Mục Lục

Danh Mục Hình

Danh Mục Bảng

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Công nghệ toàn cầu hiện nay vẫn luôn là điểm nóng, là chủ đề được bàn tán và nghiên cứu trong nhiều lĩnh vực. Đặc biệt là ứng dụng Trí Tuệ Nhân Tạo (AI) vào những vấn đề trong cuộc sống hàng ngày. Có thể giúp chúng ta thu được thông tin chuyên sâu hữu ích từ dữ liệu được đưa vào. Chúng ta có thể sử dụng những thông tin này để phân tích dự đoán nhằm giảm chi phí hoạt động, tăng độ hiệu quả từ đó giải quyết các vấn đề gặp phải một cách dễ dàng, ít tốn kém nhất.

Ngày nay, với việc dân số ngày càng tăng cao dẫn đến số lượng sinh viên nhập học mỗi năm cũng tăng lên đáng kể. Có rất nhiều trường hợp các sinh viên học đại học là để cho có, theo hướng của người khác chia sẻ, hướng dẫn chứ không tự mình tìm hiểu chính bản thân mình mà chỉ đưa ra quyết định khi mới nghe người khác góp ý. Dẫn đến việc nghỉ học, chuyển ngành, chuyển trường, v.v.. giữa chừng là việc xảy ra vô cùng nhiều trong môi trường đại học. Chính vì thế, nhóm em quyết định chọn đề tài "ỨNG DỤNG HỌC MÁY DỰ ĐOÁN KHẢ NĂNG NGHỈ HỌC CỦA SINH VIÊN TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT". Nhằm đưa ra những đánh giá, dự đoán về khả năng nghỉ học của sinh viên, dựa trên điểm số, tính cách, thái độ học tập, và các điều kiện bên ngoài khác như là khoảng cách từ nơi ở đến trường học, khả năng kinh tế gia đình,…Để có thể góp phần đưa ra lời khuyên, hướng nên theo đuổi của sinh viên một cách ổn định nhất.

1.2 Mục tiêu đề tài

Khi hoàn thành môn học và chọn đề tài thực hiện, nhóm đặt ra mục tiêu rằng sẽ dự đoán được khả năng nghỉ học của một hoặc nhiều sinh viên bằng bộ dữ liệu thu thập được.

Nhóm sử dụng dataset về điểm số của các sinh viên khoá D16-D18 do giảng viên hỗ trợ thu thập. Nhóm dựa vào đó và tiến hành các bước tiền xử lý dữ liệu kết hợp trực

quan hóa dữ liệu, trích xuất các đặc trưng của bộ dữ liệu thu được và sẽ cho ra một bảng dataset hoàn chỉnh. Tiếp đến sẽ áp dụng 3 thuật toán là Decsion Tree, Logistic Regression và Suport Vector Machine để thực hiện mục tiêu chung là hoàn thiện mô hình dùng so sánh với nhau và đưa ra kết quả so sánh giữa 3 thuật toán trên. Sau đó chọn ra mô hình có độ chính xác cao nhất để sử dụng. Nhằm phát triển theo 2 hướng sau:

1. Phát hiện sớm sinh viên gặp vấn đề:

   Việc dự đoán nếu một sinh viên có thể có xu hướng nghỉ học sẽ giúp nhà trường hoặc các cố vấn học tập có thể phát hiện sớm những sinh viên đang gặp vẫn đề và đưa ra các biện pháp nhằm giúp đỡ sinh viên.

2. Tăng hiệu quả đào tạo:

   Việc sự đoán nếu một sinh viên có xu hướng nghỉ học sẽ giúp nhà trường hoặc chương trình đào tạo có thể tăng hiệu quả đào tạo bằng cách cải thiện chương trình đào tạo hoặc kịp thời tìm kiếm các giải pháp nhằm giúp sinh viên tiếp tục học tập và đạt được kết quả của mình.

1.3 Đối tượng nghiên cứu

Sinh viên trường đại học Thủ Dầu Một.

1.4 Phạm vi nghiên cứu

Không gian: Trường đại học Thủ Dầu Một.

1.5 Các thư viện được sử dụng

1.5.1 Thư viện Pandas

Pandas là một thư viện Python cung cấp các cấu trúc dữ liệu nhanh, mạnh mẽ, linh hoạt và mang hàm ý. Tên thư viện được bắt nguồn từ panel data (bảng dữ liệu). Pandas được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc (dạng bảng, đa chiều, có tiềm năng không đồng nhất) và dữ liệu chuỗi thời gian.

Pandas rất phù hợp với nhiều loại dữ liệu khác nhau:

- 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.

Ưu điểm của thư viện Pandas:

- Dễ dàng xử lý dữ liệu mất mát, được biểu thị dưới dạng NaN, trong dữ liệu dấu phẩy động cũng như dấu phẩy tĩnh theo ý người dùng mong muốn: bỏ qua hoặc chuyển sang 0
- Khả năng thay đổi kích thước: các cột có thể được chèn và xóa khỏi DataFrame và các đối tượng chiều cao hơn
- Căn chỉnh dữ liệu tự động và rõ ràng: các đối tượng có thể được căn chỉnh rõ ràng với một bộ nhãn hoặc người dùng chỉ cần bỏ qua các nhãn và để Series, DataFrame, v.v. tự động căn chỉnh dữ liệu cho bạn trong các tính toán
- Chức năng group by mạnh mẽ, linh hoạt để thực hiện các hoạt động kết hợp phân tách áp dụng trên các tập dữ liệu, cho cả dữ liệu tổng hợp và chuyển đổi

- Dễ dàng chuyển đổi dữ liệu rời rạc (ragged), chỉ mục khác nhau (differently-indexed) trong các cấu trúc dữ liệu khác của Python và NumPy thành các đối tượng DataFrame
- Cắt lát (slicing) thông minh dựa trên nhãn, lập chỉ mục ưa thích (fancy indexing) và tập hợp lại (subsetting) các tập dữ liệu lớn
- Gộp (merging) và nối (joining) các tập dữ liệu trực quan
- Linh hoạt trong định hình lại (reshaping) và xoay (pivoting) các tập dữ liệu
- Dán nhãn phân cấp (hierarchical) của các trục (có thể có nhiều nhãn trên mỗi đánh dấu)
- Các công cụ IO mạnh mẽ để tải dữ liệu từ các tệp phẳng (flat file) như CSV và delimited, tệp Excel, cơ sở dữ liệu và lưu / tải dữ liệu từ định dạng HDF5 cực nhanh
- Chức năng theo chuỗi thời gian (time series) cụ thể: tạo phạm vi ngày và chuyển đổi tần số, thống kê cửa sổ di chuyển, dịch chuyển ngày và độ trễ.
- Tích hợp tốt với các thư viện khác của python như SciPy, Matplotlib, Plotly, v.v.
- Hiệu suất tốt

1.5.2 Thư viện Scikit-learn

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

Thư viện được cấp phép bản quyền chuẩn FreeBSD và chạy được trên nhiều nền tảng Linux và được sử dụng như một tài liệu để học tập. Sklearn được thiết kế để xử lý các thư viện số và khoa học của Python như NumPy và SciPy. Các tính năng chính của thư viện Scikit-learning bao gồm thuật toán phân loại, hồi quy và phân cụm (hỗ trợ máy vectơ, rừng ngẫu nhiên, tăng độ dốc, k-means và DBSCAN).

Scikit-learn hỗ trợ mạnh mẽ trong việc xây dựng các sản phẩm. Nghĩa là thư viện này tập trung sâu trong việc xây dựng các yếu tố: dễ sử dụng, dễ code, dễ tham khảo, dễ làm việc, hiệu quả cao.

Mặc dù được viết cho Python nhưng thực ra các thư viện nền tảng của scikit-learn lại được viết dưới các thư viện của C để tăng hiệu suất làm việc. Ví dụ như: Numpy(Tính toán ma trận), LAPACK, LibSVM và Cython.

1.5.3 Thư viện Numpy

Là một thư viện dành cho ngôn ngữ lập trình Python, hỗ trợ thêm cho mảng và ma trận lớn, nhiều chiều, cùng với một tập hợp lớn các hàm toán học cấp cao để hoạt động trên các mảng này. Tiền thân của NumPy, Numeric, ban đầu được tạo bởi Jim Hugunin với sự đóng góp của một số nhà phát triển khác. Năm 2005, Travis Oliphant đã tạo NumPy bằng cách kết hợp các tính năng của Numarray cạnh tranh vào Numeric, với nhiều sửa đổi. NumPy là phần mềm mã nguồn mở và có nhiều người đóng góp. NumPy là một dự án được tài trợ bởi NumFOCUS .

1.5.4 Thư viện Matplotlib

Matplotlib, một thư viện vẽ đồ thị cho Python vào năm 2003. Matplotlib là một thư viện vẽ đồ thị cấp thấp và là một trong những thư viện vẽ đồ thị được sử dụng rộng rãi nhất. Đây là một trong những lựa chọn đầu tiên để vẽ đồ thị để hiển thị nhanh một số dữ liệu.

Sử dụng Matplotlib, chúng ta có thể vẽ nhiều biểu đồ thú vị theo dữ liệu của mình như Biểu đồ thanh, Biểu đồ phân tán, Biểu đồ, Biểu đồ đường viền, Biểu đồ hộp, Biểu đồ hình tròn…. Chúng ta cũng có thể tùy chỉnh nhãn, màu sắc, độ dày của chi tiết biểu đồ theo chúng ta cần. Hình ảnh trên được vẽ chỉ bằng Matplotlib.

Matplotlib có một số interfaces để tương tác với thư viện matplotlib: Object-Oriented API, The Scripting Interface (pyplot), The MATLAB Interface (pylab). Pyplot và pylab đều là lightweight interfaces, tuy nhiên Pyplot cung cấp một giao diện thủ tục

các thư viện vẽ hướng đối tượng trong matplotlib. Các lệnh vẽ của nó được thiết kế tương tự với Matlab cả về cách đặt tên và ý nghĩa các đối số. Cách thiết kế này đã giúp cho việc sử dụng pyplot dễ dàng và dễ hiểu hơn vì vậy trong các bài viết về Matplotlib, tôi sẽ sử dụng giao diện pyplot thay vì hai giao diện còn lại. Nếu chúng ta muốn can thiệp sâu hơn, với nhiều tùy chỉnh hơn thì Object-Oriented API sẽ là lựa chọn thích hợp.

CHƯƠNG 2. CÁC NGHIÊN CỨU LIÊN QUAN

2.1 Nghiên cứu trong nước

2.1.1 Nghiên cứu của Trường Đại học Bách khoa Hà Nội (Trường Đại học Bách khoa Hà Nội, 2018)

Đề tài thực hiện: DỰ ĐOÁN XU THẾ CHỈ SỐ CHỨNG KHOÁN VIỆT NAM SỬ DỤNG PHÂN TÍCH HỒI QUY QUÁ TRÌNH GAUSS VÀ MÔ HÌNH TỰ HỒI QUY TRUNG BÌNH ĐỘNG [6]

Được xuất bản ngày 01/11/2018 trên tạp chí khoa học của Trường Đại học Bách khoa Hà Nội

Do 3 tác giả Huỳnh Quyết Thắng, Phùng Đình Vũ, Tống Văn Vinh thuộc Trường Đại học Bách khoa Hà Nội thực hiện

Tác giả áp dụng mô hình tự hồi quy trung bình động (ARMA: Autoregressive moving average) để dự đoán thành phần thời gian ngẫu nhiên ở một bước kế tiếp, phân tích hồi quy quá trình Gauss (GPR: Gaussian process regression) để dự đoán thành phần thời gian xu thế. Cuối cùng, kết quả dự đoán các thành phần riêng lẻ được tổng hợp lại để đưa ra kết quả dự đoán cuối cùng cho phương pháp kết hợp GPR-ARMA.

Thuật toán được sử dụng trong đề tài: Đường trung bình động hồi quy tự động (ARMA).

Với bộ dữ liệu chỉ số chứng khoán Việt Nam (VN-Index) được công khai trên các sàn giao dịch chứng khoán.

Cho ra độ chính xác của mô hình là 61,73%

2.1.2 Nghiên cứu của Trường Đại học Thương mại Hà Nội (Trường Đại học Thương mại Hà Nội, 2021)

Đề tài thực hiện: ỨNG DỤNG MỘT SỐ MÔ HÌNH HỌC MÁY TRONG DỰ BÁO CHIỀU BIẾN ĐỘNG CỦA THỊ TRƯỜNG CHỨNG KHOÁN VIỆT NAM [7]

Xuất bản ngày 13/01/2021 trên tạp chí khoa học Trường Đại học Thương mại Hà Nội.

Do nhóm tác giả gồm: Lê Văn Tuấn, Nguyễn Thu Thủy, Lê Thị Thu Giang của Bộ môn Toán thực hiện.

Tác giả sử dụng một số mô hình/thuật toán học máy để dự báo xu hướng biến động (tăng/giảm) của chỉ số thị trường chứng khoán của Việt Nam. Kết quả cho thấy, trong các mô hình hồi quy Logistic, mô hình phân tích phân biệt tuyến tính (LDA), phân tích phân biệt toàn phương (QDA) và mô hình K – lân cận (KNN): mô hình KNN(10) có độ chính xác dự báo tốt nhất.

Bộ dữ liệu: VN-Index (chỉ số đại diện cho TTCK Việt Nam), download từ trang web của Công ty Cổ phần Chứng khoán VNDIRECT.

Cho ra độ chính xác lần lượt là:

Mô hình/thuật toán	**Độ chính xác dự báo**
Hồi quy Logistic	50.39%
LDA	50%
QDA	49.60%
KNN < 0.54	< 54%

Bảng 1: Kết quả nghiên cứu của Đại học Thương Mại Hà Nội

2.2 Nghiên cứu ngoài nước

2.2.1 Nghiên cứu của Đại học Sejong Hàn Quốc, 2021

Đề tài: DỰ ĐOÁN BỎ HỌC SỚM TRONG HỌC TRỰC TUYẾN CỦA TRƯỜNG ĐẠI HỌC SEJONG HÀN QUỐC BẰNG CÁCH SỬ DỤNG HỌC MÁY

Xuất bản: 31/12/2021 đăng tải Tạp chí Quốc tế về Trực quan hoá Tin học được cấp phép theo giấy phép quốc tế.

Do Giáo sư Hee Sun Park và Tiến sĩ Seong Joon Yoo thực hiện.

Đề tài này xây dựng một mô hình dự đoán bỏ học bằng cách sử dụng mạng lưới thần kinh sâu (DNN).

Thuật toán được sử dụng: Decision Tree, Random-Forest (RF), Support Vector Machine (SVM), Deep Neural Network (DNN).

Bộ dữ liệu: Nghiên cứu này sử dụng 98.685 thông tin thống kê của sinh viên từ tháng 3 năm 2012 đến tháng 12 năm 2019 và 1.480.275 dữ liệu nhật ký được lưu trữ trong hệ thống quản lý học tập phục vụ học tập trực tuyến. Thông tin thống kê của học sinh, chẳng hạn như số lượng tuyển sinh, tình trạng học bổng, tuổi và đăng ký khóa học, được sưu tầm trong quản lý học hành chính hệ thống (ADS), và hồ sơ truy cập hàng tuần và học tập hồ sơ hoạt động được thu thập trong quản lý học tập hệ thống (LMS).

Độ chính xác các mô hình:

Decision Tree: 91%

Random-Forest: 96%

Suport Vector Machine: 81%

Deep Neural Network: 85%

2.2.1 Nghiên cứu của Đại học Bách Khoa Tây Bắc (Tây An – Trung Quốc) (Đại học Bách Khoa Tây Bắc - Trung Quốc, 2018)

Đề tài thực hiện: THUẬT TOÁN TRÍ TUỆ NHÂN TẠO NÀO DỰ ĐOÁN TỐT HƠN THỊ TRƯỜNG CHỨNG KHOÁN TRUNG QUỐC [2]

Xuất bản ngày: 25/07/2018 trên tạp chí khoa học Đại Học Bách Khoa Tây Bắc (Trung Quốc)

Do nhóm tác giả: Lin Chen, Zhilin Qiao, Minggang Wang, Chao Wang, Ruijin Du, Harry Eugene Stanley thực hiện.

Nhóm tác giả nghiên cứu thị trường tài chính, giờ đây tác giả có thể trích xuất các tính năng từ môi trường dữ liệu lớn mà không cần thông tin dự đoán trước. Ở đây, tác giả đề xuất cải thiện hơn nữa hiệu suất dự đoán này bằng cách sử dụng kết hợp mô hình dự đoán hợp đồng tương lai chỉ số chứng khoán dựa trên học sâu, bộ mã hóa tự động và máy Boltzmann bị hạn chế. tác giả sử dụng dữ liệu tần suất cao để kiểm tra hiệu suất dự đoán của học sâu và tác giả so sánh ba mạng thần kinh nhân tạo truyền thống: 1) mạng thần kinh lan truyền ngược; 2) máy học cực đoan; và 3) mạng thần kinh chức năng cơ sở xuyên tâm. tác giả sử dụng tất cả dữ liệu giao dịch tần suất cao trong 1 phút của hợp đồng tương lai CSI 300 (IF1704) trong phân tích thực nghiệm của mình và tác giả thử nghiệm ba nhóm mẫu khối lượng khác nhau để xác thực các quan sát của mình. tác giả nhận thấy rằng phương pháp học sâu để dự đoán hợp đồng tương lai chỉ số chứng khoán vượt trội so với phương pháp lan truyền ngược, máy học cực đoan và mạng thần kinh chức năng cơ sở xuyên tâm ở mức độ phù hợp và độ chính xác dự đoán theo hướng. tác giả cũng thấy rằng việc tăng lượng dữ liệu sẽ làm tăng hiệu suất dự đoán. Điều này chỉ ra rằng học sâu nắm bắt các tính năng phi tuyến tính của dữ liệu giao dịch và có thể đóng vai trò là công cụ dự đoán hợp đồng tương lai chỉ số chứng khoán mạnh mẽ cho các nhà đầu tư trên thị trường tài chính.

Thuật toán được sử dụng trong đề tài: BP(Lan truyền ngược), ELM(Mạng nơ-ron truyền thẳng một lớp ẩn), RBF(Thuật toán nội suy)

Bộ dữ liệu về chỉ số thị trường chứng khoán trung quốc CSI 300 download từ website Sở giao dịch chứng khoán Thượng Hải

Độ đo RMSE của 3 thuật toán cho ra:

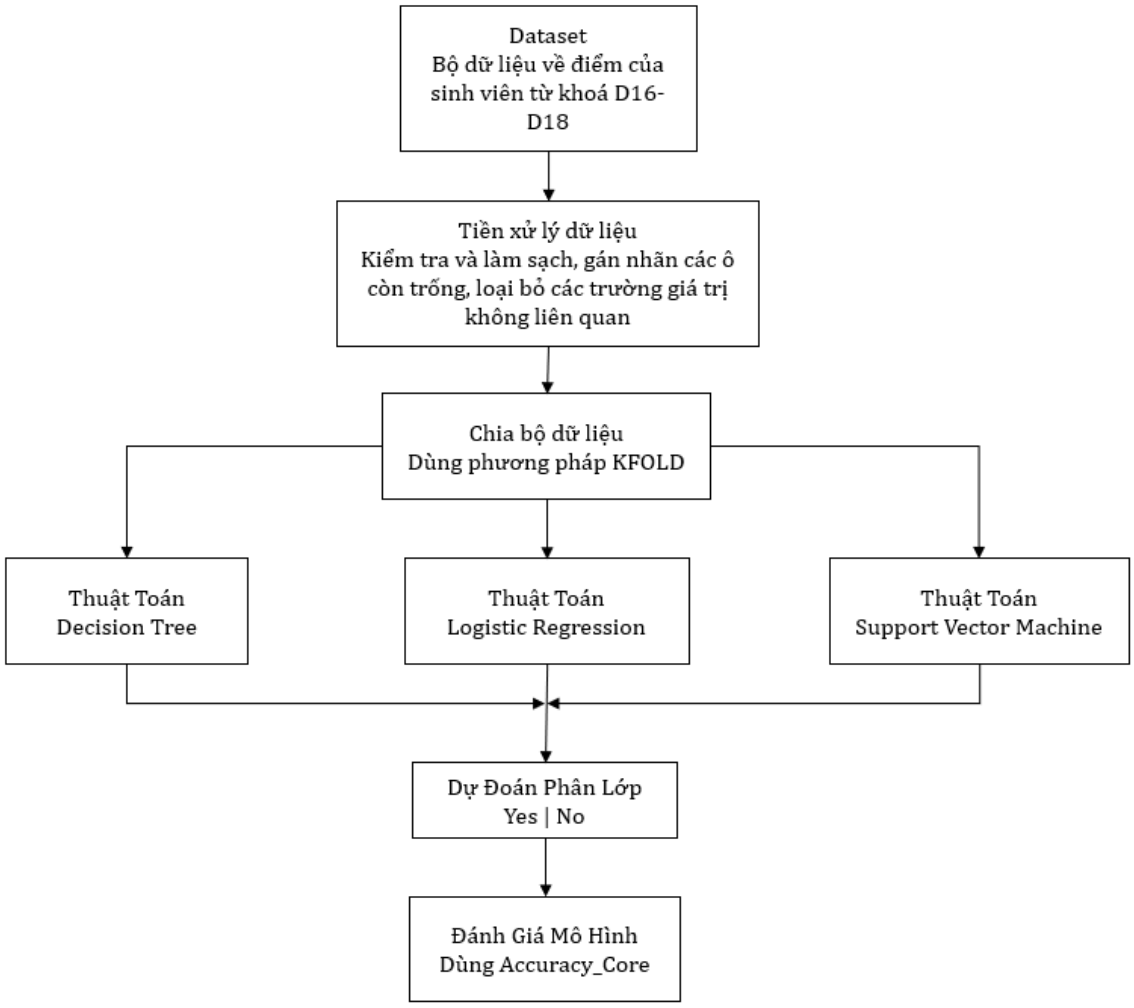
Thuật toán	Độ đo RMSE
BP	5,1203
ELM	3,2245
RBF	3,9305

Bảng 2: Kết quả nghiên cứu của Đại học Bách Khoa Tây Bắc - Trung Quốc

CHƯƠNG 3. QUY TRÌNH VÀ CÁC THUẬT TOÁN SỬ DỤNG

3.1 Xây dựng quy trình cho mô hình học máy

Khâu chuẩn bị mô hình vô cùng quan trọng do nó sẽ khái quát những việc cần làm nhằm mục đích khai phá được tri thức từ bộ dữ liệu và tiến dành dự đoán với độ chính xác cao nhất.



Hình 1: Quy trình xây dựng mô hình học máy

Quy trình gồm có 6 bước:

- **Bước 1:** Thu thập dữ liệu bao gồm điểm của sinh viên từ khoá D16-D18 trường đại học Thủ Dầu Một.
- **Bước 2:** Tiền xử lý dữ liệu bao gồm các việc: kiểm tra và làm sạch, gán nhãn các ô còn trống, loại bỏ các trường giá trị không liên quan,...
- **Bước 3:** Chia bộ dữ liệu: dùng phương pháp KFOLD với tỷ lệ 8/2
- **Bước 4:** Sử dụng thuật toán: Cây Quyết Định (Decision Tree), Hồi Quy Logistic (Logistic Regression) , Suport Vector Machine
- **Bước 5:** Tiến hành dự đoán phân lớp của mô hình dựa trên dataset
- **Bước 6:** Dùng dộ đo accuracy_core để đánh giá độ chính xác

3.2 Thuật toán Cây Quyết Định (Decision Tree).

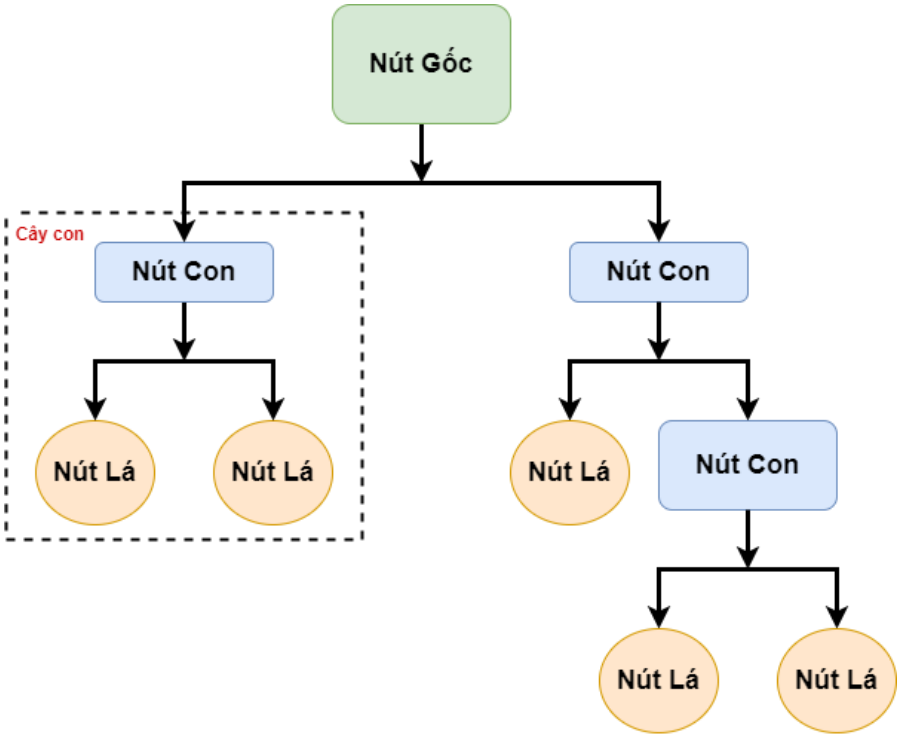
3.2.1 Thuật toán Cây Quyết Định là gì?

Cây quyết định là một phương tiện có tính mô tả dành cho việc tính toán các xác suất có điều kiện.

Thường được dùng để biểu diễn tri thức cho mục đích đưa ra quyết định hoặc kết luận.

Bắt đầu từ nút gốc, từ đó, người dùng tách thành các nút con một cách đệ quy theo luật toán học cây quyết định.

Kết quả cuối cùng là một cây quyết định trong đó mỗi nhánh biểu diễn một kịch bản có thể của quyết định và kết quả của nó.



Hình 2: Minh hoạ về cây quyết định

Ví dụ ở **Hình 2** đã mô tả được cây quyết định là một cây trong đó:

+ Nút gốc gọi là root node.
+ Những nút được tách ra từ nút gốc gọi là nút con.
+ Mỗi nút nhánh gọi là sub tree, nhằm biểu diễn một lựa chọn giữa một số khả năng.
+ Mỗi nút lá có tên là leaf node để biểu diễn một quyết định.

3.2.2. Phân loại cây quyết định

- **Cây quyết định ID3**

ID3 biểu diễn các khái niệm ở dạng các cây quyết định. Hoạt động trên biến có giá trị rời rạc.

Biểu diễn này cho phép chúng ta xác định phân loại của một đối tượng bằng cách kiểm tra các giá trị của nó trên một số thuộc tính.

ID3 sử dụng độ đo Information Gain chọn thuộc tính phân chia cây.

- **Cây quyết định C4.5**

C4.5 là thuật toán cải tiến so với ID3.

Xử lý tốt cả hai biến: rời rạc, liên tục.

Bằng cách xác định phạm vi hoặc các ngưỡng cho dữ liệu liên tục nhờ vậy dữ liệu liên tục được chuyển sang dạng rời rạc.

Cây C4.5 Sử dụng Gain Ratio thay vì Information Gain.

3.2.3. Các độ đo trong cây quyết định

- **Độ lợi thông tin Information Gain**

Độ lợi thông tin (Information Gain) là độ đo được sử dụng trong thuật toán ID3

Công thức Information Gain:

$$\mathbf{G(C, A)} = \mathbf{E(C)} - \mathbf{E(A)}$$

=> Phân hoạch lại cây quyết định với Information Gain (IG càng lớn càng tốt).

Nhược điểm: của độ đo này là có xu hướng lựa chọn thuộc tính có nhiều giá trị

- **Độ đo lựa chọn thuộc tính Gain ratio**

Sử dụng trong thuật toán C4.5

Gain ratio khắc phục nhược điểm của độ đo Information gain.

Một thuộc tính A được chọn nếu nó có giá trị Gain Ratio lớn nhất.

Công thức:

$$\text{Gain Ratio} = \frac{Information\ Gain}{Entropy}$$

3.3 Thuật toán Hồi Quy Logistic (Logistic Regression)

3.3.1 Hồi Quy Logistic là gì?

Hồi quy logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Sau đó, kỹ thuật này sử dụng mối quan hệ đã tìm được để dự đoán giá trị của những yếu tố đó dựa trên yếu tố còn lại. Dự đoán thường cho ra một số kết quả hữu hạn, như có hoặc không.

3.3.2 Tầm quan trọng của thuật toán Hồi Quy Logistic

Hồi quy logistic là một kỹ thuật quan trọng trong lĩnh vực trí tuệ nhân tạo và máy học (AI/ML). Mô hình ML là các chương trình phần mềm có thể được đào tạo để thực hiện các tác vụ xử lý dữ liệu phức tạp mà không cần sự can thiệp của con người. Mô hình ML được xây dựng bằng hồi quy logistic có thể giúp các tổ chức thu được thông tin chuyên sâu hữu ích từ dữ liệu kinh doanh của mình. Họ có thể sử dụng những thông tin chuyên sâu này để phân tích dự đoán nhằm giảm chi phí hoạt động, tăng độ hiệu quả và đổi chỉnh quy mô nhanh hơn.

Dưới đây là một số lợi ích của việc sử dụng hồi quy logistic so với các ký thuật Ml khác:

- **Tính đơn giản**

Các mô hình hồi quy logistic ít phức tạp về mặt toán học hơn các phương pháp ML khác. Do đó, chúng ta có thể triển khai chúng ngay cả khi đội ngũ của chúng ta không ai có chuyên môn sâu về ML.

- **Tốc độ**

Các mô hình hồi quy logistic có thể xử lý khối lượng lớn dữ liệu ở tốc độ cao bởi chúng cần ít khả năng điện toán hơn, chẳng hạn như bộ nhớ và sức mạnh xử lý. Điều này khiến các mô hình hồi quy logistic trở nên lý tưởng đối với những tổ chức đang bắt đầu với các dự án ML để đạt được một số thành tựu nhanh chóng.

- **Sự linh hoạt**

Chúng ta có thể sử dụng hồi quy logistic để tìm đáp án cho các câu hỏi có hai hoặc nhiều kết quả hữu hạn. Chúng ta cũng có thể sử dụng phương pháp này để xử lý trước dữ liệu.

- **Khả năng hiển thị**

Phân tích hồi quy logistic cung cấp cho nhà phát triển khả năng nhìn nhận các quy trình phần mềm nội bộ rõ hơn so với các kỹ thuật phân tích dữ liệu khác. Khắc phục sự cố và sửa lỗi cũng trở nên dễ dàng hơn do các phép toán ít phức tạp hơn.

3.3.3 Ứng dụng của thuật toán Hồi Quy Logistic

Hồi quy logistic có một số ứng dụng thực tế trong nhiều ngành công nghiệp khác nhau.

- **Sản xuất**

Các công ty sản xuất áp dụng phân tích hồi quy logistic để ước tính xác suất xảy ra sự cố ở bộ phận trong máy móc. Sau đó, họ sẽ lên lịch bảo trì dựa trên xác suất đã ước tính này để giảm thiểu sự cố trong tương lai.

- **Chăm sóc sức khỏe**

Các nhà nghiên cứu y khoa lên kế hoạch điều trị và chăm sóc dự phòng bằng cách dự đoán khả năng mắc bệnh ở bệnh nhân. Họ sử dụng các mô hình hồi quy logistic để so sánh tác động của tiền sử gia đình hoặc của bộ gen lên bệnh tật.

- **Tài chính**

Các công ty tài chính phải phân tích các giao dịch tài chính để đề phòng gian lận, xem xét các đơn xin vay và đơn bảo hiểm để đề phòng rủi ro. Những vấn đề này phù hợp với mô hình hồi quy logistic bởi chúng có kết quả cụ thể, chẳng hạn như rủi ro cao hoặc rủi ro thấp và gian lận hoặc không gian lận.

- **Bộ phận tiếp thị**

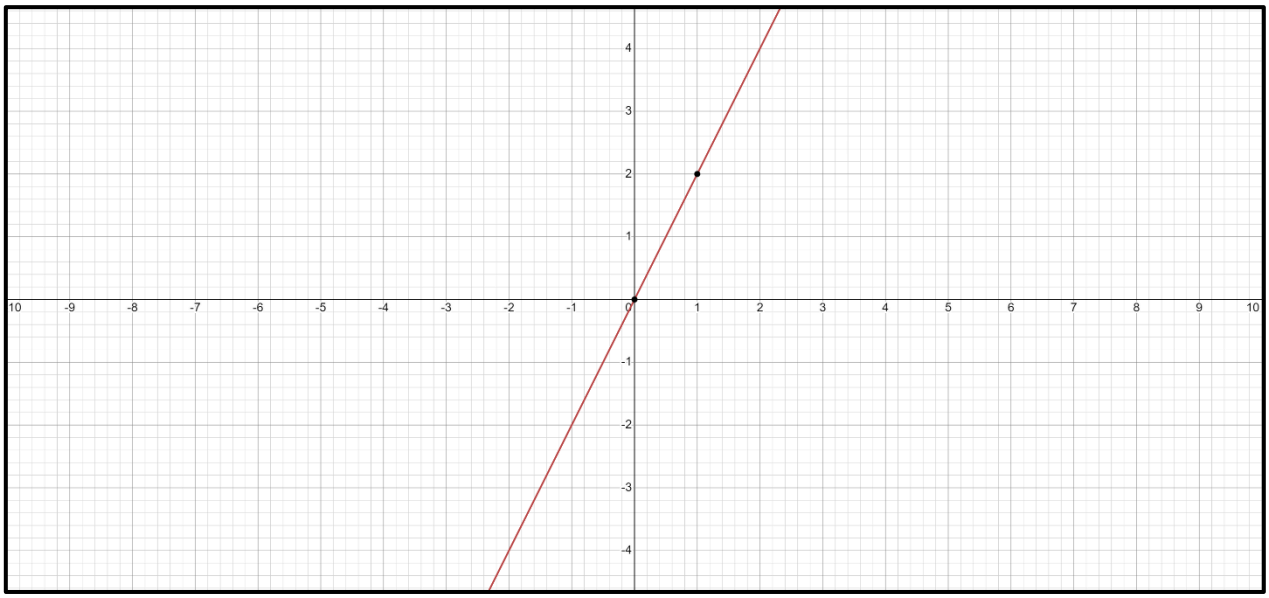
Các công cụ quảng cáo trực tuyến sử dụng mô hình hồi quy logistic để dự đoán xem người dùng sẽ nhấp vào một quảng cáo hay không. Kết quả là, các nhà tiếp thị có thể phân tích phản ứng của người dùng đối với những từ ngữ và hình ảnh khác nhau, tạo ra các quảng cáo hiệu suất cao có khả năng thu hút khách hàng.

3.3.4 Mô hình hoạt động của thuật toán Hồi Quy Logistic

Phương trình

Trong toán học, phương trình cho ta mối quan hệ giữa hai biến: *x* và *y*. Chúng ta có thể sử dụng các phương trình hoặc hàm này để vẽ đồ thị theo trục x và trục y bằng cách nhập các giá trị khác nhau của *x* và *y*.

Ví dụ: nếu chúng ta vẽ đồ thị cho hàm *y* = 2**x*, chúng ta sẽ có một đường thẳng như hình dưới đây. Do đó hàm này còn được gọi là hàm tuyến tính.



Hình 3: Đồ thị phương trình Y= 2*x

Biến hồi quy logistic

Trong thống kê, biến là các yếu tố dữ liệu hoặc thuộc tính có giá trị khác nhau. Bất kỳ phân tích nào cũng có một số biến nhất định là biến độc lập hoặc biến giải thích. Những thuộc tính này là nguyên nhân của một kết quả. Các biến khác là biến phụ thuộc hoặc biến đáp ứng; giá trị của chúng phụ thuộc vào các biến độc lập.

Nhìn chung, hồi quy logistic khám phá cách các biến độc lập ảnh hưởng đến một biến phụ thuộc bằng cách xem xét các giá trị dữ liệu lịch sử của cả hai biến.

Hàm hồi quy logistic

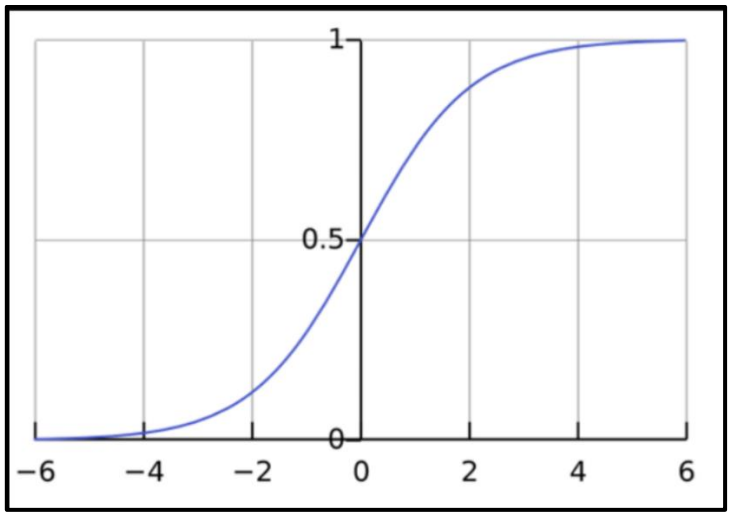
Hồi quy logistic là một mô hình thống kê sử dụng hàm logistic, hay hàm logit trong toán học làm phương trình giữa *x* và *y*. Hàm logit ánh xạ *y* làm hàm sigmoid của *x*.

Sigmoid Function là gì?

Hàm số Sigmoid

$$f(x) = \frac{1}{1+e^{-x}}$$

Sigmoid Function (Hàm Sigmoid) còn được gọi là đường cong Sigmoid. Đây là một hàm toán học có đặc trưng là đường cong hình chữ S. Nó thể hiện cho sự biến đổi các giá trị giữa phạm vi 0 và 1. Nó là một trong những hàm kích hoạt (activation function) phi tuyến tính được sử dụng rộng rãi nhất.



Hình 4: Đồ thị hàm số Sigmoid

Sigmoid Function dùng để làm gì?

Tất cả các hàm Sigmoid đều có một đặc điểm chung. Chúng có thể chuyển những con số đầu vào thành một phạm vi nhỏ nhất định. Cụ thể, các con số đầu vào sẽ chuyển thành từ 0 đến 1 hoặc -1 và 1. Nghĩa là, Hàm Sigmoid dùng để chuyển một giá trị thực thành một giá trị kiểu xác suất.

Hàm Sigmoid sẽ nhận đầu vào (input) và thực hiện những công việc sau:

- Nếu biến đầu vào âm, hàm Sigmoid sẽ chuyển gần như tất cả thành một số gần với 0.
- Với gần như tất cả đầu vào dương, hàm Sigmoid sẽ biến đầu vào thành một số gần với 1.
- Trường hợp đầu vào tương đối gần 0, hàm Sigmoid sẽ chúng thành số bất kỳ từ 0 đến 1.

3.3.5 Phân tích thuạt toán Hồi Quy Logistic với nhiều biến độc lập

Trong nhiều trường hợp, nhiều biến giải thích ảnh hưởng đến giá trị của biến phụ thuộc. Để lập mô hình các tập dữ liệu đầu vào như vậy, công thức hồi quy logistic phải

giả định mối quan hệ tuyến tính giữa các biến độc lập khác nhau. Chúng ta có thể sửa đổi hàm sigmoid và tính toán biến đầu ra cuối cùng như sau.

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\beta 0} + \boldsymbol{\beta 1 x 1} + \boldsymbol{\beta 2 x 2} + \cdots \boldsymbol{\beta n x n})$$

+ Ký hiệu β đại diện cho hệ số hồi quy.
+ Mô hình logit có thể đảo ngược tính toán các giá trị hệ số này khi chúng ta cho nó một tập dữ liệu thực nghiệm đủ lớn có các giá trị đã xác định của cả hai biến phụ thuộc và biến độc lập.

3.3.6 Log của tỷ số odds

Mô hình logit cũng có thể xác định tỷ số thành công trên thất bại hay log của tỷ số odds.

Ví dụ: nếu đang chơi poker với bạn bè và thắng bốn ván trên mười ván, tỷ số chiến thắng của bạn là bốn phần sáu, hoặc 4/6 và đó là tỷ số thành công trên thất bại của bạn. Mặt khác, xác suất thắng là 4/10.

Về mặt toán học, tỷ số odds về mặt xác suất là p/(1 - p) và log của tỷ số odds là log (p/(1 - p)). Có thể biểu diễn hàm logistic bằng log của tỷ số odds

Phương trình Log Odds Logit:

$$\textbf{Logit Funtion} = \log\left(\frac{\mathbf{p}}{\mathbf{1}-\mathbf{p}}\right)$$

3.3.7 Phân loại thuật toán Hồi Quy Logistic

Có ba cách tiếp cận phân tích hồi quy logistic dựa trên kết quả của biến phụ thuộc.

- **Hồi Quy Logistic nhị phân**

Hồi quy logistic nhị phân phù hợp với các vấn đề phân lớp nhị phân chỉ có hai kết quả có thể xảy ra. Biến phụ thuộc chỉ có thể có hai giá trị, chẳng hạn như có và không hoặc 0 và 1.

- **Hồi Quy Logistic đa thức**

Hồi quy đa thức có thể phân tích các vấn đề có một số kết quả có thể xảy ra, miễn là số kết quả hữu hạn.

Ví dụ: kỹ thuật này có thể dự đoán xem giá nhà sẽ tăng 25%, 50%, 75% hay 100% dựa trên dữ liệu dân số, nhưng sẽ không thể dự đoán được giá trị chính xác của một ngôi nhà.

- **Hồi Quy Logistic thứ tự**

Hồi quy logistic thứ tự, hay mô hình logit có thứ tự, là một loại hồi quy đa thức đặc biệt cho các vấn đề trong đó các số đại diện cho các bậc chứ không phải là giá trị thực tế.

3.3.8 Mô tả mục tiêu và các tiêu chí so sánh

Sử dụng thư viện Matplotlib và Seaborn để trực quan hoá dữ liệu, vẽ biểu đồ nhiệt so sánh mức độ liên quan giữa các cột, sau đó loại bỏ những cột không liên quan và đưa dataset "sạch" này vào mô hình.

Sử dụng 2 thuật toán phân lớp là Hồi Quy Logistic và Cây quyết định để cho máy học sau đó so sánh độ chính xác của từng mô hình.

Từ 10559 dòng dữ liệu sẽ lấy ngẫu nhiên ra 80% để thực hiện train và 20% còn lại dùng để test sau đó sẽ dùng "F1 Core", "Accuracy Core" để tính độ chính xác.

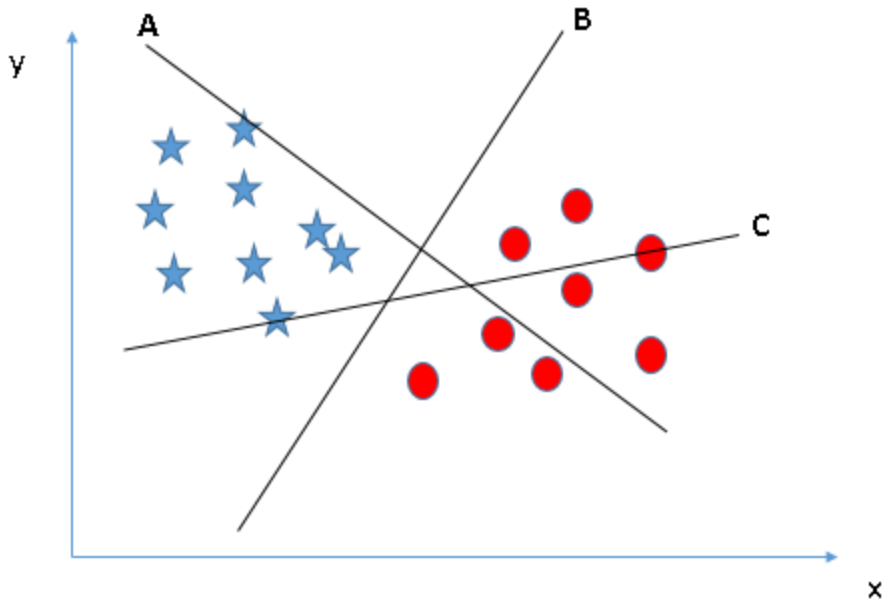
3.4 Thuật toán Suport Vector Machine

3.4.1 Suport Vector Machine là gì?

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi 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

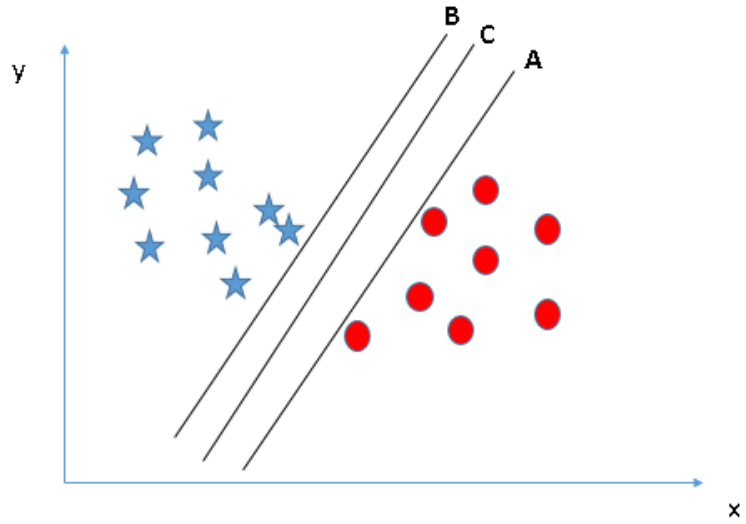
"đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

Để xác định đúng hyper-plane, chúng ta phả xác định theo lần lượt các quy tắc sau đây.



Hình 5: Quy tắc 1

- Quy tắc 1: Ở hình trên, ta dễ dàng thấy đường hyper-plane tốt nhất chính là đường B, vì đường B có thể chia 2 phần Ngôi sao và Hình tròn thành 2 phần riêng biệt nhau.

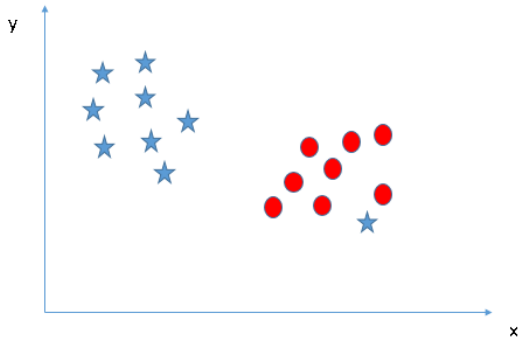


Hình 6: Quy tắc 2

- Quy tắc 2: Ở hình này ta có thể thấy cả 3 đường hyper-plane (A, B, C) đều thoả mãn quy tắc thứ nhất. Nhưng ở quy tắc 2 đó là xác định khoảng cách Lớn nhất từ điểm gần nhất của một lớp nào đó đến đường hyper-plane. Khoảng cách này được gọi là "Margin". Trở lại hình trên, trong 3 đường hyper-plane đó đường có khoảng cách Margin lớn nhất là đường C.

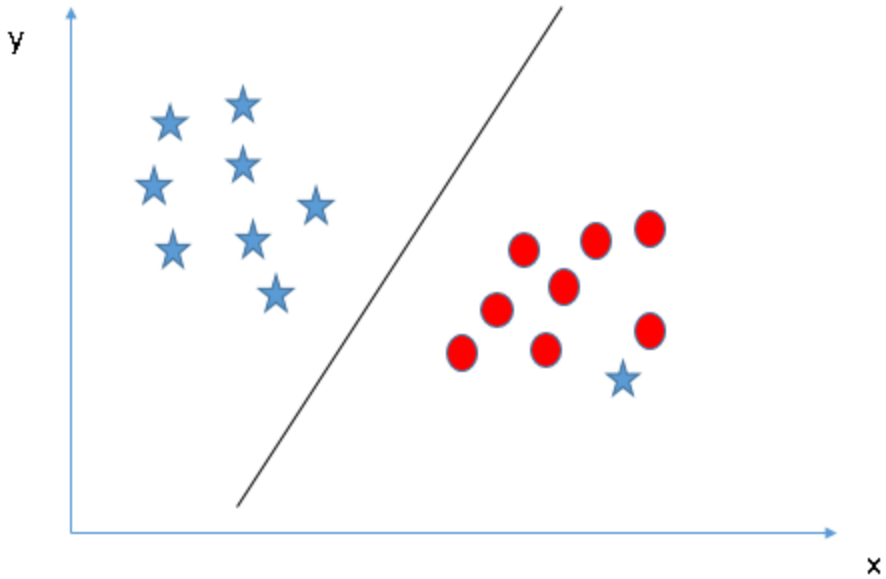
  * Lưu Ý: Nếu chọn nhầm hyper-plane có Margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu.

Với trường hợp là hình bên dưới, ta không thể chia thành 2 lớp riêng biệt với 1 đường thẳng để tạo 1 phần chỉ có Ngôi sao và 1 phần chỉ có Hình tròn.



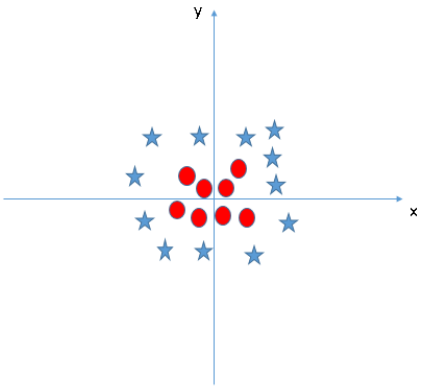
Hình 7: Trường hợp ngoại lệ 1

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



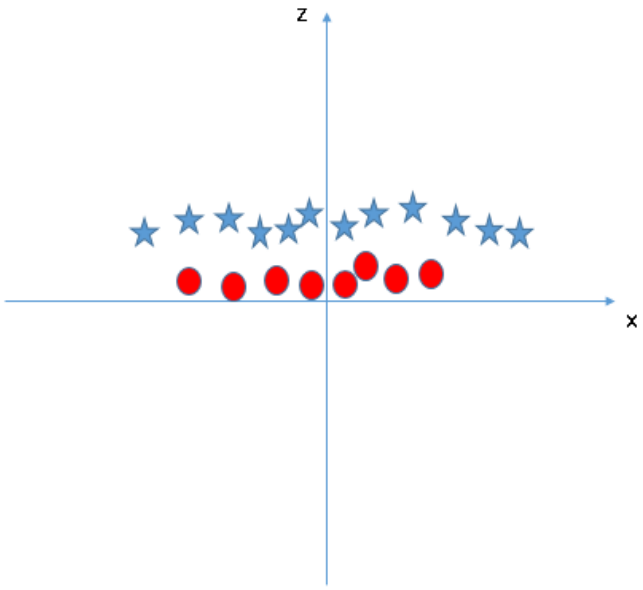
Hình 8: Đáp án của trường hợp ngoại lệ 1

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 2 lớp riêng biệt? Nãy giờ, chúng ta chỉ nói đến việc nhìn vào các đường tuyến tính hyper-plane.



Hình 9: Trường hợp ngoại lệ 2

SVM có thể giải quyết vấn đề này khá đơn giản, bằng việc nó sẽ them một tính năng vào. Ở đây, chúng ta sẽ them tính năng "z = x^2+y^2". Bây giờ dữ liệu sẽ được biến đỗi theo trục X và Z như sau.

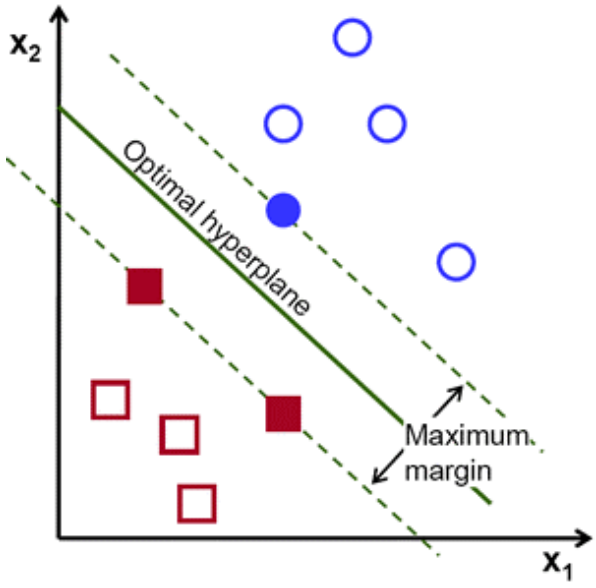


Hình 10: Đáp án của trường hợp ngoại lệ 2

Trong hình trên, các điểm cần xét là tất cả dữ liệu trên trục Z là số dương vì nó là tổng bình phương của trục 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 thì (Z, X).

Ta không cần phải thêm tính năng bằng tay một cách thủ công bởi vì, trong 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 thấm và biến đổi nó thành không gian có chiều cao hơn, các tính năng này được gọi là kernel.

3.4.2 Margin trong Supor Vector Machine (SVM)?



Hình 11: 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

CHƯƠNG 4. XÂY DỰNG MÔ HÌNH

4.1 Giới thiệu về tập dữ liệu (Dataset)

Dataset được thu thập tại trường đại học Thủ Dầu Một gồm nhiều file điểm của sinh viên được lưu lại.

4.2 Cấu hình máy để thực nghiệm

- Môi trường: Ngôn ngữ: Python 3.7
- Sản phẩm được viết và thực thi bằng IDE Visual Studio Code.
- Chạy trên thiết bị: Asus TUF Gaming F15 FX506LH (HN002T), CPU Intel core i5-10300H, Ram 8GB, SSD 512GB, GTX 1650 4GB, Window 11.

4.3 Tiền xử lý dữ liệu

4.3.1 Các trường của dataset

Bộ dữ liệu bao gồm 4 sheet (Nam1, Nam2, Nam3, Nam4).

1. Sheet Nam1 gồm 12 cột và 153 dòng.
2. Sheet Nam2 gồm 18 cột và 153 dòng.
3. Sheet Nam3 gồm 29 cột và 153 dòng.
4. Sheet Nam4 gồm 38 cột và 153 dòng.

Câu lệnh in ra toàn bộ dataset:

```
ulr = "Data_preprocessing_finish.xlsx"

data = pd.read_excel(ulr, sheet_name="Nam1")

print(data)
```

Bảng 3: Câu lệnh in ra toàn bộ dataset (Sheet Nam1)

	Mon1	Mon2	Mon3	Mon4	Mon5	Mon6	Mon1	Mon2	Mon3	Mon4	Mon5	KQ
0	0	5.6	0	5.2	6.5	7.4	6.7	5.6	5.3	7.0	5.3	1
1	0	8.9	0	6.2	9.0	5.9	8.5	6.5	8.6	6.6	8.3	1
2	0	8.3	0	5.8	5.9	5.6	8.4	6.3	5.2	7.0	6.1	1
3	0	8.1	0	5.4	6.8	6.0	8.5	6.8	5.7	7.7	6.1	1
4	0	7.2	0	6.0	6.7	5.5	8.2	5.8	5.1	7.3	6.9	1
…	…	…	…	…	…	…	…	…	…	…	…	…
147	6.5	6.8	0	8.0	6.8	5.5	7.3	5.3	4.5	5.2	7.3	1
148	6.5	5.8	0	7.3	6.8	5.4	7.8	6.0	7.3	5.1	7.0	1
149	5.8	7.0	0	7.3	7.3	6.0	8.5	6.3	6.3	5.7	7.5	1
150	6.0	5.8	0	5.0	5.0	5.3	0	5.8	0	0	7.5	0
151	5.5	6.8	0	3.8	5.5	5.4	6.0	5.3	8.0	5.3	7.3	1

Bảng 4: Kết quả in ra màn hình toàn bộ dataset (Nam1)

Sau đó sẽ dùng hàm info() được hỗ trợ bởi thư viện Sklearn để xem khái quát về dataset.

```
# Khái quát về dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 152 entries, 0 to 151
Data columns (total 38 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Mon1    152 non-null    float64
 1   Mon2    152 non-null    float64
 2   Mon3    152 non-null    float64
 3   Mon4    152 non-null    float64
 4   Mon5    152 non-null    float64
```

```
 5   Mon6    152 non-null    float64
 6   Mon1.1  152 non-null    float64
 7   Mon2.1  152 non-null    float64
 8   Mon3.1  152 non-null    float64
 9   Mon4.1  152 non-null    float64
 10  Mon5.1  152 non-null    float64
 11  Mon1.2  152 non-null    float64
 12  Mon2.2  152 non-null    float64
 13  Mon3.2  152 non-null    float64
 14  Mon4.2  152 non-null    float64
 15  Mon5.2  152 non-null    float64
 16  Mon6.1  152 non-null    float64
 17  Mon1.3  152 non-null    float64
 18  Mon2.3  152 non-null    float64
 19  Mon3.3  152 non-null    float64
 20  Mon4.3  152 non-null    float64
 21  Mon5.3  152 non-null    float64
 22  Mon6.2  152 non-null    float64
 23  Mon1.4  152 non-null    float64
 24  Mon2.4  152 non-null    float64
 25  Mon3.4  152 non-null    float64
 26  Mon4.4  152 non-null    float64
 27  Mon5.4  152 non-null    float64
 28  Mon1.5  152 non-null    float64
 29  Mon2.5  152 non-null    float64
 30  Mon3.5  152 non-null    float64
 31  Mon4.5  152 non-null    float64
 32  Mon5.5  152 non-null    float64
 33  Mon6.3  152 non-null    float64
 34  Mon1.6  152 non-null    float64
 35  Mon2.6  152 non-null    float64
 36  Mon3.6  152 non-null    float64
 37  KQ      152 non-null    int64
dtypes: float64(37), int64(1)
memory usage: 45.2 KB
None
```

Bảng 5: Câu lệnh và output khái quát về dataset bằng hàm info

4.3.2 Kiểm tra các giá trị rỗng trong dataset

```
# Kiểm tra xem có giá trị nào trong dataset bị Null hay không
data.isnull().sum()
```

```
Mon1      0
Mon2      0
Mon3      0
Mon4      0
Mon5      0
Mon6      0
Mon1.1    0
Mon2.1    0
Mon3.1    0
Mon4.1    0
Mon5.1    0
Mon1.2    0
Mon2.2    0
Mon3.2    0
Mon4.2    0
Mon5.2    0
Mon6.1    0
Mon1.3    0
Mon2.3    0
Mon3.3    0
Mon4.3    0
Mon5.3    0
Mon6.2    0
```

Mon1.4 0 *Mon2.4 0* *Mon3.4 0* *Mon4.4 0* *Mon5.4 0* *Mon1.5 0* *Mon2.5 0* *Mon3.5 0* *Mon4.5 0* *Mon5.5 0* *Mon6.3 0* *Mon1.6 0* *Mon2.6 0* *Mon3.6 0* *KQ 0* *dtype: int64*

Bảng 6: Câu lệnh Kiểm tra giá trị Null trong dataset

→ Kết quả trả về không có giá trị Null trong dataset

 # Kiểm tra xem có giá trị nào trong dataset bị NaN hay không *data.isna().sum()*
Noisinh 0 *Mon1 0* *Mon2 0* *Mon3 0* *Mon4 0* *Mon5 0*

```
Mon6      0
Mon1.1    0
Mon2.1    0
Mon3.1    0
Mon4.1    0
Mon5.1    0
KQ        0
dtype: int64
```

Bảng 7: Câu lệnh Kiểm tra giá trị NaN trong dataset

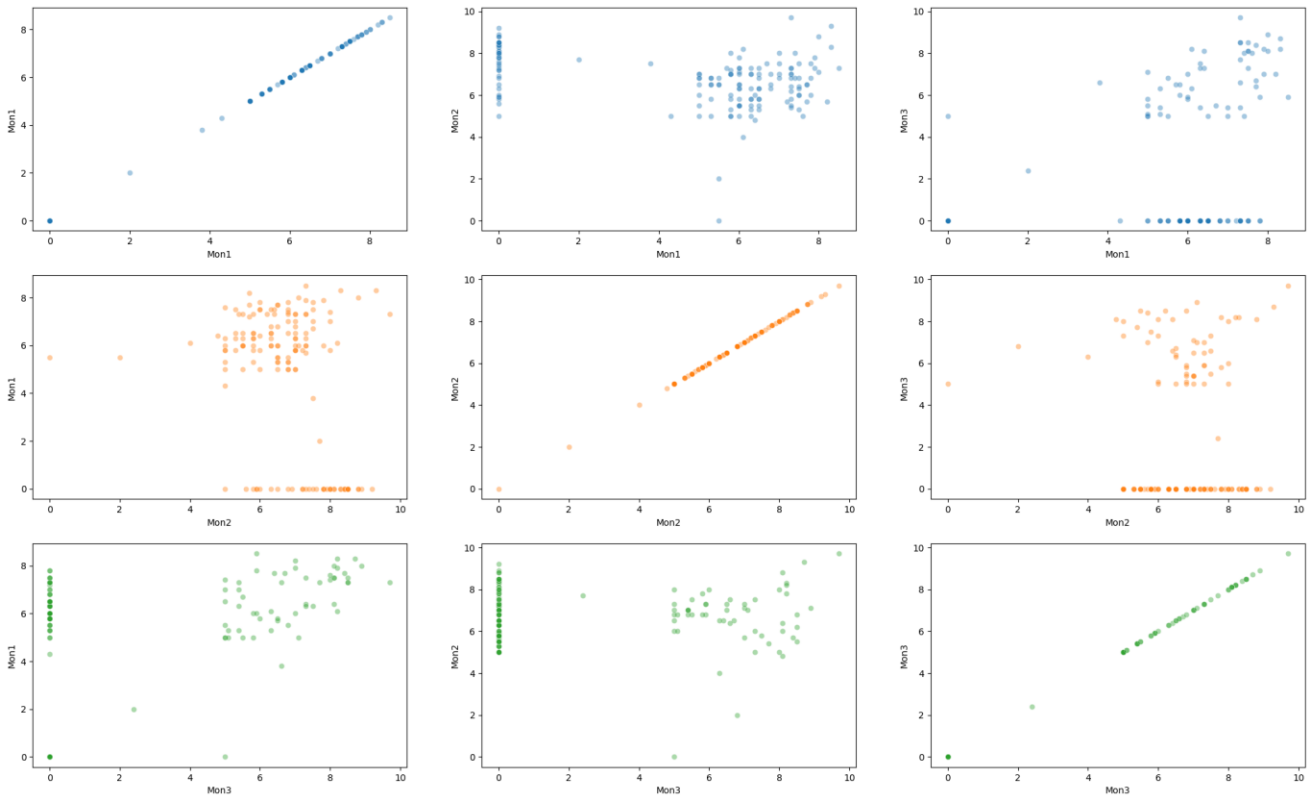
→ Kết quả trả về không có giá trị NaN trong dataset

4.3.3 Trực quan hoá dữ liệu

Dùng thư viện matplotlib để vẽ biểu đồ phân tán các giá trị của từng trường dữ liệu

```
# Vẽ biểu đồ phân tán toàn bộ các giá trị trong dataset
import seaborn as sns
plt.subplots(4,4,figsize=(25, 15))
n = 0
for x, c in zip(['Mon1', 'Mon2', 'Mon3'], list(sns.color_palette())):
   for y in ['Mon1', 'Mon2', 'Mon3']:
      plt.subplot(3,3,n+1)
      sns.scatterplot(x = x, y = y, data = data, color = c, alpha = 0.4)
      n += 1
```

Bảng 8: Câu lệnh vẽ biểu đồ phân tán giá trị các cột trong dataset



Hình 12: Kết quả in ra màn hình kết quả phân tán của dữ liệu

Hình 12 là kết quả của biểu đồ phân tán cho ra thấy rằng các giá trị tăng tuyến tính qua từng mốc khác nhau. Các giá trị từng cột khi phân tán thì không có giá trị nào bị lệch khỏi đường 'tuyến tính' quá xa, cho thấy rằng không có giá trị Outliers tồn tại trong bộ dữ liệu

4.4 Thực hiện mô hình

Tiến hành phân chia bộ dữ liệu theo phương pháp K-FOLD với tỷ lệ là 8/2 tương ứng với 80% bộ dữ liệu sẽ được dùng để cho máy học và dùng 20% còn lại kiểm tra.

```
# Phân chia dataset
array = data.values
X = array[:,0:data.shape[1]-1]
```

```
Y = array[:,data.shape[1]-1]
Y = Y.astype('int')
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
test_size=validation_size, random_state=seed)
```

Bảng 9: Câu lệnh phân chia dataset theo phương pháp Hold-Out tỷ lệ 8/2

4.5 So sánh độ chính xác của 3 thuật toán

Thư viện Sklearn hỗ trợ thuật toán Cây Quyết Định bằng hàm có tên DecisionTreeClassifier() dựa theo độ đo 'Entropy' với độ sâu tối đa là 3. Thuật toán Hồi Quy Logistic với tên hàm là LogisticRegression(), và thuật toán Suport Vector Machine với tên hàm là SVC().

```
models = []
models.append(('DTC', DecisionTreeClassifier()))
models.append(('LR', LogisticRegression()))
models.append(('SVM', SVC()))

results = []
names = []
for name, model in models:
   kfold = model_selection.KFold(n_splits=10, random_state=None)
   cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold,
   scoring=scoring)
   results.append(cv_results)
   names.append(name)
   msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
   print(msg)
```

>>Output

DTC: 0.892949 (0.111511)

LR: 0.925000 (0.058333)

SVM: 0.975000 (0.038188)

Bảng 10: Câu lệnh triển khai quá trình chia dữ liệu và do độ chính xác của 3 thuật toán

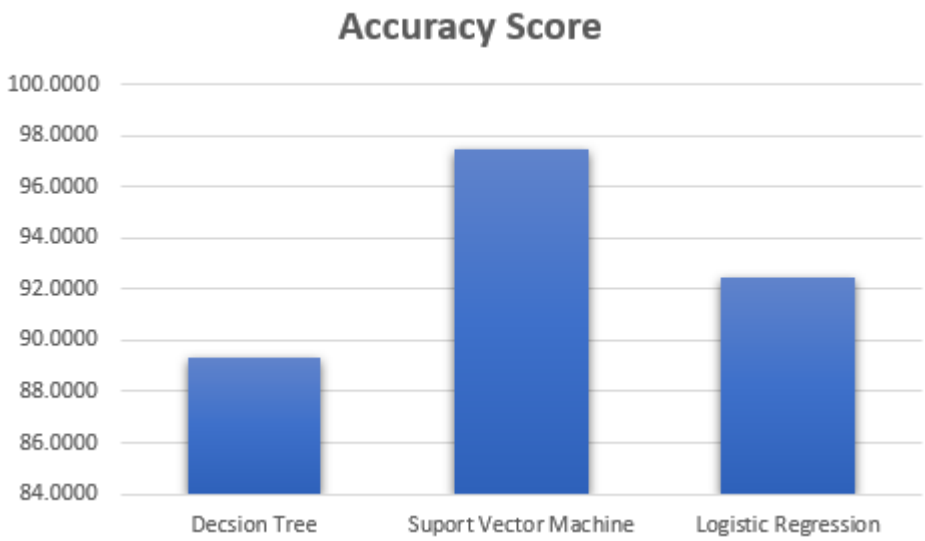
Sau khi đưa bộ dữ liệu train và test vào thuật toán Decision Tree và thang đo độ chính xác là 'Accuracy_Core' cho ra kết quả 89,2949% độ chính xác, thuật toán Logistic Regression cho ra kết quả là 92,5000% độ chính xác, cuối cùng là thuật toán Suport Vector Machine là 97,5000%.

4.6 So sánh kết quả sau học máy

Với cùng bộ dữ liệu đã được trích chọn thuộc tính và đưa vào 3 thuật toán khác nhau, sau khi trải qua bước học máy, thì cho ra 3 kết quả tính phần trăm độ chính xác dự đoán được với thang đo độ chính xác là Accuracy_Core

Độ đo / Thuật toán	Accuracy_Core
Decsion Tree	89,2949%
Suport Vector Machine	97,5000%
Logistic Regression	92,5000%

Bảng 11: Bảng so sánh các kết quả sau quá trình học máy



Hình 13: Biểu đồ cột so sánh kết quả độ chính xác của các thuật toán

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Từ bộ dữ liệu nhóm được giảng viên hỗ trợ thu thập từ trường đại học Thủ Dầu Một gồm nhiều file điểm của sinh viên được lưu lại. Tiếp đến nhóm tiến hành các bước của quy trình xây dựng mô hình và tiến hành thực nghiệm, đánh giá kết quả của mô hình.

Kết quả so sánh các thuật toán Logistic Regression, Decsion Tree và Suport Vector Machine cho thấy thuật toán Supor Vector Machine có độ chính xác tốt nhất.

5.2 Hướng phát triển trong tương lai

- Thêm giao diện trực quan cho người sử dụng.
- Thu thập thêm nhiều dữ liệu thực về con người như hành động, cảm xúc, tính cách,… cộng với việc phân tích dữ liệu rõ ràng hơn và sử dụng thêm các thuật toán Neural Network như: RNN, ANN,... để tăng độ chính xác cho mô hình Machine Learning
- Dùng thêm PCA để giảm kích thước số chiều của dữ liệu từ đó tăng thêm tốc độ xử lý và độ chính xác của mô hình

TÀI LIỆU THAM KHẢO

[1] Dataset. (2022, 12 23). *Địa chỉ file dữ liệu*. From File dữ liệu dưới dạng csv: https://drive.google.com/file/d/1Jjm7Mw4yRZUKcVLMBdk-2opeCG8t35P4/view?usp=sharing

[2] Đại học Bách Khoa Tây Bắc - Trung Quốc. (2018, 07 25). *Ieeexplore*. From Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8419702

[3] Đại học Thành Quân Quán - Hàn Quốc. (2021, 05 06). *Ieeexplore*. From A Machine Learning-Based Early Warning System for the Housing and Stock Markets: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9424620

[4] Kaggle.com. (2022, 10 23). *Kaggle Dataset*. From Kaggle datasets download -d meerashareef/apple-revenue-from-1980-to-2022: https://www.kaggle.com/datasets/meerashareef/apple-revenue-from-1980-to-2022

[5] Trần Cao Minh. (2022, 12 23). *Thực thi tại Google Colab*. From Soure Đồ Án Môn Học: https://colab.research.google.com/drive/1ZaYXAas8tWO-Ga2AjWBWV4SkamFpn0Ml?usp=sharing

[6] Trường Đại học Bách khoa Hà Nội. (2018, 11 01). *ResearchGate*. From Vietnam Stock Index Trend Prediction using Gaussian Process Regression and Autoregressive Moving Average Model: https://www.researchgate.net/publication/330743514_Vietnam_Stock_Index_Trend_Prediction_using_Gaussian_Process_Regression_and_Autoregressive_Moving_Average_Model