụng VIP (thẻ đen). Thẻ tín dụng dạng đó khách hàng có thể trải nghiệm các dịch vụ VIP ở khách sạn năm sao, đi vé thương gia, mua trả góp dễ dàng, miễn phí dịch vụ cao cấp,...

Chính vì sự tiện lợi và các dịch vụ tốt như vậy của thẻ tín dụng, nên xảy ra tình trạng gian lận giao dịch trong thẻ tín dụng ngày càng nhiều. Và các biện pháp để gian lận cũng ngày càng tinh vi hơn. Nên trong thời đại hiện nay, việc phát hiện ra gian lận thẻ tín dụng ngày càng quan trọng, và cần thiết không chỉ ở ngân hàng mà còn ở quốc gia.

2.2.2 Tầm quan trọng của phát hiện gian lận

Việc phát hiện gian lận thẻ tín dụng quan trọng vì:

Tránh thất thoát nguồn tiền ngân hàng: việc phát hiện gian lận thẻ tín dụng sẽ gây ra thất thoát nguồn tiền vì ngân hàng sẽ không thu hồi được tiền bị thất thoát do gian lận. Khi tiền bị rút ra hoặc sử dụng trái phép, ngân hàng sẽ rất khó thu hồi lại số tiền này, dẫn đến thất thoát tài chính nghiêm trọng. Điều này không chỉ ảnh hưởng đến lợi nhuận mà còn có thể làm mất uy tín của ngân hàng đối với khách hàng.

Bảo vệ khách hàng: khi phát hiện sớm gian lận, họ có thể ngăn chặn kịp thời và bảo vệ tài sản khách hàng. Điều này giúp duy trì lòng tin của khách hàng đối với dịch vụ của ngân hàng, bởi họ biết rằng tài khoản của mình luôn được giám sát và bảo vệ.

Giảm thiểu rủi ro pháp lý và tuân thủ quy định: Các ngân hàng phải tuân thủ nhiều quy định pháp luật liên quan đến bảo vệ thông tin và tài sản của khách hàng. Việc không

phát hiện và ngăn chặn kịp thời các hành vi gian lận có thể dẫn đến việc vi phạm các quy định này, gây ra rủi ro pháp lý và các khoản phạt không mong muốn.

Nâng cao uy tín và niềm tin thị trường: Vì những ngân hàng có hệ thống phát hiện gian lận tốt sẽ nâng cao uy tín trên thị trường. Điều này làm tăng niềm tin cho khách hàng và tăng mức độ uy tín ngân hàng khi mở cho khách hàng mới.

Tối ưu hóa chi phí: Phát hiện sớm và ngăn chặn gian lận giúp ngân hàng tiết kiệm được chi phí xử lý hậu quả sau khi gian lận đã xảy ra. Chi phí liên quan đến điều tra, hoàn tiền cho khách hàng, và khắc phục hậu quả thường rất cao, nên việc ngăn chặn từ đầu sẽ giúp tối ưu hóa nguồn lực và chi phí.

2.2.3 Các phương pháp truyền thống phát hiện gian lận

Giám sát giao dịch thủ công: Nhân viên ngân hàng hoặc các tổ chức tài chính sẽ giám sát thủ công các giao dịch khách hàng để phát hiện các giao dịch bất thường. Họ là những người được đào tạo hoặc có chuyên môn cao.

Quy tắc và cảnh báo cố định: Các ngân hàng thiết lập các quy tắc dựa trên các mẫu giao dịch gian lận đã biết. Ví dụ: nếu khách hàng thực hiện nhiều giao dịch ở nhiều quốc gia khác nhau trong một thời gian ngắn, hoặc nhiều giao dịch giá trị thấp liên tục,... thì đó là giao dịch gian lận.

Phân tích lịch sử giao dịch: Xem xét các mẫu giao dịch trong quá khứ của khách hàng để xác định các hành vi bất thường. Nếu một giao dịch không khớp với các mẫu giao dịch thông thường của khách hàng, nó có thể được đánh dấu để kiểm tra thêm.

Phân tích và theo dõi địa lý: Xác định vị trí địa lý của các giao dịch. Nếu có sự chênh lệch lớn về địa lý giữa các giao dịch trong một khoảng thời gian ngắn, điều này có thể được coi là một dấu hiệu của gian lận.

Giám sát tần suất giao dịch: Theo dõi số lượng giao dịch trong một khoảng thời gian ngắn. Nếu có sự gia tăng đột ngột trong tần suất giao dịch, điều này có thể là dấu hiệu của hoạt động gian lận.

Kiểm tra định kỳ tài khoản: Thực hiện các cuộc kiểm tra định kỳ và so sánh các giao dịch hiện tại với các mẫu giao dịch đã biết để phát hiện sự khác biệt hoặc bất thường.

Xác minh hai yếu tố: Áp dụng phương pháp xác minh hai yếu tố khi thực hiện các giao dịch đáng ngờ hoặc có giá trị cao. Điều này giúp tăng cường bảo mật và giảm nguy cơ gian lận.

Phản hồi từ khách hàng: Khuyến khích khách hàng thông báo ngay khi phát hiện các giao dịch bất thường hoặc không được phép trên tài khoản của họ.

2.2.4 Ưu điểm và hạn chế của phương pháp truyền thống

Ưu điểm của các phương pháp truyền thống phát hiện gian lận thẻ tín dụng:

Dễ triển khai và sử dụng: Các phương pháp như giám sát giao dịch thủ công và quy tắc cố định thường dễ dàng triển khai và không yêu cầu công nghệ phức tạp.

Hiểu biết từ kinh nghiệm thực tiễn: Nhân viên ngân hàng và chuyên gia tài chính có thể dựa vào kinh nghiệm và kiến thức cá nhân để phát hiện các giao dịch bất thường.

Chi phí thấp: Các phương pháp truyền thống thường ít tốn kém hơn so với việc triển khai các hệ thống phát hiện gian lận hiện đại dựa trên trí tuệ nhân tạo hoặc học máy.

Kiểm soát tốt các trường hợp phổ biến: Đối với các loại gian lận đã biết và phổ biến, các quy tắc cố định và giám sát thủ công có thể hoạt động khá hiệu quả.

Hạn chế của các phương pháp truyền thống phát hiện gian lận thẻ tín dụng:

Phản ứng chậm: Các phương pháp này thường không thể phản ứng ngay lập tức với các giao dịch gian lận, đặc biệt là khi cần đến sự can thiệp của con người.

Khó phát hiện các kiểu gian lận mới: Gian lận ngày càng trở nên tinh vi và đa dạng, làm cho các phương pháp dựa trên quy tắc cố định khó có thể phát hiện được các hình thức mới.

Phụ thuộc nhiều vào con người: Giám sát giao dịch thủ công yêu cầu sự tham gia liên tục của nhân viên, dễ dẫn đến sai sót do mệt mỏi hoặc thiếu sót.

Tỷ lệ báo động giả cao: Các quy tắc cố định có thể tạo ra nhiều báo động giả, gây phiền hà cho khách hàng và làm mất thời gian của nhân viên ngân hàng.

Không hiệu quả với khối lượng dữ liệu lớn: Khi số lượng giao dịch tăng lên, việc giám sát thủ công và phân tích theo quy tắc cố định trở nên kém hiệu quả.

Thiếu khả năng tự học và thích nghi: Các phương pháp truyền thống không có khả năng tự học hỏi và thích nghi với các xu hướng mới, trái ngược với các hệ thống học máy hiện đại có thể cải thiện qua thời gian.

2.2.5 Sự ra đời của việc ứng dụng học máy trong phát hiện gian lận thẻ tín dụng

Trong bối cảnh hiện nay, việc giao dịch bằng thẻ tín dụng ngày càng phổ biến, kéo theo việc gian lận ngày càng nhiều. Các biện pháp phát hiện gian lận thủ công ngày càng kém hiệu quả vì các biện pháp gian lận ngày càng tinh vi. Cho nên, ứng dụng học máy cho việc phát hiện gian lận thẻ tín dụng, vì khả năng phát hiện chính xác hơn, phát hiện tốt hơn cho các gian lận mới, và nhanh chóng, có thể phát hiện sớm gian lận để ngăn chặn.

2.2.5.1 Ưu điểm của học máy

Khả năng tự học và cải thiện: Hệ thống học máy có thể tự học từ dữ liệu lịch sử và liên tục cải thiện độ chính xác của mình khi có thêm dữ liệu mới.

Xử lý khối lượng dữ liệu lớn: Học máy có khả năng xử lý và phân tích một lượng lớn dữ liệu một cách nhanh chóng và hiệu quả, giúp phát hiện các mẫu giao dịch gian lận phức tạp.

Phát hiện các mô hình phức tạp: Các thuật toán học máy có khả năng nhận diện các mô hình và mối quan hệ phức tạp trong dữ liệu mà con người và các phương pháp truyền thống khó có thể nhận ra.

2.2.5.2 Triển khai và ứng dụng

Các tổ chức tài chính và ngân hàng đã bắt đầu triển khai các hệ thống phát hiện gian lận dựa trên học máy để:

Tự động hóa quá trình phát hiện gian lận: Giảm thiểu sự phụ thuộc vào con người và tăng tốc độ phản ứng.

Tối ưu hóa độ chính xác: Giảm tỷ lệ báo động giả và tăng khả năng phát hiện các giao dịch gian lận thực sự.

Cải thiện trải nghiệm khách hàng: Bằng cách giảm thiểu các can thiệp không cần thiết và bảo vệ tài khoản của khách hàng một cách hiệu quả hơn.

2.2.5.3 Thách thức và cơ hội

Mặc dù học máy mang lại nhiều lợi ích, việc triển khai cũng đối mặt với một số thách thức như:

Yêu cầu về dữ liệu chất lượng cao: Hệ thống học máy yêu cầu một lượng lớn dữ liệu chất lượng cao để huấn luyện và duy trì hiệu quả.

Độ phức tạp của mô hình: Các mô hình học máy có thể trở nên rất phức tạp và khó hiểu, làm tăng thách thức trong việc giải thích và điều chỉnh mô hình.

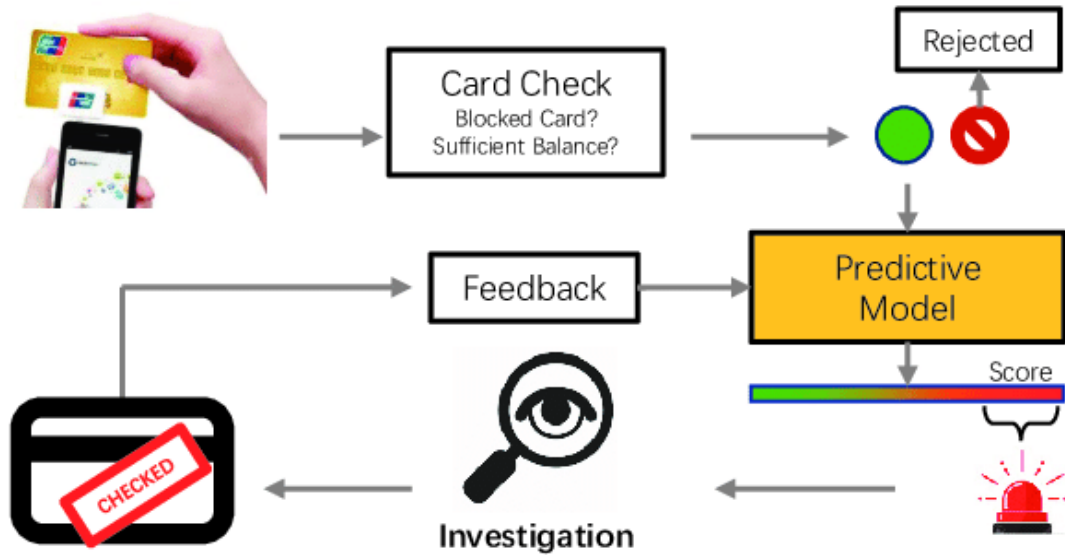
2.3 Kết luận chương

Học máy đang ngày càng phát triển trong thời đại ngày nay. Các phương pháp học máy cũng đang ngày càng phổ biến trong mọi thứ trong cuộc sống, ngay cả việc phát hiện gian lận thẻ tín dụng. Các phương pháp phát hiện giao dịch thẻ tín dụng gian lận cũng ngày càng phổ biến và khả quan hơn. Do đó, đề án này tập trung nghiên cứu huấn luyện mô hình học máy phát hiện gian lận thẻ tín dụng. Đây cũng là một phần quan trọng của đề tài lớn hơn là xây dựng ứng dụng phát hiện gian lận thẻ tín dụng.

CHƯƠNG 3: THIẾT KẾ, CẢI THIỆN VÀ ĐÁNH GIÁ KẾT QUẢ

Chương này sẽ cho biết về mô hình bài toán, dữ liệu sử dụng. Sau đó cho biết về việc tiền xử lý dữ liệu, phân đoạn dữ liệu, chia dữ liệu thành tập huấn luyện và kiểm tra. Và cho biết kết quả huấn luyện và đánh giá kết quả của từng mô hình.

3.1 Mô hình chung của bài toán



Hình 3. 1 Quy trình phát hiện

(nguồn: K-Nearest Neighbor(KNN) Algorithm) [14]

Thu thập dữ liệu:

Dữ liệu giao dịch: Bao gồm thông tin chi tiết về các giao dịch như số tiền, thời gian, địa điểm, loại giao dịch, thông tin về người dùng, và các đặc điểm khác.

Dữ liệu nhãn: Thông tin về việc giao dịch có phải là gian lận hay không, thường được xác định qua các báo cáo và điều tra trước đó.

Dữ liệu nhãn thường ở dạng nhị phân 0 hoặc 1.

Tiền xử lý dữ liệu

Làm sạch dữ liệu: Loại bỏ hoặc xử lý các giá trị thiếu, không hợp lệ, hoặc lỗi.

Biến đổi dữ liệu: Chuẩn hóa hoặc chuẩn hóa dữ liệu để đảm bảo các thuộc tính có tỷ lệ tương thích.

Chuyển đổi thuộc tính: Tạo ra các thuộc tính mới hoặc mã hóa các thuộc tính dạng văn bản (ví dụ: mã hóa one-hot).

Phân chia dữ liệu

Dữ liệu huấn luyện: Sử dụng để huấn luyện mô hình.

Dữ liệu kiểm tra và xác thực: Sử dụng để kiểm tra và đánh giá mô hình sau khi huấn luyện.

Lựa chọn và xây dựng mô hình

Thuật toán học có giám sát: Các mô hình phổ biến như Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, và Support Vector Machines.

Thuật toán học sâu: Mạng nơ-ron (Neural Networks) và các biến thể như Convolutional Neural Networks (CNNs) và Recurrent Neural Networks (RNNs).

Thuật toán học không giám sát: Các kỹ thuật như Clustering (K-Means), Principal Component Analysis (PCA), và Autoencoders.

Huấn luyện mô hình

Quá trình học: Sử dụng dữ liệu huấn luyện để tìm ra các mẫu và mối quan hệ trong dữ liệu, điều chỉnh các tham số của mô hình để tối ưu hóa hiệu suất.

Đánh giá mô hình: Sử dụng dữ liệu kiểm tra để đánh giá hiệu suất của mô hình thông qua các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity), và điểm F1 (F1-score).

Tinh chỉnh và tối ưu hóa

Điều chỉnh tham số (Hyperparameter Tuning): Sử dụng các kỹ thuật như Grid Search hoặc Random Search để tìm ra các tham số tối ưu cho mô hình.

Cross-Validation: Sử dụng k-fold cross-validation để đánh giá độ ổn định và khả năng tổng quát hóa của mô hình.

Triển khai mô hình

Hệ thống phát hiện thời gian thực: Triển khai mô hình trong hệ thống xử lý giao dịch thời gian thực để phát hiện các giao dịch gian lận ngay khi chúng xảy ra.

Giám sát và bảo trì: Theo dõi hiệu suất của mô hình trong môi trường thực tế, cập nhật và huấn luyện lại mô hình khi có thêm dữ liệu mới hoặc khi điều kiện thay đổi.

Đánh giá và cải thiện liên tục

Phân tích báo động giả và nhầm lẫn: Liên tục phân tích các trường hợp báo động giả và nhầm lẫn để cải thiện độ chính xác của mô hình.

Cập nhật mô hình: Sử dụng dữ liệu mới và các kỹ thuật học liên tục để duy trì và nâng cao hiệu suất của mô hình theo thời gian.

3.2 Thu thập dữ liệu

3.2.1 Giới thiệu chung về các tập dữ liệu sử dụng

Trong đồ án này, các bộ dữ liệu được sử dụng bao gồm: Credit Card Fraud Detection Predictive Models

3.2.2 Các tập dữ liệu công bố

Tập dữ liệu Credit Card Fraud Detection Predictive Models của tác giả GABRIEL PREDA.

Bộ dữ liệu được công bố công khai trên trang web <https://www.kaggle.com/> vào 3 năm trước.

Link bộ dữ liệu: <https://www.kaggle.com/code/gpreda/credit-card-fraud-detection-predictive-models/input>

Bộ dữ liệu có 31 trường, và có 284807 dòng dữ liệu

Mô tả các trường

Bảng 3. 1 Mô tả các trường

STT	Tên trường	Mô tả
1	Time	<p>Đây là thời gian đầu tiên mà giao dịch được thực hiện tính từ giao dịch đầu tiên trong bộ dữ liệu. Số thời gian được biểu diễn trong dữ liệu là số giây của giao dịch tiếp theo tính từ giao dịch đầu tiên trong tập dữ liệu.</p> <p>Số thời gian trên được trôi qua tính bằng giây.</p>
2	V1 – V28	<p>Đây là các trường dữ liệu ẩn (có thể đã được giảm chiều), được xử lý từ thông tin gốc dưới phương pháp Phân tích thành phần chính (PCA) hoặc các phương pháp khác.</p>

STT	Tên trường	Mô tả
		<p>Trong các trường này, có thể là dữ liệu về thông tin giao dịch, thông tin cá nhân của người dùng. Dữ liệu trong này đã được ẩn danh tính để đảm bảo an toàn thông tin nhạy cảm của người dùng trong tập dữ liệu.</p> <p>Việc xử lý này là để đảm bảo an toàn và an ninh các thông tin cá nhân của người dùng và vẫn giữ lại được các thông tin quan trọng, vì phương pháp PCA không thể giải ngược để khôi phục hoàn toàn.</p>
3	Amount	<p>Số tiền của mỗi phiên giao dịch, trong này là số dương.</p> <p>Biết số tiền cao nhất là 25691.16, và số tiền thấp nhất là 0.</p>
4	Class	<p>Trường này chứa nhãn cho biết liệu giao dịch có gian lận hay không?</p> <p>Trường này là biến mục tiêu của mô hình phân loại, được biểu diễn dưới dạng mã nhị phân:</p> <ul style="list-style-type: none"> - 1 là giao dịch gian lận - 0 là không gian lận

3.2.3 Tổng kết bộ dữ liệu

Bộ dữ liệu có tổng cộng 492 quan sát gian lận trong tổng số 284807 quan sát đã thực hiện trong gian lận thẻ tín dụng, tỷ lệ 0.1727%

3.3 Tiền xử lý dữ liệu

3.3.1 Phân đoạn dữ liệu

Trong đề án này, bộ dữ liệu Credit Card Fraud Detection Predictive Models từ trên trang web <https://www.kaggle.com/> đã được phân đoạn sẵn. Điều này giúp đảm bảo tính nhất quán, độ chính xác và tiết kiệm thời gian trong quá trình nghiên cứu và phân tích.

Bộ dữ liệu đã được phân đoạn theo 2 Class là 0 là không gian lận, 1 là có gian lận.

3.3.2 Kiểm tra và làm sạch dữ liệu

Dữ liệu đầu vào sẽ được kiểm tra để xem có dữ liệu rỗng hay không bằng phương pháp sau và sử dụng trên python



```

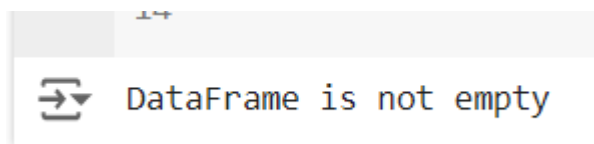
1  import pandas as pd
2
3  # Đường dẫn đến tệp dữ liệu trên Google Drive
4  file_path = '/content/drive/MyDrive/DoAnTotNghiep/creditcard.csv'
5
6  # Đọc dữ liệu từ tệp
7  df = pd.read_csv(file_path)
8
9  # Kiểm tra xem DataFrame có chứa giá trị rỗng không
10 if df.isnull().values.any():
11     print("DataFrame contains empty values")
12 else:
13     print("DataFrame is not empty")
14

```

Kết quả sau khi kiểm tra

Hình 3. 2 Kiểm tra dữ liệu trống

Trong mô hình này, tôi sử dụng bộ dữ Credit Card Fraud Detection Predictive



```

→ DataFrame is not empty

```

Hình 3. 3 Kết quả sau kiểm tra

Models đã được công bố sẵn trên <https://www.kaggle.com/> . Bộ dữ liệu này đã được loại bỏ thiếu, nhiễu, ngoại lệ trước khi công bố để phù hợp với bài toán nghiên cứu của chúng tôi.

3.3.3 Kết quả dữ liệu sau làm sạch

Dữ liệu đã được làm sạch.

Bộ dữ liệu đầy đủ, không có dữ liệu thiếu, nhiễu, ngoại lai.

3.4 Huấn luyện mô hình

3.4.1 Chuẩn hóa dữ liệu

Trong tập dataset sử dụng, dữ liệu đã được chuẩn hóa thành 2 nhãn khác nhau ở cột ‘Class’. Trong dữ liệu đã được chuẩn hóa, các giao dịch gian lận được gán nhãn là 1, còn giao dịch không gian lận được gán nhãn là 0.

Vì tập dữ liệu trên đã được chuẩn hóa khi được thu thập từ trên Kaggel. Nên em sẽ không tiếp tục thực hiện các bước chuẩn hóa nâng cao, vì dữ liệu đã được chuẩn hóa ở mức gán nhãn 0 cho giao dịch không gian lận, 1 cho giao dịch có gian lận.

Dữ liệu đại diện cho class 0

		B		C		D		E		F		G		H		I		J		K		L		M		N		O		P		Q		R		S		T		U		V		W		X		Y		Z		AA		AB		AC		AD		AE	
1	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class																														
2	0	-1.3598	-0.0728	2.53635	1.37818	-0.3383	0.46239	0.2396	0.0987	0.30579	0.09079	-0.5516	-0.6178	-0.9814	-0.5112	1.48818	-0.4704	0.20797	0.02579	0.40399	0.25141	-0.0183	0.27784	-0.1105	0.06693	0.12854	-0.1891	0.13356	-0.0211	149.62	0																														
3	0	0.19186	0.26615	0.16648	0.48315	0.06002	-0.0624	-0.0786	0.0851	-0.2554	-0.157	1.61273	1.06524	-0.4891	-0.1438	0.63556	0.40592	-0.1148	-0.1834	-0.1458	-0.0691	-0.2258	-0.0587	0.10129	-0.3598	0.16717	0.12589	-0.009	0.01472	2.99	0																														
4	1	-1.3584	-1.3402	1.77321	0.37978	-0.5032	1.8005	0.79146	0.24768	-1.5147	0.20764	0.6245	0.06608	0.71729	-0.1659	2.34586	-2.8901	1.10907	-0.1214	-2.2619	0.52498	0.248	0.77168	0.90941	-0.6893	-0.3276	-0.1391	-0.0554	-0.0596	378.66	0																														
5	1	-0.9063	-0.1852	1.79299	-0.8633	-0.0103	1.2472	0.23761	0.37744	-1.087	-0.055	-0.2265	0.17823	0.50776	-0.2879	-0.6314	-1.0596	-0.6841	1.96578	-1.2326	-0.208	-0.1083	0.00527	-0.1903	-1.1756	0.64738	-0.2219	0.06272	0.06146	123.5	0																														
6	2	-1.1582	0.87774	1.54872	0.40303	-0.4072	0.69592	0.59294	-0.2705	0.81774	0.75307	-0.8228	0.5382	1.34589	-1.1197	0.17512	-0.4514	-0.237	-0.0382	0.80349	0.40854	-0.0094	0.79828	-0.1375	0.14127	-0.206	0.50229	0.21942	0.21515	69.99	0																														
7	2	-0.426	0.96052	1.14111	-0.1683	0.42099	-0.0297	0.47462	0.26031	-0.5687	-0.3714	1.94126	0.35989	-0.3581	-0.1971	0.51762	0.40173	-0.0581	0.08865	-0.0332	0.08497	-0.2093	-0.5598	-0.0264	-0.3714	-0.2328	0.10591	0.35394	0.06108	3.67	0																														
8	4	1.22966	0.141	0.04537	1.20261	0.19188	0.27271	-0.0052	0.08121	0.46496	-0.0993	-1.4169	-0.1538	-0.7511	0.16737	0.05014	-0.4438	0.00282	-0.612	-0.0456	-0.2196	-0.1677	-0.2707	-0.1541	-0.7801	0.75014	-0.2572	0.03451	0.00517	4.99	0																														
9	7	-0.6443	1.41796	1.07438	-0.4922	0.94893	0.42812	1.12063	-0.8079	0.61537	1.24938	-0.6195	0.29147	1.75796	-1.3239	0.88613	-0.7061	-1.2221	-0.3582	0.3245	-0.1567	1.94347	-1.0155	0.0575	-0.6497	-0.4153	-0.5016	-1.2069	-1.0853	40.8	0																														
10	7	-0.8943	0.28616	-0.1132	-0.2715	2.6966	3.72182	0.37015	0.85108	-0.392	-0.4104	-0.7051	-0.1105	-0.2863	0.07438	-0.3288	-0.2101	-0.4998	0.11876	0.57033	0.05274	-0.0734	-0.2681	-0.2042	0.01159	0.3732	-0.5842	0.01175	0.1424	93.2	0																														
11	9	-0.3393	1.11959	1.04437	-0.2222	0.49506	-0.2489	0.65156	0.69554	0.7367	-0.3669	1.01781	0.83639	1.00684	-0.4435	0.15022	0.73945	-0.541	0.47668	0.45177	0.20371	-0.2469	-0.6338	-0.1208	-0.385	-0.0697	0.0942	0.24622	0.06308	3.68	0																														
12	10	1.44094	-1.1763	0.91386	-1.3757	-1.9714	-0.6292	-1.4232	0.04848	-1.7204	1.62696	1.19964	-0.8714	-0.5139	-0.095	0.23093	0.03197	0.25341	0.85434	-0.2214	-0.3872	-0.0093	0.31389	0.02774	0.50501	0.25137	-0.1295	0.04285	0.01625	7.8	0																														
13	10	0.38498	0.61611	-0.8743	-0.094	2.92458	3.31703	0.47045	0.53825	-0.5589	0.30976	-0.2591	-0.3261	-0.09	0.36283	0.9289	-0.1295	-0.81	0.35999	0.70766	0.12599	0.04992	0.23842	0.00913	0.99671	-0.7673	-0.4922	0.04247	-0.5453	9.99	0																														
14	11	-1.25	-1.2216	0.38593	-1.2349	1.4854	-0.7532	0.6894	-0.2275	-2.094	1.32373	0.22767	-0.2427	1.20542	-0.3176	0.72567	-0.8156	0.87394	-0.8478	-0.6832	-0.1028	-0.2318	-0.4833	0.08467	0.39283	0.10113	-0.355	0.02642	0.04242	121.5	0																														
15	11	1.06057	0.28772	0.82861	2.71252	-0.1784	0.35754	-0.0967	0.11598	-0.2211	0.46023	-0.7737	0.32339	-0.011	-0.1785	-0.6556	-0.1999	0.12401	-0.9895	0.9829	-0.1532	-0.0399	0.07441	-0.0714	0.10474	0.54626	0.10409	0.02149	0.02129	27.5	0																														
16	12	-2.7919	-0.3278	1.64175	1.76747	-0.1366	0.8078	-0.4229	-1.9071	0.75571	1.15109	0.84456	0.76294	0.37045	-0.735	0.4068	-0.3031	-0.1559	0.77827	2.22187	-1.5821	1.15196	0.22218	1.02059	0.02832	-0.2327	-0.2556	-0.1648	-0.3902	58.8	0																														
17	12	-0.7524	0.34549	2.05732	-1.4688	-1.1584	-0.0778	-0.6086	0.0036	-0.4362	0.74773	-0.794	-0.7704	1.04703	-1.0666	1.10695	1.60011	-0.2793	-0.42	0.43254	0.26345	0.49962	1.35365	-0.2596	-0.0651	-0.0391	-0.0871	-0.181	0.12999	15.99	0																														
18	12	1.10322	-0.0403	1.26733	1.28999	-0.736	0.28897	-0.5861	0.18938	0.78233	-0.268	-0.4503	0.93671	0.7038	-0.4698	0.35457	-0.2496	-0.0092	-0.5959	-0.5757	-0.1139	-0.0246	0.196	0.0138	0.10378	0.3643	-0.3823	0.09281	0.03705	12.99	0																														
19	13	-0.4369	0.91897	0.92459	-0.7272	0.91568	-0.1279	0.70764	0.89796	0.6653	-0.738	0.3241	0.27719	0.25362	0.2919	-0.1845	1.14317	-0.9287	0.68047	0.02544	-0.047	0.1948	-0.0726	0.1559	-0.9884	-0.3424	-0.049	0.07699	0.13102	0.99	0																														
20	14	-5.4013	-0.4501	1.1863	1.73624	0.30411	-1.7634	-1.5597	0.16084	1.23309	0.34517	0.91723	0.97012	-0.2666	-0.4791	0.5266	0.472	0.7255	0.07508	-0.4069	-2.1968	-0.5036	0.98446	2.45859	0.04212	-0.4816	-0.6213	0.39205	0.94599	46.8	0																														
21	15	1.49294	-1.0293	0.45479	-1.438	-1.5554	-0.721	-1.0807	-0.0531	-1.9787	1.63888	1.07754	-0.832	-0.417	0.05201	-0.043	-0.1664	0.30424	0.55443	0.05423	-0.3879	-0.1776	-0.1751	0.04	0.29581	0.33293	-0.2204	0.0223	0.0076	5	0																														
22	16	0.09488	-1.3618	1.02522	0.83416	-1.1912	1.39911	0.8798	0.44529	-0.4462	0.58652	1.01915	1.26933	0.42048	-0.3727	-0.808	-0.2646	0.51588	0.62585	-1.3004	-0.1383	-0.2596	-0.572	-0.0509	0.5732	-0.0509	0.3042	0.072	-0.4622	0.06655	0.0635	231.71	0																												
23	17	0.9625	0.28246	-0.1715	2.1092	1.12957	1.89994	0.10771	0.5215	-1.1913	0.724	1.69033	0.49677	-0.8364	0.98574	0.71091	-0.6022	0.40248	1.7372	0.0276	0.2693	0.144	0.40249	-0.0485	-1.5719	0.99691	0.15966	0.01637	-0.146	34.09	0																														
24	18	1.16662	0.50212	-0.0673	2.26157	-0.4288	0.08947	0.24115	0.13808	-0.9892	0.92217	0.74479	-0.5314	-0.2193	1.12687	0.00308	0.42442	-0.4545	-0.0989	-0.8166	-0.3072	0.0187	-0.062	-0.1039	-0.3704	0.6032	0.10556	-0.0405	-0.0114	2.28	0																														
25	18	0.24749	0.27767	1.18547	-0.0926	-1.3144	-0.1501	-0.9464	-1.6179	1.54407	-0.8299	-0.5832	0.52493	-0.4534	0.08139	1.5552	-1.9999	0.78313	0.43662	2.17781	-0.231	1.65018	0.20045	-0.1854	0.42307	0.20209	-0.2278	0.39893	0.25048	22.75	0																														
26	22	1.9465	-0.0449	0.4056	-1.0131	2.94197	2.95505	-0.0631	0.85555	0.04997	0.57374	-0.0813	-0.2157	0.04418	0.0339	1.19072	0.57884	-0.9757	0.04406	0.4898	-0.2167	-0.5795	-0.7992	0.8708	0.98342	0.3212	0.14965	0.70752	0.0146	0.89	0																														
27	22	0.7043	-0.1215	1.32022	0.41001	0.2652	-0.9595	0.54099	-0.1048	0.47596	1.49445	0.8596	-0.1805	-0.6552	0.2798	-0.2117	0.2333	0.01075	-0.4685	0.50575	-0.5867	-0.4036	0.2274	0.74243	0.39893	0.24921	0.2744	0.55997	0.24323	26.43	0																														
28	23	1.17328	0.5535	0.28391	1.13356	-0.1726	-0.9181	0.36902	-0.3273	-0.2467	-0.0461	-0.1434	0.97935	1.49229	0.10142	0.76148	-0.0146	-0.5116	-0.2251	-0.3909	0.02788	0.067	0.22781	-0.1505	-0.43505	0.72482	-0.3371	0.01637	0.03004	41.88	0																														
29	23	1.32271	-0.174	0.43456	0.57604	-0.8368	-0.8311	-0.2649	-0.221	-1.0714	0.86856	0.6415	-0.1113	0.36149	0.17195	0.78217	-1.3559	-0.2169	1.27177	-1.2406	-0.523	-0.2844	-0.3234	-0.0377	0.34715	0.59594	-0.2802	0.04234	0.02882	16	0																														
30	23	0.4143	0.8054	1.72745	1.74737	-0.0744	-0.2003	0.74023	-0.0292	0.5954	-0.9462	-0.0121	0.7689	0.63595	-0.0863	0.7089	-1.4059	0.7559	-0.9429	0.54397	0.09731	0.07724	0.45733	-0.0386	0.64252	-0.1839	-0.2775	0.32699	0.15266	53	0																														
31	23	1.05059	-0.1753	1.26613	1.18611	-0.786	0.57344	-0.7671	0.40105	0.6895	0.0647	1.04029	1.00562	-0.542	-0.0399	-0.2187	0.00448	-0.1998	0.04239	-0.2778	-0.178	0.01386	0.21373	0.01446	0.00295	0.29404	-0.3951	0.98146	0.02422	12.99	0																														
32	24	1.23743	0.06104	0.38053	0.76156	-0.3598	-0.4941	0.00649	-0.1339	0.43881	-0.2074	-0.9292	0.52711	0.34868	-0.1525	-0.2184	-0.1916	-0.1166	-0.6338	0.34842	-0.0664	-0.2457	-0.5309	-0.0443	0.07917	0.50914	0.28886	-0.0227	0.01184	17.28	0																														
33	25	1.11401	0.08555	-0.4937	1.33576	-0.3002	-0.0108	-0.1188	0.18882	0.20569	0.08226	1.13356	0.6267	-1.4928	0.52079	0.6746	-0.5291	0.15826	-0.3988	-0.1457	-0.2738	-0.0532	-0.0048	-0.0315	0.19805	0.95501	0.3377	0.02906	0.00445	4.45	0																														
34	26	0.5299	0.87389	1.34725	0.14548	0.41421	0.1002																																																						

Dữ liệu đại diện cho class 1

time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	Amount	Class	Label
406	-2.3122	1.96199	-1.6099	3.99791	-1.02221	1.42605	-2.6374	1.39166	-2.7701	-2.7723	3.20203	-2.8999	-0.5952	-4.2893	0.98972	-1.1407	-2.8301	-0.0168	0.41696	0.12691	0.51723	-0.035	-0.6552	0.3202	0.04452	0.17784	0.26115	-0.1433	0	1		
472	-2.0435	-3.1573	1.08846	2.28864	1.35981	1.0648	0.32557	-0.0678	-0.271	-0.8386	-0.4146	-0.5031	0.76765	-1.692	2.00683	0.66678	0.59972	1.25532	0.28334	2.10234	0.6617	0.43548	1.37597	-0.2938	0.2798	-0.1454	-0.2528	0.03576	529	1		
4462	-2.3033	1.75925	-0.3597	2.33024	-0.8216	0.07758	0.56232	-0.3991	-0.2383	-1.0254	2.03291	-0.5601	0.02294	-1.4701	-0.6988	-2.2822	-0.7818	-2.6157	-1.3344	-0.43	-0.2942	-0.9324	0.17273	-0.0873	-0.1561	-0.5426	0.03957	-0.153	239.93	1		
6986	-4.398	1.53537	-2.5928	2.67979	-1.1281	1.7095	-0.9602	-0.2488	-0.2478	-4.8016	4.89584	-10.913	0.18437	-0.7711	-0.0073	7.3891	-12.598	-5.1015	0.30833	-0.1718	0.57357	0.17697	-0.4362	-0.0539	0.25241	-0.6575	-0.8271	0.84957	59	1		
7519	1.23424	3.01974	-4.2046	4.7328	0.9542	1.5777	1.71344	-0.4664	1.2825	2.4475	2.10134	-4.6956	1.46438	-0.6793	-0.2392	2.58185	6.73938	3.04249	-2.7219	0.00906	-0.3791	-0.7042	-0.6568	-1.6327	1.4889	0.5968	-0.01	0.14679	1	1		
7526	0.00943	4.13784	-6.2407	6.67573	0.76831	-3.8351	-1.6317	0.15461	-2.7959	-6.1879	5.66439	-9.8545	-0.3062	-10.691	-0.6385	-2.042	-1.1291	0.11645	-1.9347	0.48838	0.36451	-0.6081	-0.3595	0.12894	1.48848	0.50796	0.73582	0.51357	1	1		
7535	0.02678	4.13246	-6.5606	6.34856	1.23967	-2.5135	-1.6891	0.30325	-1.1394	-0.6455	6.75463	-9.8482	0.70272	-10.734	-1.3795	-1.639	-1.7464	0.77674	-1.3274	0.58774	0.37051	-0.5768	-0.6696	-0.7599	1.60506	0.54068	0.73704	0.4967	1	1		
7543	0.32959	3.71289	-5.7759	6.07827	1.66786	-2.4202	-0.8129	0.13308	-2.2143	-5.1345	4.56072	-8.6737	-0.7975	-0.1772	-0.257	0.8717	1.31301	0.77391	-2.3706	0.26977	0.15662	-0.6525	-0.5516	-0.7165	1.41572	0.55526	0.53051	0.40447	1	1		
7551	0.31646	3.80968	-5.6152	6.04745	1.55403	-2.6514	-0.7466	0.03559	-2.6787	-4.9595	6.43965	-7.5201	0.38835	-0.2523	-1.3652	0.5024	0.78443	1.4843	-1.808	0.38831	0.20883	-0.5117	-0.3638	-0.2198	1.47475	0.49119	0.51887	0.40253	1	1		
7610	0.72595	2.90059	-5.33	4.00789	-1.7304	-1.7322	-3.9586	1.00373	-0.4961	-4.625	5.58972	-7.1482	1.68045	-0.2103	0.49529	-3.5665	-4.8303	-0.6491	2.26112	0.50455	0.59867	0.10554	0.60105	-0.3647	-1.8431	0.35191	0.59455	0.09937	-1	1		
7672	0.70271	2.42643	-5.2345	4.41666	-1.7708	-2.6676	-8.7871	0.91134	-0.1662	-5.0092	6.67573	-8.1672	0.63856	-6.7633	1.29686	-3.8118	-3.7541	-1.0492	1.5542	0.42274	0.55118	-0.0098	0.7217	0.47325	-1.9593	0.31948	0.60048	0.12931	1	1		
7740	1.02387	2.00149	-4.7968	3.81919	-1.2718	-1.7347	-0.0592	0.8898	0.41538	-3.9558	3.57205	-7.1865	0.14724	-3.2493	1.67833	-2.6415	-1.3121	-0.3917	1.11826	0.20414	0.34328	-0.0542	0.70965	-0.3722	-0.0321	0.36678	0.39517	0.02021	1	1		
7891	-1.5855	3.26158	-4.1374	2.3571	-1.405	-1.8794	-3.5137	1.51561	-1.2072	-6.2346	5.45075	-7.3337	1.98119	-6.6081	-0.4811	-2.6025	-4.8351	-0.553	0.35195	0.31596	0.50154	-0.5499	-0.0766	-0.4256	0.12364	0.32198	0.26403	0.13282	1	1		
8090	-1.7832	3.40279	-3.8227	2.62537	-1.9764	-2.7317	-3.4306	1.4132	-0.7769	-6.1959	4.36071	-8.2433	0.34578	-5.5906	0.26558	-3.0385	-2.2145	-1.1136	-0.2654	0.36409	0.45403	-0.5775	0.04597	-0.4617	0.04415	0.30357	0.33096	0.24376	1	1		
8169	0.85732	4.00591	-7.4239	7.38024	0.97337	-2.7308	-1.4965	0.54302	-2.5512	-3.9442	6.35508	-7.3097	0.74845	-0.056	-0.6489	-1.0731	1.5245	1.83136	-0.0897	0.4833	0.37503	0.1454	0.2406	-0.2346	-1.0049	0.43583	0.81332	1.4847	1	1		
8408	-1.8133	4.91785	-5.9261	5.7015	1.20439	-3.0351	-1.7134	0.56126	-3.7964	-7.4548	7.38806	-10.475	-0.3793	-11.737	-0.207	-2.4424	-3.5355	0.13036	-0.2715	0.57666	0.61564	-0.4064	-0.737	-0.2796	1.10677	0.32389	0.89477	0.56952	1	1		
8415	-0.2515	4.31352	-6.8914	6.7968	0.6163	-2.9663	-2.4367	0.48933	-3.716	-6.8108	7.62009	-10.285	-0.3424	-11.543	-1.335	-2.6893	-3.2044	0.08652	-1.3145	0.63271	0.53689	-0.5461	-0.6052	-0.2637	1.59992	0.52357	0.89102	0.57274	1	1		
8451	0.3148	2.60687	-5.92	4.5225	-2.315	-2.2784	-0.6841	1.20227	-0.6947	-5.5263	6.66244	-8.5255	0.74275	-7.6787	0.59307	-4.4781	-8.8443	-1.1027	2.17739	0.56271	0.74331	0.06404	0.67784	0.08301	-1.911	0.32219	0.62087	0.18503	1	1		
8529	0.4474	2.48185	-5.6968	4.45992	-2.4438	-2.1185	-4.7161	1.2468	-0.7163	-5.3903	6.45419	-8.4853	0.63528	-7.0199	0.59981	-4.5499	-6.2894	-1.5393	2.26296	0.54981	0.75605	0.14017	0.66541	0.13146	-1.9082	0.33481	0.74853	0.17541	1	1		
8614	-2.1699	3.63965	-4.5085	2.73067	-2.1227	-2.341	-4.2353	1.70354	-1.3053	-6.7167	6.35361	-8.6016	0.44993	-7.5062	-0.4381	-3.6945	-6.3048	-1.2676	0.35799	0.50078	0.6451	-0.5035	-0.0005	0.0717	0.09201	0.3085	0.55259	0.29985	1	1		
8757	-1.8638	3.44264	-4.4683	2.80534	-2.1184	-2.3323	-2.6212	1.70168	-1.4394	-6.9999	6.31621	-8.6708	0.31602	-7.4177	-0.4365	-3.6528	-6.2931	-1.2432	0.36481	0.36092	0.66793	-0.5162	-0.1212	0.07061	0.0585	0.30488	0.50760	0.12088	1	1		
8808	-4.6172	1.69569	-3.1144	4.3282	-1.8738	-0.9899	-4.5773	0.47222	0.47202	-5.578	4.80232	-10.833	0.1043	-4.4054	-0.8076	-7.5523	-9.8026	-4.1206	1.74051	-0.039	0.48183	0.14602	0.11704	-0.2176	-0.1388	-0.4245	-1.002	0.89078	1.1	1		
8870	2.6618	5.85639	-7.6536	6.37974	-0.0607	-7.3135	-3.1036	1.77849	-8.3312	-7.1916	7.10299	-9.9287	-0.0675	-10.924	-1.6979	2.3794	-2.7751	0.2738	-1.3822	0.3991	0.73477	-0.4359	-0.3846	-0.296	1.00793	0.4132	0.26028	0.3034	1	1		
8886	-2.5359	5.79364	-7.6185	6.36583	-0.0652	-3.1364	-3.1046	1.62323	-3.8787	-7.2978	7.07167	-10.001	-0.2079	-10.861	-1.6968	-2.6069	-2.7601	0.2581	-1.3856	0.4087	0.71672	-0.4481	-0.4024	-0.2888	1.01175	0.42596	0.41314	0.3082	1	1		
9064	-3.4901	0.25856	-4.4896	4.85389	-6.9745	6.82838	5.43127	-1.9467	-0.7757	-1.9678	4.6904	-6.998	1.45401	-3.738	0.31774	2.0135	-1.1381	-1.1838	1.66339	-0.0426	-1.0524	0.20482	-2.119	0.17028	-0.3938	0.29637	1.98991	-0.9005	1809.68	1	1	
11080	-2.1255	5.97356	-1.1035	9.00715	-1.6895	-2.8544	-7.8104	0.20387	-5.9028	-12.841	12.0189	-17.769	-0.431	-19.214	-0.9625	-10.267	-15.503	-5.4949	-0.4105	1.49377	1.64652	-0.2785	-0.6648	-1.1646	1.7018	0.69081	2.11975	1.10893	1	1		
11092	0.37827	3.9148	-5.7269	6.09414	1.69888	-2.8073	-0.5911	-0.1235	-2.5307	-5.1531	0.85409	-7.8395	1.37182	-6.5347	-0.7396	0.6632	0.89193	0.97868	-2.0055	0.44044	0.1499	-0.602	-0.0137	-0.4031	1.56845	0.52188	0.52794	0.41191	1	1		
11131	-1.4266	4.14189	-9.8041	6.66627	-4.7495	-2.0731	-10.09	2.79158	-3.2496	-11.42	10.853	-15.969	0.54669	-14.691	0.91234	12.227	-18.587	-6.9208	3.167	1.41068	1.88568	0.40781	0.60561	-0.7693	-1.7483	0.50204	1.97726	0.71161	1	1		
11629	-3.8912	7.06982	-11.426	8.60756	-2.0657	-2.9683	-8.1386	2.79393	-6.2728	-13.193	11.6197	-17.632	-0.3552	-18.822	-1.2831	-10.031	-15.227	-5.322	-0.5018	1.38262	1.75709	-0.1897	-0.5098	-1.1893	1.18854	0.60524	1.88153	0.87526	1	1		
11635	0.91914	4.19963	-7.5356	7.42694	1.1822	-2.8867	-1.341	0.36393	-2.2032	-4.1378	4.57011	-7.6292	1.73392	-9.4404	-0.0234	-1.234	1.63201	1.31573	-0.2872	0.35543	0.31609	0.05518	0.21069	-0.4179	-0.9112	0.46652	0.62739	1.51785	1	1		
12093	-4.6968	2.69387	-4.7551	5.46768	-1.5568	-1.5494	-4.1042	0.55393	-1.4985	-4.595	5.27551	-11.349	0.37455	-8.1387	0.54857	-6.6536	-10.247	-1.9111	0.99149	-0.159	0.5739	-0.0802	0.31841	-0.2459	0.33824	0.03227	-1.5085	0.60807	0	1		
12095	-4.7277	3.04447	-5.5964	5.92819	-2.1908	-1.5293	-4.4874	0.91639	-1.307	-4.1389	5.14941	-11.124	0.54207	-7.8409	0.74383	-6.7771	-9.9318	-4.093	1.50492	-0.2078	0.65999	0.25498	0.62884	-0.2381	-0.6713	-0.0336	-1.3318	0.7057	30.39	1		
12393	-4.064	3.10052	-1.1885	3.26463	-1.9808	0.32035	-0.5649	-3.2775	-2.82093	1.01511	0.18719	-7.0045	0.87271	-6.2206	-0.9044	-3.0751	-5.0447	-1.7181	-0.6625	-0.5319	1.68967	-0.0788	0.19373	-0.4795	-0.5666	-0.4099	-3.0383	-0.6366	179.66	1	1	
12597	-2.5896	7.01671	-13.705	10.3432	-2.9545	-3.0551	-3.3043	3.49567	-5.6542	-11.854	11.6692	-17.229	0.05557	-18.494	-0.3042	-10.629	-14.441	-5.1055	1.27023	1.48886	1.88774	0.334	0.28769	-0.7784	-1.6468	0.48754	1.42771	0.58317	1	1		
13126	-2.88	5.22544	-11.063	6.68995	-5.7999	-2.244	-11.2	4.01472	-4.2933	-11.562	10.4468	-15.479	0.73444	-13.884	0.82144	-11																

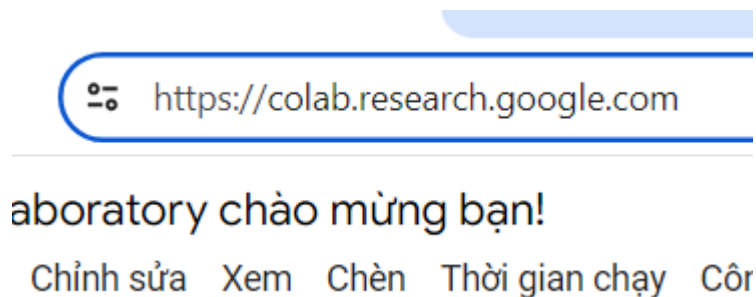
là đạt, không thì cần phải xem xét lại. Và để kiểm nghiệm được độ chính xác của mô hình này, người ta dùng tập Testing set. Khác với Training set, Testing set chỉ gồm các giá trị input (TD, GC, YKNTK, TI, và RRCN) mà không có giá trị output (YDM).

3.4.2.3 Quy trình chia dữ liệu trên mô hình thực tế

Ở đây, để thực hiện chia dữ liệu thành tập huấn luyện và tập kiểm tra, mình dùng thư viện `sklearn.model_selection` và hàm `train_test_split` có sẵn trong python để chia dữ liệu.

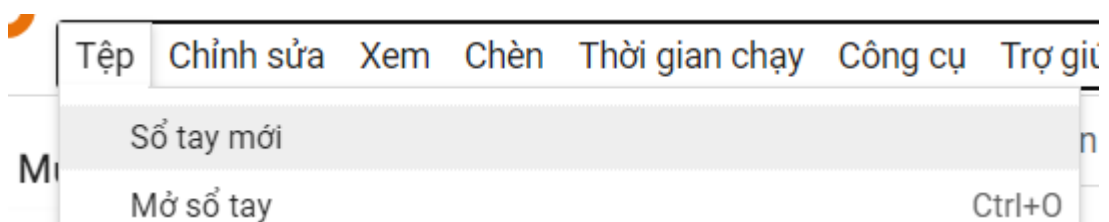
Trong bài toán thực tế trên đồ án này, mình chia tỷ lệ 70% cho tập huấn luyện và 30% cho tập kiểm tra.

Truy cập vào trang google colab trên địa chỉ <https://colab.research.google.com/>



Hình 3. 6 Trang chủ colab

Tạo sổ tay mới: ấn Tập → Sổ tay mới Khai báo dataset cần chia trong google driver



Hình 3. 7 Mở sổ tay mới

```

5
6 # Kết nối với Google Drive
7 drive.mount('/content/drive')
8
9 # Đường dẫn đến file CSV trên Google Drive
0 file_path = '/content/drive/MyDrive/DoAnTotNghiep/creditcard.csv'
1
2 # Đọc file CSV
3 df = pd.read_csv(file_path)

```

Hình 3. 8 Khai báo dataset

Tách các biến và nhãn

```

16 # X_train, X_test, y_train, y_test = train_test_split
17 # Tách các features và nhãn
18 X_resampled = df.drop('Class', axis=1) # features
19 y_resampled = df['Class'] # nhãn
20 |

```

Hình 3. 9 Tách các biến và nhãn data

Tính toán tỷ lệ mẫu kiểm tra dựa trên số lượng mẫu sau khi resampling

```

# Tính toán tỷ lệ mẫu kiểm tra dựa trên số lượng mẫu sau khi resampling
test_ratio = 0.3 # tỷ lệ mẫu kiểm tra mong muốn
num_total_samples = len(X_resampled)
num_test_samples = int(test_ratio * num_total_samples)

```

Hình 3. 10 Tỷ lệ mẫu

Chia dữ liệu thành tập huấn luyện và kiểm tra

```

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ mẫu kiểm tra đã tính toán
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=num_test_samples, shuffle=True, random_state=42)

```

Hình 3. 11 Chia data

Kích thước sau khi chia với tỷ lệ 30% cho tập kiểm tra, 70% cho tập huấn luyện.

```

Kích thước mẫu kiểm tra: 85442
Kích thước mẫu huấn luyện: 199365

```

Hình 3. 12 Kích thước sau khi chia

3.4.3 Mô hình phân lớp sử dụng

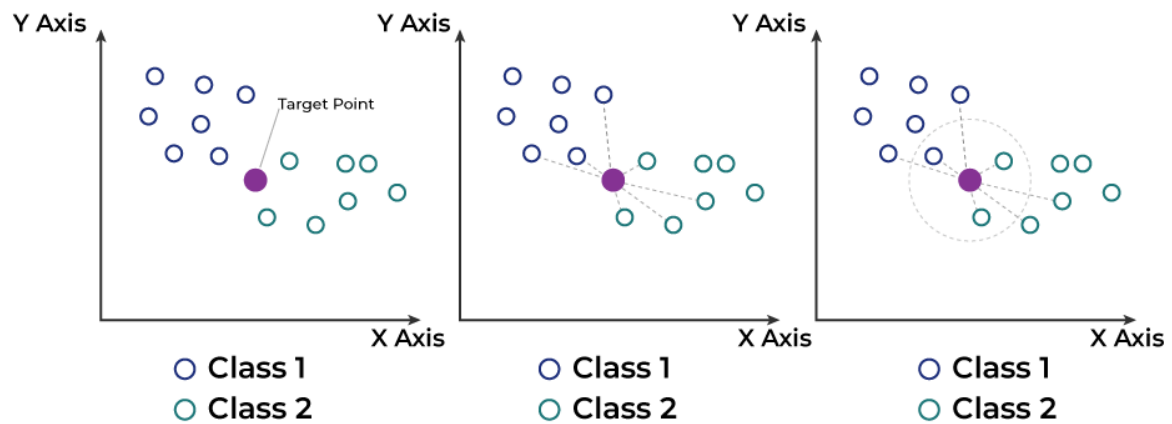
3.4.3.1 KNN

KNN hoạt động dựa trên nguyên tắc là tìm K hàng xóm gần nhất trên các điểm dữ liệu mới, từ đó đưa ra được kết luận.

Các bước thực hiện trong KNN

- Đo khoảng cách đến điểm dữ liệu gần đó, thường dùng Euclid.
- Xác định K là hàng xóm gần nó xem cái nào phổ biến nhất, K do người thiết kế chọn.

- Xác định xem điểm dữ liệu nào phổ biến nhất hoặc trung bình điểm dữ liệu quanh nó để đưa ra kết luận.



Hình 3. 13 Mô tả KNN

(nguồn: K-Nearest Neighbor(KNN) Algorithm) [13]

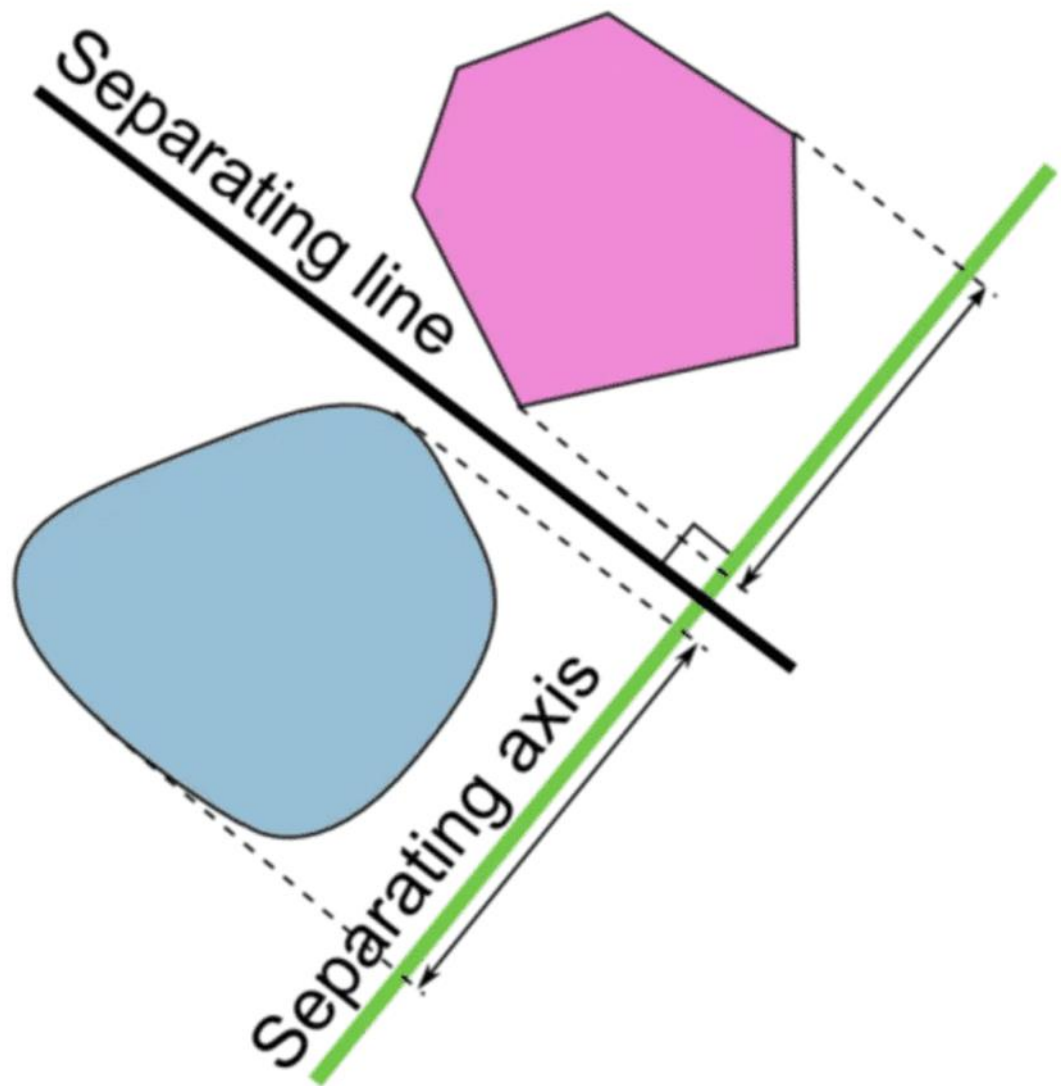
Vì thế, KNN dùng được cho cả phân loại và hồi quy.

3.4.3.2 SVM

Mục tiêu của SVM là tìm ra một siêu phẳng trong không gian N chiều (ứng với N đặc trưng) chia dữ liệu thành hai phần tương ứng với lớp của chúng. Nói theo ngôn ngữ của đại số tuyến tính, siêu phẳng này phải có lẽ cực đại và phân chia hai bao lồi và cách đều chúng.

Để phân chia hai lớp dữ liệu, rõ ràng là có rất nhiều siêu phẳng có thể làm được điều này. Mặc dù vậy, mục tiêu của chúng ta là tìm ra siêu phẳng có lẽ rộng nhất tức là có khoảng cách tới các điểm của hai lớp là lớn nhất. Hình dưới đây là một ví dụ trực quan về điều đó.

[6]

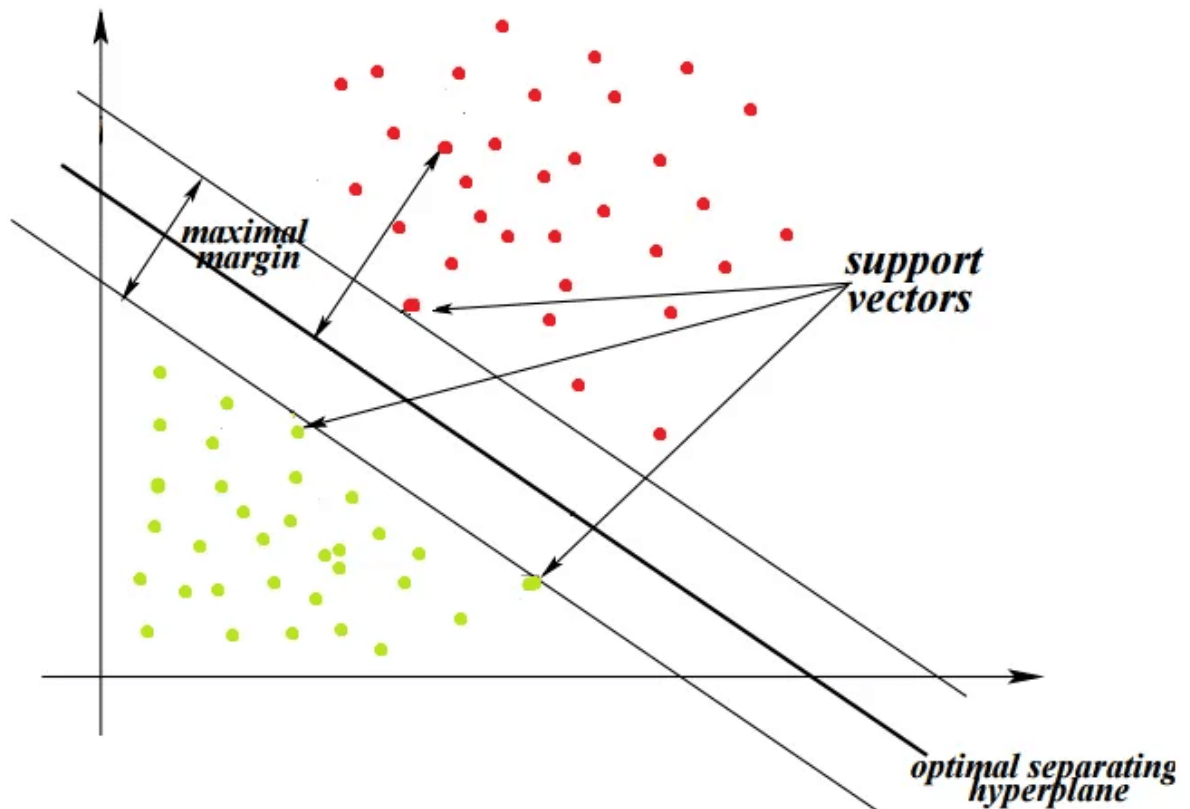


Hình 3. 14 Siêu phẳng SVM

(Nguồn: SVM quá khó hiểu! Hãy đọc bài này) [6]

Các véc tơ hỗ trợ

Một điểm trong không gian véc tơ có thể được coi là một véc tơ từ gốc tọa độ tới điểm đó. Các điểm dữ liệu nằm trên hoặc gần nhất với siêu phẳng được gọi là véc tơ hỗ trợ, chúng ảnh hưởng đến vị trí và hướng của siêu phẳng. Các véc tơ này được sử dụng để tối ưu hóa lề và nếu xóa các điểm này, vị trí của siêu phẳng sẽ thay đổi. Một điểm lưu ý nữa đó là các véc tơ hỗ trợ phải cách đều siêu phẳng. [6]



Hình 3. 15 Vector hỗ trợ SVM

(Nguồn: SVM quá khó hiểu! Hãy đọc bài này) [6]

3.4.3.3 Random Forest

Thuật toán này dựa trên ý tưởng là xây dựng nhiều cây quyết định thành một rừng cây quyết định để giảm hiện tượng overfitting trong học máy.

Nguyên tắc hoạt động:

Chọn mẫu ngẫu nhiên: Với mỗi cây quyết định trong rừng, một mẫu con dữ liệu được lấy ngẫu nhiên từ tập dữ liệu huấn luyện. Điều này giúp đa dạng hóa các cây quyết định trong rừng.

Xây dựng cây quyết định: Mỗi cây quyết định được xây dựng bằng cách chọn thuộc tính tốt nhất từ một tập hợp ngẫu nhiên của các thuộc tính và sử dụng thuộc tính đó để phân chia tập dữ liệu thành các nhóm. Quá trình này được lặp lại đến khi cây quyết định đạt được một tiêu chí dừng.

Tính toán dự đoán: Sau khi cây quyết định được xây dựng, mỗi mẫu dữ liệu trong tập kiểm tra được đưa qua mỗi cây quyết định và được dự đoán một nhãn. Trong trường

hợp phân loại, phần đông phiếu bầu của tất cả các cây được sử dụng làm dự đoán cuối cùng. Trong trường hợp dự đoán, trung bình của các dự đoán từ tất cả các cây được sử dụng.

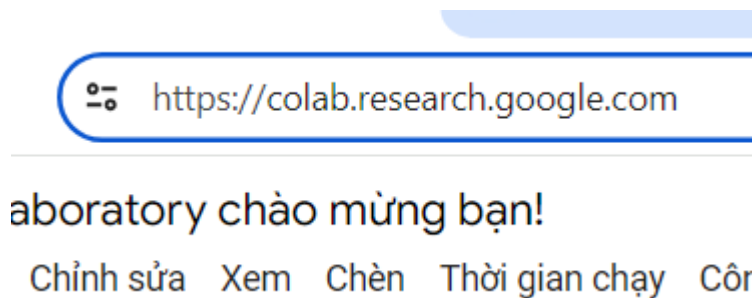
Tính toán độ chính xác: Độ chính xác của Random Forest được tính bằng cách so sánh dự đoán của mô hình với nhãn thực tế trong tập kiểm tra hoặc tập dữ liệu không nhìn thấy.

Tinh chỉnh tham số (nếu cần): Các tham số của mô hình như số cây, độ sâu của cây, và số lượng mẫu con có thể được tinh chỉnh bằng cách sử dụng kỹ thuật tinh chỉnh siêu tham số như Grid Search hoặc Random Search để cải thiện hiệu suất của mô hình.

3.5 Triển khai mô hình và huấn luyện

3.5.1 Môi trường cài đặt

Để sử dụng Google Colab cho việc thử nghiệm mô hình, đầu tiên, truy cập vào trang google colab trên địa chỉ <https://colab.research.google.com/> như hình 3.1



Hình 3. 16 Trang chủ colab

Sau đó, chạy câu lệnh như hình 3.2 dưới đây để kết nối với google driver

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Hình 3. 17 Kết nối gg driver

Vậy là đã xong các bước thiết lập ban đầu để cài đặt Colab trong môi trường thiết kế. Các tập dữ liệu và các chương trình sẽ được lưu ở trong google driver và sẽ được thao tác trực tiếp với google driver.

3.5.2 Phương pháp xây dựng bài toán

Ban đầu, cần phải đưa tệp dữ liệu sau khi đã trải qua quá trình thu thập và tiền xử lý, chuẩn hóa như ở trên Chương 2 mình đã nói. Bài toán gồm các hàm chính

Hàm `df = pd.read_csv` sẽ đọc dữ liệu từ file CSV để đưa vào bài toán.

Hàm `X_resampled` dùng để chọn các biến cần sử dụng làm biến đầu vào.

Hàm `y_resampled` dùng để gán nhãn dữ liệu cho mô hình.

Cụm hàm `test_ratio`, `num_total_samples`, `num_test_samples` dùng để tính đoán độ dài dữ liệu cần thiết cho tập kiểm tra.

Hàm `train_test_split` dùng để chia dữ liệu thành tập huấn luyện và kiểm tra.

Hàm `knn_model` được dùng để khởi tạo và huấn luyện mô hình KNN. Các mô hình khác sẽ có hàm khởi tạo và huấn luyện khác nhau nhưng đều có `_model` phía sau.

Hàm `y_pred` dùng để dự đoán nhãn trên tập kiểm tra.

Hàm `classification_report(y_test, y_pred)` dùng để đánh giá mô hình.

Hàm `predicted_class` dùng để dự đoán nhãn dữ liệu nhập bên ngoài.

Hàm `confusion_matrix` để tạo ma trận nhầm lẫn giữa thực tế và dự đoán.

Chi tiết các đoạn mã chương trình từng mô hình thuật toán ở phần phụ lục.

3.5.3 Quy trình huấn luyện mô hình bài toán

3.5.3.1 Các tham số trong thư viện sử dụng

3.5.3.1.1 Các thư viện sử dụng chung trong bài toán

`Import pandas as pd`: Dùng để thao tác dữ liệu, tạo bảng dataframe và Series.

`From sklearn.model_selection import train_test_split`: Chia dữ liệu thành tập train (huấn luyện) và test (đánh giá).

`From sklearn.metrics import classification_report`: Dùng để in ra báo cáo phân loại trong tập kiểm tra.

`Confusion_matrix`: Tạo ma trận nhầm lẫn để trực quan hóa hiệu suất mô hình (đúng, sai cho từng lớp).

`Import matplotlib.pyplot as plt`: Tạo biểu đồ tĩnh, động và tương tác trong Python.

Import seaborn as sns: Tạo biểu đồ thống kê đẹp mắt với cú pháp đơn giản hơn Matplotlib.

Import time: Làm việc với thời gian trong Python như lấy thời gian hiện tại, tạm dừng, định dạng thời gian.

3.5.3.1.2 Thư viện cho các thuật toán

a) KNN

From sklearn.neighbors import kneighborsclassifier: Dùng để khởi tạo mô hình KNN với các thao tác trong mô hình

b) SVM

From sklearn.svm import SVC: Thuật toán SVM để phân loại dữ liệu.

Hinge_loss: Tính toán hàm mất mát hinge (đo sai lệch dự đoán).

Randomoversampler: Cân bằng số lượng mẫu giữa các lớp (xử lý mất cân bằng lớp).

c) Random Forest

From sklearn.ensemble import randomforestclassifier: Thư viện này cung cấp thuật toán Rừng ngẫu nhiên (Random Forest) để phân loại dữ liệu. Thuật toán này tạo ra nhiều mô hình cây quyết định ngẫu nhiên và kết hợp dự đoán của chúng để cải thiện độ chính xác và giảm thiểu quá phụ thuộc.

From imblearn.over_sampling import randomoversampler: Thư viện này cung cấp kỹ thuật lấy mẫu quá mức ngẫu nhiên (Random oversampling) để xử lý vấn đề mất cân bằng lớp (class imbalance) trong dữ liệu. Kỹ thuật này tạo ra các bản sao ngẫu nhiên của các mẫu thuộc nhóm thiểu số để cân bằng số lượng mẫu với các nhóm khác, giúp mô hình phân loại hiệu quả hơn.

3.5.3.2 Các hàm trong quá trình xây dựng bài toán

Bước 1, dữ liệu được load từ google driver qua hàm `df = pd.read_csv` để lấy dữ liệu đầu vào.

Bước 2, `X_resampled = df` và `y_resampled = df` sẽ lấy dữ liệu và phân chia thành các biến dùng để dự đoán và biến nhãn, sau đó hàm `num_test_samples` sẽ lấy số lượng mẫu của tập kiểm tra đưa vào hàm `train_test_split` để chia dữ liệu thành tập huấn luyện và tập kiểm tra.

Bước 3, hàm `knn_model` bắt đầu khởi tạo và huấn luyện mô hình, `_model` có thể thay đổi theo tên mô hình sử dụng.

Bước 4, hàm `classification_report` sẽ dùng để đánh giá mô hình học máy.

Bước 5, hàm `predicted_class` dùng để huấn luyện mô hình dựa trên dữ liệu được nhập từ bên ngoài và xuất ra kết quả huấn luyện.

Ở thuật toán SVM và Random Forest các bước tương tự như trên. Chỉ khác rằng hàm khởi tạo và huấn luyện mô hình trên SVM là `svm_model`. Còn với Random Forest là khởi tạo và huấn luyện mô hình trên hàm `rf_model`.

3.6 Phương pháp đánh giá kết quả

3.6.1 Cách đo độ đánh giá

Để đánh giá và so sánh hiệu quả của từng phương pháp học máy, ta sử dụng phương pháp Accuracy (Độ chính xác). Độ chính xác được định nghĩa là tỷ lệ giữa số dự đoán đúng và tổng số dự đoán được thực hiện.

Công thức tính Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Trong đó:

TP (True Positives): Số lượng dự đoán đúng mà thực tế cũng đúng (dự đoán đúng dương tính).

TN (True Negatives): Số lượng dự đoán sai mà thực tế cũng sai (dự đoán đúng âm tính).

FP (False Positives): Số lượng dự đoán đúng nhưng thực tế sai (dự đoán sai dương tính).

FN (False Negatives): Số lượng dự đoán sai nhưng thực tế đúng (dự đoán sai âm tính).

Ngoài ra, còn sử dụng phương pháp F1-score gồm có các thông số đánh giá tổng thể như: Precision, Recall. F1 – score là thông số tổng quát của 2 cái Precision và Recall.

Precision (Độ chính xác):

Precision là tỷ lệ số lượng mẫu được phân loại chính xác là thuộc lớp dự đoán so với tổng số lượng mẫu được dự đoán là thuộc lớp đó. Nói cách khác, Precision cho biết tỷ lệ các mẫu được mô hình dự đoán là dương tính thực sự là dương tính.

- Công thức tính: $Precision = \frac{TP}{TP+FP}$
- TP (True Positives): Số lượng dự đoán đúng mà thực tế cũng đúng (dự đoán đúng dương tính).
- FP (False Positives): Số lượng dự đoán đúng nhưng thực tế sai (dự đoán sai dương tính).

Recall (Độ bao quát):

Recall là tỷ lệ số lượng mẫu thuộc lớp thực sự được mô hình phân loại chính xác là thuộc lớp đó so với tổng số lượng mẫu thuộc lớp đó. Nói cách khác, Recall cho biết tỷ lệ các mẫu dương tính thực sự được mô hình nhận diện là dương tính.

- Công thức tính: $Recall = \frac{TP}{TP+FN}$
- TP (True Positives): Số lượng dự đoán đúng mà thực tế cũng đúng (dự đoán đúng dương tính).
- FN (False Negatives): Số lượng dự đoán sai nhưng thực tế đúng (dự đoán sai âm tính).

F1-score:

F1-score là thước đo hiệu suất được tính toán từ Precision và Recall, nhằm cân bằng giữa hai chỉ số này. F1-score có giá trị từ 0 đến 1, với 1 là giá trị cao nhất, cho biết mô hình có hiệu suất phân loại tốt nhất.

- Công thức tính: $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$

3.6.2 Phương pháp đánh giá

Để đánh giá khách quan được các mô hình huấn luyện khác nhau, phương pháp được sử dụng là phương pháp "Báo cáo phân loại" (Classification Report). Phương pháp trên đánh giá các thông số của mô hình trong tập kiểm tra với các tiêu chí:

Precision (Độ chính xác): Tỷ lệ phần trăm dự đoán dương tính chính xác.

Recall (Độ thu hồi): Tỷ lệ phần trăm mẫu dương tính được dự đoán chính xác.

F1-score: Điểm cân bằng giữa precision và recall.

Accuracy (Độ chính xác): là một thước đo đơn giản và trực quan để đánh giá hiệu suất của mô hình phân loại. Nó thể hiện tỷ lệ phần trăm số lượng mẫu được dự đoán đúng so với tổng số mẫu trong tập dữ liệu.

3.7 Kết quả thực nghiệm

3.7.1 Kết quả thực nghiệm trên thuật toán KNN

Bảng 3. 2 Đánh giá mô hình trên KNN

Lớp	Precision	Recall	F1-score	Support
0	1,00	1,00	1,00	85306
1	0,50	0,01	0,03	136
Accuracy	1,0			85442

Như bảng 3.2, ta thấy như sau:

Precision (Độ chính xác của dự đoán):

Của lớp 0 đạt 1,00 nghĩa là nhận dạng được 100% các dữ liệu thuộc lớp 0 là lớp không gian lận.

Nhưng lớp 1 chỉ được 0.50 nghĩa là mô hình này chỉ dự đoán chính xác được 50/50 các giao dịch là có gian lận.

Recall (Tỷ lệ nhận dạng):

Lớp 0: 1,00 có nghĩa là mô hình có thể dự đoán được đúng thực sự 100% là giao dịch không có gian lận

Lớp 1: 0,01 nghĩa là mô hình nhận diện bao quát được tỷ lệ gian lận rất thấp, vì chỉ nhận diện thực sự được 1% giao dịch gian lận, 99% còn lại là bị bỏ qua hết. Với mô hình học máy nhận diện như thế là nhận diện gian lận rất kém.

Accuracy (độ chính xác):

Ở bảng 3.1, accuracy đạt được 1.0 nghĩa là mô hình KNN hoạt động trên tập kiểm tra của dữ liệu có thể dự đoán đúng được hầu hết tất cả các mẫu dữ liệu trong tập kiểm tra.

Dự đoán nhãn của mô hình KNN với dữ liệu tùy chọn được nhập từ bàn phím và các biến dữ liệu cố định kết hợp:

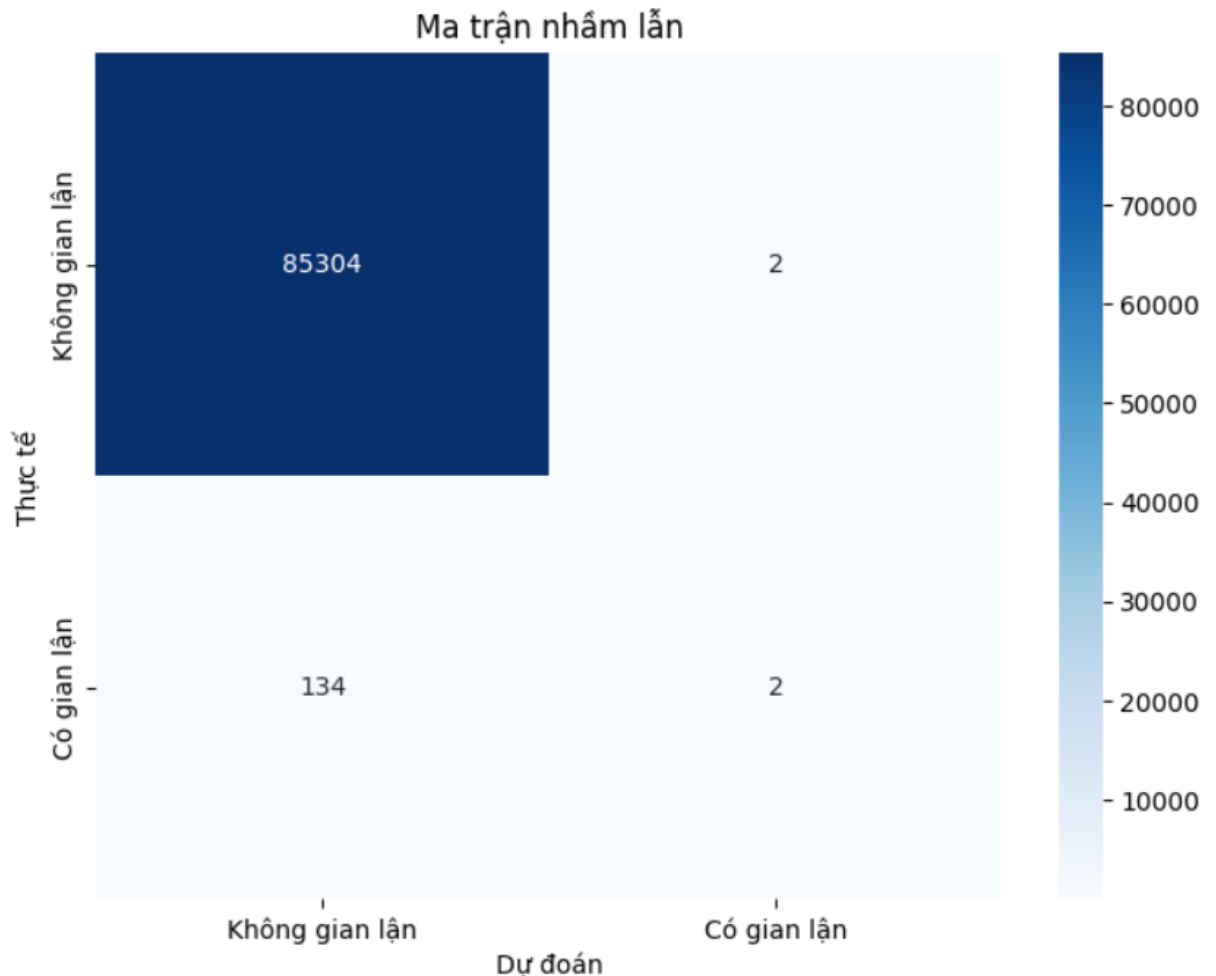
- Các biến dữ liệu cố định

```
default_V25 = 0.128539  
default_V26 = 0.125895  
default_V27 = -0.008983  
default_V28 = 0.014724
```

- Kết quả huấn luyện dữ liệu từ bàn phím

```
Nhập thời gian (Time) của giao dịch: 22  
Nhập số tiền (Amount) của giao dịch: 30  
Dự đoán: Không gian lận
```

Ma trận nhầm lẫn giữa thực tế và dự đoán trên KNN



Hình 3. 18 Ma trận nhầm lẫn KNN

Theo hình 3.18, thì:

- Giá trị trong ô (1, 1): Giá trị trong ô (1, 1) là 85.304. Điều này có nghĩa là 85.304 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán đúng là thuộc lớp "Có gian lận".
- Giá trị trong ô (1, 2): Giá trị trong ô (1, 2) là 2. Điều này có nghĩa là 2 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán sai là thuộc lớp "Không gian lận".
- Giá trị trong ô (2, 1): Giá trị trong ô (2, 1) là 134. Điều này có nghĩa là 134 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán sai là thuộc lớp "Có gian lận".
- Giá trị trong ô (2, 2): Giá trị trong ô (2, 2) là 2. Điều này có nghĩa là 2 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán đúng là thuộc lớp "Không gian lận".

3.7.2 Kết quả thực nghiệm trên thuật toán SVM

Bảng 3. 3 Đánh giá mô hình trên SVM

Lớp	Precision	Recall	F1-score	Support
0	1,00	1,00	1,00	85306
1	0,00	0,00	0,00	136
Accuracy	1,0			85442

Như bảng 3.3, ta thấy như sau:

Precision (Độ chính xác của dự đoán):

Của lớp 0 đạt 1,00 nghĩa là dự đoán được 100% các dữ liệu thuộc lớp 0 là lớp không gian lận.

Nhưng lớp 1 là 0,0 nghĩa là mô hình này không thể dự đoán được giao dịch có gian lận.

Recall (Tỉ lệ nhận dạng):

Lớp 0: 1,00 có nghĩa là mô hình có thể nhận diện được đúng thực sự 100% là giao dịch không có gian lận

Lớp 1: 0,00 nghĩa là mô hình bỏ qua hết các giao dịch có gian lận khi bao quát, không thể tìm ra được các giao dịch gian lận

Accuracy (độ chính xác):

Ở bảng 3.2, accuracy đạt được 1.0 nghĩa là mô hình hoạt động trên tập kiểm tra của dữ liệu có thể dự đoán đúng được hầu hết tất cả các mẫu dữ liệu trong tập kiểm tra.

Dự đoán nhãn của mô hình SVM với dữ liệu tùy chọn được nhập từ bàn phím và các biến dữ liệu cố định kết hợp:

- Các biến dữ liệu cố định:

```
default_V25 = 0.128539
default_V26 = 0.125895
default_V27 = -0.008983
default_V28 = 0.014724
```

- Kết quả hàm Loss

Hàm mất mát trên tập huấn luyện: 0.003615040996105172
 Hàm mất mát trên tập kiểm tra: 0.003226923034413009

- Dữ liệu huấn luyện nhập từ bàn phím

Nhập thời gian (Time) của giao dịch: 5
 Nhập số tiền (Amount) của giao dịch: 22
 Dự đoán: Không gian lận

Ma trận nhầm lẫn giữa thực tế và dự đoán trên SVM



Hình 3. 19 Ma trận nhầm lẫn SVM

Như trên hình 3.19, ta thấy:

- Giá trị trong ô (1, 1): Giá trị trong ô (1, 1) là 85.306. Điều này có nghĩa là 85.306 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán đúng là thuộc lớp "Có gian lận".

- Giá trị trong ô (1, 2): Giá trị trong ô (1, 2) là 0. Điều này có nghĩa là 0 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán sai là thuộc lớp "Không gian lận".
- Giá trị trong ô (2, 1): Giá trị trong ô (2, 1) là 136. Điều này có nghĩa là 136 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán sai là thuộc lớp "Có gian lận".
- Giá trị trong ô (2, 2): Giá trị trong ô (2, 2) là 0. Điều này có nghĩa là 0 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán đúng là thuộc lớp "Không gian lận".

3.7.3 Kết quả thực nghiệm trên thuật toán Random Forest

Bảng 3. 4 Đánh giá mô hình random forest

Lớp	Precision	Recall	F1-score	Support
0	1,00	1,00	1,00	85306
1	1,00	0,21	0,34	136
Accuracy	1,0			85442

Như bảng 3.4, ta thấy như sau:

Precision (Độ chính xác của dự đoán):

Của lớp 0 đạt 1,00 nghĩa là dự đoán được được 100% các dữ liệu thuộc lớp 0 là lớp không gian lận.

Lớp 1 là 1,00 nghĩa là dự đoán được được 100% các dữ liệu thuộc lớp 1 là lớp có gian lận.

Recall (Tỉ lệ nhận dạng):

Lớp 0: 1,00 có nghĩa là mô hình có thể nhận diện được đúng thực sự 100% là giao dịch không có gian lận.

Lớp 1: 0,21 nghĩa là mô hình nhận diện được khi bao quát mô hình ở mức khoảng 20 đến 21%, còn 80% còn lại bị bỏ qua không nhận diện được là giao dịch có gian lận.

Accuracy (độ chính xác):

Ở bảng 3.3, accuracy đạt được 1.0 nghĩa là mô hình hoạt động trên tập kiểm tra của dữ liệu có thể dự đoán đúng được hầu hết tất cả các mẫu dữ liệu trong tập kiểm tra.

Dự đoán nhãn của mô hình Random Forest với dữ liệu tùy chọn được nhập từ bàn phím và các biến dữ liệu cố định kết hợp:

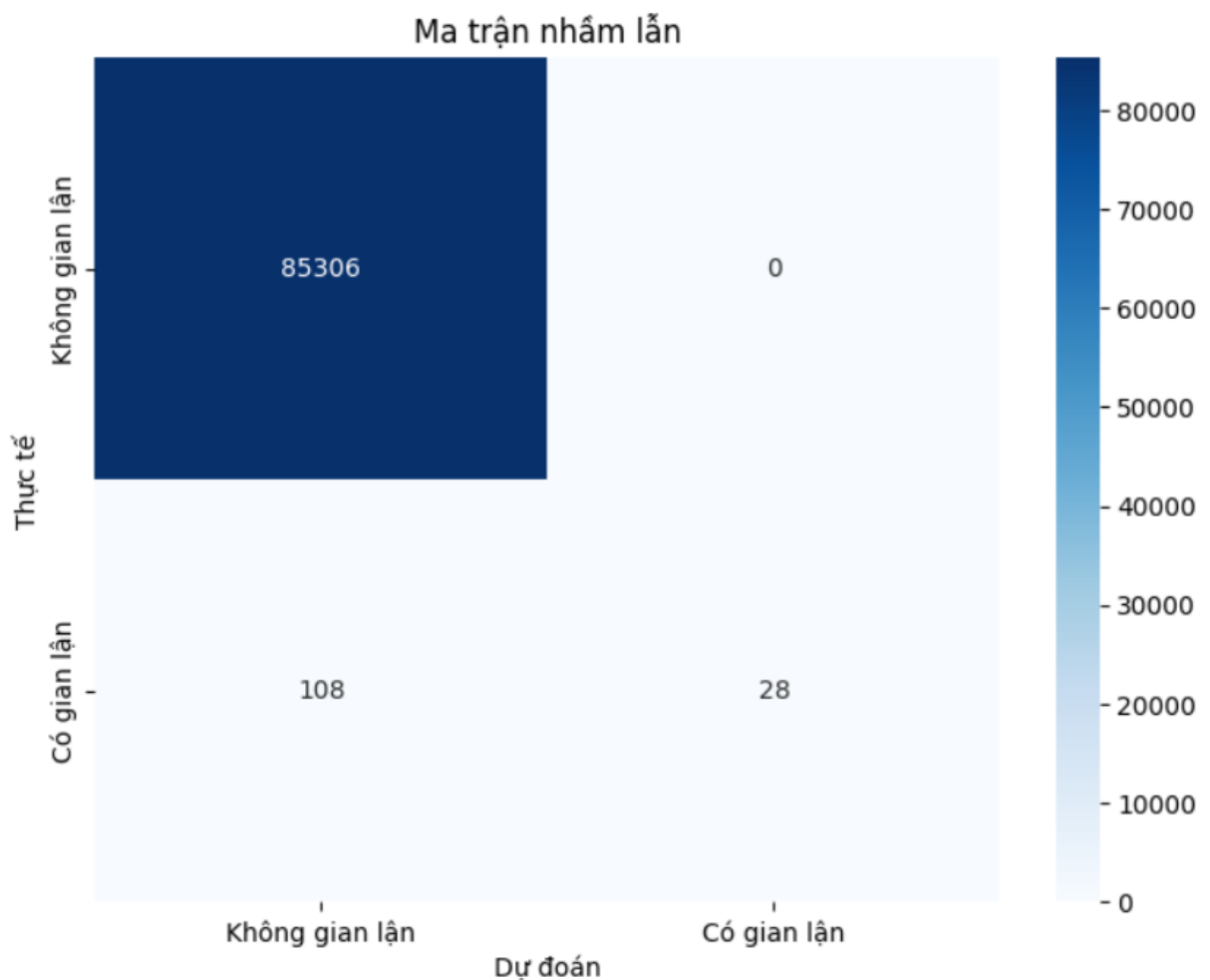
- Các biến dữ liệu cố định

```
default_V25 = 0.128539
default_V26 = 0.125895
default_V27 = -0.008983
default_V28 = 0.014724
```

- Dữ liệu nhập từ bàn phím và dự đoán

```
Nhập thời gian (Time) của giao dịch: 3
Nhập số tiền (Amount) của giao dịch: 25
Dự đoán: Không gian lận
```

Ma trận nhầm lẫn random forest



Hình 3. 20 Ma trận nhầm lẫn random forest

Như trên hình 3.20, ta thấy:

- Giá trị trong ô (1, 1): Giá trị trong ô (1, 1) là 85.306. Điều này có nghĩa là 85.306 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán đúng là thuộc lớp "Có gian lận".
- Giá trị trong ô (1, 2): Giá trị trong ô (1, 2) là 0. Điều này có nghĩa là 0 mẫu thực sự thuộc lớp "Có gian lận" được mô hình dự đoán sai là thuộc lớp "Không gian lận".
- Giá trị trong ô (2, 1): Giá trị trong ô (2, 1) là 108. Điều này có nghĩa là 108 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán sai là thuộc lớp "Có gian lận".
- Giá trị trong ô (2, 2): Giá trị trong ô (2, 2) là 28. Điều này có nghĩa là 28 mẫu thực sự thuộc lớp "Không gian lận" được mô hình dự đoán đúng là thuộc lớp "Không gian lận".

3.7.4 Kết quả đánh giá chung

Qua phân đánh giá ở trên, có thể thấy:

- Mô hình KNN cho khả năng phát hiện thẻ tín dụng có giao dịch gian lận tốt hơn SVM nhiều vì KNN nó dựa theo hàng xóm gần nhất của nó.
- Mô hình SVM các giao dịch gian lận bị bỏ qua hết, không thể phát hiện gian lận. Nguyên nhân có thể do mô hình bị hiện tượng overfitting. Nguyên nhân dẫn đến hiện tượng trên là do tập dữ liệu huấn luyện bất đối xứng, nhãn 0 chiếm đa số nên dẫn đến SVM bị overfitting.
- Mô hình Random Forest phát hiện được hầu hết các giao dịch gian lận. Vì mô hình đó sử dụng nhiều cây quyết định nên mỗi cây sẽ dự đoán và bình chọn, từ đó mô hình giảm được vấn đề overfitting và dự đoán tốt trên dữ liệu bất đối xứng.

3.8 Kết luận chương

Trong chương này, em đã thu thập, tiền xử lý và huấn luyện được ba mô hình học máy và so sánh kết quả để tìm ra mô hình cho hiệu suất tốt nhất. KNN có thể cho hiệu suất tốt vì nó dự đoán được các giao dịch gian lận được 50/50. SVM thì cho hiệu suất kém nhất vì không có khả năng dự đoán các giao dịch gian lận. Mô hình cho hiệu suất tốt nhất là Random Forest vì dự đoán được 100% các giao dịch gian lận.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài “Xây Dựng Ứng Dụng Phát Hiện Gian Lận Thẻ Tín Dụng Bằng Kỹ Thuật Học Máy” đã đạt được các mục tiêu đề ra, bao gồm biết xây dựng thuật toán học máy phát hiện gian lận thẻ tín dụng, đánh giá được thuật toán nào tối ưu cho bài toán nhất. Quá trình trên đã tìm được thuật toán tối ưu nhất cho phát hiện gian lận thẻ tín dụng, cũng như tìm ra được khuyết điểm của các thuật toán học máy khác nhau cho việc xây dựng bài toán phát hiện gian lận thẻ tín dụng.

Ưu điểm của bài toán đã xây dựng trên là đã biết được quy trình hoạt động của các thuật toán phân loại, phân lớp; biết cách xây dựng một bài toán phát hiện gian lận thẻ tín dụng. Biết được quá trình huấn luyện mô hình học máy trên dữ liệu huấn luyện. Biết cách đánh giá kết quả các mô hình học máy khác nhau, tìm hiểu kết quả xem mô hình nào tối ưu nhất cho bài toán để có thể đưa ra kết luận về mô hình tốt nhất cho các ứng dụng lớn hơn.

Hướng phát triển:

Bài toán đã xây dựng trên vẫn còn tồn tại một số nhược điểm cần khắc phục. Chưa thể dự đoán được gian lận hoàn toàn chính xác, vẫn bị gặp vấn đề dự đoán sai, hiệu suất dự đoán chưa tốt và có thể không dự đoán được gian lận. Do đó, cần một bộ dữ liệu huấn luyện tốt hơn để có thể khắc phục được các vấn đề nêu trên để bài toán đã xây dựng có thể hoạt động tốt hơn, cũng như dự đoán chính xác gian lận hơn. Có thể bổ sung thêm nhiều dữ liệu gian lận cho việc huấn luyện hiệu quả hơn.

Vì thời gian có hạn cho nên đề tài mới chỉ dừng lại ở mức độ nghiên cứu về các thuật toán học máy, chưa có ứng dụng sử dụng các kỹ thuật nêu trên. Nhưng bài toán có tiềm năng lớn trong xây dựng ứng dụng phát hiện và ngăn chặn giao dịch thẻ tín dụng gian lận.

Trong tương lai, để đưa bài toán vào ứng dụng thực tế, cần phải tập trung vào nhiều hướng phát triển quan trọng. Trước tiên, cần cải thiện dữ liệu huấn luyện cho thêm các dữ liệu giao dịch gian lận để mô hình có thể học tốt hơn. Sau đó, có thể sử dụng các mô hình học máy khác cho bài toán như: mạng nơron nhân tạo, deep learning (học sâu), AI (trí tuệ nhân tạo), ... Đây đều là những mô hình học máy tiên tiến, cho hiệu suất cực tốt, có thể dự đoán tốt được giao dịch gian lận. Ngoài ra, để triển khai mô hình vào thực tế, cần thiết kế

bài toán theo dạng ứng dụng nhúng để có thể tích hợp vào hệ thống ngân hàng, các ứng dụng giao dịch ngân hàng; để có thể thông báo cho khách hàng và nhân viên ngân hàng biết giao dịch đó là giao dịch có gian lận.

Tóm lại, dù bài toán xây dựng ứng dụng phát hiện gian lận thẻ tín dụng bằng kỹ thuật học máy đã hoàn thành tốt mục tiêu đề ra, là tìm được thuật toán học máy nào tối ưu nhất cho việc phát hiện gian lận thẻ tín dụng, và tìm ra mô hình tốt nhất cho việc xây dựng bài toán. Tuy nhiên, bài toán trên vẫn còn nhiều nhược điểm cần khắc phục. Việc liên tục nâng cấp và cải tiến bài toán giúp ngân hàng ngày càng hiệu quả trong việc phát hiện các giao dịch thẻ tín dụng gian lận, từ đó nếu bài toán được đưa vào ứng dụng thực tế, có thể giúp ngân hàng phát hiện và ngăn chặn giao dịch thẻ tín dụng gian lận.

TÀI LIỆU THAM KHẢO

- [1] <https://anasbrital98.github.io/blog/2021/Random-Forest/> truy cập ngày 25/04/2024
- [2] <https://en.wikipedia.org/wiki/Scikit-learn> truy cập ngày 25/04/2024
- [3] <https://erx.vn/lam-quen-voi-thu-vien-pandas-va-dataframe-7122812051.html> truy cập ngày 10/04/2024
- [4] <https://fptshop.com.vn/tin-tuc/danh-gia/google-colab-167087> truy cập ngày 11/4/2024
- [5] <https://journals.sagepub.com/doi/abs/10.3102/1076998619832248> truy cập ngày 11/04/2024
- [6] <https://machinelearningcoban.com/2017/01/08/knn/#k-nearest-neighbor> truy cập ngày 11/04/2024
- [7] <https://trituenhantao.io/kien-thuc/svm-qua-kho-hieu-hay-doc-bai-nay/> truy cập ngày 20/04/2024
- [8] <https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVEImB> truy cập ngày 15/04/2024
- [9] <https://viblo.asia/p/pandas-python-tutorial-XL6lAxaDZek> truy cập ngày 22/04/2024
- [10] <https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz> truy cập ngày 20/4/2024
- [11] <https://viblo.asia/p/support-vector-machine-trong-hoc-may-mot-cai-nhin-don-gian-hon-XQZkxoQmewA> truy cập ngày 05/05/2024
- [12] <https://vietnamnet.vn/nguoi-dung-viet-phan-anh-bi-thiet-hai-hon-300-ty-dong-vi-lua-dao-truc-tuyen-2280269.html> truy cập ngày 18/05/2024
- [13] <https://www.geeksforgeeks.org/k-nearest-neighbours/> truy cập ngày 11/05/2024
- [14] https://www.researchgate.net/figure/A-simple-credit-card-fraud-detection-method-22_fig1_370391680 truy cập ngày 22/04/2024

PHỤ LỤC

Code trên KNN:

<https://colab.research.google.com/drive/1Xh4xdFu9fWJp72WMF89sb9ErrStm61Vm?usp=sharing>

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc dữ liệu từ file CSV
file_path = '/content/drive/MyDrive/DoAnTotNghiep/creditcard.csv' #
Đường dẫn đến file CSV trên Google Drive
df = pd.read_csv(file_path)

# Chọn các thuộc tính cần sử dụng
X_resampled = df[['Time', 'Amount', 'V25', 'V26', 'V27', 'V28']] #
features
y_resampled = df['Class'] # nhãn

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
test_ratio = 0.3 # tỷ lệ mẫu kiểm tra mong muốn
num_total_samples = len(X_resampled)
num_test_samples = int(test_ratio * num_total_samples)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ mẫu kiểm
tra đã tính toán
X_train, X_test, y_train, y_test = train_test_split(X_resampled,
y_resampled, test_size=num_test_samples, shuffle=True, random_state=42)

# Khởi tạo mô hình KNN với số láng giềng là 5
knn_model = KNeighborsClassifier(n_neighbors=5)

# Huấn luyện mô hình
knn_model.fit(X_train, y_train)

# Dự đoán nhãn của dữ liệu kiểm tra
y_pred = knn_model.predict(X_test)

# Đánh giá mô hình
print("Báo cáo phân loại trên tập kiểm tra:")
print(classification_report(y_test, y_pred))

# Nhập biến time và amount từ bàn phím
time = float(input("Nhập thời gian (Time) của giao dịch: "))
amount = float(input("Nhập số tiền (Amount) của giao dịch: "))
```

```

default_V25 = 0.128539
default_V26 = 0.125895
default_V27 = -0.008983
default_V28 = 0.014724

# Dự đoán nhãn của dữ liệu nhập từ bàn phím
predicted_class = knn_model.predict([[time, amount, default_V25,
default_V26, default_V27, default_V28]])

# In ra kết quả phân loại
if predicted_class == 0:
    print("Dự đoán: Không gian lận")
elif predicted_class == 1:
    print("Dự đoán: Có gian lận")
else:
    print("Không thể dự đoán kết quả.")

# Vẽ ma trận nhầm lẫn
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=['Không
gian lận', 'Có gian lận'], yticklabels=['Không gian lận', 'Có gian lận'])
plt.xlabel('Dự đoán')
plt.ylabel('Thực tế')
plt.title('Ma trận nhầm lẫn')
plt.show()

```

Code trên SVM: https://colab.research.google.com/drive/1HynEowPsnfjfw syl7-q_jhK-sBR0q6GP?usp=sharing

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, hinge_loss
from imblearn.over_sampling import RandomOverSampler
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc dữ liệu từ file CSV
file_path = '/content/drive/MyDrive/DoAnTotNghiep/creditcard.csv'
df = pd.read_csv(file_path)

# Chọn các thuộc tính cần sử dụng
X_resampled = df[['Time', 'Amount', 'V25', 'V26', 'V27', 'V28']] #
features
y_resampled = df['Class'] # nhãn

```



```

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
test_ratio = 0.3
num_total_samples = len(X_resampled)
num_test_samples = int(test_ratio * num_total_samples)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ mẫu kiểm
tra đã tính toán
X_train, X_test, y_train, y_test = train_test_split(X_resampled,
y_resampled, test_size=num_test_samples, shuffle=True, random_state=42)

# Khởi tạo mô hình SVM với kernel RBF và tham số C
svm_model = SVC(kernel='rbf', C=10)

# Huấn luyện mô hình
svm_model.fit(X_train, y_train)

# Dự đoán nhãn của dữ liệu kiểm tra
y_pred = svm_model.predict(X_test)

# Đánh giá mô hình
print("Báo cáo phân loại trên tập kiểm tra:")
print(classification_report(y_test, y_pred))

# Tính và in ra hàm mất mát (loss)
train_loss = hinge_loss(y_train, svm_model.decision_function(X_train))
test_loss = hinge_loss(y_test, svm_model.decision_function(X_test))
print(f"Hàm mất mát trên tập huấn luyện: {train_loss}")
print(f"Hàm mất mát trên tập kiểm tra: {test_loss}")

# Nhập biến time và amount từ bàn phím
time = float(input("Nhập thời gian (Time) của giao dịch: "))
amount = float(input("Nhập số tiền (Amount) của giao dịch: "))

default_V25 = 0.128539
default_V26 = 0.125895
default_V27 = -0.008983
default_V28 = 0.014724

# Dự đoán nhãn của dữ liệu nhập từ bàn phím
predicted_class = svm_model.predict([[time, amount, default_V25,
default_V26, default_V27, default_V28]])

# In ra kết quả phân loại
if predicted_class == 0:
    print("Dự đoán: Không gian lận")
elif predicted_class == 1:
    print("Dự đoán: Có gian lận")
else:
    print("Không thể dự đoán kết quả.")

```

```
# Vẽ ma trận nhầm lẫn
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=['Không gian lận', 'Có gian lận'], yticklabels=['Không gian lận', 'Có gian lận'])
plt.xlabel('Dự đoán')
plt.ylabel('Thực tế')
plt.title('Ma trận nhầm lẫn')
plt.show()
```

Code trên Random Forest:

<https://colab.research.google.com/drive/1uBvtEUbGrQu0mcNjvPNCyXHmGSOxu-Mo?usp=sharing>

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from imblearn.over_sampling import RandomOverSampler
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc dữ liệu từ file CSV
file_path = '/content/drive/MyDrive/DoAnTotNghiep/creditcard.csv' #
Đường dẫn đến file CSV trên Google Drive
df = pd.read_csv(file_path)

# Chọn các thuộc tính cần sử dụng
X_resampled = df[['Time', 'Amount', 'V25', 'V26', 'V27', 'V28']] #
features
y_resampled = df['Class'] # nhãn

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
test_ratio = 0.3 # tỷ lệ mẫu kiểm tra mong muốn
num_total_samples = len(X_resampled)
num_test_samples = int(test_ratio * num_total_samples)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ mẫu kiểm tra đã tính toán
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=num_test_samples, shuffle=True, random_state=42)

# Khởi tạo mô hình Random Forest với số cây là 100 (có thể thay đổi tùy ý)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Huấn luyện mô hình
rf_model.fit(X_train, y_train)
```

```

# Dự đoán nhãn của dữ liệu kiểm tra
y_pred = rf_model.predict(X_test)

# Đánh giá mô hình
print("Báo cáo phân loại trên tập kiểm tra:")
print(classification_report(y_test, y_pred))

# Nhập biến time và amount từ bàn phím
time = float(input("Nhập thời gian (Time) của giao dịch: "))
amount = float(input("Nhập số tiền (Amount) của giao dịch: "))

# Định nghĩa các giá trị cố định cho các biến V1 đến V28
default_V25 = 0.128539
default_V26 = 0.125895
default_V27 = -0.008983
default_V28 = 0.014724

# Dự đoán nhãn của dữ liệu nhập từ bàn phím
predicted_class = rf_model.predict([[time, amount, default_V25,
default_V26, default_V27, default_V28]])

# In ra kết quả phân loại
if predicted_class == 0:
    print("Dự đoán: Không gian lận")
elif predicted_class == 1:
    print("Dự đoán: Có gian lận")
else:
    print("Không thể dự đoán kết quả.")

# Vẽ ma trận nhầm lẫn
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=['Không gian lận', 'Có gian lận'], yticklabels=['Không gian lận', 'Có gian lận'])
plt.xlabel('Dự đoán')
plt.ylabel('Thực tế')
plt.title('Ma trận nhầm lẫn')
plt.show()

```