TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

VIỆN KỸ THUẬT – CÔNG NGHỆ

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN THAM GIA

CUỘC THI SINH VIÊN NGHIÊN CỨU KHOA HỌC NĂM HỌC 2022-2023

ỨNG DỤNG HỌC MÁY DỰ ĐOÁN KHẢ NĂNG NGHỈ HỌC CỦA SINH VIÊN TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

Sinh viên/Nhóm Sinh viên thực hiện:

Nguyễn Hữu Nghĩa - 2124802050013

Giảng viên hướng dẫn:

ThS. Hồ Ngọc Trung Kiên

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

VIỆN KỸ THUẬT – CÔNG NGHỆ

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN THAM GIA

CUỘC THI SINH VIÊN NGHIÊN CỨU KHOA HỌC NĂM HỌC 2022-2023

ỨNG DỤNG HỌC MÁY DỰ ĐOÁN KHẢ NĂNG NGHỈ HỌC CỦA SINH VIÊN TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

STT	Họ và tên SV	Giới tính	Dân tộc	Lớp, Viện	SV năm thứ/ Số năm đào tạo	Ngành học	Ghi chú
1	Nguyễn Hữu Nghĩa	Nam	Kinh	D21TTNT01, viện kỹ thuật - công nghệ	2/4	Trí tuệ nhân tạo	SV thực hiện chính

Người hướng dẫn: ThS. Hồ Ngọc Trung Kiên

UBND TỈNH BÌNH DƯƠNG
TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung:

- Tên đề tài: Ứng dụng học máy dự đoán khả năng nghỉ học của sinh viên trường đại học Thủ Dầu Một.

- Sinh viên/ nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV	Lớp	Viện	Năm thứ/ Số năm đào tạo
1	Nguyễn Hữu Nghĩa	2124802050013	D21TTNT01	KT-CN	**2.5/4**

- Người hướng dẫn: ThS. Hồ Ngọc Trung Kiên

2. Mục tiêu đề tài:

Khoanh vùng, đưa ra dự đoán về khả năng nghỉ học của sinh viên trên điểm số, ngành học, số tuổi, kinh tế, khoảng cách từ nơi ở so với trường học,…Từ đó nhà trường có thể đưa ra những phương án tuyển sinh, tư vấn ngành học cho sinh viên để giảm thiểu tối đa số lượng sinh viên nghỉ học sau một thời gian.

4. Kết quả nghiên cứu:

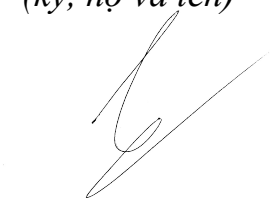
Ứng dụng dự đoán khả năng nghỉ học của từng sinh viên.

5. Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:

Giáo dục và đào tạo: Khả năng ứng dụng hỗ trợ phát hiện kịp thời sinh viên có khả năng nghỉ học, để nhà trường có thể đưa ra những phương pháp giúp đỡ sinh viên.

6. Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài *(ghi rõ họ tên tác giả, nhan đề và các yếu tố về xuất bản nếu có)* hoặc nhận xét, đánh giá của cơ sở đã áp dụng các kết quả nghiên cứu *(nếu có)*:

Ngày 17 tháng 04 năm 2022
Sinh viên chịu trách nhiệm chính thực hiện đề tài
(ký, họ và tên)



Nguyễn Hữu Nghiu

Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài *(phần này do người hướng dẫn ghi):*

.....
.....
.....
.....
.....
.....
.....
.....

Ngày 17 tháng 4 năm 2023

Xác nhận của lãnh đạo viện
(ký, họ và tên)

Người hướng dẫn
(ký, họ và tên)

UBND TỈNH BÌNH DƯƠNG
TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

THÔNG TIN VỀ SINH VIÊN

CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI

I. SƠ LƯỢC VỀ SINH VIÊN:

Họ và tên: Nguyễn Hữu Nghĩa

Sinh ngày: 6 tháng 06 năm 2003

Nơi sinh: Ninh Thuận

Lớp: D21TTNT01 Khóa: 2021-2026

Khoa: Viện kĩ thuật – công nghệ

Địa chỉ liên hệ: 424/39/8 tổ 4, khu phố 4, Phường Phú Hoà, tp Thủ Dầu Một, tỉnh Bình Dương.

Điện thoại: 0338001365 Email: nghia.nhn662003@gmail.com

II. QUÁ TRÌNH HỌC TẬP (kê khai thành tích của sinh viên từ năm thứ 1 đến năm đang học):

**** Năm thứ 1:***

Ngành học: Trí tuệ nhân tạo và khoa học dữ liệu Viện: Kỹ thuật – công nghệ

Kết quả xếp loại học tập: khá

Sơ lược thành tích:

Ngày 17 tháng 04 năm 2023

Xác nhận của lãnh đạo viện
(ký, họ và tên)

Sinh viên chịu trách nhiệm chính thực hiện đề tài
(ký, họ và tên)

Nguyễn Hữu Nghĩa

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

TỔNG QUAN ĐỀ TÀI

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.

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.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Sinh viên trường đại học Thủ Dầu Một.

Phạm vi nghiên cứu:

- Trường đại học Thủ Dầu Một.

4. Cơ cấu nghiên cứu

Bài nghiên cứu gồm 3 chương:

Chương 1: Các nghiên cứu liên quan.

Chương 2: Quy trình và các thuật toán sử dụng

Chương 3: Xây dựng và đánh giá độ chính xác của từng mô hình.

5. Phương pháp nghiên cứu

Thu thập dữ liệu về điểm số trong 4 năm học của sinh viên trường đại học Thủ Dầu Một từ khoá D16 đến D18. Sau đó áp dụng các thuật toán Machine Learning như Decision Tree, Logistic Regression và Suport Vector Machine để dự đoán và đưa ra kết quả là một sinh viên nào đó có khả năng nghỉ học hay không.

6. Các nghiên cứu liên quan

Nghiên cứu trong nước

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 Gaus 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%

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.

Nghiên cứu ngoài nước

Nghiên cứu của Đại học Sejong Hàn Quốc, 2021

Đề tài thực hiện: **Dự đoán bỏ học sớm tron ghọc 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%

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.

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 1. CƠ SỞ LÝ THUYẾT

1.1. Các thư viện được sử dụng

1.1.1. 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

1.1.2. 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.

1.1.3. 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 .

1.1.4. 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.

1.2. Các thuật toán sử dụng

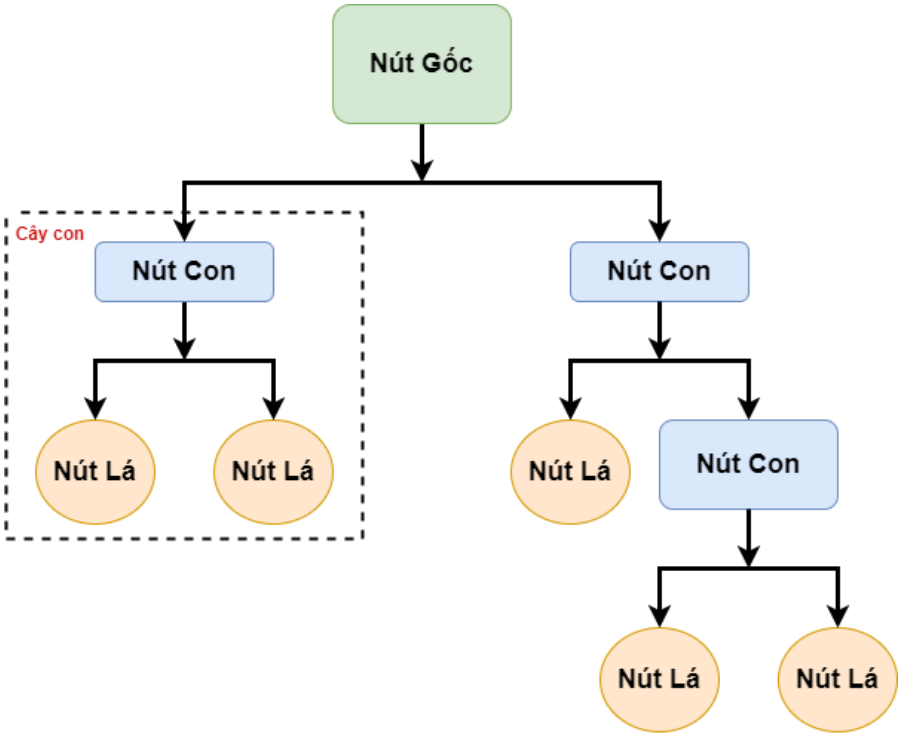
1.2.1. Thuật toán Cây Quyết Định (Decision Tree)

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 1: Minh hoạ về cây quyết định.

Ví dụ ở **Hình 1** 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.

1.2.2. Thuật toán Hồi Quy Logistic (Logistic Regression)

- **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.

- **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.

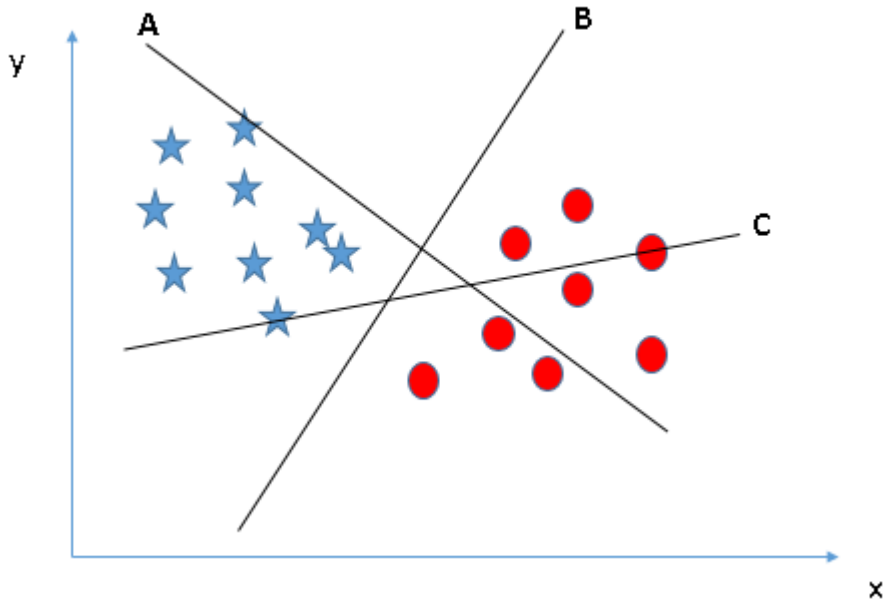
Khả năng hiển thị

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

1.2.3. Thuật toán Suport Vector Machine (SVM)

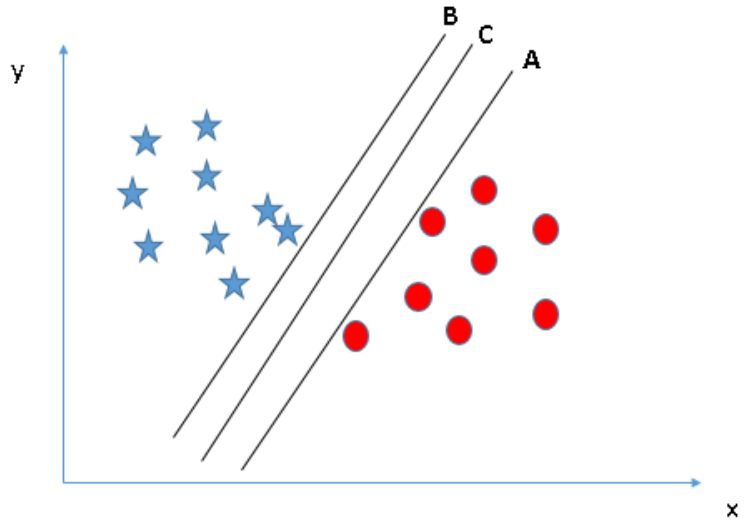
SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần 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 2: 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.

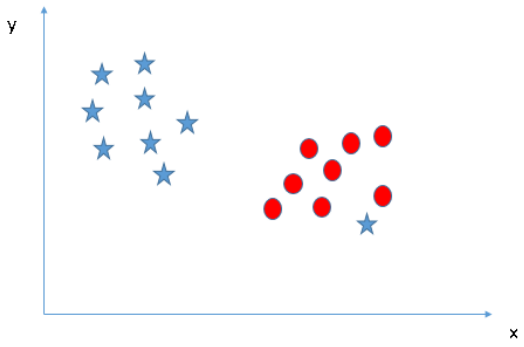


Hình 3: Quy tắc 2.

- 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.

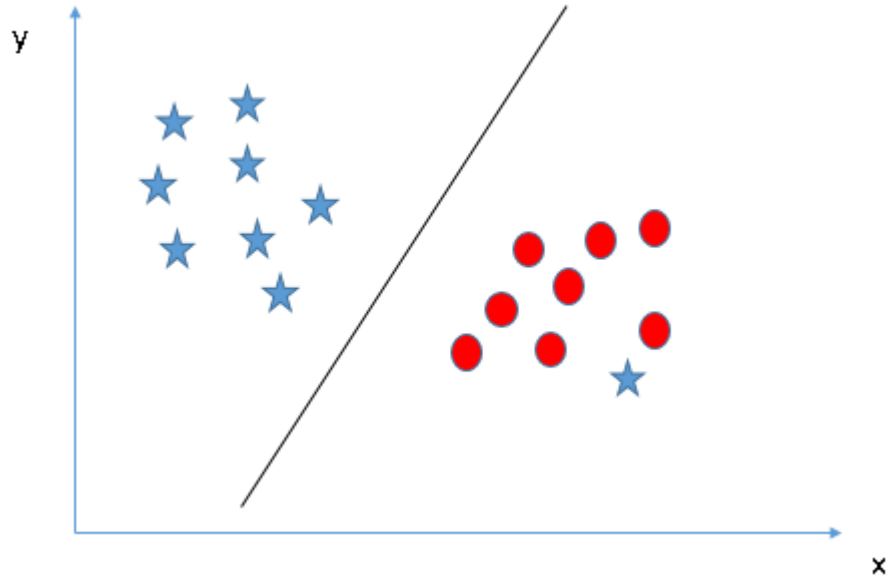
* Lưu Ý: Nếu chọn nhầm hyper-plane có Margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu.

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.



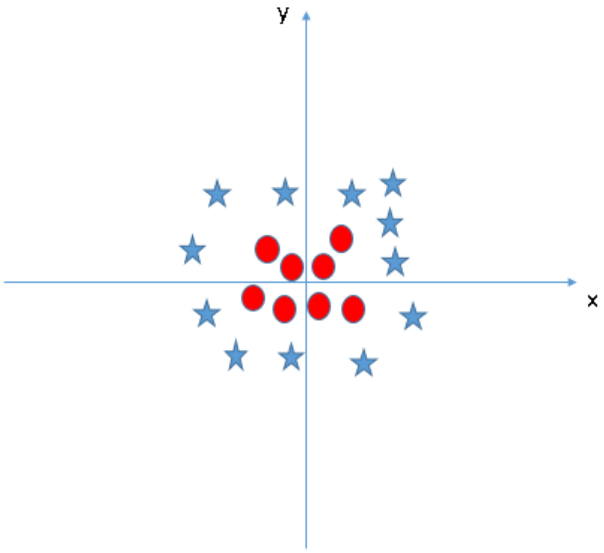
Hình 4: Trường hợp ngoại lệ 1.

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.



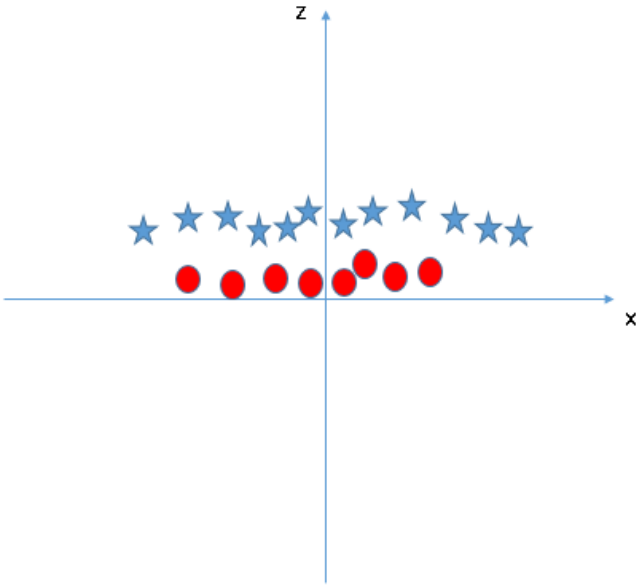
Hình 5: Đáp án của trường hợp ngoại lệ 1.

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 6: 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.

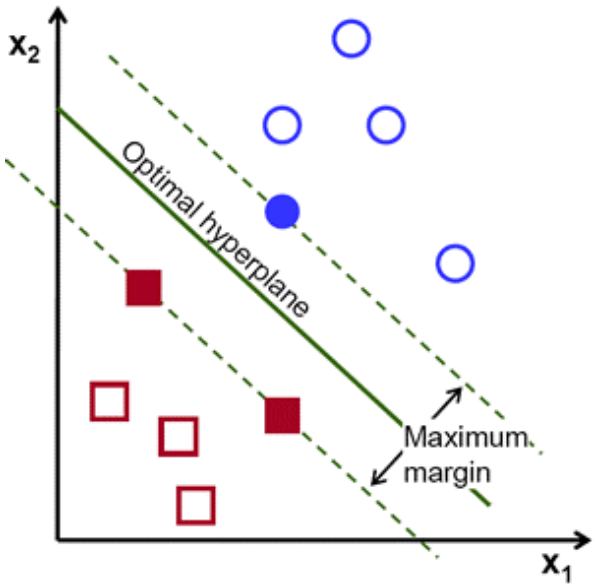


Hình 7: Đáp án của trường hợp ngoại lệ 2.

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.

- **Margin trong Supor Vector Machine (SVM)?**



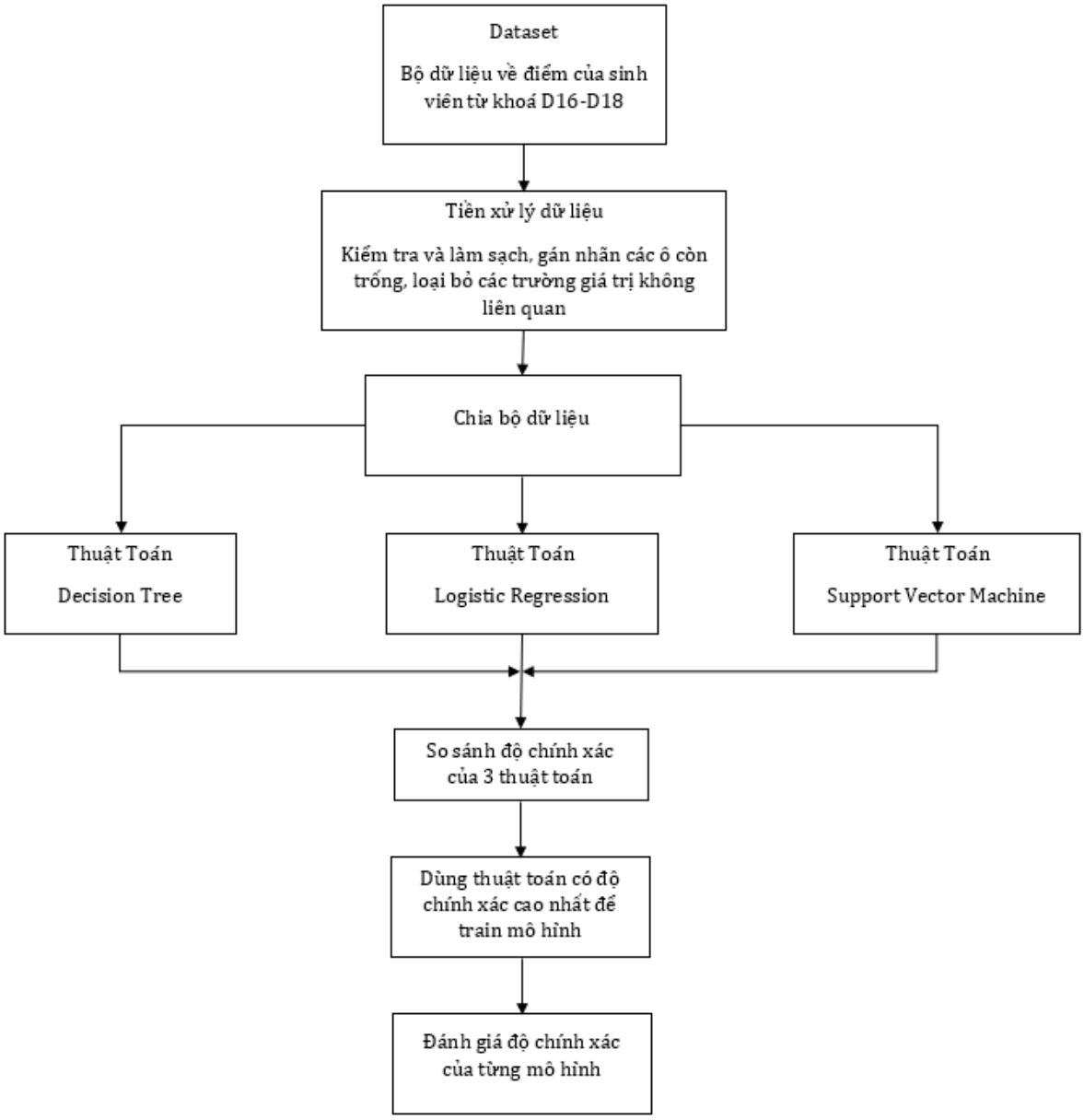
Hình 8: Margin trong SVM.

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 2. XÂY DỰNG VÀ ĐÁNH GIÁ ĐỘ CHÍNH XÁC CỦA TỪNG MÔ HÌNH

2.1. Xây dựng mô hình học máy

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



Hình 9: 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 ở trường đại học Thủ Dầu Một.

Bước 2: Tiền xử lý dữ liệu bao gồm việc: kiểm tra và làm sạch, gán nhãn các ô còn trống, loại bỏ các trường không liên quan,…

Bước 3: Dùng "train_test_split()" của thư viện Sklearn để chia bộ dữ liệu với "validation_size = 0.20" tương đương với 80% bộ dữ liệu là dung để train, và 20% còn lại sẽ dùng để test.

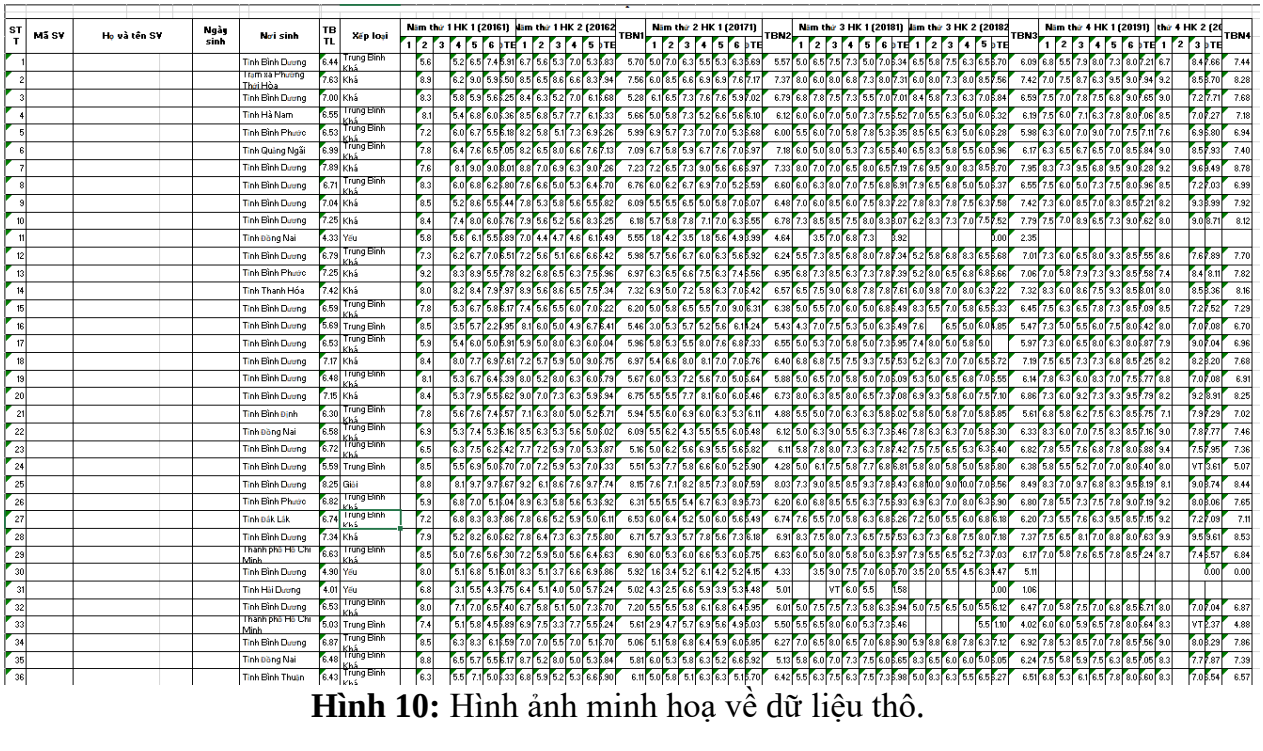
Bước 4: Áp dụng 3 thuật toán trên là Decision Tree, Logistic Regression và Suport Vector Machine và sử dụng phương pháp "cross_val_score()" trong thư viện Sklearn để đánh giá và so sánh độ chính xác của 3 thuật toán trên.

Bước 5: Dùng thuật toán có độ chính xác cao nhất để train model.

Bước 6: Dùng Accuracy_score() để xem và đánh giá độ chính xác của 4 model sau khi train.

2.2. Giới thiệu về tập dữ liệu

Dataset được thu thập tại trường đại học Thủ Dầu Một gồm nhiều file điểm của sinh viên từ khoá D16-D18 được lưu lại, hay còn gọi là "dữ liệu thô".



Hình 10: Hình ảnh minh hoạ về dữ liệu thô.

2.3. Tiền sử lý dữ liệu

2.3.1. Các bước tiền xử lý dữ liệu

Bước 1: Xoá bỏ các trường không liên quan, không cần thiết, không có tác dụng trong việc train mô hình.

Bước 2: Gộp tất cả các file dữ liệu thô thành một file duy nhất.

Bước 3: Xử dụng Microsoft Excel để xử lý file dataset

- Dùng hàm “Replace()” để thay thế các ô không có điểm hoặc “VT” hoặc “CT” thành điểm “0”.

- Thêm một trường “KQ” chỉ chứa “0” hoặc “1”, nếu là “0” thì là “Có khả năng nghỉ học”, ngược lại nếu là “1” thì sẽ là “Không có khả năng nghỉ học”. Trường “KQ” sẽ được điền dựa theo công thức: Nếu trường “TBN4” bé hơn hoặc bằng 5 và có ít hơn hoặc bằng 3 trường chứa điểm 0 trong tất cả các trường từ năm 1 tới năm 4 thì trường “KQ” được điền “1”, ngược lại điền “0”.

Mon1	Mon2	Mon3	Mon4	Mon5	Mon6	Mon1	Mon2	Mon3	Mon4	Mon5	KQ
0	5.6	0	5.2	6.5	7.4	6.7	5.6	5.3	7.0	5.3	1
0	8.9	0	6.2	9.0	5.9	8.5	6.5	8.6	6.6	8.3	1
0	8.3	0	5.8	5.9	5.6	8.4	6.3	5.2	7.0	6.1	1
0	8.1	0	5.4	6.8	6.0	8.5	6.8	5.7	7.7	6.1	1
…	…	…	…	…	…	…	…	…	…	…	…
6.5	5.8	0	7.3	6.8	5.4	7.8	6.0	7.3	5.1	7.0	1
5.8	7.0	0	7.3	7.3	6.0	8.5	6.3	6.3	5.7	7.5	1
6.0	5.8	0	5.0	5.0	5.3	0	5.8	0	0	7.5	0
5.5	6.8	0	3.8	5.5	5.4	6.0	5.3	8.0	5.3	7.3	0

Bảng 3: Dữ liệu sau khi qua 4 bước tiền xử lý.

Bước 5: Chia file sau khi điền các ô không có điểm hoặc “VT” hoặc “CT” thành điểm 0 và thêm trường “KQ” thành 5 sheet:

- Sheet 1: tên “Data” là sheet chứa toàn bộ điểm của sinh viên từ năm 1 đến năm 4 và trường “KQ”

- Sheet 2: tên “Nam1” là sheet chỉ chứa điểm của sinh viên trong năm đầu tiên và trường “KQ”, ngoại trừ trường “ĐTB” và “TBN1”.

- Sheet 3: tên “Nam2” là sheet chứa điểm của sinh viên từ năm 1 đến năm 2 và trường “KQ”, ngoại trừ các trường “ĐTB”, “TBN1” và “TBN2”.

- Sheet 4: tên “Nam3” là sheet chứa điểm của sinh viên từ năm 1 đến năm 3 và trường “KQ”, ngoại trừ các trường “ĐTB”, “TBN1”, “TBN2” và “TBN3”.

- Sheet 5: tên "Nam4" là sheet chứa điểm của sinh viên từ năm 1 đến năm 4 và trường "KQ", ngoại trừ các trường "ĐTB", "TBN1", "TBN2", "TBN3" và "TBN4".

2.4. Thực hiện phân chia bộ dữ liệu

Với bộ dữ liệu được chia thành 4 sheet khác nhau, trừ sheet "Data", nhóm sẽ có 4 bộ dữ liệu từ các sheet Nam1 đến Nam4 để train ra 4 mô hình tương ứng từng năm, từ năm 1 – năm 4.

Tiến hành phân chia bộ dữ liệu bằng hàm "train_test_split()" của thư viện Sklearn với tỷ lệ 8/2 tương ứng với 80% bộ dữ liệu dùng để train và 20% còn lại dùng để test.

```
# 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_1, X_validation_1, Y_train_1, Y_validation_1= model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)
```

Bảng 4: Câu lệnh phân chia dữ liệu sheet "Nam1".

```
#Output
X_train_1 =
[[7.2  5.7  0.   ... 6.5  8.2  6.  ]
 [6.3  6.3  0.   ... 6.   6.3  6.3 ]
 [7.3  5.4  7.7  ... 5.   7.5  6.8 ]
 ...
 [5.8  7.   0.   ... 5.5  8.5  7.7 ]
 [7.   8.   5.   ... 7.8  6.18 7.2 ]
 [0.   5.9  0.   ... 5.8  5.6  5.3 ]]
X_validation_1 =
[[5.   7.   5.   3.8  5.8  7.5  6.6  6.5  6.3  5.55 6.9 ]
 [5.8  5.8  0.   6.3  8.5  6.3  8.   5.1  5.5  5.5  6.9 ]
 [7.4  8.   8.   6.5  6.   7.   4.7  9.   5.3  8.6  7.8 ]
 [0.   7.3  0.   6.2  6.7  7.   7.2  5.6  5.1  6.6  6.6 ]
 [6.4  6.   7.3  6.8  6.   9.   7.3  6.8  5.3  6.   6.8 ]
 [6.8  6.5  0.   7.5  8.   5.   7.   5.8  7.   5.4  6.8 ]
…
 [5.   6.8  5.5  5.1  6.1  5.   7.6  5.3  7.   5.91 7.8 ]
 [7.8  6.   0.   9.5  7.5  7.5  6.5  7.5  6.8  7.5  7.  ]
 [6.   7.3  0.   5.5  7.8  7.   5.5  6.   6.1  5.8  5.4 ]
 [6.5  6.8  0.   8.   6.8  5.5  7.3  5.3  4.5  5.2  7.3 ]
 [6.5  5.3  0.   8.   8.   3.8  0.   5.   4.5  5.5  5.14]]
```

```
Y_train _1=

[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1
1 0 0 1 1 0 1 1 1 1]


Y_validation_1 =
[0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1
1 1 1 0]
```

Bảng 5: Output sau khi phân chia bộ dữ liệu từ sheet “Nam1”.

```
# 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_2, X_validation_2, Y_train_2, Y_validation_2= model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)
```

Bảng 6: Câu lệnh phân chia dữ liệu sheet “Nam2".

#Output

X_train _2=
[[7.2 5.7 0. ... 8.2 6. 5.]
[6.3 6.3 0. ... 5.8 8.3 8.3]
[7.3 5.4 7.7 ... 7.5 6.8 5.1]
...
[5.8 7. 0. ... 6.8 6.4 8.7]
[7. 8. 5. ... 5.3 7. 7.8]
[0. 5.9 0. ... 6.7 6.3 8.9]]

X_validation_2 =
[[5. 7. 5. 3.8 5.8 7.5 6.6 6.5 6.3 5.55 6.9 6.5
5.5 6.8 5. 0. 6.3]
[5.8 5.8 0. 6.3 8.5 6.3 8. 5.1 5.5 5.5 6.9 6.
8.4 5.7 7. 7.7 6.5]
[7.4 8. 8. 6.5 6. 7. 4.7 9. 5.3 8.6 7.8 4.7
9. 5.3 8.6 7.8 7.]
[0. 7.3 0. 6.2 6.7 7. 7.2 5.6 5.1 6.6 6.6 5.7
5.6 6.7 6. 6.3 5.6]
[6.4 6. 7.3 6.8 6. 9. 7.3 6.8 5.3 6. 6.8 7.3
6.8 5.3 6. 6.8 9.]
[6.8 6.5 0. 7.5 8. 5. 7. 5.8 7. 5.4 6.8 7.5
6.5 7. 6.8 5.4 6.8]
…
[5. 6.8 5.5 5.1 6.1 5. 7.6 5.3 7. 5.91 7.8 7.6
5.5 5. 6. 5.3 7.]
[7.8 6. 0. 9.5 7.5 7.5 6.5 7.5 6.8 7.5 7. 0.
6. 6. 5.8 6.1 5.5]
[6. 7.3 0. 5.5 7.8 7. 5.5 6. 6.1 5.8 5.4 7.5
8.6 6.3 7.3 6.5 6.]
[6.5 6.8 0. 8. 6.8 5.5 7.3 5.3 4.5 5.2 7.3 6.5
7.3 6. 5. 5.2 7.3]
[6.5 5.3 0. 8. 8. 3.8 0. 5. 4.5 5.5 5.14 0. 0.
0. 0. 0. 0.]]

Y_train _2=
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
1 0 0 1 1 0 1 1 1 1]

Y_validation _2=
[0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1
1 1 1 0]

Bảng 7: Output sau khi phân chia bộ dữ liệu từ sheet “Nam2”.

```
# 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_3, X_validation_3, Y_train_3, Y_validation_3= model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)
```

Bảng 8: Câu lệnh phân chia dữ liệu sheet “Nam3"**.**

```
#Output

X_train_3=
[[7.2 5.7 0.  ... 2.5 6.  6.2]
 [6.3 6.3 0.  ... 8.  6.6 6.9]
 [7.3 5.4 7.7 ... 7.9 5.1 6.2]
...
 [5.8 7.  0.  ... 8.  8.3 8. ]
 [7.  8.  5.  ... 7.  7.3 8. ]
 [0.  5.9 0.  ... 7.  8.  6.3]]


X_validation_3 =
[[ 5.   7.   5.   3.8  5.8  7.5  6.6  6.5  6.3  5.55
  6.9  6.5  6.8  5.   0.   6.3  5.3  5.8  6.   4.2  0.
  6.   0. 5.2  7.   4.2  0.  ]
 [ 5.8  5.8  0.   6.3  8.5  6.3  8.   5.1  5.5  5.5
  6.9  6.  8.4  5.7  7.   7.7  6.5  7.5  6.5  8.5  7.4
  8.   8.9  9.   9.   8.5  9.5  8.3 ]
[ 7.4  8.   8.   6.5  6.   7.   4.7  9.   5.3  8.6  7.8
  4.7  9.   5.3  8.6  7.8  7.   6.5  6.3  7.8 10.   7.
 10.   9.5  7.5  6.   7.3  8.3 ]
…
 [ 6.   7.3  0.   5.5  7.8  7.   5.5  6.   6.1  5.8  5.4
  7.5  8.6  6.3  7.3  6.5  6.   5.   8.8  7.   5.5  5.3
  6.   6.   8.6  8.   7.6  8.3 ]
  [ 6.5  6.8  0. 8.  6.8  5.5  7.3  5.3  4.5  5.2  7.3
  6.5  7.3  6.   5.   5.2  7.3  7.   5.8  0.   5.   5.
  6.   6.2  5.5  6.8  0.   6.5 ]
 [ 6.5  5.3  0.   8.   8.   3.8  0.   5.   4.5  5.5
  5.14 0. 0.   0.   0.   0.   0.   0.   0.   0.   0.
  0.   0. 0.   0.   0.   0.  ]]
```

```
Y_train_3 =
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1
1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
 1 0 0 1 1 0 1 1 1 1]


Y_validation_3 =
[0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1
1 1 1 0]
```

Bảng 9: Output sau khi phân chia bộ dữ liệu từ sheet “Nam3”.

```
# 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_4, X_validation_4, Y_train_4, Y_validation_4= model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)
```

Bảng 10: Câu lệnh phân chia dữ liệu sheet "Nam4"**.**

```
#Output

X_train_4 =
[[7.2  5.7  0.   ... 8.3  8.5  7.57]
 [6.3  6.3  0.   ... 8.4  8.   8.  ]
 [7.3  5.4  7.7  ... 6.4  7.7  6.59]
...
 [5.8  7.   0.   ... 8.1  5.3  7.5 ]
 [7.   8.   5.   ... 7.8  8.8  8.8 ]
 [0.   5.9  0.   ... 9.2  0.   8.  ]]

 X_validation_4 =
 [[5.   7.   5.   ... 6.3  8.3  7.3 ]
 [5.8  5.8  0.   ... 7.4  8.   8.9 ]
 [7.4  8.   8.   ... 9.   8.5  8.46]
  ...
 [6.   7.3  0.   ... 5.5  5.3  6.  ]
 [6.5  6.8  0.   ... 5.5  6.84 7.3 ]
 [6.5  5.3  0.   ... 0.   0.   0.  ]]
```

```
Y_train_4 =
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1
 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0
 1 1 1 1 1 0 0 1 1 0 1 1 1 1]


Y_validation_4 =
[0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1
1 1 1 0]
```

Bảng 11: Output sau khi phân chia bộ dữ liệu từ sheet "Nam4".

CHƯƠNG 3. XÂY DỰNG MÔ HÌNH VÀ THỰC NGHIỆM

3.1. 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.

3.2. Bộ dữ liệu đưa vào thực nghiệm

Sau khi dùng hàm "train_test_split()" cho ra được 4 bộ dữ liệu train và test. 2 bộ dữ liệu này sẽ được đưa vào quá trình áp dụng thuật toán và so sánh độ chính xác của 3 thuật toán đó trên từng bộ dữ liệu.

3.3. Áp dụng 3 thuật toán và so sánh độ chính xác

Dùng "cross_val_score()" trong thư viện Sklearn để so sánh độ chính xác của 3 thuật toán trên.

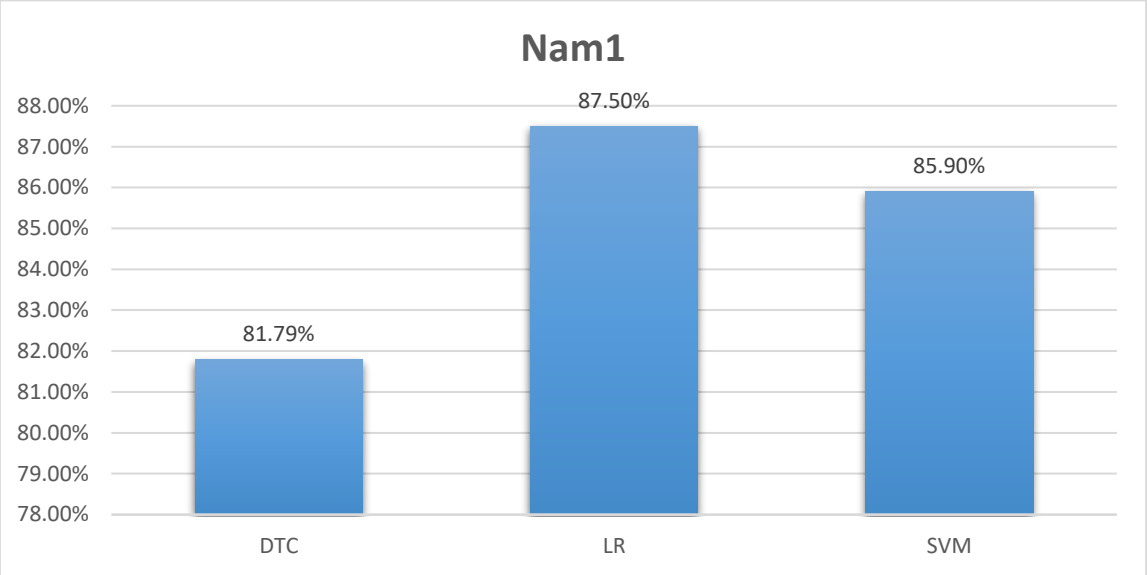
```
models = []
models.append(('DTC', DecisionTreeClassifier()))
models.append(('LR', LogisticRegression()))
models.append(('SVM', SVC()))
results = []
names = []
for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state=7)
    cv_results = model_selection.cross_val_score(model, 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)
```

Bảng 12: Câu lệnh sử cross_val_score() để so sánh độ chính xác của 3 thuật toán.

```
DTC: 0.817949 (0.050572)
LR: 0.875000 (0.107044)
```

SVM: 0.858974 (0.084216)

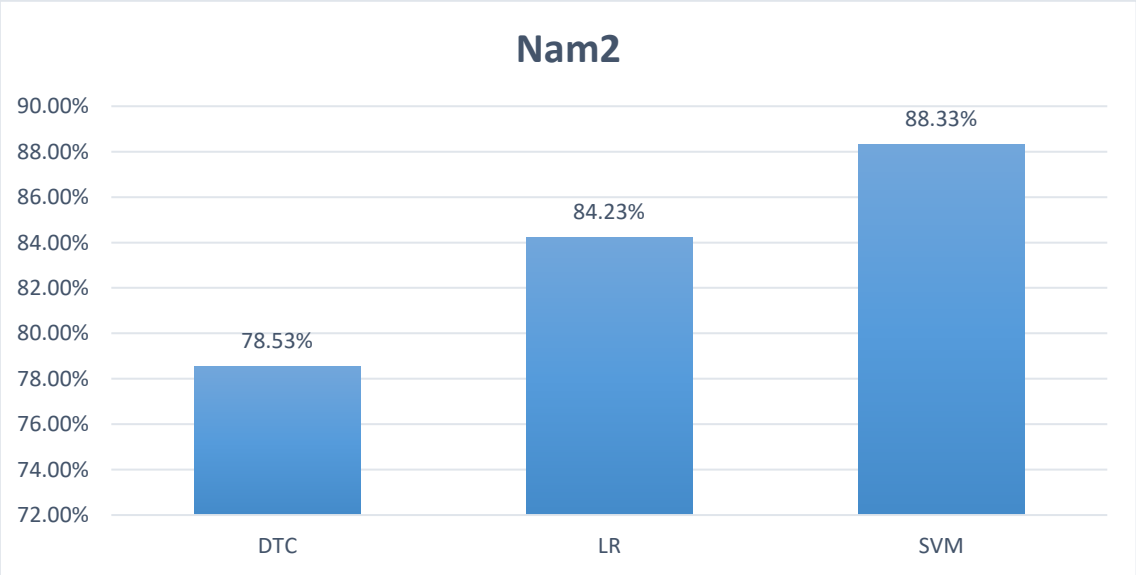
Bảng 13: Output độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam1".



Hình 11: Output độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam1".

DTC: 0.785256 (0.054412)
LR: 0.842308 (0.087575)
SVM: 0.883333 (0.100000)

Bảng 14: Output độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam2".



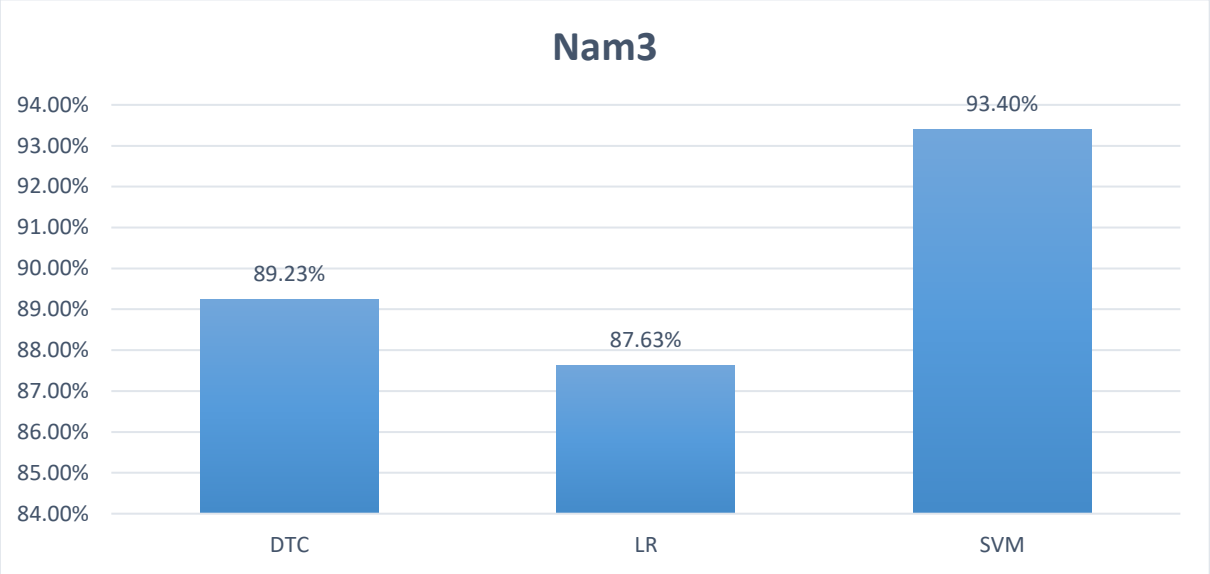
Hình 12: Độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam2".

DTC: 0.892308 (0.065359)

LR: 0.876282 (0.076228)

SVM: 0.933974 (0.062219)

Bảng 15: Output độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam3".



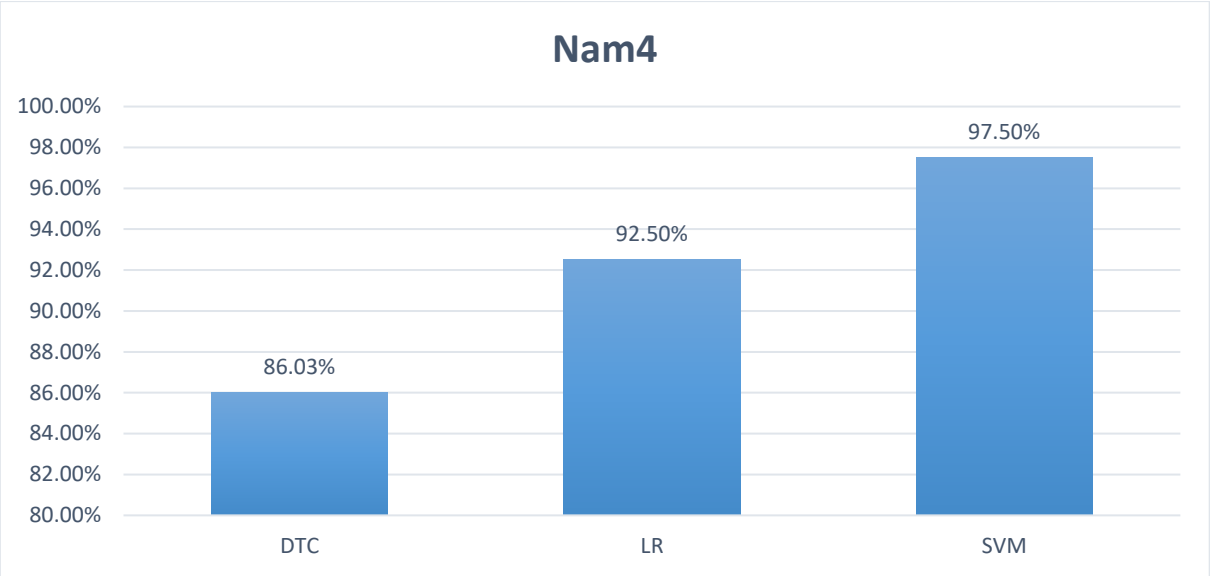
Hình 13: Độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam3".

DTC: 0.860256 (0.103909)

LR: 0.925000 (0.058333)

SVM: 0.975000 (0.038188)

Bảng 16: Output độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam4".



Hình 14: Độ chính xác của 3 thuật toán từ bộ dữ liệu "Nam4".

Sau áp dụng 3 thuật toán vào 4 bộ dữ liệu nhóm có được, có thể thấy thuật toán Suport Vector Machine có độ chính xác cao nhất lên đến 97, 5%

3.4. Thực hiện mô hình

Nhóm sử dụng thuật toán có độ chính xác cao nhất đó là Suport Machine Vector (SVM) để train mô hình. Với 4 bộ dữ liệu thì nhóm sẽ có thể train ra 4 mô hình tương ứng từ Nam1 đến Nam4.

```
import numpy as np
import pickle
model = SVC().fit(X_train_1, Y_train_1)
#save model
pickle.dump(model, open('nam1.pkl', 'wb'))
```

Bảng 17: Câu lệnh train và lưu lại mô hình "nam1.pkl".

```
import numpy as np
import pickle
model = SVC().fit(X_train_2, Y_train_2)
#save model
pickle.dump(model, open('nam1.pkl', 'wb'))
```

Bảng 18: Câu lệnh train và lưu lại mô hình "nam2.pkl".

```
import numpy as np
```

```
import pickle
model = SVC().fit(X_train_3, Y_train_3)
#save model
pickle.dump(model, open('nam1.pkl', 'wb'))
```

Bảng 19: Câu lệnh train và lưu lại mô hình "nam3.pkl".

```
import numpy as np
import pickle
model = SVC().fit(X_train_4, Y_train_4)
#save model
pickle.dump(model, open('nam1.pkl', 'wb'))
```

Bảng 20: Câu lệnh train và lưu lại mô hình "nam4.pkl".

3.5. Đánh giá độ chính xác của từng mô hình

Nhóm sử dụng hàm "accuracy_score()" để đánh giá độ chính xác của từng mô hình.

```
# used_accuracy_score()
import pickle
model = pickle.load(open('nam1.pkl', 'rb'))
accuracy_score = model.score(X_validation_1, Y_validation_1)
print("accuracy_score = ", accuracy_score)
```

```
#output
accuracy_score =  0.8709677419354839
```

Bảng 21: Câu lệnh và output độ chính xác của mô hình "nam1.pkl" sau khi train.

```
# used_accuracy_score()
import pickle
model = pickle.load(open('nam2.pkl', 'rb'))
accuracy_score = model.score(X_validation_2, Y_validation_2)
print("accuracy_score = ", accuracy_score)
```

```
#output
accuracy_score =  0.8709677419354839
```

Bảng 22: Câu lệnh và output độ chính xác của mô hình "nam2.pkl" sau khi train.

```
# used_accuracy_score()
import pickle
model = pickle.load(open('nam3.pkl', 'rb'))
accuracy_score = model.score(X_validation_3, Y_validation_3)
print("accuracy_score = ", accuracy_score)
```

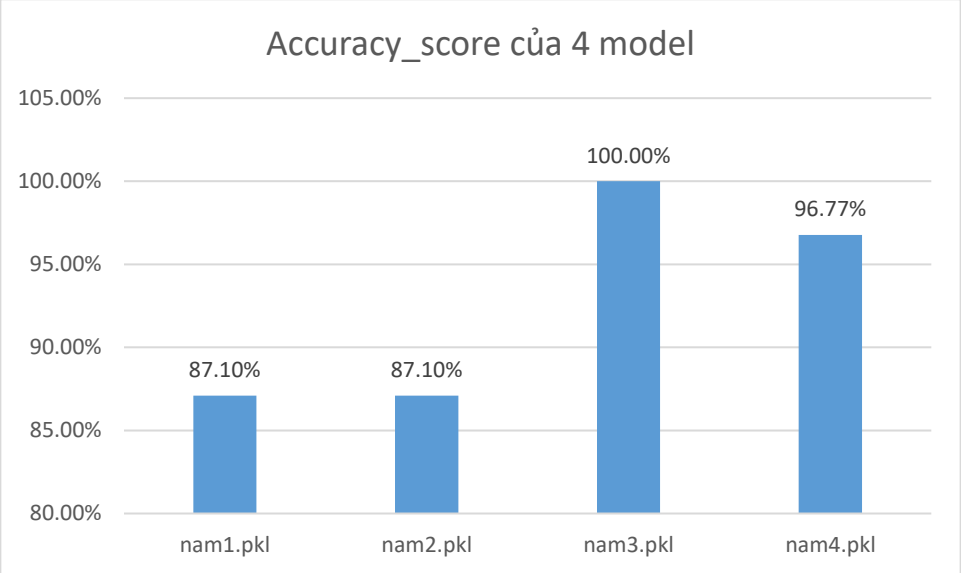
```
#output
accuracy_score =  1.0
```

Bảng 23: Câu lệnh và output độ chính xác của mô hình "Nam3.pkl" sau khi train.

```
# used_accuracy_score()
import pickle
model = pickle.load(open('nam4.pkl', 'rb'))
accuracy_score = model.score(X_validation_4, Y_validation_4)
print("accuracy_score = ", accuracy_score)
```

```
#output
accuracy_score =  0.967741935483871
```

Bảng 24: Câu lệnh và output độ chính xác của mô hình "Nam4.pkl" sau khi train.



Hình 15: Accuracy_score của 4 model sau khi train.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Kết quả đạt được

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.

Hạn chế

Trong 4 mô hình được train, mô hình "nam3.pkl" cho ra kết quả accuracy cao nhất. Và mô hình "nam1.pkl", "nam2.pkl" lại cho ra kết quả accuracy giống nhau. Điều này cho thấy, dữ liệu nhóm thu thập và qua các bước tiền xử lý vẫn chưa được chính xác mà tối ưu nhất có thể.

2. Hướng phát triển đề tài 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]. M. Lorch, S.J. Teach, Facial nerve palsy: etiology and approach to diagnosis and treatment, Pediatr. Emerg. Care. 26 (2010) 763–769.

[2]. Strapasson RAP, Herrera LM, Melani RFH. Forensic Facial Reconstruction: Relationship Between the Alar Cartilage and Piriform Aperture. J Forensic Sci. 2017 Nov;62(6):1460-1465. doi: 10.1111/1556-4029.13494.

[3]. Claes P, Vandermeulen D, De Greef S, Willems G, Suetens P. Statistically Deformable Face Models for Cranio-Facial Reconstruction. Journal of Computing and Information Technology. 2006; 14:21–30

[4]. Châu, Ma, Nguyễn Đình Tư, & Đinh Quang Huy. " Tái tạo khuôn mặt 3 chiều từ hộp sọ." VNU Journal of Science: Natural Sciences and Technology [Online], 27.4 (2011)

[5]. L. Văn-Chung, T. Hiệp-Hòa, L.-K. Triều-Hưng, N. Minh-Đức, N. Lương-Thọ, Ứng dụng công nghệ 3-D thực tại ảo mô phỏng cơ thể người trong hoạt động giảng dạy, học tập và nghiên cứu, (2017)

[6]. Jarabo A, Masia B, Marco J, Gutierrez D. Recent advances in transient imaging: A computer graphics and vision perspective. Visual Informatics 1(1): 65-79, 2017.

[7]. Borji A. Negative results in computer vision: A perspective. Image and Vision Computing 69:1-8, 2018.

[8]. Heimberger M, Horgan J, Hughes C, McDonald J, Yogamani S. Computer vision in automated parking systems: Design, implementation and challenges. Image and Vision Computing 68: 88-101, 2017.

[9]. Zein Y, Darwiche M, Mokhiamar O. GPS tracking system for autonomous vehicles. Alexandria Engineering Journal 57(4):3127-3137, 2018

[10]. Cohen SA, Hopkins D. Autonomous vehicles and the future of urban tourism. Annals of Tourism Research 74:33-42, 2019.

[11]. Egger J, Wallner J, Gall M, Chen X, Schwenzer-Zimmerer K, Reinbacher K, Schmalstieg D. Computer-aided position planning of miniplates to treat facial bone defects. PLoS One. 2017 Aug 17;12(8):e0182839. doi: 10.1371/journal.pone.0182839. eCollection 2017.

[12]. Chen X, Xu L, Li X, Egger J. Computer-aided implant design for the restoration of cranial defects. Sci Rep. 2017 Jun 23;7(1):4199. doi: 10.1038/s41598-017-04454-6.

[13]. Bedaka AK, Mahmoud AM, Lee SC, Lin CY. Autonomous Robot-Guided Inspection System Based on Offline Programming and RGB-D Model. Sensors (Basel). 2018 Nov 16;18(11). pii: E4008. doi: 10.3390/s18114008.

[14]. Khosravan N, Celik H, Turkbey B, Jones EC, Wood B, Bagci U. A collaborative computer aided diagnosis (C-CAD) system with eye-tracking, sparse attentional model, and deep learning. Med Image Anal. 2019 Jan;51:101-115. doi: 10.1016/j.media.2018.10.010. Epub 2018 Oct 28

[15]. Yang L, Noguchi N. Human detection for a robot tractor using omni-directional stereo vision. Computers and Electronics in Agriculture 89:116-125, 2012.

[16]. Atrevi DF, Vivet D, Duculty F, Emile B. A very simple framework for 3-D human poses estimation using a single 2D image: Comparison of geometric moments descriptors. Pattern Recognition 71:389-401, 2017.

[17]. Beringer M, Spohn F, Hildebrandt A, Wacker J, Recio G. Reliability and validity of machine vision for the assessment of facial expressions. Cognitive Systems Research 56:119-132, 2019

[18]. Jain DK, Shamsolmoali P, Sehdev P. Extended deep neural network for facial emotion recognition. Pattern Recognition Letters 120:69-74, 2019.

[19]. Lekdioui K, Messoussi R, Ruichek Y, Chaabi Y, Touahni R. Facial decomposition for expression recognition using texture/shape descriptors and SVM classifier. Signal Processing: Image Communication 58 :300-312, 2017.

[20]. Ritto FG, Schmitt ARM, Pimentel T, Canellas JV, Medeiros PJ. Comparison of the accuracy of maxillary position between conventional model surgery and virtual surgical planning. International Journal of Oral and Maxillofacial Surgery 47(2):160-166, 2018.

[21]. Bartella AK, Kamal M, Scholl I, Schiffer S, Steegmann J, Ketelsen D, Hölzle F, Lethaus B. Virtual reality in preoperative imaging in maxillofacial surgery: implementation of “the next level”? British Journal of Oral and Maxillofacial Surgery, In Press, 2019.

[22]. Quero G, Lapergola A, Soler L, Shabaz M, Hostettler A, Collins T, Marescaux J, Mutter D, Diana M, Pessaux P. Virtual and Augmented Reality in Oncologic Liver Surgery. Surgical Oncology Clinics of North America 28(1):31-44, 2019

[23]. Poker A, Pescarini E, Nduka C, Kannan RY. McLaughlin's legacy in the current treatment of facial palsy. Br J Oral Maxillofac Surg. 2019 Jul 23. pii: S0266-4356(19)30274-8. doi: 10.1016/j.bjoms.2019.07.009

[24]. Pereira LM, Obara K, Dias JM, Menacho MO, Lavado EL, Cardoso JR. Facial exercise therapy for facial palsy: systematic review and meta-analysis. Clin Rehabil. 2011 Jul;25(7):649-58. doi: 10.1177/0269215510395634. Epub 2011 Mar 7.

[25]. Babl FE, Mackay M, Dalziel SR. Facial nerve palsy in children. J Paediatr Child Health. 2019 Jul;55(7):878-879. doi: 10.1111/jpc.14500

[26]. Flynn C, Stavness I, Lloyd J, Fels S. A finite element model of the face including an orthotropic skin model under in vivo tension. Comput Methods Biomech Biomed Engin. 2015;18(6):571-82. doi: 10.1080/10255842.2013.820720. Epub 2013 Aug 6.

[27]. Dao TT, Fan AX, Dakpé S, Pouletaut P, Rachik M, Ho Ba Tho MC. Image-based Skeletal Muscle Coordination: Case Study on a Subject Specific Facial Mimic Simulation Journal of Mechanics in Medicine and Biology, 18(2) 1850020, 2018

[28]. Fan AX, Dakpé S, Dao TT, Pouletaut P, Rachik M, Ho Ba Tho MC. MRI-based Finite Element Modeling of Facial Mimics: a Case Study on the Paired Zygomaticus Major Muscles Computer Methods in Biomechanics and Biomedical Engineering 20(9):919-928, 2017.

[29]. Strapasson RAP, Herrera LM, Melani RFH. Forensic Facial Reconstruction: Relationship Between the Alar Cartilage and Piriform Aperture. J Forensic Sci. 2017 Nov;62(6):1460-1465. doi: 10.1111/1556-4029.13494.

[30]. Gietzen T, Brylka R, Achenbach J, Zum Hebel K, Schömer E, Botsch M, Schwanecke U, Schulze R. A method for automatic forensic facial reconstruction based on dense statistics of soft tissue thickness. PLoS One. 2019 Jan 23;14(1):e0210257. doi: 10.1371/journal.pone.0210257. eCollection 2019

[31]. Herrera LM, Strapasson RAP, Zanin AA, da Silva JVL, Melani RFH. Comparison Among Manual Facial Approximations Conducted by Two Methodological Approaches of Face Prediction. J Forensic Sci. 2017 Sep;62(5):1279-1285. doi: 10.1111/1556-4029.13435. Epub 2017 Feb 23

[32]. Starbuck JM, Ghoneima A, Kula K. Facial tissue depths in children with cleft lip and palate. J Forensic Sci. 2015 Mar;60(2):274-84. doi: 10.1111/1556-4029.12645. Epub 2014 Nov 28

[33]. Claes P, Vandermeulen D, De Greef S, Willems G, Suetens P. Statistically Deformable Face Models for Cranio-Facial Reconstruction. Journal of Computing and Information Technology. 2006; 14:21–30

[34]. Wilkinson C. Facial reconstruction—anatomical art or artistic anatomy? Journal of Anatomy. 2010; 216(2):235–50.

[35]. Turner W, Brown R, Kelliher T, Tu P, Taister M, Miller K. A novel method of automated skull registration for forensic facial approximation. Forensic Science International. 2005; 154:149–158

[36]. Rynn C, Wilkinson C, Peters HL. Prediction of nasal morphology from the skull. Forensic Science, Medicine, and Pathology. 2010; 6(1):20–34

[37]. Shui W, Zhou M, Maddock S, He T, Wang X, Deng Q. A PCA-Based Method for Determining Craniofacial Relationship and Sexual Dimorphism of Facial Shapes. Computers in Biology and Medicine. 90:33–49, 2017.

[38]. LeCun Y, Bengio Y, Hinton G. Deep learning, Nature 521 (7553): 436–444, 2015.

[39]. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33(4):917–963, 2019.

[40]. Ma L, Liu Y, Zhang X, Ye Y, Yin G, Johnson BA. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing 152:166-177, 2019

[41]. Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J. A guide to deep learning in healthcare. Nature Medicine 25: 24–29, 2019.

[42]. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, 1998.

[43]. Eickenberg M, Gramfort A, Varoquaux G, Thirion B. Seeing it all: Convolutional network layers map the function of the human visual system. NeuroImage 152:184-194, 2017

[44]. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 9(8):1735–1780, 1997.

[45]. Dao TT. From Deep Learning to Transfer Learning for the Prediction of Skeletal Muscle Forces. Medical & Biological Engineering & Computing 57(5):1049–1058, 2019

[46]. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F, The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository., J. Digit. Imaging. 26 (2013) 1045–57. doi:10.1007/s10278-013-9622-7, 11.3. Danh mục các công trình đã công bố thuộc lĩnh vực của đề tài của chủ nhiệm và những thành viên tham gia nghiên cứu (họ và tên tác giả; bài báo; ấn phẩm; các yếu tố về xuất bản)

[47]. T N Nguyen, V D Tran, H Q Nguyen, D P Nguyen, T T Dao (2021). Enhanced Head-Skull Shape Learning using Statistical Modeling and Topological Features. Medical & Biological Engineering & Computing. In Press.

[48]. Nguyen, T. N., Dakpe, S., Tho, M. C. H. B., & Dao, T. T. (2021). Kinect-driven Patient-specific Head, Skull, and Muscle Network Modelling for Facial Palsy Patients. Computer Methods and Programs in Biomedicine, 200, 105846.

[49]. H Q Nguyen, T-N-T Nguyen, T Q D Pham, V X Tran, V D Nguyen, T T Dao. Crack Propagation in the Tibia Bone within Total Knee Replacement Using the eXtended Finite Element Method. Applied Sciences. 2021; 11(10):4435. https://doi.org/10.3390/app11104435

[50]. Nguyen, T. N., Tran, V. D., Nguyen, H. Q., & Dao, T. T. (2020). A statistical shape modeling approach for predicting subject-specific human skull from head surface. Medical & Biological Engineering & Computing, 58(10), 2355-2373.

[51]. H Q Nguyen, T T Dao, A Rassineux, MC Ho Ba Tho. 2018. Material driven mesh of the lumbar spine derived from CT data. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, Vol.6, No.2, pp. 128-136.

[52]. TN. Nguyen, VD. Tran, HQ. Nguyen, and TT. Dao (2020). Virtual Human Skull Model: Prediction from Head Surface. In: VPH2020 Conference, Paris 26-28 August 2020.

[53]. H Q Nguyen, T T Dao, MC Ho Ba Tho (2019). Multimodal Medical Imaging Fusion For Developping Patient-Specific Lumbar Spine Models In: 25th Congress of the European Society of Biomechanics (ESB 2019), Vienna, Austria, July 7-10, 2019.