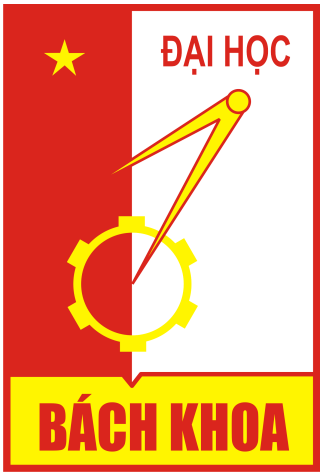
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



Ứng dụng phát hiện dữ liệu ngoại lệ vào gian lận tài chính.

ĐỒ ÁN 1

Chuyên ngành: HỆ THỐNG THÔNG TIN QUẢN LÝ

Chuyên sâu: Toán ứng dụng

Giảng viên hướng dẫn: TS. Nguyễn Hải Sơn

Sinh viên thực hiện: Lê Thị Ngọc Khánh

Lớp: Hệ Thống Thông Tin Quản Lý – K64

HÀ NỘI – 2022

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục tiêu và nội dung của đồ án

.....

.....

.....

.....

.....

.....

2. Kết quả đạt được

.....

.....

.....

.....

.....

.....

3. Ý thức làm việc của sinh viên

.....

.....

.....

.....

.....

.....

Hà Nội, ngày ... tháng ... năm 2022

Giảng viên hướng dẫn

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành đến TS. Nguyễn Hải Sơn đã luôn chỉ bảo, hướng dẫn để em có thể hoàn thành tốt báo co Đồ án 1 này. Em xin chân thành cảm ơn thầy.

Hà Nội, tháng 08 năm 2022

Sinh viên

Lê Thị Ngọc Khánh

Tóm tắt nội dung Đồ án

1. Tổng quan một số kiến thức cần chuẩn bị.
2. Giới thiệu một số phương pháp phát hiện ngoại lệ.
3. Ứng dụng của phát hiện ngoại lệ vào phát hiện gian lận với dữ liệu tài chính, minh hoạ trên bộ dữ liệu kiểm toán.
4. Kết luận.

Mục lục

Bảng ký hiệu và chữ viết tắt **3**

Mở đầu **4**

Chương 1 Kiến thức chuẩn bị **5**

1.1 Các khái niệm về dữ liệu 5

1.1.1 Dữ liệu ngoại lệ - Outlier 5

1.1.2 Khai phá dữ liệu - Data mining 7

1.2 Gian lận tài chính 8

Chương 2 Một số phương pháp phát hiện dữ liệu ngoại lệ **10**

2.1 Phương pháp ngây thơ (naive methods) 10

2.1.1 Phương pháp phân cụm (Clustering methods) 10

2.2 Một số phương pháp Machine Learning - ML 16

2.2.1 Thuật toán Rừng cách ly (Isolation forest) - IF 17

2.2.2 Thuật toán Yếu tố ngoại lệ địa phương (Local outlier factor) - LOF 20

2.2.3 Thuật toán Máy vector hỗ trợ (Support vector machine) - SVM 22

Chương 3 Ứng dụng phát hiện ngoại lệ vào phát hiện gian lận với dữ liệu tài chính **25**

3.1 Tiền xử lý dữ liệu. Các phương pháp 26

3.1.1 Tiền xử lý dữ liệu 26

3.1.2 Các phương pháp 26

3.1.3 Các chỉ số đánh giá 28

3.2 Ứng dụng trên tập dữ liệu kiểm toán 29

3.2.1 Mô tả dữ liệu 29
3.2.2 Kết quả chạy dữ liệu với các mô hình 32
3.3 Đánh giá kết quả đạt được 34
Chương 4 Kết luận **35**
Tài liệu tham khảo **37**

Danh sách bảng

3.1 Kết quả chạy thuật toán HDBSCAN- dữ liệu kiểm toán 32
3.2 Kết quả chạy thuật toán Isolation forest- dữ liệu kiểm toán 32
3.3 Kết quả chạy thuật toán LOF- dữ liệu tài chính với k- neighbors=200 32
3.4 Kết quả chạy thuật toán LOF- dữ liệu kiểm toán với k- neighbors=100 33
3.5 Kết quả chạy thuật toán LOF- dữ liệu kiểm toán với k- neighbors=300 33
3.6 Kết quả chạy thuật toán SVM- dữ liệu kiểm toán với phương pháp svm.LinearSVC 33
3.7 Kết quả chạy thuật toán SVM- dữ liệu kiểm toán với phương pháp svm.SVC 34
3.8 Bảng kết quả 34

Danh sách hình vẽ

1.1 Quang phổ từ dữ liệu bình thường đến ngoại lệ 7
1.2 Kim tự tháp DIKW 7

2.1 Cùng một tập dữ liệu ngẫu nhiên được chia thành 3 cụm và 6 cụm 11
2.2 Minh họa thuật toán DBSCAN 13
2.3 Đồ thị cây cô đặc từ tập dữ liệu ngẫu nhiên 15
2.4 Trái: Tập dữ liệu ngẫu nhiên, điểm đỏ là bất thường. Phải: Cây liên kết với sự phân vùng này. 18
2.5 Trái: Phân nhánh cho điểm dữ liệu bất thường. Phải: Phân nhánh cho điểm dữ liệu bình thường 19
2.6 Ví dụ về khoảng cách có thể tiếp cận: p_1 và p_2 21
2.7 Minh họa thuật toán SVM trong không gian 2 chiều 23

3.1 Confusion matrix 28
3.2 5 bản ghi đầu tập dữ liệu kiểm toán 30
3.3 Biểu đồ tương quan dữ liệu kiểm toán 31
3.4 Biểu đồ tương quan giữa các thuộc tính với thuộc tính Risk . . . 31

Bảng ký hiệu và chữ viết tắt

DBSCAN: Density- based spatial clustering application with noise

HDBSCAN: Hierarchical density based spatial- clustering of application with noise

IF: Isolation forest

LOF: Local outlier factor

SVM: Support vector machine

ML: Machine learning

LRD: Local reachability density

Mở đầu

Phát hiện dữ liệu ngoại lệ (outlier) đã được sử dụng trong nhiều thế kỷ để phát hiện và loại bỏ các quan sát bất thường khỏi dữ liệu. Các điểm dữ liệu ngoại lệ có thể phát sinh do lỗi cơ học, thay đổi hành vi của hệ thống, hành vi gian lận, lỗi của con người, lỗi thiết bị hoặc đơn giản là do sự sai lệch tự nhiên trong tổng thể. Việc phát hiện dữ liệu ngoại lệ có thể xác định lỗi hệ thống và gian lận trước khi chúng leo thang với những hậu quả xấu có thể xảy ra, và có thể làm sạch dữ liệu trước khi phân tích. Phát hiện ngoại lệ hiện nay ứng dụng trong rất nhiều lĩnh vực như: Phát hiện xâm nhập, cảm biến, chẩn đoán y tế, khoa học trái đất, ... Đặc biệt nó có vai trò lớn trong phát hiện gian lận tài chính. Vì hầu hết các giao dịch ngày nay được thực hiện theo cách không trực tiếp do đó việc xảy ra gian lận trong các giao dịch ngày càng trở nên phổ biến.

Trong đồ án này, em giới thiệu về một số phương pháp để phát hiện ngoại lệ. Và minh hoạ nó cho việc phát hiện gian lận trong tập dữ liệu tài chính. Mục đích của đồ án này là tìm hiểu và sử dụng một số thuật toán phát hiện dữ liệu ngoại lệ trong việc phát hiện gian lận tài chính, từ đó so sánh hiệu quả của các thuật toán đối với một bộ dữ liệu.

Đồ án gồm 4 chương. Trong chương 1, em sẽ giới thiệu qua một số khái niệm cơ bản về dữ liệu và gian lận tài chính. Một số phương pháp phát hiện ngoại lệ sẽ được trình bày ở Chương 2. Trong Chương 3, em ứng dụng các thuật toán đã nêu để xây dựng mô hình tìm kiếm ngoại lệ ứng dụng cho phát hiện gian lận trong bộ dữ liệu tài chính cụ thể là bộ dữ liệu kiểm toán, và đưa ra một vài nhận xét với các kết quả tìm được. Cuối cùng, Chương 4 kết thúc đồ án với một số kết luận.

Chương 1

Kiến thức chuẩn bị

1.1 Các khái niệm về dữ liệu

1.1.1 Dữ liệu ngoại lệ - Outlier

Dữ liệu là thông tin như sự kiện hay con số được sử dụng để phân tích điều gì đó hoặc đưa ra quyết định.

Một ngoại lệ là một điểm dữ liệu mà khác biệt so với dữ liệu còn lại. Hawkins định nghĩa : Một dữ liệu ngoại lệ là một quan sát mà tách biệt rất nhiều so với những quan sát khác để khơi dậy nghi ngờ rằng nó được tạo bởi cơ chế khác.

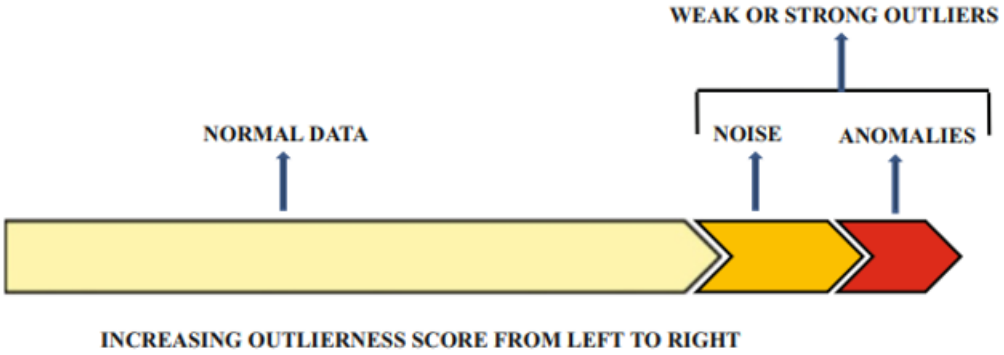
Các yếu tố ngoại lệ cũng được gọi là bất thường, bất hoà, sai lệch, hoặc dị thường trong khai phá và thống kê dữ liệu. Trong hầu hết các ứng dụng, dữ liệu được tạo bởi một hoặc nhiều quy trình, dữ liệu có thể phản ánh hoạt động trong hệ thống hay là những quan sát thu được về các thực thể. Khi quá trình tạo dữ liệu có bất thường sẽ dẫn đến tạo ra các điểm ngoại lệ. Do đó một điểm ngoại lệ thường gồm những thông tin hữu ích về đặc điểm bất thường của hệ thống và các thực thể tác động đến quá trình tạo dữ liệu. Sự nhận thấy những đặc điểm bất thường như vậy cung cấp các thông tin chi tiết hữu ích về những bất thường trong ứng dụng. Ví dụ như trong các ứng dụng phát hiện xâm nhập, gian lận thẻ tín dụng, sự kiện cảm biến, chuẩn đoán ý tế, ...Trong tất cả các ứng dụng này, dữ liệu có mô hình bình thường, và những điểm bất thường được xem như sai lệch so với mô hình bình thường này. Những điểm dữ liệu bình thường được

gọi là nội số (inliers).

Trong một số ứng dụng như phát hiện xâm nhập hay gian lận, dữ liệu ngoại lệ tương ứng với tập gồm nhiều điểm dữ liệu hơn là những điểm dữ liệu riêng lẻ và những điểm này còn được gọi là tập dị thường (collective anomalies). Đầu ra của thuật toán phát hiện ngoại lệ có thể một trong hai dạng:

- Điểm số ngoại lệ - Outlier score: Hầu hết các thuật toán phát hiện ngoại lệ đều đưa ra điểm số định lượng mức độ ngoại lệ của mỗi điểm dữ liệu. Điểm này cũng có thể được sử dụng để xếp hạng dữ liệu theo thứ tự xu hướng ngoại lệ của chúng. Đây là một dạng output rất chung, giữ lại tất cả thông tin được cung cấp bởi một thuật toán cụ thể, nhưng nó không cung cấp một mô tả ngắn gọn về số lượng các điểm dữ liệu cần xem xét ngoại lệ.
- Nhãn nhị phân - Binary label: Nhãn nhị phân cho biết một điểm dữ liệu có phải là ngoại lệ hay không. Một vài thuật toán có thể trực tiếp trả về nhãn nhị phân, tuy nhiên từ điểm số ngoại lệ cũng có thể chuyển đổi sang nhãn nhị phân. Điều này thường thực hiện bằng cách đặt các ngưỡng (threshold) cho điểm số ngoại lệ và ngưỡng được chọn dựa trên thống kê phân phối của các điểm số. Một nhãn nhị phân chứa ít thông tin hơn cơ chế tính điểm, nhưng kết quả cuối cùng thường cần cho việc ra quyết định trong các ứng dụng thực tế.

Thông thường, các thuật toán phát hiện ngoại lệ sử dụng một số thước đo đã định lượng về mức độ ngoại lệ của điểm dữ liệu như khoảng cách dựa trên hàng xóm gần nhất hoặc sự phù hợp với phân phối dữ liệu. Mọi điểm dữ liệu nằm trên một chuỗi liên tục từ dữ liệu bình thường đến nhiễu, và cuối cùng là dị thường, như minh hoạ ở hình 1.1.



Hình 1.1: Quang phổ từ dữ liệu bình thường đến ngoại lệ

1.1.2 Khai phá dữ liệu - Data mining

Theo Bernstein (2009), dữ liệu, thông tin, kiến thức và trí tuệ tạo thành một sơ đồ hình chóp. Dữ liệu thô là một quan sát không có giá trị qua quá trình xử lý sẽ trở thành thông tin, sau đó nó trở thành một chỉ dẫn và cung cấp kiến thức. Cuối cùng dẫn đến trí tuệ là khả năng nhìn và dự đoán kết quả lâu dài. Do đó, dữ liệu thô như dữ liệu giao dịch ngân hàng có thể trở thành trí tuệ và có khả năng dự đoán qua toàn bộ quá trình lọc, giảm, ... là các bước xử lý dữ liệu khác nhau.

Hình 1.2: Kim tự tháp DIKW

Thật vậy, dữ liệu không quan tâm đến việc được thu thập vô hạn mà không có bất kỳ sự biến đổi nào. Đó chỉ là bước đầu tiên trong toàn bộ quá trình phân

tích dữ liệu với mục tiêu cuối cùng là dự đoán cho tương lai và cải thiện sự hiểu biết về dữ liệu. Quá trình này được gọi là khai phá dữ liệu (data mining).

Data mining là một quá trình nhằm khám phá các mẫu trong tập dữ liệu sử dụng kỹ thuật khác nhau của toán học và khoa học máy tính. Ý tưởng là tìm thông tin từ dữ liệu để khám phá kiến thức từ cơ sở dữ liệu.
Các bước của quá trình khai phá dữ liệu:

1. Làm sạch dữ liệu (Data cleaning): Làm sạch cơ sở dữ liệu bằng cách loại bỏ các dữ liệu không nhất quán.
2. Tích hợp dữ liệu (Data integration): Thu thập dữ liệu từ các nguồn khác nhau.
3. Lựa chọn dữ liệu (Data selection): Chọn dữ liệu phù hợp với chủ đề.
4. Chuyển đổi dữ liệu (Data transformation): Chuyển đổi dữ liệu để xử lý và khai phá.
5. Khai phá dữ liệu (Data mining): Đưa ra mô hình bằng việc sử dụng toán học và khoa học máy tính.
6. Đánh giá mẫu (Pattern evaluation): Đánh giá mô hình.
7. Trình bày hiểu biết (Knowledge presentation): Trình bày kiến thức thu được cho người sử dụng.

Trong đồ án này em sẽ tập trung ở bước thứ 5 của quá trình khai phá dữ liệu và sử dụng các kỹ thuật phát hiện outlier ứng dụng vào phát hiện gian lận trong tập dữ liệu tài chính.

1.2 Gian lận tài chính

Hiện nay, trong hệ thống giao dịch, hai loại mô hình phát hiện giân lận khác nhau được sử dụng:

- Về mặt lịch sử, các kỹ thuật phát hiện gian lận đầu tiên là dựa trên các quy tắc và sơ đồ nhận dạng đã được thiết lập theo cách thủ công. Các phương pháp này có độ chính xác vừa phải và mất nhiều thời gian.
- Phát hiện gian lận dựa trên Machine learning (ML) nhằm mục đích tự động phát hiện gian lận trong hoặc gần thời gian thực bằng cách xác định mối tương quan ẩn trong dữ liệu.

Một phương tiện tốt để ngăn chặn các giao dịch gian lận này là khai thác dữ liệu và nhận dạng mẫu, ví dụ sử dụng các phương pháp phân cụm. Đối với điều này, càng có nhiều mẫu giao dịch, thì các phương pháp này càng hiệu quả. Thật vậy, có một vài yêu cầu cho các phương pháp phát hiện gian lận dựa trên AI. Đầu tiên là lượng dữ liệu: các mô hình được huấn luyện trên một tập dữ liệu lớn có độ chính xác tốt hơn. Thứ hai là chất lượng của dữ liệu và đặc biệt là dữ liệu được sắp xếp chính xác.

Hơn nữa, có hai kỹ thuật chính của machine learning: học không giám sát (unsupervised) và học có giám sát (supervised). Loại đầu tiên không cần nhãn trong mẫu dữ liệu trái ngược với loại thứ hai. Trong đồ án này, em sẽ sử dụng thuật toán học có giám sát là: Máy vector hỗ trợ (support vector machine); và thuật toán học không giám sát: Rừng cách ly (isolation forest), yếu tố ngoại lệ địa phương (local outlier factor). Em cũng sẽ triển khai thuật toán ngây thơ (naïve): clustering (phân cụm) hiện tại là mô hình được sử dụng nhiều nhất trong phát hiện gian lận tài chính.

Với bộ dữ liệu em chọn thực nghiệm trong đồ án này là dữ liệu có nhãn tuy nhiên em vẫn sẽ sử dụng các thuật toán (cả thuật toán học có giám sát và học không giám sát) để minh hoạ. Một minh chứng cho sự phù hợp của các phương pháp học không giám sát là trên thực tế đa số các tập dữ liệu là không có nhãn.

Chương 2

Một số phương pháp phát hiện dữ liệu ngoại lệ

Trong phần này mô tả một số phương pháp phát hiện ngoại lệ. Đầu tiên, phương pháp phân cụm (clustering) là phương pháp trực quan và sau đó là 3 phương pháp Machine learning: Support vector machine, Isolation forest và Local outlier factors.

2.1 Phương pháp ngây thơ (naive methods)

Thuật toán ngây thơ đối lập với các thuật toán ML là các thuật toán trực quan được mã hoá một cách tự nhiên. Trong phần này, em sẽ giới thiệu hai thuật toán: thuật toán phân cụm với DBSCAN và HDBSCAN.

2.1.1 Phương pháp phân cụm (Clustering methods)

Phân cụm dữ liệu là một dạng của vấn đề khai phá dữ liệu. Mục đích của phân cụm dữ liệu là xác định các cụm trong toàn bộ phân phối của tập dữ liệu. Trong trường hợp một chiều hoặc hai chiều, một đồ thị có thể được tạo ra và các cụm xuất hiện khá rõ ràng.

Hình 2.1: Cùng một tập dữ liệu ngẫu nhiên được chia thành 3 cụm và 6 cụm

Ví dụ ở hình 2.1 cho thấy một tập các điểm ngẫu nhiên áp dụng thuật toán phân cụm k- means. k-means sẽ nhóm các điểm theo khoảng cách của chúng với nhau. Thuật toán có một siêu tham số (hyperparameter) phải đặt trước khi chạy: k - số lượng các cụm. Trong phần bên trái của hình 2.1, áp dụng k- means với 3 cụm, và hình bên phải là 6 cụm.

Trên thực tế, tồn tại nhiều phương pháp phân cụm sử dụng các cách tiếp cận khác nhau. Thật vậy, để kết nối một điểm đến một cụm, chúng ta có thể sử dụng hai phương pháp:

- Dựa trên xác suất (A probability- based)
- Dựa trên khoảng cách (A distance- based)

Trong phần này, em sẽ sử dụng cách tiếp cận dựa trên khoảng cách cho phương pháp phân cụm.

Các thuật toán phân cụm phổ biến nhất có thể chia thành hai loại nhỏ liên quan đến vấn đề phát hiện ngoại lệ:

- Loại đầu tiên yêu cầu số lượng cụm được cung cấp trong các tham số. Như các thuật toán k- means, affinity propagation (lan truyền áp lực), spectral clustering (phân cụm quang phổ) và các thuật toán agglomerative (phân cụm tích tụ). Ví dụ, hình 2.1 cho thấy các tập dữ liệu giống nhau được chia thành số cụm khác nhau;
- Loại thứ hai, có các thuật toán DBSCAN và HDBSCAN.

Luận điểm của bài này là so sánh nhiều phương pháp phát hiện ngoại lệ. Có thể đưa ra rằng không nên gán các giá trị ngoại lệ trong tập dữ liệu cho một cụm. Do đó, không thể lấy số lượng cụm làm tham số. Đó là lý do tại sao chỉ các thuật toán của loại thứ 2 mới thích hợp. Vì vậy, em sẽ tìm hiểu thuật toán DBSCAN và HDBSCAN.

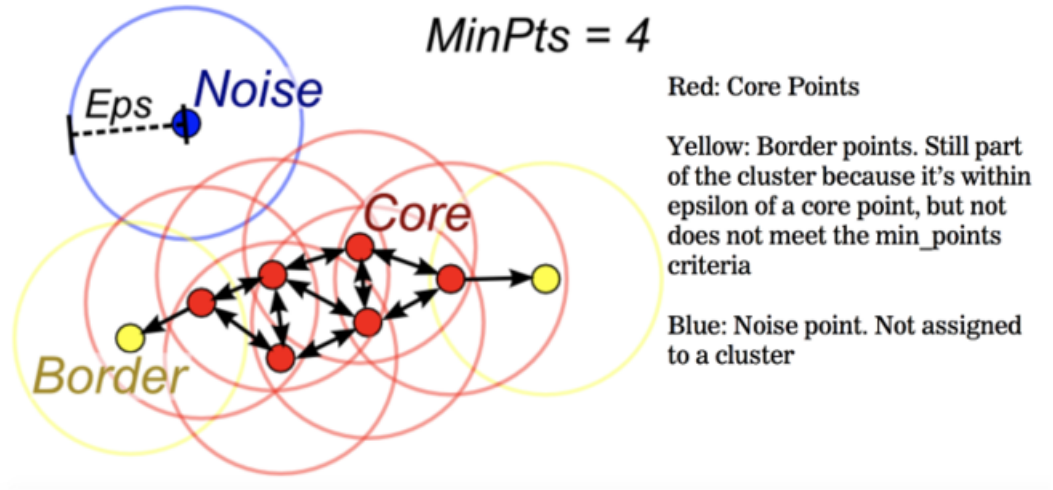
Thuật toán DBSCAN- Density- based spatial clustering application with noise

Ứng dụng phân cụm không gian dựa trên mật độ với nhiễu (DBSCAN) lần đầu tiên được trình bày tại hội nghị khai phá dữ liệu KDD'96 (Ester và cộng sự, 1996). Thuật toán này trở nên rất phổ biến và hiện được sử dụng trong nhiều ứng dụng thực tế.
Mô hình dựa trên mật độ này có hai siêu tham số (hyperparameter):

- *epsilon* (ϵ): Khoảng cách lớn nhất giữa hai điểm để coi chúng là hàng xóm của nhau.
- *min points*: Số điểm tối thiểu để tạo thành một cụm.

Sự lựa chọn những tham số này đóng vai trò lớn trong độ chính xác của mô hình. Thật vậy, nếu ϵ quá lớn, tất cả các điểm có thể kết thúc trong một cụm và việc phát hiện ngoại lệ sẽ phức tạp. Cũng giống như nếu *min points* quá lớn, chúng ta có thể kết thúc mà không có bất kì cụm nào và một tập dữ liệu chỉ được hình thành bởi các điểm ngoại lệ.

Đối với các đánh giá thực nghiệm, Ester và cộng sự (1996) đề nghị lấy số điểm tối thiểu (*min points*) bằng 2 lần số chiều của tập dữ liệu. Trong ví dụ dưới đây (hình 2.2) nếu tập dữ liệu là 2 chiều thì số điểm tối thiểu là 4. Việc xác định ϵ phức tạp hơn và phụ thuộc vào phần trăm nhiễu (các điểm ngoại lệ) mong muốn. Biến này sau đó sẽ được suy ra phụ thuộc vào tập dữ liệu thử nghiệm.



Hình 2.2: Minh họa thuật toán DBSCAN

Trong hình trên các chấm đại diện cho các điểm trong tập dữ liệu và vòng tròn xung quanh đại diện cho khoảng cách ϵ. Ở đây, số điểm tối thiểu cần thiết để tạo thành cụm là 4. Do đó, khi sàng lọc tập dữ liệu, nếu DBSCAN tìm thấy bốn điểm trong khoảng cách ϵ của nhau thì nó là một cụm.

Từ hình 2.2 căn cứ vào vị trí của các điểm dữ liệu so với cụm chúng ta có thể chia chúng thành 3 loại:

- Đối với các điểm nằm sâu bên trong cụm (màu đỏ) xem chúng là điểm lõi (core points), đây là một điểm có ít nhất số điểm tối thiểu trong vùng lân cận ϵ của chính nó.
- Các điểm biên (border points) nằm ở phần ngoài cùng của cụm (màu vàng) và nó không phải là điểm lõi của cụm nào.
- Điểm nhiễu (noise points) đây là điểm không phải điểm lõi cũng không phải điểm biên (màu xanh).

Thuật toán DBSCAN có thể được mô tả như sau:

Bước 1: Thuật toán lựa chọn một điểm dữ liệu bất kì. Sau đó tiến hành xác định các điểm lõi và điểm biên thông qua vùng lân cận ϵ bằng cách lan truyền theo liên kết chuỗi các điểm thuộc cùng một cụm.

Bước 2: Cụm hoàn toàn được xác định khi không thể mở rộng được thêm. Khi

đó lặp lại đệ quy toàn bộ quá trình với điểm khởi tạo trong số các điểm dữ liệu còn lại để xác định một cụm mới.

Thuật toán DBSCAN có nhiều ưu điểm so với các thuật toán phân cụm khác. Như đã nói, nó không yêu cầu số lượng cụm được cung cấp dưới dạng tham số. Điều này làm cho nó trở thành thuật toán hữu ích trong việc phát hiện ngoại lệ. Vì nó là thuật toán dựa trên mật độ, rất tốt để sử dụng khi có các cụm mật độ khác nhau trong cùng một tập dữ liệu. Cũng có thể tìm thấy các cụm có hình dạng tùy ý, điều này rất khó cho các thuật toán phân cụm khác.

Việc lựa chọn *min points* và ϵ có thể là một ưu điểm hoặc là một nhược điểm. Nếu tập dữ liệu được hiểu rõ thì việc chọn các tham số này rất hữu ích. Tuy nhiên, nếu tập dữ liệu không được hiểu rõ hoặc sự khác nhau giữa mật độ các cụm là quá lớn, thiết lập một ϵ thích hợp và các giá trị *min points* sẽ phức tạp. Và do đó, điều này sẽ ảnh hưởng đến kết quả của thử nghiệm.

Thuật toán HDBSCAN - Hierarchical density based spatial- clustering of application with noise:

Thuật toán phân cụm phân cấp không gian dựa trên mật độ (HDBSCAN) tương tự như thuật toán DBSCAN vì nó cũng là phương pháp phân cụm trực quan. Campello, Moulavi, và Sander giới thiệu nó vào năm 2013 tại hội nghị Châu Á- Thái Bình Dương về khám phá tri thức và khai phá dữ liệu. Nó là mở rộng của DBSCAN bằng cách chuyển đổi nó thành một thuật toán phân cụm phân cấp. Sự khác nhau giữa hai thuật toán nằm ở các tham số đầu vào: với thuật toán HDBSCAN chỉ lấy *min points* là đầu vào.

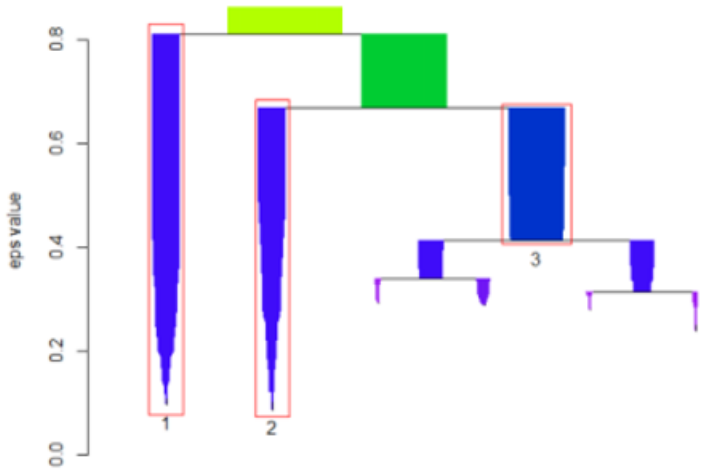
Với *mint point* thuật toán này sau đó sẽ tính một cây phân cụm (clustering tree) bao gồm tất cả các phân vùng mà DBSCAN có thể thu được cho các giá trị khác nhau của ϵ theo cách phân cấp.

Nó cũng bao gồm các node cho biết khi nào một điểm thay đổi từ điểm cốt lõi thành điểm ngoại lệ. Thật vậy, khi biết giá trị *mint points* cần thiết để tạo một cụm, khi đi xuống trong hệ phân cấp có thể quyết định tại mỗi lần phân tách rằng hai cụm mới được hình thành hoặc không. Nếu một trong các cụm

mới được tạo bởi sự phân tách có ít điểm hơn số điểm tối thiểu, thì những điểm này được phân loại là ngoại lệ.

Nhờ bước này, cây phân cụm được cô đặc. Bước tiếp theo của thuật toán HDBSCAN là trích suất các cụm. Chúng ta cần một số đo khác với khoảng cách để xem xét sự tồn tại của các cụm, chúng ta sử dụng $\lambda = \frac{1}{\epsilon}$. Mỗi cụm được gán với một λ_{birth} tương ứng với thời điểm cụm hình thành và λ_{death} là giá trị tương ứng với thời điểm khi cụm tách thành các cụm nhỏ hơn. Định nghĩa rằng, với một điểm p trong một cụm, λ_p là giá trị mà tại đó nếu nó xảy ra thì điểm này sẽ rời khỏi cụm, là một giá trị nằm trong khoảng $(\lambda_{birth}; \lambda_{death})$.

Sau đó đối với mỗi cụm, một giá trị ổn định (stability value) có thể được tính dưới dạng $\sum_{p \in cluster} (\lambda_p - \lambda_{birth})$. Đi xuống cây phân cụm một lần nữa, chúng tôi tính độ ổn định của tất cả các cụm. Nếu tổng mỗi độ ổn định của các cụm con lớn hơn độ ổn định của cụm cha thì độ ổn định của cụm được đặt là tổng các cụm con của nó. Nếu không, cụm cha được chọn và cụm con của nó sẽ tự động bị bỏ. Khi đến node gốc, HDBSCAN trả về tập các cụm thu được.



Hình 2.3: Đồ thị cây cô đặc từ tập dữ liệu ngẫu nhiên

Để hiểu được tốt hơn khái niệm này, một ví dụ được cho trong hình 2.3. Trong hình này, cây cô đặc (condensed tree) của tập hợp ngẫu nhiên là đồ thị. Chúng ta có thể thấy 3 cụm HDBSCAN được đóng khung màu đỏ. Thật vậy, đối với cụm thứ 3 vì độ ổn định của cụm cha lớn hơn tổng của các cụm con của nó

nên cụm cha đã được chọn. Tóm lại, hai phương pháp phân cụm, DBSCAN và HDBSCAN khá giống nhau. Chỉ khác nhau ở việc lựa chọn ϵ tham số sử dụng cho thuật toán DBSCAN và được tính toán trong thuật toán HDBSCAN. Tuỳ thuộc vào khả năng hiểu tập dữ liệu của người sử dụng mà một phương pháp có thể tốt hơn. Nếu có một sự hiểu biết hoàn hảo về dữ liệu, DBSCAN được ưu tiên hơn vì chúng ta có thể chọn ϵ. Ngược lại, HDBSCAN có thể là một lựa chọn tốt hơn.

2.2 Một số phương pháp Machine Learning - ML

Một cách rất đơn giản để thấy sự khác nhau giữa các thuật toán ngây thơ và các thuật toán ML:

- Các thuật toán ngây thơ lấy các quy tắc và dữ liệu là input và trả ra câu trả lời là output.
- Các thuật toán ML lấy câu trả lời và dữ liệu làm input và đưa ra các quy tắc là đầu ra.

Mục đích của ML là tính toán các giải pháp cho một chương trình mà con người không thể giải thích. Lấy một ví dụ về nhận diện khuôn mặt. Đó là một nhiệm vụ chúng ta làm hàng ngày mà không hề nghĩ đến là nếu phải giải thích cách thực hiện thì chúng ta không thể. Sau đó, không thể viết một chương trình máy tính có thể nhận diện trực tiếp các mẫu khuôn mặt của một người. Tuy nhiên với ML, chương trình có thể ghi lại các mẫu cụ thể của khuôn mặt một người như miệng, mắt, và học để nhận ra chúng.

Các thuật toán ML có điểm chung là ở các thức hoạt động của chúng. Luôn có một tập dữ liệu huấn luyện (train set) và tập dữ liệu kiểm thử (test set). Như đã giải thích trước, mục đích của thuật toán ML là "học" từ một mẫu của tập dữ liệu thường được gọi là tập huấn luyện và sau đó đưa ra output trên tập kiểm thử.

Trong phần này, em sẽ trình bày 3 thuật toán ML được sử dụng cho phát hiện outlier: SVM, IF, và LOF.

2.2.1 Thuật toán Rừng cách ly (Isolation forest) - IF

IF được thiết kế đặc biệt cho phát hiện ngoại lệ. Nó được đề xuất đầu tiên vào năm 2008 bởi Fei Tony Liu, Kai Ming Ting và Zhi- Hua Zhou. Hầu hết các kĩ thuật dùng để phát hiện dị thường thường dựa trên định nghĩa "thế nào là bình thường". Từ đó, những gì không nằm trong bộ bình thường được coi là bộ dị thường. Trong khi đó, thuật toán IF lại dùng cách tiếp cận khác. Thay vì xây dựng mô hình nhận diện bình thường, nó tìm cách cô lập các bộ dị thường trong tập dữ liệu. Ưu điểm của cách tiếp cận này là tốc độ xử lý nhanh và đòi hỏi ít bộ nhớ.

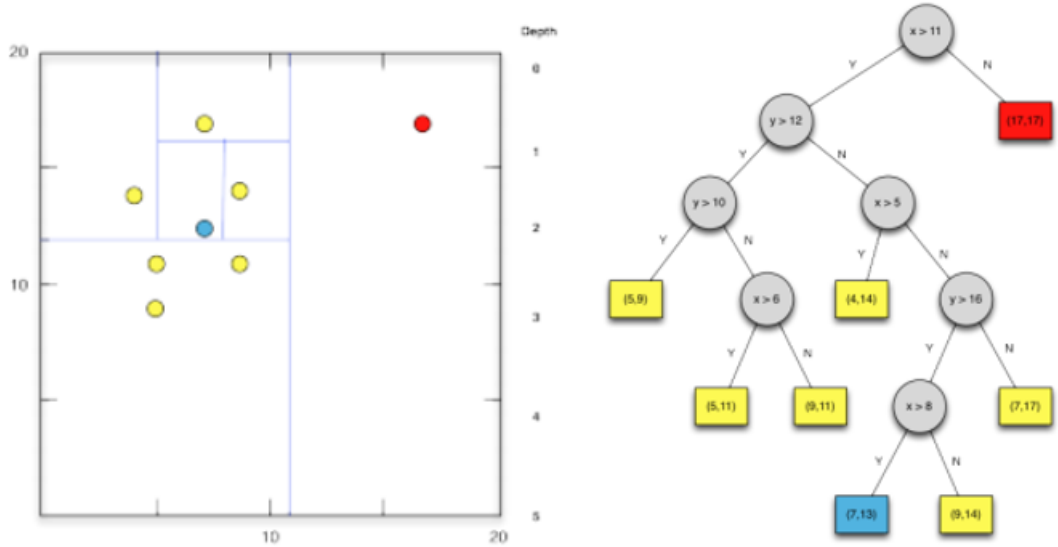
Trong thuật toán này, việc huấn luyện bao gồm việc tạo cây. Đầu tiên giới thiệu tập dữ liệu có n chiều tương ứng với n thuộc tính. Bước đầu tiên của thuật toán là lấy mẫu dữ liệu huấn luyện và xây dựng cây nhị phân (binary tree). Quá trình phân nhánh của cây này diễn ra như sau:

- Với tập dữ liệu đã cho, một mẫu dữ liệu ngẫu nhiên được chọn để xây dựng cây nhị phân.
- Chọn một chiều d tương ứng với một thuộc tính (feature) từ n thuộc tính.
- Chọn một giá trị ngẫu nhiên v_r giữa giá trị nhỏ nhất và giá trị lớn nhất của chiều cụ thể này.
- Sàng lọc dữ liệu, nếu điểm có giá trị nhỏ hơn v_r với chiều d, điểm đó sẽ được chuyển đến một nhánh bên trái, ngược lại, nó sẽ được gửi đến nhánh bên phải.

Với quá trình này, mỗi node dẫn đến hai nhánh. Sau đó, quá trình được lặp lại cho toàn bộ dữ liệu cho đến khi một điểm bị cô lập hoặc đạt được độ sâu giới hạn (nếu được chỉ định).

Sau đó, quá trình bắt đầu lại với mẫu khác của tập dữ liệu ban đầu. Khi số lượng lớn cây được xây dựng, việc huấn luyện hoàn thành. Thuật toán này hoạt động theo nguyên lý các điểm bất thường trong tập dữ liệu là "ít và khác biệt". Do đó, độ sâu của mỗi điểm là một chỉ báo tốt để tách các điểm ngoại lệ khỏi

các điểm bình thường. Thật vậy, thông thường, các outlier sẽ có độ sâu nhỏ hơn so với các điểm bình thường.



Hình 2.4: Trái: Tập dữ liệu ngẫu nhiên, điểm đỏ là bất thường. Phải: Cây liên kết với sự phân vùng này.

Bằng cách tạo tất cả những cây này, điểm số bất thường (outlier score) có thể được tính toán để bất kì điểm dữ liệu mới nào cũng có thể đi qua cây này đối với dữ liệu đã đào tạo. Phương trình điểm bất thường được cho bởi:

$$s(x,n) = 2\frac{(-E(h(x))}{c(n)} \tag{2.1}$$

Trong đó:

$E(h(x))$ là giá trị trung bình của độ sâu một điểm dữ liệu đạt đến trong tất cả các cây.

$c(n)$ là trung bình độ sâu trong một tìm kiếm không thành công trong cây tìm kiếm nhị phân (binary search tree) được cho bởi phương trình:

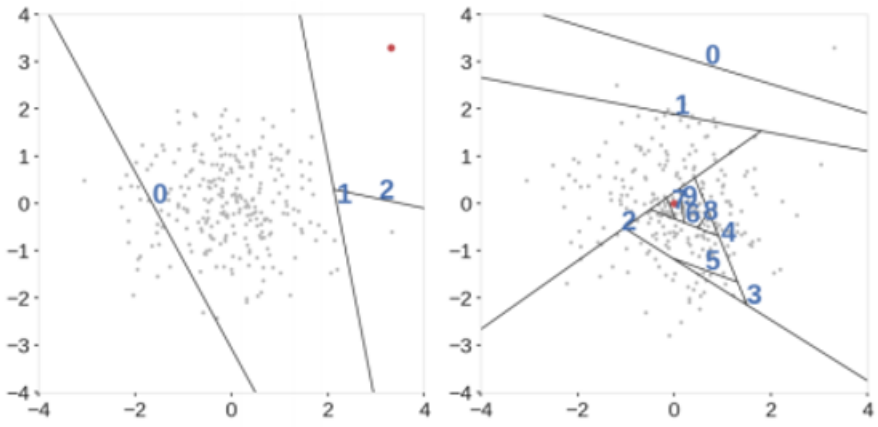
$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{2.2}$$

Với $H(i)$ là số điều hoà (harmonic) xấp xỉ bởi $log(i) + \gamma$ trong đó γ là hằng số Euler; và n là số điểm tham gia vào xây dựng cây.

Các điểm số bất thường (outlier score) được tính cho từng cây và được tính trung bình trên các cây khác nhau để nhận được điểm số bất thường cuối cùng

trong toàn bộ rừng cho một điểm dữ liệu nhất định. Sau đó, nếu điểm số này gần bằng 1 thì điểm này sẽ được gắn nhãn là ngoại lệ. Nếu điểm số gần 0 thì điểm này được gắn nhãn là điểm bình thường.

Thuật toán IF là một phương pháp tốt để tìm các điểm bất thường trong tập dữ liệu ngay cả khi tập dữ liệu nhỏ. Tuy nhiên, cách chia tập dữ liệu thực hiện khi cắt các nhánh luôn theo chiều ngang hoặc dọc. Điều này có thể dẫn đến sự sai lệch trong phát hiện các dị thường. Đây là lý do tại sao vào năm 2018, Hariri và cộng sự giới thiệu thuật toán Rừng cách ly mở rộng (Extended Isolation Forest). Thuật toán này rất giống với Isolation forest nhưng lấy ngẫu nhiên độ dốc (slope) khi chia cắt nhanh, thay vì chọn một chiều ngẫu nhiên và một giá trị ngẫu nhiên trong chiều đó.



Hình 2.5: Trái: Phân nhánh cho điểm dữ liệu bất thường. Phải: Phân nhánh cho điểm dữ liệu bình thường

Hình 2.5 Cho thấy cách phân vùng khác nhau với extended isolation forest trong tập dữ liệu ngẫu nhiên. Với thuật toán này thì dùng các đường có độ dốc để chia cắt dữ liệu thay vì chỉ là những đường ngang và dọc như trong thuật toán IF. Và từ hình 2.5 có thể thấy quá trình phân nhánh của điểm dữ liệu màu đỏ trong hình trái là điểm dữ liệu bất thường thì chỉ cần 3 lần cắt để cô lập. Còn hình bên phải điểm đỏ là một điểm bình thường thì cần nhiều lần cắt để cô lập được nó thậm chí trong trường hợp này giới hạn độ sâu của cây đã đạt đến

trước khi điểm bị cô lập hoàn toàn.

2.2.2 Thuật toán Yếu tố ngoại lệ địa phương (Local outlier factor) - LOF

Thuật toán LOF lần đầu được giới thiệu bởi Marcus Breunig và cộng sự. Trong thời gian hội nghị quốc tế về quản lý dữ liệu năm 2000. Đây là một thuật toán được tạo ra đặc biệt cho phát hiện ngoại lệ dựa trên mật độ (density- based), nó dựa trên những thuật toán khác như DBSCAN và k- nearest neighbors.

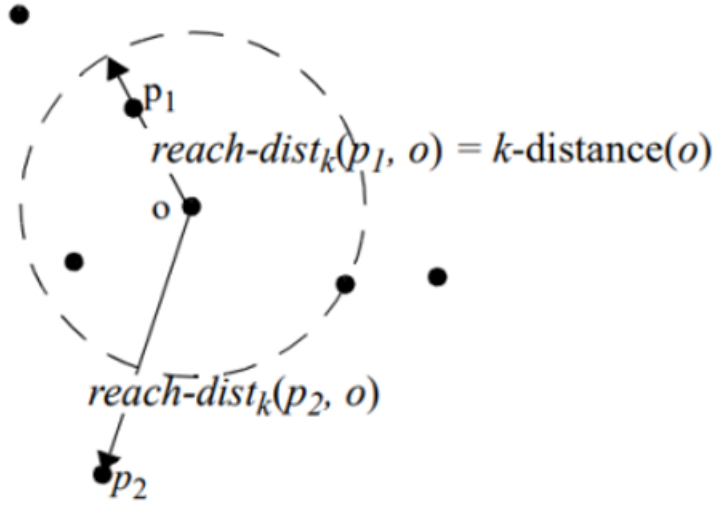
Thuật toán LOF lấy một số nguyên k làm đầu vào. Tham số này đại diện cho số hàng xóm mà thuật toán sẽ xem xét. Bước đầu tiên của LOF là tính mật độ hàng xóm của một điểm nhất định để so sánh sau đó. Tham số này quan trọng vì một số k nhỏ sẽ tạo ra tiêu điểm cục bộ hơn, trong khi một k lớn có thể dẫn đến bỏ sót các ngoại lệ.

Tham số k sau đó được sử dụng để tính $k-distance$ là khoảng cách từ một điểm đến hàng xóm thứ k của nó, tức là điểm gần nhất thứ k với điểm đã cho. Với $k-distance$ này, khoảng cách có thể tiếp cận (reachability- distance) có thể được tính toán và đưa ra theo công thức sau:

$$reachability-distance_k(A,B) = max\{k-distance(B), d(A,B)\} \tag{2.3}$$

Trong đó A và B là 2 điểm của tập dữ liệu. Nếu điểm A nằm trong k hàng xóm của điểm B, thì khoảng cách có thể tiếp cận sẽ là $k-distance(B)$. Nếu không, nó sẽ là khoảng cách Euclide giữa A và B. Để hiểu được khái niệm này hơn, hình 2.6 cho thấy hai trường hợp khác nhau.

Với trung bình tất cả $reachability-distance$ đến k hàng xóm gần nhất của một điểm được tính toán để xác định mật độ khả năng tiếp cận cục bộ (LRD) (local reachability density) của điểm đó bằng cách lấy nghịch đảo của giá trị trung bình. LRD là thước đo mật độ của k điểm gần nhất xung quanh một điểm được



Hình 2.6: Ví dụ về khoảng cách có thể tiếp cận: p_1 và p_2

tính bằng:

$$lrd_k(A) := \frac{1}{\frac{\sum_{B \in N_k(A)} reachability - distance_k(A,B)}{|N_k(A)|}} \tag{2.4}$$

Trong đó $\sum_{B \in N_k(A)} reachability - distance_k(A,B)$ là tổng tất cả các $reachability - distance$ giữa A và k điểm lân cận gần nhất của nó. Chia $|N_k(A)| = k$ để lấy giá trị trung bình. Với LRD việc tính toán hệ số ngoại lệ cục bộ LOF của một điểm có thể được tính theo công thức:

$$LOF_k(A) := \frac{\sum_{B \in N_k(A)} \frac{lrd_k(B)}{lrd_k(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k} (A) lrd_k(B)}{|N_k(A)|.lrd_k(A)} \tag{2.5}$$

Vì vậy, trong phương trình nếu mật độ của các điểm lân cận và điểm đang xét gần như bằng nhau, có thể nói chúng khá giống nhau. Nếu mật độ của các vùng lân cận nhỏ hơn mật độ của điểm đang xét có thể nói rằng điểm đó là một inlier tức là nằm trong cụm, và nếu mật độ của các điểm lân cận nhiều hơn điểm đang xét có thể nói điểm đó là một ngoại lệ.

$LOF \sim 1 \Rightarrow$ điểm dữ liệu tương đồng

$LOF < 1 \Rightarrow$ inlier (giống với điểm dữ liệu nằm trong cụm)

$LOF > 1 \Rightarrow$ outlier

Điểm mạnh của thuật toán này là tính đặc trưng cục bộ của nó. Thật vậy, một

điểm gần với một cụm dày đặc có thể được gắn nhãn là một ngoại lệ trong khi nó có cùng giá trị mật độ với các điểm trong một cụm thưa thớt. Và do đó, thuật toán LOF có thể phát hiện các ngoại lệ mà các thuật toán khác không thể.
Tuy nhiên, thực tế là nó dựa trên một thương số có thể là một nhược điểm. Thật vậy nếu giá trị LOF cho một điểm trong tập dữ liệu rất sạch là 1.1, nó có thể là một outlier, trong khi trong một tập dữ liệu đa dạng hơn điểm này có thể được coi là một inlier. Do đó, điều quan trọng là phải biết cách diễn giải các giá trị LOF.

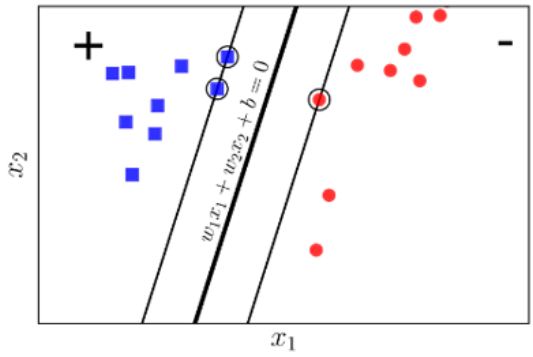
2.2.3 Thuật toán Máy vector hỗ trợ (Support vector machine) - SVM

SVM được giới thiệu lần đầu bởi Vapnik năm 1979, sau đó xuất bản năm 2006, và hiện nay là một thuật toán ML phổ biến. Theo như Srivastava và Bhambhu (2009), hiệu quả của nó thường cao hơn về độ chính xác phân lớp so với các thuật toán phân lớp khác. SVM có thể là tuyến tính hoặc phi tuyến.

SVM được sử dụng cho phân lớp và hồi quy. Với tập dữ liệu huấn luyện, thuật toán này sẽ chia các điểm vào 2 lớp. Trong trường hợp SVM tuyến tính, nó tạo ra một siêu phẳng tối ưu nếu tập dữ liệu có 2 chiều và một tập các siêu phẳng tối ưu nếu tập dữ liệu có nhiều chiều hơn.
Giả sử rằng các tập dữ liệu training set là $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$:

- x_i là một vector d chiều thể hiện đầu vào của một điểm dữ liệu.
- y_i là nhãn của điểm dữ liệu đó giả sử rằng yi nhận giá trị 1 (class 1) hoặc -1 (class 2).
- d là số chiều của dữ liệu.
- N là số điểm dữ liệu.

Để dễ hình dung, chúng ta sẽ xét trường hợp trong không gian hai chiều và các phép toán hoàn toàn có thể được tổng quát hoá lên không gian nhiều chiều.
Giả sử các điểm vuông xanh thuộc class 1, các điểm tròn đỏ thuộc class -1 và



Hình 2.7: Minh họa thuật toán SVM trong không gian 2 chiều

mặt $w^T x + b = w_1 x_1 + w_2 x_2 + b = 0$ là mặt phân chia giữa hai class (hình trên). Hơn nữa class 1 nằm về phía dương, class -1 nằm về phía âm của mặt phân chia. Nếu ngược lại, ta chỉ cần đổi dấu của w và b.

Với một điểm dữ liệu bất kì (x_n, y_n), khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(w^T x_n + b)}{||w||_2} \tag{2.6}$$

y_n luôn cùng dấu với phía của x_n. Từ đó suy ra y_n cùng dấu với $(w^T x_n + b)$, và tử số luôn là một số không âm. Mục đích của SVM là tìm một siêu phẳng tối ưu chia tập dữ liệu vào 2 lớp. Cái được gọi là siêu phẳng tối là một hàm tuyến tính với biên cực đại (maximal margin) giữa các vector của hai lớp. Với mặt phần chia như trên, margin được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó (bất kể điểm nào trong hai classes):

$$margin = \min_n \frac{y_n(w^T x_n + b)}{||w||_2} \tag{2.7}$$

Bài toán tối ưu trong SVM chính là bài toán tìm w và b sao cho $margin$ này đạt giá trị lớn nhất:

$$(w, b) = \arg\max_{w,b} \{\min_n \frac{y_n(w^T x_n + b)}{||w||_2}\} = \arg\max_{w,b} \{\frac{1}{||w||_2} \min_n y_n(w^T x_n + b)\} \tag{2.8}$$

Việc giải trực tiếp sẽ phức tạp, do vậy tìm cách để đưa nó về bài toán đơn giản hơn. Nhận xét rằng nếu thay vector w bởi kw và b bởi kb trong đó k là một hằng số dương khoảng cách từ các điểm đến mặt phân chia không đổi tức là $margin$

không đổi. Dựa trên tính chất này có thể giả sử:

$$y_n(w^T x_n + b) = 1 \tag{2.9}$$

Với những điểm nằm gần mặt phân chia nhất như hình 2.7. Như vậy với mọi n, ta có:

$$y_n(w^T x_n + b) \geq 1 \tag{2.10}$$

Vậy bài toán (2.8) có thể đưa về bài toán tối ưu có ràng buộc sau đây:

$$(w, b) = \arg\max_{w,b} \frac{1}{||w||_2} \text{ với } y_n(w^T x_n + b) \geq 1, \forall n = 1, 2, ..., N \tag{2.11}$$

Bằng một biến đổi đơn giản, có thể đưa bài toán này về bài toán dưới đây:

$$(w, b) = \arg\min_{w,b} \frac{1}{2}||w||_2^2 \text{ với } 1 - y_n(w^T x_n + b) \leq 0, \forall n = 1, 2, ..., N \tag{2.12}$$

Ở đây, lấy nghịch đảo của hàm mục tiêu, bình phương nó để được một hàm khả vi, và nhân với $\frac{1}{2}$ để biểu thức đạo hàm đẹp hơn. Trong bài toán (2.11) hàm mục tiêu là một hàm lồi. Các hàm bất đăng thức ràng buộc là các hàm tuyến tính theo w và b nên chúng cũng là các hàm lồi, do đó mà bài toán (2.11) là bài toán lồi. Lại có hàm mục tiêu $||w||_2^2 = w^T I w$ và I là một ma trận đơn vị- là một ma trận xác định dương. Từ đây có thể suy ra nghiệm cho SVM là suy nhất.

Tuy nhiên việc giải bài toán trở nên phức tạp khi số chiều d của không gian dữ liệu và số điểm dữ liệu tăng cao. Người ta thường giải bài toán đỗi ngẫu cỉa bài toán này. Và trong quá trình xây dựng bài đối ngẫu, thấy rằng SVM có thể có các đường phân chia không phải là một mặt phẳng mà có thể là mặt có hình thù phức tạp hơn.

Xác định lớp (class) cho một điểm dữ liệu mới: Sau khi tìm được mặt phân cách $w^T x + b = 0$, lớp của bất kì một điểm nào sẽ được xác định đơn giản bằng cách:

$$class(x) = sgn(w^T x + b)$$

Trong đó hàm sgn làm hàm xác định dấu, nhận giá trị 1 nếu đối số là không âm và -1 nếu ngược lại.

Chương 3

Ứng dụng phát hiện ngoại lệ vào phát hiện gian lận với dữ liệu tài chính

Các thuật toán phát hiện ngoại lệ có rất nhiều ứng dụng trong thực tế. Công dụng của nó được sử dụng để tìm các giá trị bất thường và phát hiện gian lận, nó cũng được sử dụng để tối ưu hoá một hệ thống. Ví dụ trong các phương tiện giao thông công cộng, phát hiện ngoại lệ trong dữ liệu phân phối luồng giao thông cho phép phát hiện sự cố trên các tuyến đường và có thể giúp tối ưu hoá dịch vụ.

Trong tài chính cũng vậy, phân tích dữ liệu về các giao dịch tài chính có thể được sử dụng trong mục tiêu thương mại. Các kỹ thuật phát hiện ngoại lệ có vai trò rất lớn trong phân tích dữ liệu nó giúp tìm ra những bất thường trong tập dữ liệu. Theo định nghĩa một ngoại lệ là một giá trị trong tập dữ liệu mà có khác biệt đáng kể so với những giá trị khác thể hiện sự tương phản. Một giao dịch gian lận có thể được phát hiện bởi sự tương phản của nó với những giao dịch khác, ví dụ dựa trên thời gian giao dịch hoặc số tiền giao dịch.

Trong đồ án này, em đánh giá các kỹ thuật phát hiện ngoại lệ trên tập dữ liệu kiểm toán (audit data) là một tập dữ liệu cân bằng với khoảng 800 dòng, em đã xây dựng các mô hình trên google colab với ngôn ngữ python.

3.1 Tiền xử lý dữ liệu. Các phương pháp

Với mỗi tập dữ liệu, công việc tiền xử lý là cần thiết trước khi bắt đầu với việc học nó. Mục đích để kiểm tra rằng chúng có các giá trị không tồn tại hay không và tất cả các dữ liệu đã ở đúng định dạng để chạy thuật toán chưa.

3.1.1 Tiền xử lý dữ liệu

Tập dữ liệu kiểm toán:

Trong tập dữ liệu kiểm toán, một giá trị không tồn tại đã được tìm thấy ở cột Money_Value, em đã xử lý nó bằng cách loại bỏ bản ghi đó. Hơn nữa, cột LOCATION_ID bao gồm các chuỗi, loại bỏ vì nó không có quá nhiều ý nghĩa với mô hình. Thật vậy, vị trí (location) không liên quan đến nghiên cứu nên không cần thực hiện one- hot encoding với nó.

3.1.2 Các phương pháp

Phương pháp phân cụm- Thuật toán HDBSCAN:

Đầu tiên, em thực hiện phương pháp naïve: clustering (phân cụm) với việc dùng thuật toán HDBSCAN trong chương 2 là lựa chọn tốt nhất khi chúng ta không biết tập dữ liệu hoạt động như nào với việc phân nhóm. Trong python, toàn bộ thư viện HDBSCAN có sẵn và em quyết định sử dụng nó cho chương trình của mình.

Thách thức trong phương pháp phân cụm là tìm ra kích thược tối thiểu của các cụm cho phép tạo ra ma trận phân lớp tốt nhất tức là các chỉ số *precision*, *recall*, và $F1 - \ score$ (được nói ở dưới). Với tập dữ liệu kiểm toán em đã nhận được kết quả tốt với kích thước cụm là 230.

Phương pháp Machine learing:

Sử dụng thư viện scikit- learn trong Python vì nó cung cấp việc triển khai nhiều thuật toán trong đó có: IF, LOF và SVM được sử dụng trong bài này. Đầu tiên để triển khai các phương pháp ML, các tập dữ liệu được chia thành 2 phần:

- X: Là tập không bao gồm cột Risk.
- Y: Là cột Risk.

Sau đó sử dụng train_test_split trong thư viện scikit- learn. Công cụ này chia mảng hay ma trận vào tập train và test một cách bất kỳ. Nó cũng cho người sử dụng quyết định kích thước của tập test. Công việc chuẩn bị này là chung cho hầu hết các phương pháp ML cần được huấn luyện. Em lựa chọn thiết lập kích thước tập test là 20% của mỗi tập.

Isolation Forest:

Để thực hiện xây dựng mô hình thuật toán IF, sử dụng thuật toán của scikit-learn. Tất cả các đầu vào của nó là tuỳ chọn và có thể để là mặc định, nhưng em quyết định tự đặt một số giá trị vì kết quả mà không có đầu vào là không đủ tốt:

- Số lượng của các mẫu rút ra từ X để huấn luyện mỗi ước lượng cơ sở được đặt trên độ dài của tập huấn luyện con của X;
- Mức độ contamination (nhiễu) đặt là 0.45 với tập dữ liệu kiểm toán tham số này đại diện cho tỷ lệ của outlier trong tập dữ liệu.

Local Outlier Factor:

LOF là một thuật toán học không giám sát cũng được thực hiện trong scikit-learn. Không như IF, tập dữ liệu không chia thành tập train và test. Với thuật toán này, em sử dụng hàm $fit\_predict$ từ cùng thư viện. Công cụ này có ích cho các phương pháp không giám sát vì chúng ta sẽ chỉ nhận được các nhãn kết quả của việc chạy mô hình trong các tập dữ liệu.

Giống với IF, LOF có inputs tối ưu mà em đã thử thay đổi để được kết quả tốt hơn:

- Contaimination (mức độ nhiễu) được đặt giá trị tương tự như trong thuật toán IF ở trên;
- Số lượng hàng xóm để sử dụng cho k- neighbors đặt là 200 với tập dữ liệu kiểm toán.

Support vector machine:

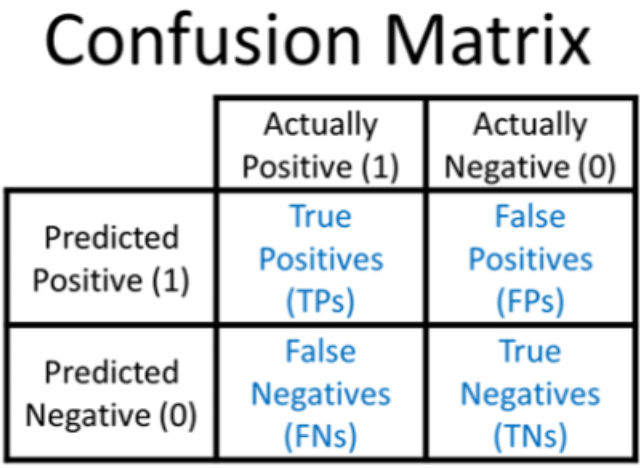
Trong trường hợp của SVM, em quyết định sử dụng hai phương pháp khác nhau:

- Linear SVC, trong đó chỉ đặt một input là số lần lặp tối đa là 1 000 000;
- svm.SVC, trong đó đặt kernel thành poly với bậc 2 và gamma để tự động.

3.1.3 Các chỉ số đánh giá

Để đánh giá hiệu suất của các thuật toán, sử dụng báo cáo phân lớp (classification report) từ thư viện scikit- learning. Công cụ này trả về bản tóm tắt về độ chính xác *accuracy*, *precision*, *recall* và $F-1$ *score* cho mỗi lớp. Ở đây các lớp 1 và 0 cho các giá trị outlier và inlier. Để hiểu tất cả các chỉ số do báo cáo trả về, trước tiên giới thiệu bốn thuộc tính để xác định kết quả phân lớp trên 1 điểm dữ liệu:

- True positive (TP): nếu thực tế là positive và dự đoán cũng là positive.
- False positive (FP): nếu thực tế là negative và dự đoán là positive.
- True negative (TN): nếu thực tế là negative và dự đoán là negative.
- False negative (FN): nếu thực tế là positive và dự đoán là negative.



Hình 3.1: Confusion matrix

Độ chính xác accuracy:

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.1}$$

Tuy nhiên trong nhiều trường hợp dữ liệu bị mất cân bằng - một lớp có số lượng lớn hơn lớp kia, *accuracy* không còn là thước đo phù hợp.

Recall:

Recall hay còn gọi là độ nhạy. Tỷ lệ những điểm true positive trong những điểm thực sự là positive.

$$recall = \frac{TP}{TP + FN} \tag{3.2}$$

Precision: *Precision* là tỷ lệ những điểm true positive trong những điểm phân loại là positive.

$$precision = \frac{TP}{TP + FP} \tag{3.3}$$

F1- score: Đối với những tập dữ liệu mất cân bằng thì *accuracy*, *recall*, *precision* không phản ánh được độ chính xác hiệu quả của thuật toán. Do đó, cần sử dụng độ đo mới $F1 - \ score$ là trung bình hài hoà của *precision* và *recall*:

$$F1 - score = \frac{2 * recall * precision}{recall + precision} \tag{3.4}$$

3.2 Ứng dụng trên tập dữ liệu kiểm toán

3.2.1 Mô tả dữ liệu

Tập dữ liệu nhỏ với khoảng 800 bản ghi, mỗi bản ghi đại diện cho một công ty.

Nó được mô tả bởi các thuộc tính như sau:

- Sector_score: giá trị điểm rủi ro lịch sử của đơn vị mục tiêu;
- Location_ID: ID của thành phố/ tỉnh;
- PARA_A, PARA_B: phát hiện có sự chênh lệch trong kế hoạch chi tiêu của cuộc thanh tra;
- Total: tổng của PARA_A và PARA_B
- Score_A, score_B, Score_B.1, score_MV, Score: điểm rủi ro lịch sử của đơn vị mục tiêu.

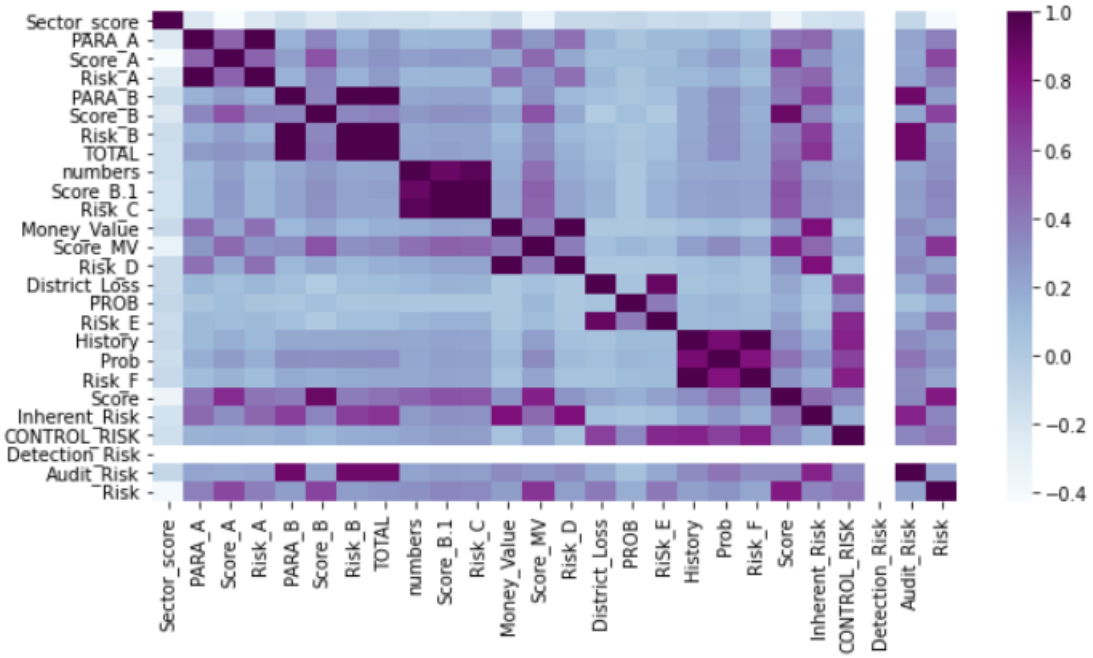
- History: lượng lỗ trung bình của công ty trong 10 năm qua;
- Money_Value: Số tiền liên quan đến các sai sót trong các cuộc kiểm toán trước đây;
- District_Loss: Số tiền lỗ của công ty trong năm qua;
- Numbers: điểm số khác biệt trong quá khứ
- Risk_A, Risk_B, Risk_C, Risk_D, Risk_E, Risk_F: lớp rủi ro được chỉ định cho một trường hợp kiểm toán;
- Audit_Risk: Tổng điểm rủi ro sử dụng quy trình phân tích;
- Inherent_Risk, Control_Risk, Detection_Risk: các rủi ro khác liên quan đến công ty
- Prob, PROB: xác suất liên quan đến công ty

Minh hoạ tập dữ liệu (5 bản ghi đầu):

	Sector_score	LOCATION_ID	PARA_A	Score_A	Risk_A	PARA_B	Score_B	Risk_B	TOTAL	numbers	...	RiSk_E	History	Prob	Risk_F	Score	Inherent_Risk	CONTROL_RISK	Detection_Risk	Audit_Risk	Risk
0	3.89	23	4.18	0.6	2.508	2.50	0.2	0.500	6.68	5.0	...	0.4	0	0.2	0.0	2.4	8.574	0.4	0.5	1.7148	1
1	3.89	6	0.00	0.2	0.000	4.83	0.2	0.966	4.83	5.0	...	0.4	0	0.2	0.0	2.0	2.554	0.4	0.5	0.5108	0
2	3.89	6	0.51	0.2	0.102	0.23	0.2	0.046	0.74	5.0	...	0.4	0	0.2	0.0	2.0	1.548	0.4	0.5	0.3096	0
3	3.89	6	0.00	0.2	0.000	10.80	0.6	6.480	10.80	6.0	...	0.4	0	0.2	0.0	4.4	17.530	0.4	0.5	3.5060	1
4	3.89	6	0.00	0.2	0.000	0.08	0.2	0.016	0.08	5.0	...	0.4	0	0.2	0.0	2.0	1.416	0.4	0.5	0.2832	0

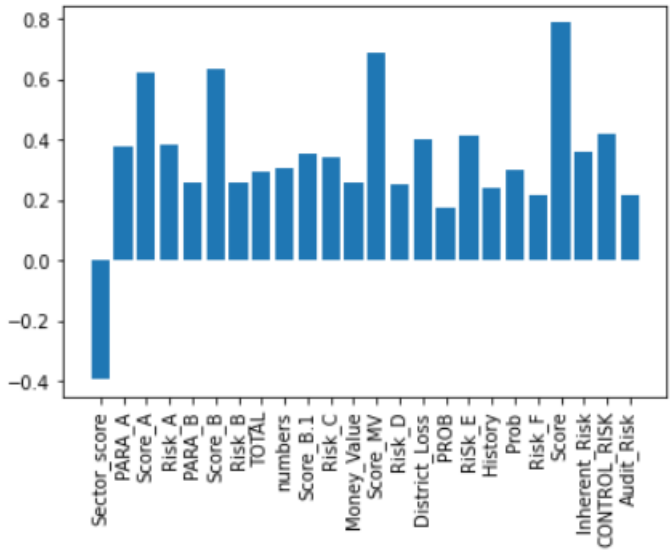
Hình 3.2: 5 bản ghi đầu tập dữ liệu kiểm toán

Các thuộc tính dùng để mô tả các công ty là các thuộc tính kinh tế cần một mức độ kiến thức nhất định để hiểu được nhưng nó không phải là trọng điểm của nghiên cứu này. Trong tập dữ liệu này, mọi dữ liệu là một số nguyên hoặc một số thực ngoại từ cột Location_ID là kiểu chuỗi và nó không có ích do vậy em quyết định loại bỏ cột này. Ngoài ra, một giá trị bị thiếu trong cột Money_Value và em quyết định xử lý missing value bằng cách loại bỏ bản ghi đó. Tập dữ liệu này khá cân bằng, có 305 công ty gian lận và 471 công ty hợp lệ.. Các thuộc tính điểm số (score) nằm trong khoảng 0 đên 10, trong khi các thuộc tính khác có các giá trị khác nhau.



Hình 3.3: Biểu đồ tương quan dữ liệu kiểm toán

Biểu đồ tương quan được trình bày ở hình 3.3 và 3.4 cho thấy mối tương quan giữa tất cả các thuộc tính. Có thể thấy rằng cột Detection_Risk không có ích, nó chỉ là các giá trị giống nhau cho tất cả các công ty: 0.5, do vậy loại bỏ thuộc tính Detection_Risk. Từ biểu đồ thì tất cả các các thuộc tính có tương quan thuận với thuộc tính Risk ngoại trừ thuộc tính Sector_score.



Hình 3.4: Biểu đồ tương quan giữa các thuộc tính với thuộc tính Risk

3.2.2 Kết quả chạy dữ liệu với các mô hình

Thuật toán HDBSCAN:

	precision	**recall**	**f1- score**	**support**
0	0.76	1.00	0.87	470
1	1.00	0.52	0.69	305
accuracy			0.81	775
macro avg	0.88	0.76	0.78	775
weighted avg	0.86	0.81	0.79	775
running time				0.0794s

Bảng 3.1: Kết quả chạy thuật toán HDBSCAN- dữ liệu kiểm toán

Thuật toán Isolation forest:

	precision	**recall**	**f1- score**	**support**
0	1.00	0.96	0.98	95
1	0.94	1.00	0.97	60
accuracy			0.97	155
macro avg	0.97	0.98	0.97	155
weighted avg	0.98	0.97	0.97	155
running time				0.4495s

Bảng 3.2: Kết quả chạy thuật toán Isolation forest- dữ liệu kiểm toán

Thuật toán Local outlier factor:

	precision	**recall**	**f1- score**	**support**
0	0.92	0.84	0.88	470
1	0.78	0.90	0.83	305
accuracy			0.86	775
macro avg	0.85	0.87	0.86	775
weighted avg	0.84	0.86	0.86	775
running time				0.05377s

Bảng 3.3: Kết quả chạy thuật toán LOF- dữ liệu tài chính với k- neighbors=200

	precision	**recall**	**f1- score**	**support**
0	0.78	0.71	0.75	470
1	0.61	0.70	0.65	305
accuracy			0.71	775
macro avg	0.70	0.70	0.70	775
weighted avg	0.72	0.71	0.71	775
running time				0.06029s

Bảng 3.4: Kết quả chạy thuật toán LOF- dữ liệu kiểm toán với k- neighbors=100

	precision	**recall**	**f1- score**	**support**
0	0.72	0.65	0.68	470
1	0.53	0.60	0.56	305
accuracy			0.63	775
macro avg	0.62	0.63	0.62	775
weighted avg	0.64	0.63	0.63	775
running time				0.08656s

Bảng 3.5: Kết quả chạy thuật toán LOF- dữ liệu kiểm toán với k- neighbors=300

Thuật toán Support vector machine:

	precision	**recall**	**f1- score**	**support**
0	0.99	0.99	0.99	95
1	0.98	0.98	0.98	60
accuracy			0.99	155
macro avg	0.99	0.99	0.99	155
weighted avg	0.99	0.99	0.99	155
running time				0.1940s

Bảng 3.6: Kết quả chạy thuật toán SVM- dữ liệu kiểm toán với phương pháp svm.LinearSVC

	precision	**recall**	**f1- score**	**support**
0	0.73	1.00	0.84	95
1	1.00	0.42	0.59	60
accuracy			0.77	155
macro avg	0.87	0.71	0.72	155
weighted avg	0.83	0.77	0.75	155
running time				0.02833s

Bảng 3.7: Kết quả chạy thuật toán SVM- dữ liệu kiểm toán với phương pháp svm.SVC

3.3 Đánh giá kết quả đạt được

Như đã trình bày kết quả chạy thuật toán trong phần trên, em sẽ đưa ra so sánh và đánh giá giữa các phương pháp khi chạy với bộ dữ liệu.

	Audit set		
	F1- score (Class 0)	**F-1 score (Class 1)**	**Accuracy**
HDBSCAN	0.87	0.69	0.81
Isolation forest	0.98	0.97	0.97
Local outlier factor	0.88	0.83	0.86
Support vector machine	0.99	0.98	0.99
Average running time	0.1942s		

Bảng 3.8: Bảng kết quả

Nhìn vào bảng kết quả được tổng hợp khi chạy các thuật toán đã giới thiệu ở phần trên có thể thấy các phương pháp đều cho ra các chỉ số đánh giá khá tốt tuy nhiên trong 4 phương pháp đã chạy với bộ dữ liệu có một phương pháp phân cụm là HDBSCAN thì cho kết quả không tốt bằng các thuật toán còn lại dựa vào các chỉ số $F_1 - score = 0.87$ với class 0, 0.69 với class 1 và $accuracy = 0.81$ thấp hơn so với độ chính xác của 3 thuật toán ML lần lượt là 0.97, 0.86, 0.99. Do đó, có thể kết luận rằng các phương pháp ML để phát hiện dữ liệu ngoại lệ là giải pháp tốt hơn các kỹ thuật ngay thơ để ứng dụng vào thực tế hiện nay.

Chương 4

Kết luận

Trong thời đại hiện này, khi thế giới và cụ thể hơn là Việt Nam đã và đang trong thời kỳ bùng nổ chuyển đổi số thì dữ liệu là một nguồn tài nguyên vô cùng đáng quý và việc phát hiện dữ liệu ngoại lệ một cách đáng tin cậy là điều cần thiết, đóng vai trò rất lớn trong quá trình phân tích dữ liệu. Phát hiện các ngoại lệ có thể là để làm sạch tập dữ liệu khỏi bất kỳ lỗi nào, hiểu dữ liệu tốt hơn hay như trong đồ án này em đã trình bày phát hiện ngoại lệ còn có ứng dụng rất lớn vào phát hiện gian lận điển hình như gian lận tài chính- vấn đề phổ biến hiện nay.

Nhìn chung, đồ án này là một cái nhìn tổng quan về dữ liệu ngoại lệ và một số những mô hình đã hoạt động như thế nào trong trường hợp phát hiện gian lận với tập dữ liệu tài chính cụ thể là tập dữ liệu về kiểm toán.

Từ đồ án có thể thấy rằng thuật toán ngây thơ là kỹ thuật mang lại kết quả không tốt với tập dữ liệu. Điều này chứng minh rằng các kỹ thuật ML để phát hiện ngoại lệ là một giải pháp thay thế tốt hơn các kỹ thuật ngây thơ trong phát hiện gian lận tài chính ngày nay. Bộ dữ liệu kiểm toán em sử dụng trong đồ án này là một bộ dữ liệu nhỏ có khoảng 800 dòng dữ liệu và nó là tập khá cân bằng giữa dữ liệu bình thường và ngoại lệ vì vậy các mô hình trong bài hiện hoạt động khá tốt với nó. Em mong muốn trong các bài báo cáo hay đồ án sau của mình có thể sử dụng các kỹ thuật trong bài này để ứng dụng vào bộ dữ liệu lớn hơn và có thể là các tập dữ liệu mất cân bằng vì nó khá phù hợp với thực tế hiện

nay để từ đó có được đánh giá tổng quan, đầy đủ hơn.

Qua quá trình làm báo cáo Đồ án I về đề tài **Ứng dụng phát hiện dữ liệu ngoại lệ vào gian lận tài chính** cùng với sự giúp đỡ và góp ý tận tình của thầy Nguyễn Hải Sơn em đã tiếp nhận được rất nhiều kiến thức mới về các thuật toán, mô hình phát hiện dữ liệu ngoại lệ và qua quá trình làm việc cùng thầy em cũng rút ra cho mình rất nhiều kinh nghiệm khi làm đồ án như: khả năng đọc hiểu tài liệu trong quá trình tìm hiểu cho bài báo cáo, cách tư duy khi giải quyết vấn đề, sắp xếp thời gian và hệ thống các vấn đề cần nghiên cứu một cách tổng quan, hình thành được những thói quen làm việc có kế hoạch và các kỹ năng viết cũng như trình bày báo cáo, đồ án.

Tài liệu tham khảo

Tiếng Anh

[1] Charu C. Aggarwal *Outlier analysis, Second Edition,* Springer international publishing, 2017.

[2] Églantine Boucher, *Outlier Dectection Methods Applied to Financial Fraud,* master thesis,2020.

[3] Laughlin, "Holistic customer insight as an engine of growth", *Journal of Direct, Data and Digital Marketing Practice,* 2014

Tiếng Việt

[4] `https://machinelearningcoban.com/tabml_book/ch_data_processing/process_outliers.html`

[5] `https://vi.tr2tr.wiki/wiki/Isolation_forest`

[6] `https://machinelearningcoban.com/2017/04/09/smv/`