ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

BÁO CÁO ĐỒ ÁN
MẠNG XÃ HỘI
ĐỀ TÀI
DỰ ĐOÁN KẾT QUẢ HỌC TẬP CỦA HỌC KỲ TIẾP THEO CỦA SINH VIÊN UIT

GVHD: Nguyễn Thị Anh Thư

Nhóm Sinh viên thực hiện:

1. Nguyễn Ngọc Gia Khiêm	MSSV: 21520287
2. Ngô Thùy Yến Nhi	MSSV: 21521230
3. Mai Quốc Bảo	MSSV: 21521850
4. Võ Thị Bích Ly	MSSV: 21522317
5. Trần Kim Thanh	MSSV: 21522605
6. Ngô Kỳ Anh	MSSV: 21521825
7. Hoàng Xuân Lộc	MSSV: 22520788

TP. Hồ Chí Minh, tháng … năm ….

NHẬN XÉT CỦA GIẢNG VIÊN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu bài toán

Trong bối cảnh giáo dục hiện nay, việc đánh giá năng lực và tiềm năng của sinh viên ngày càng có vai trò quan trọng, không chỉ trong quá trình học tập mà còn trong việc đáp ứng yêu cầu tuyển dụng của doanh nghiệp.

- **Đối với nhà trường:** Dự đoán kết quả học tập giúp nhà trường đánh giá và cải thiện chất lượng chương trình đào tạo. Nhờ đó, trường có thể phát hiện sớm các vấn đề nếu có dấu hiệu sụt giảm học lực của sinh viên, từ đó triển khai các biện pháp nâng cao hiệu quả giảng dạy.

- **Đối với doanh nghiệp:** Thông tin về tiềm năng học tập của sinh viên hỗ trợ nhà tuyển dụng trong việc lựa chọn ứng viên phù hợp, đáp ứng yêu cầu công việc trong tương lai.

- **Đối với cố vấn học tập và sinh viên:** Giúp phát hiện sớm sinh viên có nguy cơ giảm sút thành tích, để đưa ra kế hoạch học tập hợp lý, góp phần tăng cơ hội thành công trong học kỳ tới.

Nhận thức được vai trò quan trọng của bài toán này, nhóm chúng tôi lựa chọn đề tài "Dự đoán điểm trung bình các kỳ học tiếp theo của sinh viên." Mục tiêu của đề tài là phân tích và khai thác dữ liệu nhằm xác định các phương pháp và mô hình đạt hiệu quả cao. Để hiện thực hóa mục tiêu, chúng tôi tiến hành tiền xử lý, phân tích dữ liệu và thực hiện các thí nghiệm toàn diện để đánh giá hiệu quả của cách tiếp cận đã lựa chọn.

1.2. Ngữ cảnh

Trong mỗi học kỳ tại UIT, có từ 9% đến 16% sinh viên có điểm trung bình học kỳ dưới 5, thuộc diện học lực yếu, và tỷ lệ này có xu hướng tăng theo thời gian. Việc kéo dài thời gian học do học lực yếu dẫn đến lãng phí thời gian và tài chính cho sinh viên và nhà trường. Vì vậy, dự đoán sớm học lực của sinh viên (đặc biệt là học lực yếu) trong kỳ tiếp theo là cần thiết để giúp nhà trường và sinh viên đưa ra các biện pháp hỗ trợ kịp thời.

1.3. Phát biểu bài toán

Input Đầu vào của bài toán bao gồm các thông tin về học kỳ, thông tin về sinh

viên và thông tin môn học. Trong mỗi học kỳ, một sinh viên có thể học nhiều môn. Bảng dữ liệu cung cấp d anh sách các môn học mà sinh viên đã đăng ký, điểm số tương ứng trong các học kỳ trước đó, và các đặc điểm của môn học. Đầu ra của bài toán là điểm dự đoán sinh viên sẽ đạt được ở các môn học trong học kỳ tiếp theo.

1.4. Thách thức của bài toán

Xử lý dữ liệu: Dữ liệu về sinh viên, môn học và điểm số có thể không đầy đủ, chứa nhiễu hoặc có các giá trị bất thường do nhiều yếu tố khác nhau, chẳng hạn như sai sót trong nhập liệu, thay đổi chương trình giảng dạy, hoặc do sự khác biệt trong cách đánh giá. Điều này có thể ảnh hưởng tiêu cực đến khả năng của mô hình trong việc nhận diện các mẫu phức tạp và đưa ra dự đoán chính xác. Thách thức nằm ở việc xử lý dữ liệu thiếu sót, cần phải loại bỏ hoặc thay thế những giá trị không hợp lệ mà vẫn đảm bảo giữ lại tính chính xác và tính đại diện của dữ liệu ban đầu.

Sự phụ thuộc vào ngữ cảnh: Điểm số của sinh viên không chỉ đơn thuần phản ánh năng lực cá nhân mà còn bị ảnh hưởng bởi các yếu tố ngữ cảnh khác nhau. Các yếu tố này bao gồm sự phù hợp với từng môn học (sinh viên có thể giỏi ở một số lĩnh vực cụ thể) và yêu cầu đặc thù của từng môn (độ khó, nội dung chuyên sâu, yêu cầu thực hành,...). Việc đánh giá cần được thực hiện một cách toàn diện và khách quan, đòi hỏi phải xem xét từ nhiều góc độ, bao gồm cả đặc điểm của sinh viên và môn học. Do đó, đề tài cần đưa ra cách tiếp cận xử lý sự phụ thuộc này, từ đó xây dựng mô hình dự đoán hiệu quả và chính xác hơn, phản ánh đúng năng lực của sinh viên trong từng ngữ cảnh học tập khác nhau.

1.5. Đối tượng và phạm vi

Đối tượng của đề tài: các sinh viên trong một hệ thống giáo dục hoặc tổ chức giảng dạy cụ thể, với dữ liệu học tập đã thu thập bao gồm thông tin về điểm số của các môn học trong các học kỳ trước. Đề tài tập trung vào nhóm sinh viên có lịch sử học tập đủ chi tiết để mô hình có thể học hỏi từ đó và đưa ra dự đoán.

Phạm vi của đề tài: xây dựng một mô hình dự đoán nhằm ước lượng điểm số của các môn học trong học kỳ tiếp theo dựa trên dữ liệu lịch sử học tập của sinh viên. Phạm vi không chỉ dừng lại ở việc dự đoán mà còn mở rộng đến việc đánh giá độ chính xác của mô hình và phân tích các yếu tố ảnh hưởng đến kết quả dự đoán. Mục đích là

cung cấp những thông tin có giá trị cho nhà trường trong việc hỗ trợ quyết định giáo dục, cũng như giúp sinh viên điều chỉnh quá trình học tập nhằm đạt được kết quả tốt hơn.

Giới hạn của đề tài: Đề tài chỉ tập trung vào việc dự đoán kết quả điểm môn học, do đó sẽ không bao gồm các khía cạnh như quản lý chương trình đào tạo, ràng buộc môn học bắt buộc, tiên quyết, hay sự lựa chọn môn học theo nhóm. Ngoài ra, đề tài cũng không xem xét đến các yếu tố phát sinh, chẳng hạn như sự thay đổi hoặc thêm mới các môn học trong chương trình đào tạo, mà chỉ dựa trên các môn học hiện có trong hệ thống dữ liệu lịch sử của sinh viên.

1.6. Mục tiêu

Nghiên cứu, khảo sát và ứng dụng các kỹ thuật từ mô hình máy học, mô hình học sâu, và hệ thống gợi ý để giải quyết nhiệm vụ dự đoán kết quả học tập của sinh viên trong các môn học tương lai. Các kỹ thuật này được triển khai nhằm nắm bắt mối quan hệ giữa dữ liệu lịch sử và khả năng thành công của sinh viên trong các môn học tiếp theo.

Phân tích ưu và nhược điểm của từng phương pháp và xác định sự kết hợp tối ưu giữa các kỹ thuật được đề xuất. Điều này bao gồm việc đánh giá hiệu quả của các thuật toán, đồng thời so sánh chúng với các phương pháp khác để chọn lọc những phương pháp mang lại hiệu quả dự đoán cao nhất.

Áp dụng trọng số cho các đặc trưng khác nhau nhằm khám phá các đặc điểm học tập của sinh viên, dựa trên kết quả học tập ở các môn học đã qua. Việc này giúp xác định những yếu tố ảnh hưởng mạnh nhất đến kết quả học tập, từ đó tăng cường khả năng dự đoán và hỗ trợ trong quá trình điều chỉnh mô hình để đạt được độ chính xác cao hơn.

CHƯƠNG 2. MÔ HÌNH GIẢI BÀI TOÁN

2.1. Tìm hiểu dữ liệu

Các bảng dữ liệu trong Dataset

2.1.1. File 01.sinhvien.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin chi tiết về các sinh viên tại UIT bao gồm thông tin cá nhân.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên là khóa chính trong bảng. Trong các bảng khác mssv có thể là khóa chính hoặc khóa phụ. Rất quan trọng khi nối các bảng lại với nhau
namsinh	Integer	Năm sinh của sinh viên	Thấp nhất là 1979, cao nhất là 2001
gioitinh	Category	Giới tính của sinh viên	1: Nam, 0: Nữ
noisinh	String	Nơi sinh của sinh viên	Chỉ ghi lại Thành phố/Tỉnh
lopsh	String	Mã lớp sinh hoạt	Các lớp có đuôi .1, .2, .3 có thể được quy lại thành một lớp
khoa	String	Tên khoa của sinh viên	Các giá trị trong cột này trùng với những chữ cái trong Mã lớp sinh hoạt (lopsh). Cần lưu ý nếu dùng sử dụng 2 cột này trong mô hình dự đoán.
hedt	String	Hệ đào tạo của sinh viên	Gồm 5 hệ tất cả: CLC, CNTN, CTTT, CQUI, KSTN
khoahoc	Category	Số thứ tự khóa đào tạo của sinh viên	Số thứ tự khóa đào tạo có thể xác định được năm mà sinh viên vào trường. Với: { "14": "2019", "13": "2020",…}.

			Việc biết được năm sinh viên vào trường rất hữu ích khi xử lý trên các bảng khác
chuyennganh2	String	Mã chuyên ngành của sinh viên.	Giá trị này gần như không có liên quan đến các bảng khác. Nội dung cũng không rõ ràng. Cần xem xét kỹ khi đưa vào mô hình dự đoán
tinhtrang	Category	Tình trạng của sinh viên	
diachi_tinhtp	String	Địa chỉ tỉnh/thành phố hiện tại của sinh viên	Sau khi làm sạch dữ liệu, chỉ giữ lại tỉnh/thành phố hiện tại của sinh viên. (Các giá trị còn lại hoặc quá ích hoặc có thiên hướng gây nhiễu)

2.1.2. File 02.diem.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin chi tiết về thành tích học tập của sinh viên trong các môn học.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không phải là khóa chính, nhưng vẫn quan trọng cho việc định danh sinh viên
ngaythi	DateTime	Ngày thực hiện bài kiểm tra	Thời gian bài kiểm tra được tiến hành, định dạng: YYYY-MM-DD
loaixn	String	Loại bài kiểm tra	Loại bài kiểm tra mà sinh viên thực hiện, ví dụ: TOEIC_LR
Listening	Integer	Điểm phần Listening	Điểm mà sinh viên đạt được trong phần Listening
Speaking	Integer	Điểm phần Speaking	Điểm mà sinh viên đạt được trong phần Speaking

Reading	Integer	Điểm phần Reading	Điểm mà sinh viên đạt được trong phần Reading
Writing	Integer	Điểm phần Writing	Điểm mà sinh viên đạt được trong phần Writing
Total	Float	Tổng điểm	Tổng điểm của sinh viên

2.1.3. File 03.sinhvien_chungchi.xlsx

Tóm tắt: Dữ liệu này cung cấp kết quả cho các bài kiểm tra tiếng Anh của sinh viên. Có thể được dùng để xác định sinh viên đã đặt chuẩn đầu ra hay chưa khi kết hợp với các khóa đào tạo lấy từ bảng sinhvien và chuẩn đầu ra tiếng anh được cung cấp trên trang chủ của UIT.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên là khóa chính và được dùng để xác định từng sinh viên
total	Float	Điểm tổng cộng mà sinh viên đạt được	Điểm mà sinh viên đạt được phụ thuộc vào khả năng của sinh viên và quy định phân loại của từng năm học.
mamh	String	Mã môn học được dùng để xác định trình độ Anh văn của sinh viên	Các mã môn học như "AVSC1", "AVSC2", "ENG01",...đại diện cho trình độ Anh văn của sinh viên. Một số dữ liệu từ cột này được điền khuyết bằng cách: nối với bảng sinhvien để lấy được thông tin số thứ tự khóa đào tạo của sinh viên từ đó xét theo điều kiện trong file ghichu để xếp loại được sinh viên thuộc về lớp nào

2.1.4. File 04.xeploaiav.xlsx

Tóm tắt: Dữ liệu này cung cấp kết quả cho bài kiểm tra tiếng Anh đầu vào của sinh viên. Có thể được dùng để xác định trình độ tiếng anh của sinh viên ngay thời điểm mới nhập học.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên là khóa chính và được dùng để xác định từng sinh viên
total	Float	Điểm tổng cộng mà sinh viên đạt được	Điểm mà sinh viên đạt được phụ thuộc vào khả năng của sinh viên và quy định phân loại của từng năm học.
mamh	String	Mã môn học được dùng để xác định trình độ Anh văn của sinh viên	Các mã môn học như "AVSC1", "AVSC2", "ENG01",...đại diện cho trình độ Anh văn của sinh viên. Một số dữ liệu từ cột này được điền khuyết bằng cách: nối với bảng sinhvien để lấy được thông tin số thứ tự khóa đào tạo của sinh viên từ đó xét theo điều kiện trong file ghichu để xếp loại được sinh viên thuộc về lớp nào

2.1.5. File 05.ThiSinh.xlsx

Tóm tắt: Dữ liệu này cung cấp phương thức và kết quả mà sinh viên dùng để xét tuyển vào trường.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên. Số lượng giá trị duy nhất: 8232
dien_tt	Category	Diện trúng tuyên của sinh viên (Tức phương thức mà sinh viên dùng để xét vào trường)	Phương thức xét tuyển của sinh viên cho biết phương thức mà sinh viên dùng để xét tuyển vào trường. Cột này rất quan trọng khi dùng để chuẩn hóa dữ liệu từ cột diem_tt.
diem_tt	Float	Điểm trúng tuyển của sinh viên	Điểm này có thể đại diện cho điểm tổng kết của sinh viên khi tốt nghiệp THPT hoặc điểm thi đánh giá năng lực của sinh viên.

			Với các sinh viên thuộc diện tuyển thẳng, giá trị này được điền bằng 0

2.1.6. File 06.giayxacnhan.xlsx

Tóm tắt: Dữ liệu này cung cấp các loại giấy xác nhận mà sinh viên nộp cho nhà trường thông qua website, hoặc yêu cầu làm lại thẻ.

Tên cột	Loại	Mô tả	Chi tiết
maloaigiay	Integer	Mã loại giấy tờ sinh viên sử dụng	Xác định loại giấy tờ mà sinh viên yêu cầu, không phải là khóa chính.
dain	Category	Trạng thái in của giấy tờ	Giá trị 1 có thể đại diện cho đã in, 0 là chưa in
baosai	Category	Trạng thái báo sai của giấy tờ	Giá trị 1 có thể đại diện cho giấy tờ bị sai, 0 là không
daphat	Category	Trạng thái phát hành của giấy tờ	Giá trị 1 có thể đại diện cho giấy tờ đã được phát, 0 là chưa
trangthai	Category	Trạng thái của giấy tờ	Giá trị cụ thể của trạng thái cần được xác định thêm
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
lydoxacnhan	String	Lý do xác nhận của sinh viên	Lý do xác nhận đại diện cho lý do
lydocapthe	String	Lý do cấp thẻ	Lý do cấp thẻ có thể liên quan đến việc cấp thẻ sinh viên
hocky	String	Học kỳ	Học kỳ xác định thời điểm cấp thẻ
daky	Category	Trạng thái đã ký	Trạng thái này có thể liên quan đến việc giấy xác nhận đã được ký
dadongdau	Category	Trạng thái đã đóng dấu	Trạng thái này có thể liên quan đến việc đóng dấu trên giấy tờ
ngayphat	String	Ngày phát giấy tờ	Ngày phát có thể đại diện cho thời gian giấy tờ được phát hành

2.1.7. File 08.XLHV.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin về các xử lý học vụ của sinh viên và năm học cũng như học kỳ mà nó được thực hiện.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
tinhtrang	Category	Trạng thái của quyết định xử lý học vụ	Trạng thái này quyết định mức độ của xử lý học vụ mà sinh viên phải chịu. Có khả năng là một đặc trưng quan trọng vì theo quan sát, phần lớn xử lý học vụ có liên quan đến kết quả học tập của sinh viên.
lydo	String	Lý do xử lý học vụ sinh viên	Lý do xác định tình trạng của sinh viên có thể đại diện cho lý do tình trạng của sinh viên
hocky	Integer	Học kỳ	Học kỳ xác định thời điểm cập nhật tình trạng
namhoc	Integer	Năm học	Năm học xác định năm cập nhật tình trạng
soqd	String	Số quy định về xử lý học vụ của sinh viên	Số quy định xác định loại quy định mà xử lý học vụ này dựa trên

2.1.8. File 10.diemrl.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin về số điểm rèn luyện sinh viên tích lũy theo học kỳ của năm học. Dữ liệu này tương tự như dữ liệu ở file diemrl.xlsx nhưng phạm vi dữ liệu hẹp hơn (từ năm 2013 đến năm 2020).

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên là khóa chính và được dùng để xác định từng sinh viên
lopsh	Category	Lớp sinh hoạt của sinh viên	Lớp sinh hoạt của sinh viên là một định danh quan trọng để xác định từng nhóm sinh viên
hocky	Integer	Học kỳ	Học kỳ xác định thời điểm cập nhật điểm rèn luyện

namhoc	Integer	Năm học	Năm học xác định thời điểm cập nhật điểm rèn luyện

2.1.9. File 12.baoluu.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin và trạng thái bảo lưu của sinh viên, cũng như thời gian mà trạng thái bảo lưu được lưu trữ.

Tên cột	Loại	Mô tả	Chi tiết
masv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
tinhtrang	Category	Trạng thái của sinh viên	Trạng thái thể hiện trạng thái của việc bảo lưu
hocky	Integer	Học kỳ	Học kỳ có thể liên quan đến thời điểm cập nhật tình trạng của sinh viên
namhoc	Integer	Năm học	Năm học có thể liên quan đến thời điểm cập nhật tình trạng của sinh viên
soqd	String	Số quyết định	Số quyết định có thể thể hiện quyết định cụ thể liên quan đến tình trạng của sinh viên
ngayqd	datetime	Ngày quyết định	Ngày quyết định có thể thể hiện thời điểm mà quyết định về tình trạng của sinh viên được thực hiện

2.1.10. File 14.totnghiep.xlsx

Tóm tắt: Dữ liệu này cung cấp thời gian và xếp loại tốt nghiệp của sinh viên.

Tên cột	Loại	Mô tả	Chi tiết
id	Integer	Mã định danh duy nhất cho mỗi bản ghi	Không có
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên

xeploai	Category	Xếp loại tốt nghiệp (ví dụ: Khá, Giỏi,...)	Không có
soquyetdinh	String	Số quyết định về kết quả tốt nghiệp	Không có
ngaycapvb	String	Ngày cấp văn bản quyết định về kết quả tốt nghiệp (DD/MM/YYYY)	Không có

2.1.11. File diem_Thu.xlsx

Tóm tắt: Dữ liệu này cung toàn bộ kết quả từ điểm quá trình, thực hành, giữa kỳ , cuối kỳ và số tín chỉ của từng môn học mà sinh viên tham gia. Đồng thời chúng ta cũng có thông tin trạng thái của môn học.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
mamh	String	Mã môn học	Mã môn học, đại diện cho mỗi môn học duy nhất.
malop	String	Mã lớp	Mã lớp, đại diện cho mỗi lớp học duy nhất.
sotc	Integer	Số tín chỉ của môn học	Số tín chỉ cho mỗi môn học, cho trọng số của số điểm học phần của môn học đó. Rất quan trọng trong việc tính toán giá trị cuối cùng. Những môn sotc = 0 là những môn khogn6 tính vào điểm ĐTB cũng như điểm tích lũy
hocky	Integer	Học kỳ	Học kỳ trong năm học, giúp xác định thời điểm một môn học được học.
namhoc	Integer	Năm học	Năm học, cho biết thời điểm mà sinh viên tham gia môn học.
diem_qt	Float	Điểm quá trình	Điểm quá trình, điểm mà sinh viên nhận được trong suốt thời gian học môn học.

diem_th	Float	Điểm thực hành	Điểm thực hành, điểm mà sinh viên nhận được từ các hoạt động thực hành.
diem_gk	Float	Điểm giữa kỳ	Điểm giữa kỳ, điểm mà sinh viên nhận được từ bài kiểm tra giữa kỳ.
diem_ck	Float	Điểm cuối kỳ	Điểm cuối kỳ, điểm mà sinh viên nhận được từ bài thi cuối kỳ.
diem_hp	Float	Điểm học phần	Điểm học phần, điểm tổng hợp mà sinh viên nhận được từ môn học.
trangthai	Integer	Trạng thái của môn học (có thể là đã qua, không qua, học lại, miễn thi,...)	Trạng thái của môn học, 0: hủy; 1: bình thường; 2: trả nợ; 3: cải thiện; 4: Miễn; 5: Hoãn. Để tính điểm TB HK/NH, lọc trangthai = 1,2,3. Để tính ĐTB toàn khóa, chọn trangthai = 1.
tinhtrang	Integer	Tình trạng hiện tại của sinh viên (có thể là đang học, đã tốt nghiệp, bỏ học, nghỉ học,...)	Tình trạng hiện tại của sinh viên, có thể cho biết sinh viên đang học, đã tốt nghiệp, bỏ học, ...

2.1.12. File diemrl.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin về số điểm rèn luyện sinh viên tích lũy theo học kỳ của năm học. Dữ liệu này tương tự như dữ liệu ở file 10.diemrl.xlsx nhưng phạm vi dữ liệu rộng hơn (từ năm 2009 đến năm 2022).

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên là khóa chính và được dùng để xác định từng sinh viên
lopsh	Category	Lớp sinh hoạt của sinh viên	Lớp sinh hoạt của sinh viên là một định danh quan trọng để xác định từng nhóm sinh viên
hocky	Integer	Học kỳ	Học kỳ xác định thời điểm cập nhật điểm rèn luyện

namhoc	Integer	Năm học	Năm học xác định thời điểm cập nhật điểm rèn luyện

2.1.13. File sinhvien_dtb_hocky.xlsx

Tóm tắt: Dữ liệu này ghi lại kết quả học tập (điểm trung bình) theo học kỳ và năm học của sinh viên cũng như số tín chỉ mà sinh viên tích lũy được trong học kỳ đó.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
hocky	Integer	Học kỳ	Học kỳ trong năm học, giúp xác định thời điểm học kỳ
namhoc	Integer	Năm học	Năm học, cho biết thời điểm mà sinh viên tham gia học kỳ
dtbhk	Float	Điểm trung bình học kỳ	Điểm trung bình của sinh viên trong học kỳ
sotchk	Integer	Số tín chỉ học kỳ	Số tín chỉ mà sinh viên đã hoàn thành trong học kỳ

2.1.14. File sinhvien_dtb_toankhoa.xlsx

Tóm tắt: Dữ liệu này ghi lại kết quả học tập (điểm trung bình) tổng kết của sinh viên cũng như số tín chỉ mà sinh viên tích lũy được trong toàn khóa học.

Tên cột	Loại	Mô tả	Chi tiết
mssv	String	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
dtb_toankhoa	Float	Điểm trung bình toàn khóa	Điểm trung bình của sinh viên trong toàn bộ khóa học. Đặc trưng chính trong quá trình huấn luyện và kiểm thử
dtb_tichluy	Float	Điểm trung bình tích lũy	Điểm trung bình tích lũy của sinh viên qua các học kỳ

sotc_tichluy	integer	Số tín chỉ tích lũy	Số tín chỉ mà sinh viên đã hoàn thành trong toàn khóa học

2.1.15. File uit_hocphi_miengiam.xlsx

Tóm tắt: Dữ liệu này cung cấp thông tin sinh viên và lý do sinh viên được miễn giảm học phí.

Tên cột	Loại	Mô tả	Chi tiết
id	int64	Index của dữ liệu	Index tự tăng, không có ý nghĩa đặc biệt
mssv	object	Mã số sinh viên	Mã số sinh viên không là khóa chính và được dùng để xác định từng sinh viên
hocky	int64	Học kỳ mà sinh viên được miễn giảm học phí	Có thể giúp xác định mốc thời gian khi sinh viên được miễn giảm học phí
namhoc	int64	Năm học mà sinh viên được miễn giảm học phí	Có thể giúp xác định mốc thời gian khi sinh viên được miễn giảm học phí
doituong	object	Đối tượng được miễn giảm	Lý do mà sinh viên được miễn giảm học phí
miengiam	int64	Số tiền được miễn giảm	Số tiền mà sinh viên được miễn giảm, có thể là % hoặc số tiền cụ thể
ghichu	object	Ghi chú về lý do miễn giảm	Mô tả chi tiết hơn về lý do miễn giảm học phí cho sinh viên

Ghi chú:

Mức xếp loại trúng tuyển của sinh viên

0: Trúng tuyển theo phương thức THPT có điểm thi THPT < 20 hoặc ĐGNL < 600

1: Trúng tuyển theo phương thức CUTUYEN, 30A, THPT có điểm thi 20 THPT < 22 hoặc 600 ≤ ĐGNL < 750

2: Trúng tuyển theo phương thức THPT có điểm thi 20 THPT < 22 hoặc 750 ≤ ĐGNL < 900

3: Trúng tuyển theo phương thức THPT có điểm thi 20 THPT < 22 hoặc 900 ≤ ĐGNL < 1000

4: Trúng tuyển theo phương thức CCQT, TT-BỘ, ƯT-BỘ, ƯT-ĐHQG, THPT có điểm thi 20 THPT < 22 hoặc ĐGNL 1000

7 vùng kinh tế của Việt Nam

1. Vùng Trung du và miền núi phía Bắc

2. Đồng bằng Bắc Bộ hay đồng bằng sông Hồng

3. Bắc Trung Bộ

4. Vùng duyên hải Nam Trung Bộ

5. Vùng Tây Nguyên

6. Vùng Đông Nam Bộ

7. Vùng đồng bằng sông Cửu Long

Mức xếp loại anh văn đầu vào của sinh viên

0: Chứng chỉ TOEIC < 300 hoặc điểm bài thi anh văn đầu vào < 40

1: Chứng chỉ 300 ≤ TOEIC 345 hoặc điểm bài thi anh văn đầu vào 40 [điểm] 60

2: Chứng chỉ 350 ≤ TOEIC 395 hoặc điểm bài thi anh văn đầu vào 60 [điểm] 70

3: Chứng chỉ 400 ≤ TOEIC 445 hoặc điểm bài thi anh văn đầu vào 70 [điểm] 80

4: Chứng chỉ 500 ≤ TOEIC 555 hoặc điểm bài thi anh văn đầu vào 80 [điểm] 90

5: Chứng chỉ TOEIC 555 hoặc điểm bài thi anh văn đầu vào [điểm] 90

Mức xếp loại điểm rèn luyện của sinh viên

Điểm rèn luyện tích luỹ được tính bằng trung bình cộng điểm rèn luyện các học kỳ trước đó của sinh viên. Điểm số sau khi tính trung bình được phân loại thành các mức khác nhau.

0: đrl < 35

1: 35 ≤ đrl < 50

2: 50 ≤ đrl < 65

3: 65 ≤ đrl < 80

4: 80 ≤ đrl < 90

5: đrl 90

2.2. Tiền xử lý dữ liệu

2.2.1. Tiền xử lý tổng quát

Đối với mỗi tệp dữ liệu định dạng Excel trong , thực hiện lần lượt các bước sau để làm sạch và cải thiện chất lượng dữ liệu:

- Xóa các khoảng trắng ở tên cột (header).
- Loại bỏ các dữ liệu trùng lặp bằng phương pháp drop_duplicates
- Xóa các cột không cần thiết
- Loại bỏ các kí tự đặc biệt và giá trị rỗng

2.2.1.1. Bảng sinhvien.xlsx

Các bước tiền xử lý bảng sinh viên:

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'namsinh', 'noisinh', 'diachi_tinhtp', 'tinhtrang']

```
for file in files:
 if file in '01.sinhvien.xlsx':
    df = pd.read_excel(file)
    tmp = ['id','namsinh','noisinh','diachi_tinhtp','tinhtrang']
    df.drop(columns=tmp, inplace=True, errors='ignore')
    df.info()
    df.to_excel(file, index=False)
    break
```

- Xây dựng hàm làm sạch tên và chuẩn hóa về định dạng chung.

```
def cleaning(name):
    if pd.isna(name) or name.strip() == "":
        return ""
    name = name.strip()
    name = unidecode.unidecode(name).title()
    name = name.replace("'", "").replace('"', '').strip()
    return name
```

  - Kiểm tra giá trị rỗng: Nếu name là NaN hoặc chuỗi rỗng, hàm trả về "".
  - Loại bỏ khoảng trắng ở đầu và cuối chuỗi.
  - Loại bỏ dấu và chuẩn hóa chữ cái đầu: chuyển chuỗi name thành chữ cái Latin không dấu, đồng thời viết hoa chữ cái đầu của mỗi từ.
  - Loại bỏ dấu ngoặc đơn và ngoặc kép.
- Xây dựng hàm tìm tên gần khớp nhất với tên đã làm sạch từ hàm cleaning() trong danh sách mapp_flat.

```
def get_closest_match(name):
    cleaned_name = cleaning(name)
    if not cleaned_name:
        return None
    match, score = process.extractOne(cleaned_name, mapp_flat)
    return match if score >= 50 else None
```

  - Làm sạch tên: Gọi cleaning(name) để chuẩn hóa tên đầu vào.

  - Kiểm tra tên rỗng: Nếu cleaned_name là chuỗi rỗng, trả về None.
  - Tìm tên gần khớp nhất: process.extractOne(cleaned_name, mapp_flat) tìm tên có độ tương đồng cao nhất giữa cleaned_name và các tên trong mapp_flat, trả về cả match (tên gần khớp nhất) và score (điểm tương đồng).
  - Lọc tên theo điểm khớp: Nếu score >= 50, trả về match; nếu không, trả về None.
- Xử lý thuộc tính khu vực hedt và khoa:
  - Đọc dữ liệu từ mỗi tệp Excel trong danh sách files và lưu vào DataFrame df.
  - Nếu DataFrame chứa cột hedt, thực hiện one-hot encoding cho hedt, tạo ra các cột mới với tiền tố hedt_.
  - Nếu DataFrame chứa cột khoa, thực hiện one-hot encoding cho khoa, tạo ra các cột mới với tiền tố khoa_.
  - Làm sạch tên: Nếu DataFrame có cột noisinh, hàm cleaning được áp dụng để làm sạch giá trị của cột.
  - Xác định khu vực: Sử dụng hàm get_closest_match để xác định khu vực (khuvuc) tương ứng với mỗi noisinh.
  - Ánh xạ ngược: Giá trị khuvuc được ánh xạ ngược từ reverse_mapping (chưa được định nghĩa trong mã nhưng có thể là một từ điển để chuyển giá trị khu vực về mã số).
  - Gán giá trị mặc định: Nếu khuvuc có giá trị NaN, thay thế bằng -1 và chuyển sang kiểu Int64 để cho phép chứa giá trị NaN.
  - Xác định các giá trị noisinh không thể xác định được khuvuc (khuvuc = -1). Nếu có bất kỳ giá trị nào, in ra các giá trị noisinh đó dưới dạng thông báo lỗi để kiểm tra.
  - Sau khi hoàn thành các bước trên, DataFrame df được lưu lại vào tệp Excel ban đầu.

```
for file in files:
    df = pd.read_excel(file)

    # One hot encoding

    if 'hedt' in df.columns:
        df = pd.get_dummies(df, columns=['hedt'], prefix='hedt')
    if 'khoa' in df.columns:
        df = pd.get_dummies(df, columns=['khoa'], prefix='khoa')

    if 'noisinh' in df.columns:
        df['noisinh'] = df['noisinh'].apply(cleaning)
        df['khuvuc'] = df['noisinh'].apply(get_closest_match)
        df['khuvuc'] = df['khuvuc'].map(reverse_mapping)
        df['khuvuc'] = df['khuvuc'].fillna(-1).astype('Int64')
        unexpected_values = df[df['khuvuc'] == -1]['noisinh'].unique()
        if unexpected_values.size > 0:
            print(f"Test Error: {unexpected_values}")
    df.to_excel(file, index=False)
```

2.2.1.2. Bảng xeploaiav.xlsx

Các bước tiền xử lý bảng xếp loại anh văn:

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'listening', 'reading', 'mamh', 'ghichu']

```
for file in files:
    if file in '04.xeploaiav.xlsx':
        df = pd.read_excel(file)
        tmp = ['id','listening','reading','mamh','ghichu']
        df.drop(columns=tmp, inplace=True, errors='ignore')
        df.info()
        df.to_excel(file, index=False)
        break
```

- Tạo cột phân loại xl_av dựa trên tổng điểm như sau:
    - Dưới 40: gán xl_av là 0.
    - Từ 40 đến dưới 60: gán xl_av là 1.
    - Từ 60 đến dưới 70: gán xl_av là 2.
    - Từ 70 đến dưới 80: gán xl_av là 3.

- Từ 80 đến dưới 90: gán xl_av là 4.
- Từ 90 trở lên: gán xl_av là 5.

```
for index, row in df.iterrows():
    total = row['total']
    if total < 40:
        df.at[index, 'xl_av'] = 0
    elif 40 <= total < 60:
        df.at[index, 'xl_av'] = 1
    elif 60 <= total < 70:
        df.at[index, 'xl_av'] = 2
    elif 70 <= total < 80:
        df.at[index, 'xl_av'] = 3
    elif 80 <= total < 90:
        df.at[index, 'xl_av'] = 4
    elif total >= 90:
        df.at[index, 'xl_av'] = 5
```

2.2.1.3. Bảng thisinh.xlsx

Các bước tiền xử lý bảng thí sinh:

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['lop12_matinh', 'lop12_matruong', 'TEN_TRUONG', 'dien_tt', 'diem_tt']

```
for file in files:
   if file in '05.ThiSinh.xlsx':
      df = pd.read_excel(file)
      tmp = ['lop12_matinh','lop12_matruong','TEN_TRUONG','dien_tt','diem_tt']
      df.drop(columns=tmp, inplace=True, errors='ignore')
      df.info()
      df.to_excel(file, index=False)
      break
```

- Tạo thêm một cột xeploai_tt dựa trên điểm thi DGNL hoặc THPT:

```
if 'dien_tt' in df.columns and 'diem_tt' in df.columns:
    df['xeploai_tt'] = None

    for index, row in df.iterrows():
        if row['dien_tt'] == 'ĐGNL':
            dgnl_score = row['diem_tt']
            if dgnl_score < 600:
                df.at[index, 'xeploai_tt'] = 0
            elif 600 <= dgnl_score < 750:
                df.at[index, 'xeploai_tt'] = 1
            elif 750 <= dgnl_score < 900:
                df.at[index, 'xeploai_tt'] = 2
            elif 900 <= dgnl_score < 1000:
                df.at[index, 'xeploai_tt'] = 3
            elif dgnl_score >= 1000:
                df.at[index, 'xeploai_tt'] = 4
        else:
            thpt_score = row['diem_tt']
            if thpt_score < 20:
                df.at[index, 'xeploai_tt'] = 0
            elif 20 <= thpt_score < 22:
                df.at[index, 'xeploai_tt'] = 1
            elif 22 <= thpt_score < 24:
                df.at[index, 'xeploai_tt'] = 2
            elif 24 <= thpt_score < 26:
                df.at[index, 'xeploai_tt'] = 3
            elif thpt_score >= 26:
                df.at[index, 'xeploai_tt'] = 4
```

2.2.1.4. Bảng diemrl.xlsx

Các bước tiền xử lý bảng điểm rèn luyện:

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'ghichu', 'drl']

```
for file in files:
   if file in '10.diemrl.xlsx':
      df = pd.read_excel(file)
      tmp = ['id','ghichu','drl']
      df.drop(columns=tmp, inplace=True, errors='ignore')
      df.info()
      df.to_excel(file, index=False)
      break
```

- Tạo cột phân loại điểm rèn luyện drltl dựa trên tổng điểm như sau:
  - Dưới 35: gán drltl là 0.
  - Từ 35 đến dưới 50: gán drltl là 1.
  - Từ 50 đến dưới 65: gán drltl là 2.
  - Từ 65 đến dưới 80: gán drltl là 3.
  - Từ 80 đến dưới 90: gán drltl là 4.
  - Từ 90 trở lên: gán drltl là 5.

```
for index, row in df.iterrows():
    drl_score = row['drl']
    if drl_score < 35:
        df.at[index, 'drltl'] = 0
    elif 35 <= drl_score < 50:
        df.at[index, 'drltl'] = 1
    elif 50 <= drl_score < 65:
        df.at[index, 'drltl'] = 2
    elif 65 <= drl_score < 80:
        df.at[index, 'drltl'] = 3
    elif 80 <= drl_score < 90:
        df.at[index, 'drltl'] = 4
    elif drl_score >= 90:
        df.at[index, 'drltl'] = 5
```

- Kết hợp 2 bảng drl

```
diemrl = pd.read_excel('diemrl.xlsx')
diemrl2 = pd.read_excel('10.diemrl.xlsx')

if 'mssv' in diemrl.columns and 'mssv' in diemrl2.columns:
    combined_df = pd.concat([diemrl, diemrl2], ignore_index=True)

    combined_df.info()

    combined_df.to_excel('combined_diemrl.xlsx', index=False)
```

- Tạo thêm 1 cột drltl để phân loại điểm rèn luyện:

```
if 'drl' in df.columns:
    df['drltl'] = None

    for index, row in df.iterrows():
        drl_score = row['drl']
        if drl_score < 35:
            df.at[index, 'drltl'] = 0
        elif 35 <= drl_score < 50:
            df.at[index, 'drltl'] = 1
        elif 50 <= drl_score < 65:
            df.at[index, 'drltl'] = 2
        elif 65 <= drl_score < 80:
            df.at[index, 'drltl'] = 3
        elif 80 <= drl_score < 90:
            df.at[index, 'drltl'] = 4
        elif drl_score >= 90:
            df.at[index, 'drltl'] = 5
```

2.2.2. Bảng dữ liệu sau khi xử lý

STT	Tên thuộc tính	Ý nghĩa thuộc tính	Kiểu dữ liệu	Ghi chú
1	mssv	Mã số sinh viên	object	
2	gioitinh	Giới tính	float64	1: nam 0: nữ
3	lopsh	Lớp sinh hoạt	object	
4	khoahoc	Khóa học	float64	
5	chuyennganh2	Chuyên ngành 2	object	
6	hedt_ CLC	Hệ đào tạo CLC	bool	
7	hedt_ CNTN	Hệ đào tạo CNTN	bool	
8	hedt_ CQUI	Hệ đào tạo chính quy	bool	
9	hedt_ CTTT	Hệ đào tạo CTTT	bool	
10	hedt_ KSTN	Hệ đào tạo KSTN	bool	
11	khoa_ CNPM	Khoa CNPM	bool	
12	khoa_ HTTT	Khoa HTTT	bool	

13	khoa_ KHMT	Khoa KHMT	bool	
14	khoa_ KTMT	Khoa KTPM	bool	
15	khoa_ KTTT	Khoa KTTT	bool	
16	khoa_ MMT&TT	Khoa MMT&TT	bool	
17	khuvuc	Khu vực sinh sống của sinh viên	float64	
18	xl_av	Xếp loại anh văn đầu vào của mỗi sinh viên	float64	Sinh viên được xếp loại anh văn đầu vào dựa vào kết quả bài thi anh văn đầu vào ở trường hoặc các chứng chỉ tiếng Anh khác. Mức độ tiếng Anh đầu vào đươc đánh giá theo mức độ tang dần từ 0 → 5
19	xeploai_tt	Xếp loại trúng tuyển của sinh viên	float64	Sinh viên được xếp loại dựa trên diện trúng tuyển và kết quả đầu vào, được chia theo giá trị từ 0 → 4
20	hocky	Học kỳ	float64	
21	namhoc	Năm học	float64	
22	drltl	Điểm rèn luyện tích lũy	float64	Đánh giá theo 6 mức tăng dần từ 0 → 5.

23	dtbhk	Điểm trung bình học kỳ	float64	
24	sotchk	Số tín chỉ học kỳ	float64	
25	dtbhk_truoc	Điểm trung bình học kỳ trước	float64	

2.3. Khám phá dữ liệu

Chuyển thuộc tính dtbhk thành thuộc tính đầu ra 'xeploai' như sau:

- Dưới 5: gán xeploai là 0.
- Từ 5 đến dưới 6.5: gán xeploai là 1.
- Từ 6.5 đến dưới 8: gán xeploai là 2.
- Từ 8 đến dưới 9: gán xeploai là 3.
- Từ 9 trở lên: gán drltl là 4.

```
def danh_gia_diem(score):
    if score < 5:
        return 0
    elif 5 <= score < 6.5:
        return 1
    elif 6.5 <= score < 8:
        return 2
    elif 8 <= score < 9:
        return 3
    else:
        return 4

df['xeploai'] = df['dtbhk'].apply(danh_gia_diem)
```

Gộp các thuộc tính hệ đào tào và khoa thành 1 thuộc tính:

- Thuộc tính hệ đào tạo:

```
def determine_hedt(row):
    if row['hedt_ CLC']:
        return 'CLC'
    elif row['hedt_ CNTN']:
        return 'CNTN'
    elif row['hedt_ CQUI']:
        return 'CQUI'
    elif row['hedt_ CTTT']:
        return 'CTTT'
    else:
        return 'Other'
```

- Thuộc tính khoa:

```
def determine_khoa(row):
    for col in row.index:
        if 'khoa_' in col and row[col] == True:
            return col.replace('khoa_', '').strip()
    return 'Other'
```

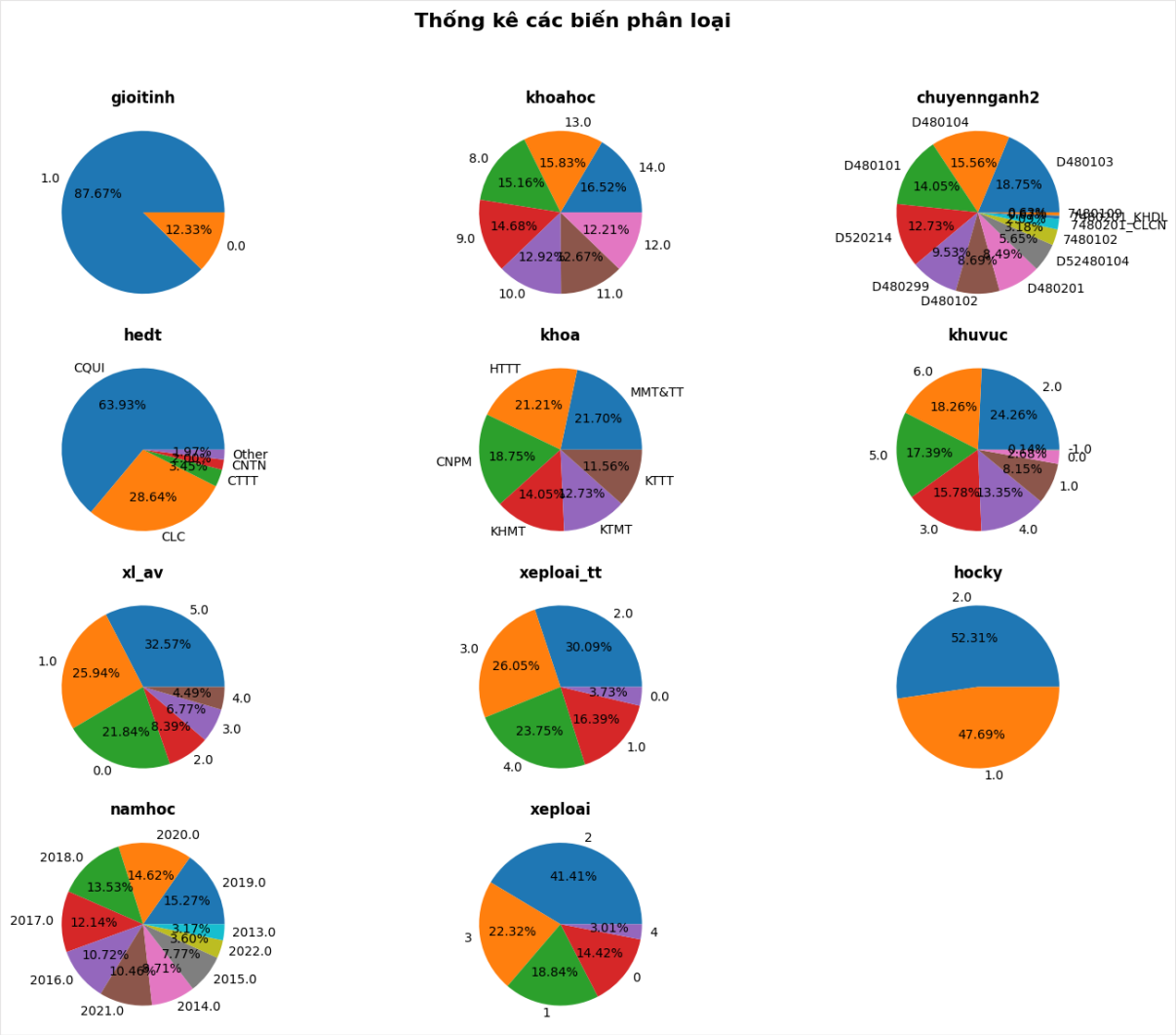
2.3.1. Phân tích đơn biến từng thuộc tính

Để phân tích sự ảnh hưởng của các thuộc tính tới kết quả học tập của sinh viên, ta cần xác định các thuộc tính phân tích:

- Thuộc tính phân loại: ['gioitinh', 'khoahoc', 'chuyennganh2', 'hedt', 'khoa', 'khuvuc', 'xl_av', 'xeploai_tt', 'hocky', 'namhoc', 'xeploai']
- Thuộc tính tuyến tính: ['drltl', 'sotchk', 'dtbhk_truoc']

a. Thuộc tính phân loại:

Tiến hành tạo các biểu đồ cho các thuộc tính phân loại:



- Giới tính (gioitinh): Biểu đồ cho thấy phần lớn dữ liệu thuộc về một giới tính (87.67% là một giới tính cụ thể), trong khi giới tính còn lại chiếm 12.33%. Điều này có thể gợi ý sự mất cân bằng giới tính trong tập dữ liệu hoặc trong đối tượng khảo sát.
- Khóa học (khoahoc): Các khóa học được phân bổ khá đồng đều với một số khóa học chiếm tỷ lệ cao hơn một chút. Điều này cho thấy sự đa dạng trong việc chọn lựa khóa học của các sinh viên, nhưng không có sự tập trung rõ rệt vào một khóa học cụ thể.
- Chuyên ngành 2 (chuyennganh2): Biểu đồ này có nhiều nhãn nhỏ và phức tạp, cho thấy nhiều chuyên ngành với tỷ lệ phân bố khác nhau. Điều này phản ánh tính đa dạng trong lựa chọn chuyên ngành của sinh viên.
- Hệ đào tạo (hedt): Hệ CQUI chiếm tỷ lệ lớn nhất với 63.93%, tiếp theo là hệ CLC với 28.64%. Các hệ khác chỉ chiếm tỷ lệ nhỏ. Điều này cho thấy hầu

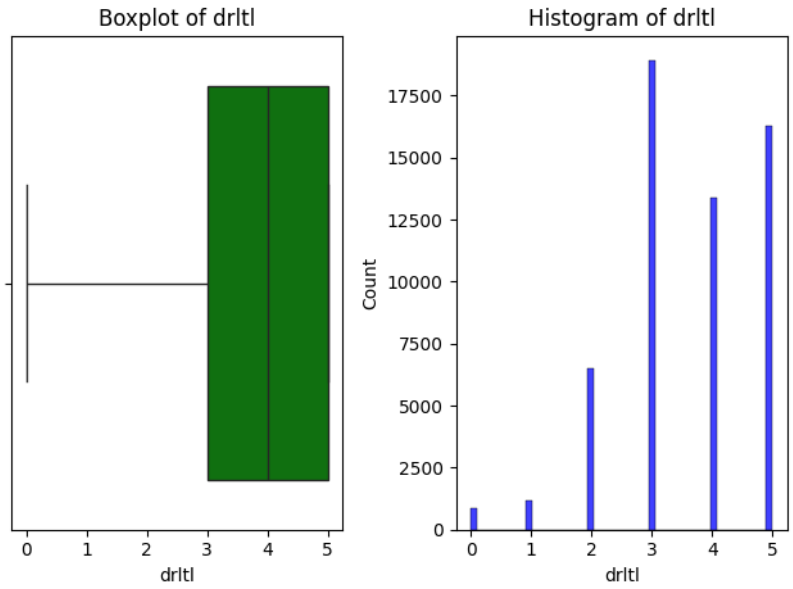
hết sinh viên nằm trong các hệ đào tạo chính như CQUI và CLC. Khoa (khoa): Phân bổ sinh viên ở các khoa cũng tương đối đồng đều, với một số khoa như HHT, MMT&TT, CNPM chiếm tỷ lệ cao hơn một chút. Điều này thể hiện sự phân bổ đa dạng của sinh viên giữa các khoa.

- Khu vực (khuvuc): Sinh viên phân bố ở các khu vực khá đồng đều, với các khu vực có tỷ lệ lớn hơn là 1 và 2. Điều này có thể phản ánh sự đa dạng địa lý của sinh viên trong dữ liệu.
- Xếp loại AV (xl_av): Các mức xếp loại AV được phân bổ đồng đều, nhưng chủ yếu tập trung vào các mức cao (5.0 và 4.5). Điều này có thể cho thấy sinh viên có trình độ ngoại ngữ tốt.
- Xếp loại TT (xeploai_tt): Xếp loại TT cũng có sự phân bố đồng đều giữa các mức, với mức 3.0 và 2.0 chiếm tỷ lệ lớn nhất. Điều này cho thấy sự đa dạng về thành tích học tập của sinh viên.
- Học kỳ (hocky): Hầu hết sinh viên tập trung vào học kỳ 2 (52.31%), còn lại là học kỳ 1. Điều này là do thời điểm khảo sát được thực hiện trong học kỳ 1 nên chưa có dữ liệu học kỳ 2 của năm hiện được khảo sát
- Năm học (namhoc): Sinh viên phân bố ở các năm học từ 2017 đến 2022, với một số năm học gần đây chiếm tỷ lệ lớn hơn. Điều này có thể do sự tập trung của dữ liệu gần đây hoặc gia tăng số lượng của sinh viên mới.
- Xếp loại (xeploai): Xếp loại phân bố đa dạng, với mức 2 (xếp loại khá) chiếm tỷ lệ cao nhất (40.56%). Bên cạnh đó, mức 4 (xếp loại xuất sắc) chiếm rất ít (4.11%), gây ra tình trạng mất cân bằng dữ liệu, ảnh hưởng tới độ chính xác của mô hình dự đoán

b. Thuộc tính tuyến tính:

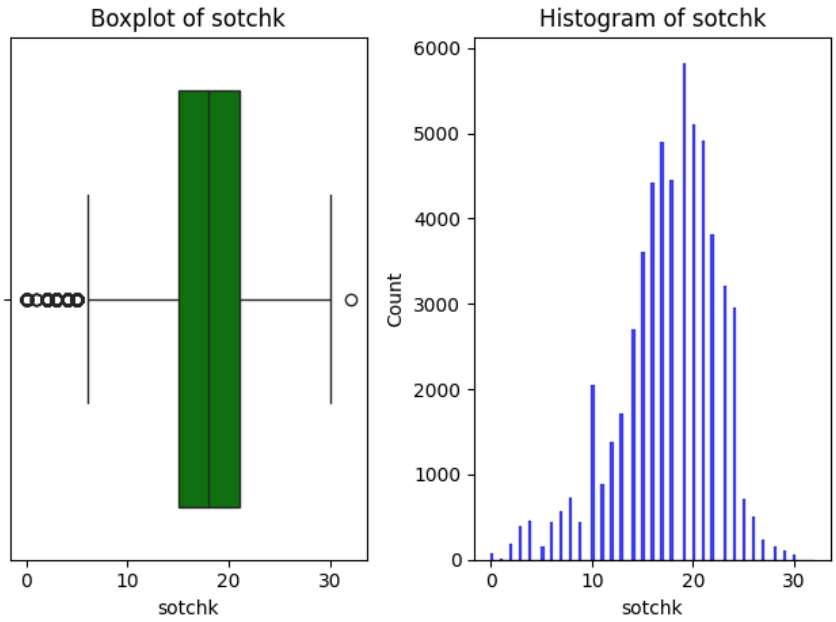
Tiến hành vẽ boxplot và histogram:

- Thuộc tính drltl:



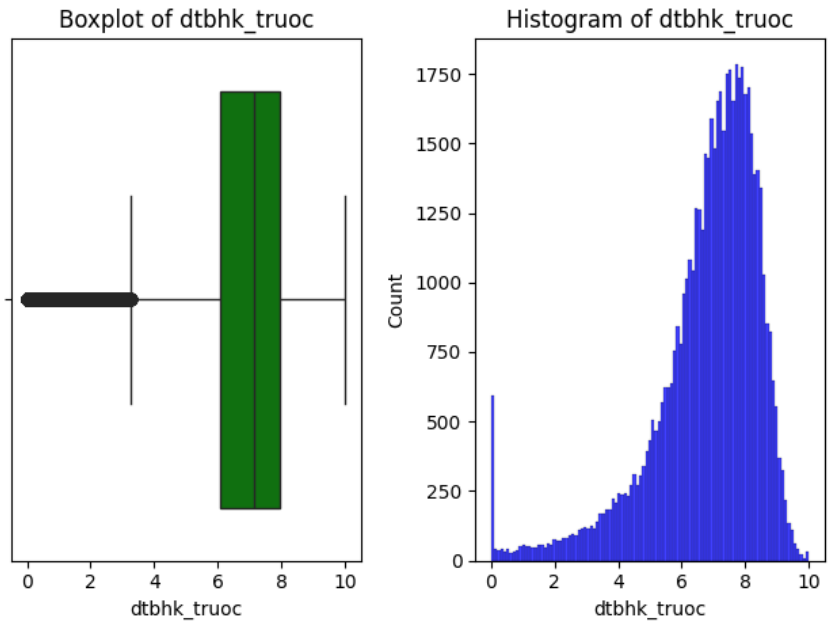
- Boxplot: Điểm rèn luyện tích lũy chủ yếu tập trung ở mức cao, từ khoảng 3 đến 5. Điều này cho thấy đa phần sinh viên có điểm rèn luyện tích lũy tốt. Không có điểm dữ liệu bất thường hoặc quá thấp, điều này có thể chỉ ra rằng hầu hết sinh viên đạt yêu cầu về mặt rèn luyện.
- Histogram: Biểu đồ tần suất cho thấy rằng các điểm rèn luyện cao hơn (gần mức 5) phổ biến hơn, với số lượng sinh viên giảm dần khi điểm rèn luyện thấp đi. Điều này có thể cho thấy sự nỗ lực và tính kỷ luật cao trong đa số sinh viên.

- Thuộc tính sotchk:



- Boxplot: Có một số điểm ngoại lệ nằm dưới mức tín chỉ bình thường số tín chỉ dưới tối thiểu đa số là do sinh viên năm cuối đăng ký, vẫn có một số trường hợp (khá ít) không phải sinh viên năm cuối do học lại hoặc học cải thiện, tuy vậy phần lớn sinh viên đăng ký từ khoảng 10 đến 30 tín chỉ mỗi học kỳ. Phần đa dữ liệu tập trung ở mức trung bình, cho thấy sinh viên thường có số tín chỉ nằm trong ngưỡng quy định hoặc mong muốn để đảm bảo tiến độ học tập.
- Histogram: Biểu đồ phân bố có dạng hình chuông, cho thấy rằng phần lớn sinh viên đăng ký một lượng tín chỉ trung bình, với số lượng sinh viên giảm dần khi số tín chỉ quá cao hoặc quá thấp. Điều này có thể gợi ý rằng sinh viên có xu hướng đăng ký số tín chỉ phù hợp với sức học và thời gian.

- Thuộc tính dtbhk_truoc



- Boxplot: Kì đầu chưa có thông tin dtbhk_truoc nên được mặc định bằng 0 , có một số điểm ngoại lệ nằm dưới mức điểm trung bình, tuy nhiên phần lớn sinh viên có điểm trung bình học kỳ trước từ khoảng 5 đến 8. Phần đa dữ liệu tập trung ở mức trung bình, cho thấy hầu hết sinh viên đạt điểm trong ngưỡng phổ biến, phân phối chuẩn điểm hình chuông(bell-curved), đảm bảo mức độ tiến bộ học tập
- Histogram: Biểu đồ phân bố có dạng hơi lệch phải, cho thấy phần lớn

sinh viên đạt điểm trung bình học kỳ trước trong khoảng 6 đến 8, với số lượng sinh viên giảm dần khi điểm số giảm xuống hoặc tăng lên ngoài khoảng này. Điều này gợi ý rằng sinh viên thường có xu hướng đạt điểm cao hoặc trung bình, phản ánh một mức độ nỗ lực và sự ổn định trong học tập.

2.3.2. Trích chọn đặc trưng

2.3.2.1. Các phương pháp trích chọn đặc trưng

Filter method sử dụng các chỉ số thống kê hoặc độ liên kết giữa các đặc trưng và nhãn mục tiêu để đánh giá mức độ quan trọng của mỗi đặc trưng, trong đó có:

- Correlation Matrix: kiểm tra mối quan hệ giữa các đặc trưng và giữa đặc trưng với nhãn mục tiêu thông qua **hệ số tương quan (correlation coefficient)**

- Mutual Information: đo lường khả năng liên kết mạnh mẽ giữa các thuộc tính và có thể giúp mô hình phân loại tốt hơn.

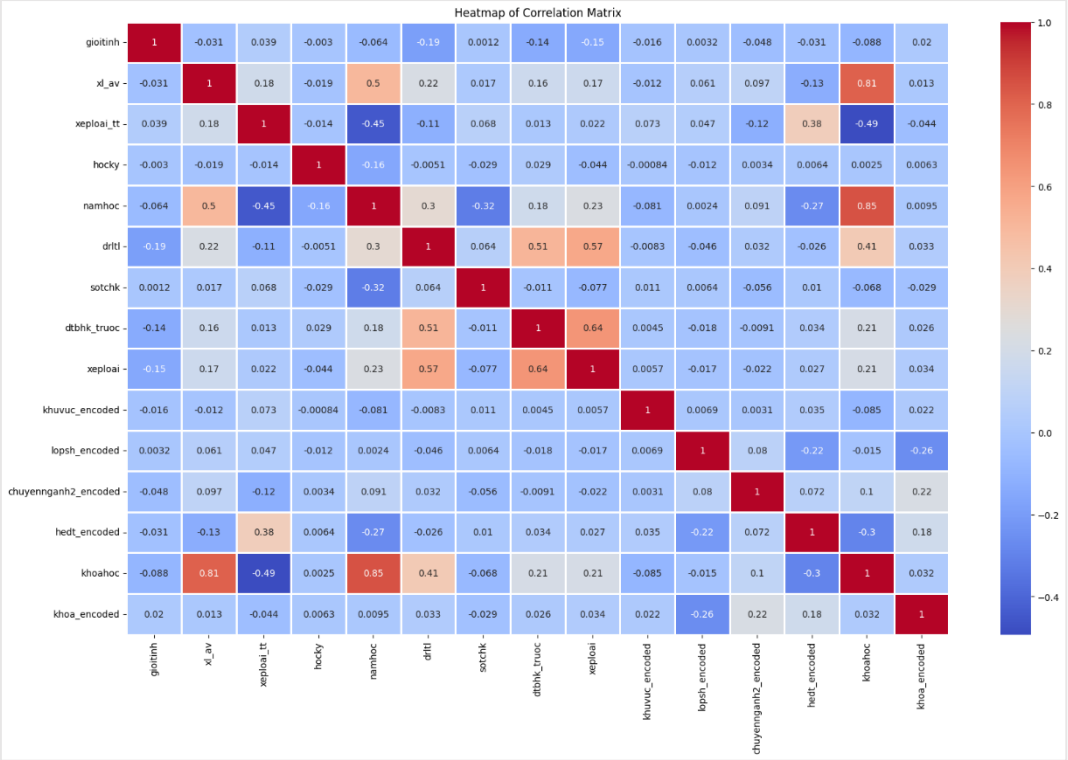
- Chi-Square Test (Chi-Square): sử dụng kiểm định **chi-square** để kiểm tra mối quan hệ giữa các thuộc tính. Trong đó, nếu cặp đặc trưng có giá trị p-value dưới ngưỡng và mức độ liên kết Cramer's V cao thì cặp đặc trưng đó có sự liên kết mạnh mẹ với nhau

Wrapper Method chọn lựa đặc trưng dựa trên khả năng của mô hình học máy để, trong đó nhóm sử dụng **SelectKBest** để chọn ra k đặc trưng tốt nhất dựa trên hiệu suất của mô hình.

2.3.2.2. Tiến hành thực hiện

Đầu tiên, quan sát các thuộc tính qua ma trận tương quan **(Correlation Matrix)**

```
#Lấy các thuộc tính đã mã hóa
selected_columns = ['gioitinh', 'xl_av', 'xeploai_tt', 'hocky', 'namhoc', 'drltl', 'sotchk',
       'dtbhk_truoc', 'xeploai', 'khuvuc_encoded',
       'lopsh_encoded', 'chuyennganh2_encoded', 'hedt_encoded', 'khoahoc',
       'khoa_encoded']

df = df[selected_columns]
```

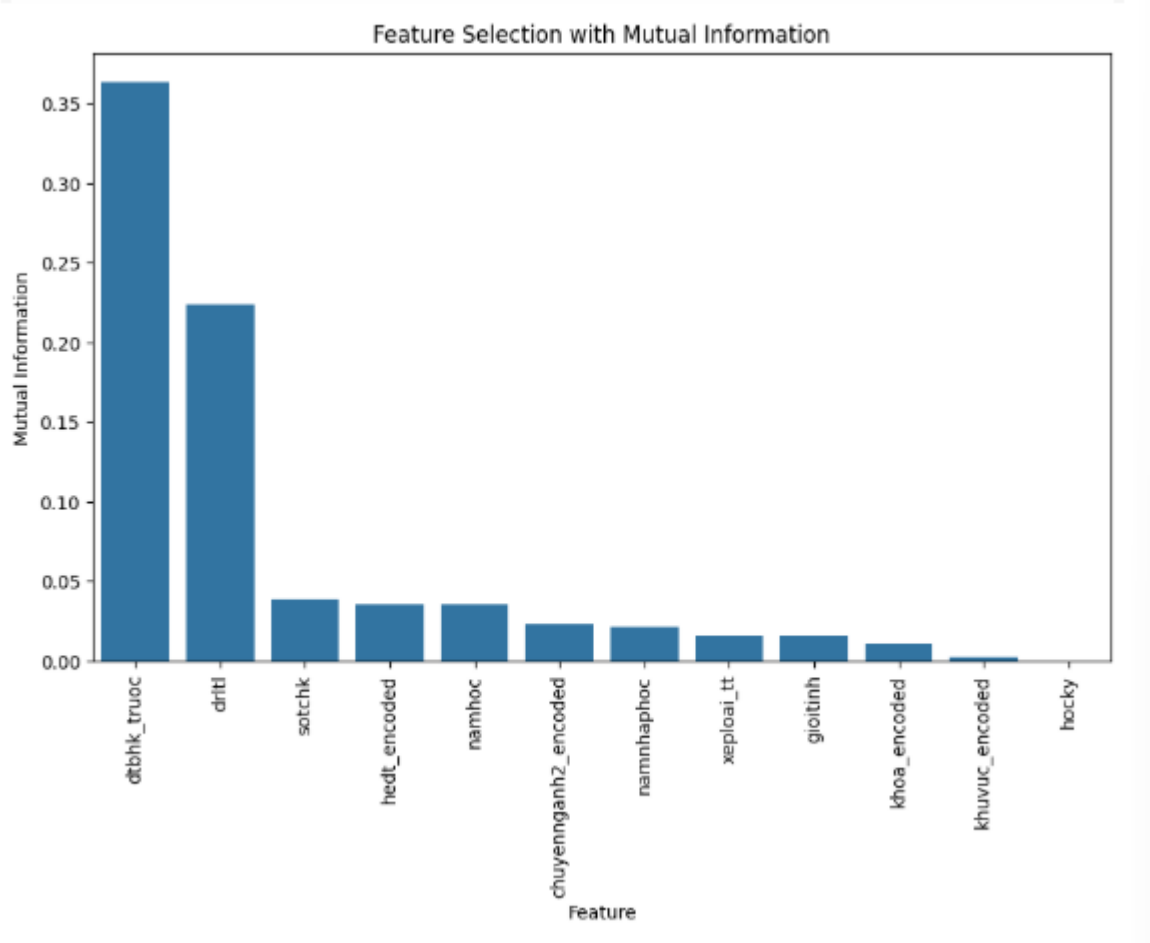


Nhận xét:

+ Thuộc tính khuvuc ảnh hưởng không đáng kể (mức độ tương quan 0.0057) => loại bỏ cột khuvuc

+ Các thuộc tính đầu vào namhoc, khoahoc, xlav có độ tương quan cao (lớn hơn 0.5) nên nhóm chỉ chọn 1 đặc trưng duy nhất do các thuộc tính dư thừa không đóng góp đáng kể vào khả năng dự đoán đầu ra của mô hình => Chọn thuộc tính namhoc

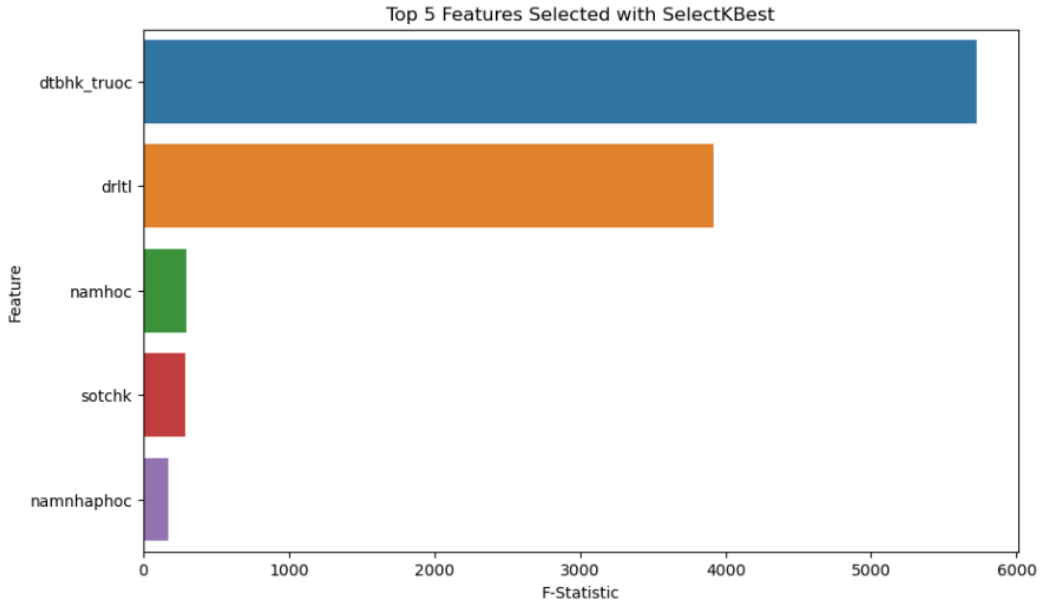
Tiếp theo, sử dụng **Mutual Information** để lựa chọn đặc trưng có mối quan hệ mạnh mẽ với nhãn và loại bỏ các thuộc tính dư thừa:



Nhận xét:

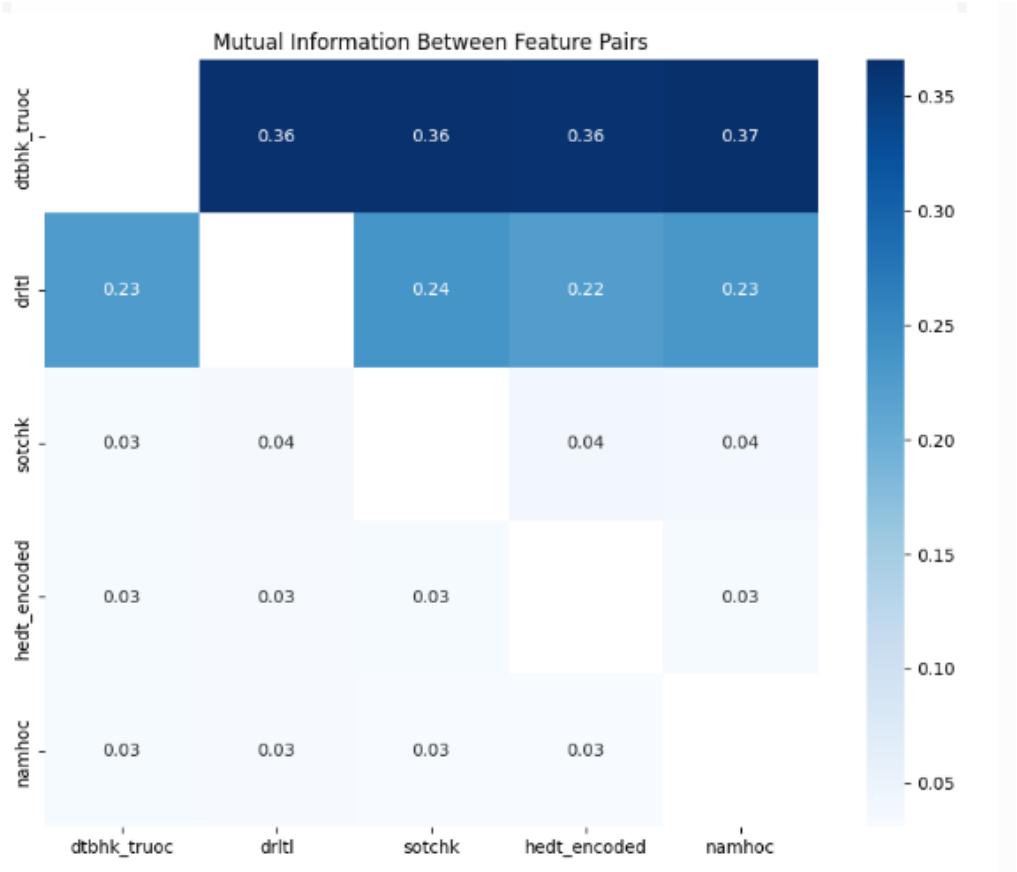
Có thể nhận thấy rằng thuộc tính **dtbhk_truoc** là yếu tố quyết định đến **xeploai**, trong khi các thuộc tính khác như **drltl**, **sotchk**, **hedt**, và **namhoc** cũng có sự ảnh hưởng nhất định với **MI > 0.2**.

Bên cạnh đó, khi nhóm sử dụng phương pháp trích chọn đặc trưng **Select K Best** cũng cho ra kết quả tương tự



2.3.3. Phân tích về mối quan hệ giữa các thuộc tính trong đồ thị mạng

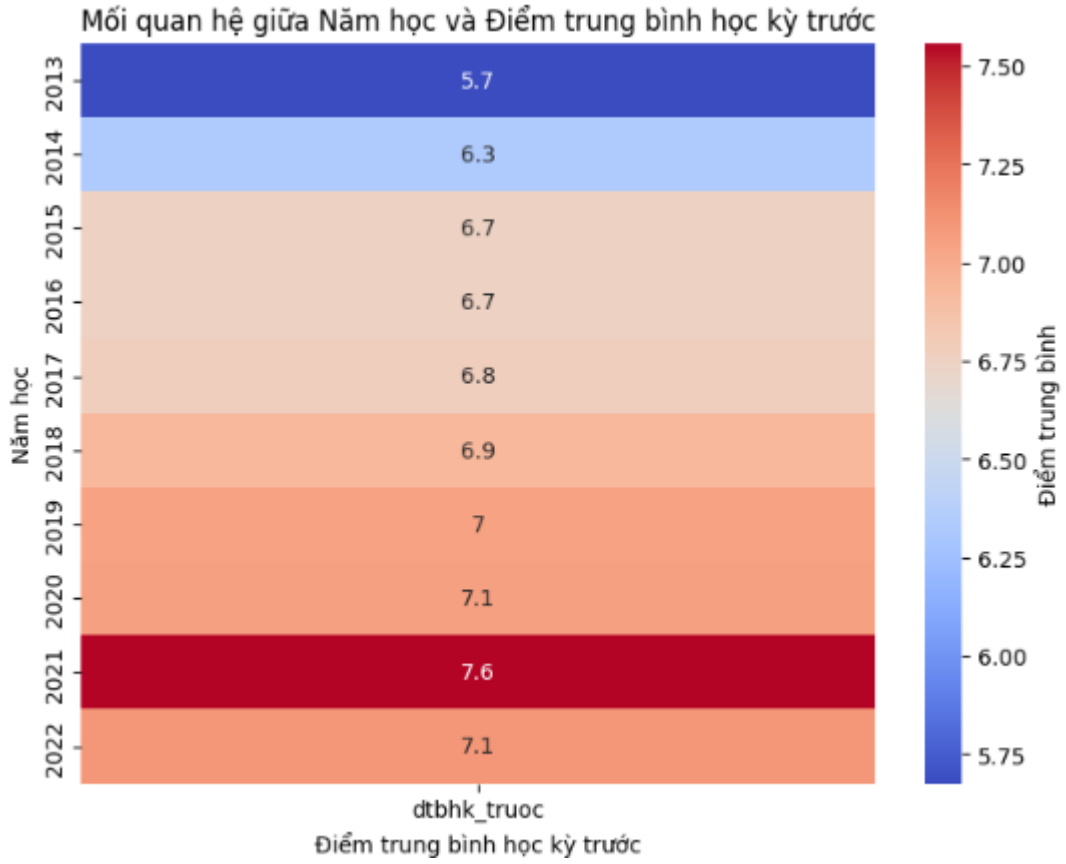
Sau khi đã trích chọn đặc trưng, nhóm tiến hành phân tích sâu hơn về mối quan hệ giữa các thuộc tính đầu vào:



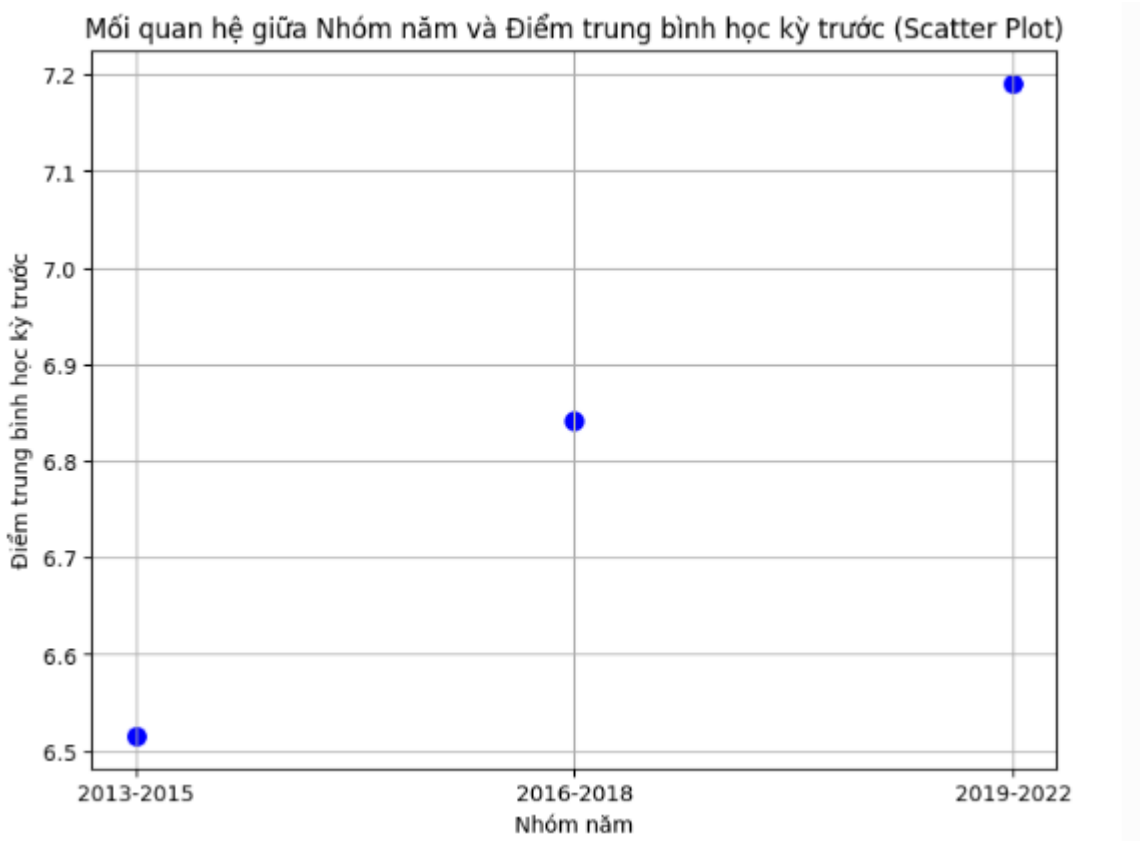
Dựa vào biểu đồ trên, có thể thấy dtbhk_truoc và namhoc có MI cao (0.37),

chứng tỏ đây là một cặp có quan hệ chặt chẽ. Bên cạnh đó sotchk và drltl cũng có MI cao hơn so với các cặp quan hệ khác (0.04)

Để chứng minh cho điều đó, nhóm tiến hành phân tích các mối liên hệ giữa các thuộc tính. Đầu tiên, đối với dtbhk_truoc và namhoc:



Nhận xét: có thể thấy rõ xu hướng quan hệ tuyến tính giữa 2 biến này theo từng năm. Cụ thể, trong những năm gần đây, dtbhktruoc có xu hướng tăng dần, với các giá trị cao hơn so với các năm trước đó. Đây có thể phản ánh sự cải thiện trong chất lượng học tập của sinh viên qua các năm. Có thể chứng minh rõ ràng hơn qua việc phân khoảng các năm và thể hiện ở biểu đồ phân tán dưới đây



Ngoài ra, nhóm cũng thực hiện kiểm định Chisquare để chứng minh liên hệ giữa 2 biến và thấy rằng p_value < 0.5 (mức ý nghĩa) cho thấy namhoc ảnh hưởng đến dtbhktruoc, bên cạnh đó chỉ số cramer v nhằm đo mức độ mạnh yếu của các liên kết với chỉ số 0.02 cho thấy mối liên hệ khá mạnh ở 2 biến này

```
dtb_contingency = pd.crosstab(df['dtbhk_truoc'], df['namhoc'])

dtb_chi2, dtb_p, _, _ = chi2_contingency(dtb_contingency)

{
    "chi2_statistic": dtb_chi2,
    "p_value": dtb_p
}
```

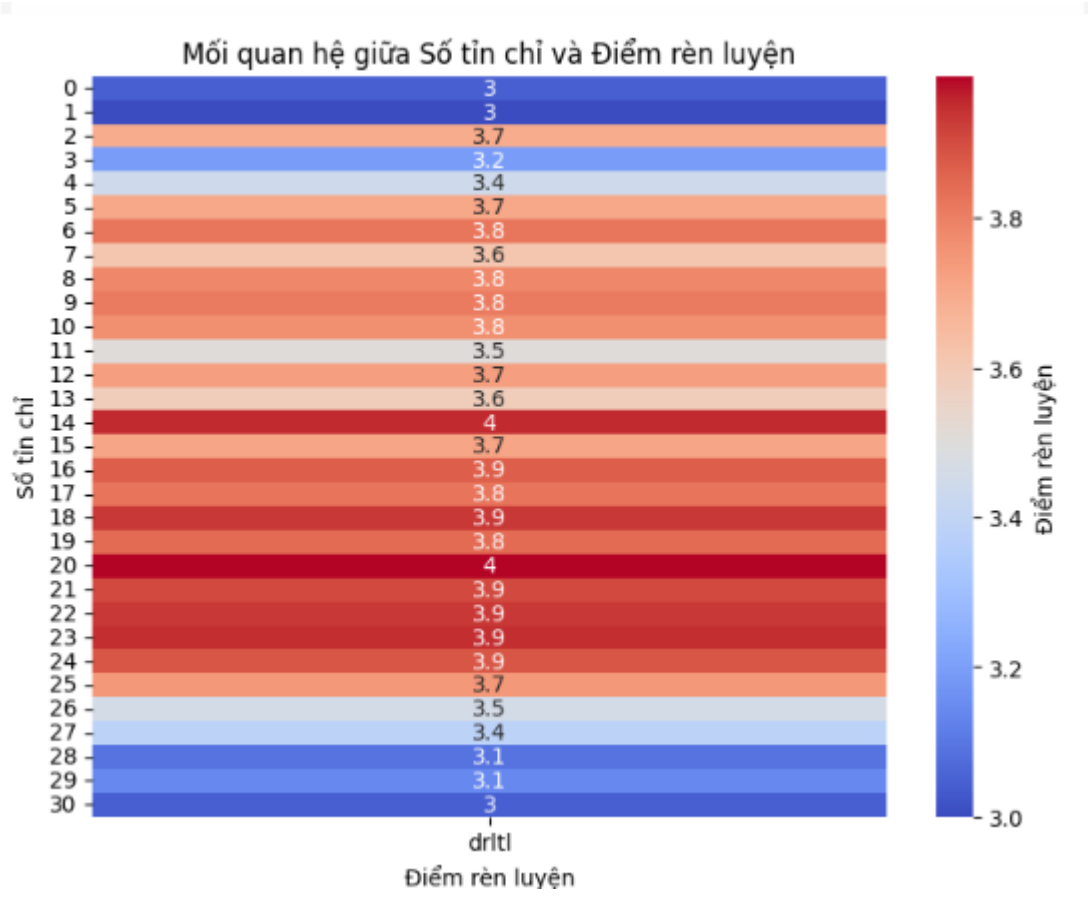
```
Out[13]:
{'chi2_statistic': 14129.254850515426, 'p_value': 3.6682185101371154e-267}
```

```
In [14]:
def cramers_v(chi2, n, dof):
    return np.sqrt(chi2 / (n * dof))

n = df.shape[0]
dtb_dof = dtb_contingency.shape[0] - 1

dtb_cramers_v = cramers_v(dtb_chi2, n, dtb_dof)

{
    "Hedt và xeploai": dtb_cramers_v
}
```

```
Out[14]:
{'Hedt và xeploai': 0.020209587529910384}
```

Thực hiện tương tự với mối liên hệ giữa sotckhk và drltl, nhóm phân tích thấy rằng:

- Ở mức độ tín chỉ đạt ngưỡng quy định thông thường của chương trình đào tạo (từ 14 đến 24), điểm rèn luyện thường ở mức cao nhất, đa phần nằm ở xếp loại tốt.
- Trong khi đó, các khoảng nằm ngoài (>24 và <14) lại có drltl thấp hơn, có thể là do sinh viên sắp tốt nghiệp/ không tập trung học đủ số lượng môn hoặc quá tải dẫn kết quả học tập kỳ trước thấp, ảnh hưởng đến điểm rèn luyện kỳ này.



Bên cạnh đó, chỉ số p_value < 0.5 (mức ý nghĩa) cho thấy namhoc ảnh hưởng đến dtbhktruoc, cũng cố bởi chỉ số cramer v - đo mức độ mạnh yếu của các liên kết với chỉ số 0.07 cho thấy mối liên hệ rất mạnh ở 2 biến này.

```
In [17]:
dtb_contingency = pd.crosstab(df['drltl'], df['sotchk'])

dtb_chi2, dtb_p, _, _ = chi2_contingency(dtb_contingency)

{
    "chi2_statistic": dtb_chi2,
    "p_value": dtb_p
}

Out[17]:
{'chi2_statistic': 1109.5411669457355, 'p_value': 4.7106086336993525e-146}
In [18]:
def cramers_v(chi2, n, dof):
    return np.sqrt(chi2 / (n * dof))

n = df.shape[0]
dtb_dof = dtb_contingency.shape[0] - 1

dtb_cramers_v = cramers_v(dtb_chi2, n, dtb_dof)

{
    "Sotckhk và drltl": dtb_cramers_v
}

Out[18]:
{'Sotckhk và drltl': 0.07867707129335524}
```

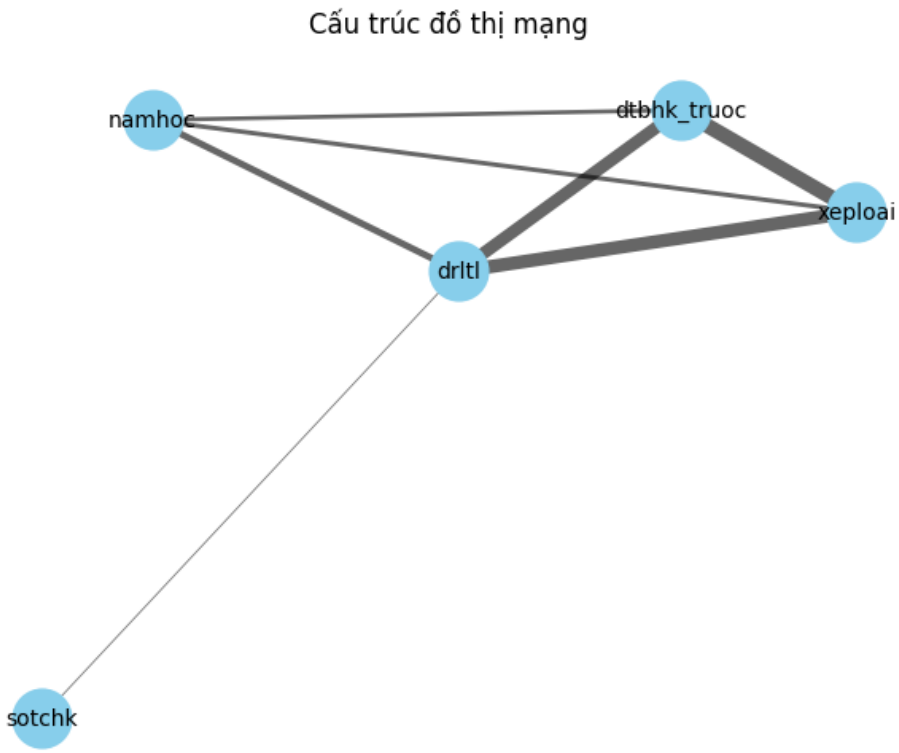
2.4. Xây dựng đồ thị mạng

2.4.1. Trực quan hóa đồ thị

Đồ thị mạng vô hướng biểu diễn các mối quan hệ giữa các biến trong dữ liệu. Cụ thể:

- Nút (node): Các thuộc tính trong tập dữ liệu, gồm gioitinh, lopsh_encoded, khuvuc_encoded...

- Cạnh (edge): Biểu diễn mối quan hệ của các thuộc tính. Độ dày và màu sắc của cạnh biểu thị độ mạnh yếu của mối quan hệ đó. Ví dụ, cạnh dày hơn và màu đậm hơn thường đại diện cho mối quan hệ mạnh hơn.



2.4.2. Các phương pháp biến đổi đồ thị mạng để đưa vào Machine Learning/ Deep Learning

Nhóm thực hiện các cách:

- **Vector hóa dữ liệu**: thông qua DictVectorizer, chuyển đổi dữ liệu từ danh sách các cặp thuộc tính có liên kết mạnh mẽ với nhau thành ma trận mạng (sparse matrix)
- Tạo **ma trận kề** dựa trên mối quan hệ giữa các đối tượng lân cận (các thông tin như tổng tất cả các nút kết nối với một nút, các cạnh liền kề của đồ thị mạng)
- Sử dụng **phân cụm phân cấp** (hierarchical clustering) kết hợp với average pooling để xây dựng một cấu trúc mạng đa tầng từ dữ liệu ban đầu. Cụ thể, thực hiện lặp lại việc xây dựng cấu trúc tầng, mỗi tầng sẽ gom cụm đặc trưng dựa trên đồ thị ban đầu, với mỗi cụm, tính giá trị trung bình của các đặc trưng trong cụm đó trên tất cả các mẫu. [1]

Trong quá trình đánh giá và tinh chỉnh mô hình, nhóm so sánh và kết luận rằng **Ma trận kề** và **Hierarchical clustering** là hai phương pháp hiệu quả nhất trong

việc giữ lại thông tin cấu trúc của dữ liệu mạng. **DictVectorizer** kết quả thấp hơn so với hai phương pháp ban nãy vì một số thông tin quan trọng về mối quan hệ giữa các nút bị mất đi trong quá trình "nén" về vector.

Phương pháp	Mô hình	Average Accuracy	Mean Std
DictVectorizer	XGBoost	0.9058	0.0347
	Random Forest	0.9274	0.0292
Ma trận kề	XGBoost	0.9487	0.0035
	Random Forest	0.965	0.0032
Hierarchical clustering	XGBoost	0.9487	0.0035
	Random Forest	0.965	0.0032

Do đó, nhóm chọn hướng tiếp cận **Ma trận kề** và **Hierarchical clustering.**

2.5. Cân bằng dữ liệu

- Phương pháp SMOTENN:

SMOTE-ENN (Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors) là một phương pháp kết hợp giữa kỹ thuật over-sampling và under-sampling để xử lý vấn đề dữ liệu không cân bằng. Ý tưởng chính của SMOTE-ENN là sử dụng SMOTE để tăng số lượng mẫu của lớp thiểu số và sau đó sử dụng ENN để làm sạch dữ liệu bằng cách loại bỏ các mẫu không phù hợp. Đầu tiên, SMOTE tạo ra các mẫu tổng hợp cho lớp thiểu số bằng cách nội suy giữa các mẫu hiện có. Sau đó, phương pháp ENN được áp dụng để làm sạch tập dữ liệu kết quả. ENN loại bỏ các mẫu có nhãn khác với phần lớn nhãn trong số k hàng xóm gần nhất, đảm bảo rằng chỉ các mẫu rõ ràng và không nhiễu được giữ lại. Kết hợp giữa SMOTE và ENN giúp nâng cao chất lượng các mẫu từ lớp thiểu số trong khi đồng thời giảm thiểu các mẫu nhiễu hoặc gần biên quyết định, cải thiện tính khả thi và hiệu suất của mô hình học máy

2.6. Khai thác dữ liệu mạng

2.6.1. Dataset

Dữ liệu gốc gồm 35849 dòng, sau khi thực hiện SMOTEENN, số mẫu giảm

xuống còn 24496 mẫu do đã loại bỏ đi mẫu nhiễu trong quá trình under-sampling

Với 24,496 mẫu trong X sau khi lấy mẫu, chia dữ liệu thành 10 phần (folds) và sử dụng 10-fold cross validation để dánh giá

2.6.2. Hướng tiếp cận Machine Learning

- XGBoost

Thuật toán Extreme Gradient Boosting (XGBoost) là một thuật toán học máy dựa trên cây quyết định và học tăng cường. Cụ thể, XGBoost tạo ra các cây quyết định bằng cách sử dụng kỹ thuật gradient descent, bắt đầu với một ngưỡng mặc định và cập nhật các ngưỡng liên tục bằng cách giảm thiểu các sai số (residuals) trong quá trình xây dựng cây, trong đó sai số là sự chênh lệch giữa giá trị quan sát và giá trị dự đoán. Mỗi cây bắt đầu với một lá duy nhất và tất cả các sai số đều được đi đến lá đó [2]

- RandomForest

Random Forest là một phương pháp học máy mạnh mẽ, sử dụng lấy mẫu bootstrap để xây dựng nhiều cây quyết định độc lập. Mỗi cây được huấn luyện trên một tập con dữ liệu ngẫu nhiên và một tập con ngẫu nhiên các đặc trưng, dự đoán cuối cùng được quyết định bằng bầu chọn đa số từ tất cả các cây trong rừng.

- AdaBoost

Thuật toán AdaBoost (Adaptive Boosting) là một trong những kỹ thuật học tăng cường phổ biến nhất trong học máy, được thiết kế để cải thiện khả năng phân loại của các bộ phân loại yếu. AdaBoost bắt đầu bằng việc gán trọng số ban đầu cho từng mẫu trong tập huấn luyện. Các bộ phân loại yếu được huấn luyện trên tập này, đồng thời được điều chỉnh trọng số dựa trên kết quả dự đoán của từng bộ. Các bộ phân loại này thường là các mô hình đơn giản và dễ huấn luyện. Sau khi huấn luyện, mỗi bộ phân loại yếu được đánh giá dựa trên tỷ lệ lỗi, giúp xác định mức độ sai phân. Các mẫu được phân loại sai sẽ có trọng số tăng lên, trong khi các mẫu được phân loại đúng sẽ có trọng số giảm xuống, giúp bộ phân loại tập trung vào các mẫu khó phân loại hơn. Cuối cùng, các bộ phân loại yếu được kết hợp lại thành một bộ phân loại mạnh bằng cách tính tổng có trọng số của các dự đoán từ các bộ yếu. Bộ phân loại mạnh này được sử dụng để dự

đoán trên dữ liệu mới sau khi huấn luyện[3]

- SVM:

SVM (Support Vector Machine) là một phương pháp học có giám sát có khả năng phân loại cả dữ liệu tuyến tính và phi tuyến. Phương pháp này hoạt động bằng cách chuyển đổi dữ liệu huấn luyện gốc vào không gian N chiều (ứng với N đặc trưng) và xây dựng siêu phẳng trong không gian mới, được xác định thông qua tìm ra siêu phẳng có lề (margin) rộng nhất, tức là có khoảng cách tới các điểm của hai lớp là lớn nhất. để cung cấp khoảng cách lớn nhất giữa các lớp.

2.6.3. Hướng tiếp cận Deep Learning

- Sử dụng mô hình Feed Forward Neural Network (FNN) với cơ chế Attention Layer để xử lý và kết hợp hai cặp đặc trưng (drltl, sotchk) và (dtbhktruoc, namhoc) từ dữ liệu mạng.

- Sử dụng mô hình Graph Convolutional Network (GCN) với đầu vào là ma trận kề gồm các nút và danh sách cạnh liền kề, mô hình này sẽ lan truyền thông tin giữa các nút, qua đó, học được các đặc trưng từ cấu trúc đồ thị và dữ liệu liên quan.

2.6.4. Kết quả thực nghiệm

- Kịch bản thực nghiệm: Nhóm đề xuất kịch bản thực nghiệm tập trung vào việc phân chia dữ liệu theo thời gian, Dữ liệu học kỳ 1 được sử dụng làm đặc trưng đầu vào (features), trong khi dữ liệu học kỳ 2 được sử dụng làm nhãn mục tiêu (labels). Nhóm thực hiện mô phỏng tình huống thực tế, bằng cách sử dụng kết quả của học kỳ 1 là cơ sở để dự đoán kết quả học kỳ 2. Dữ liệu sau đó được chia thành hai phần: dữ liệu huấn luyện bao gồm các dòng có năm học nhỏ hơn 2021, và dữ liệu kiểm tra bao gồm các dòng có năm học từ 2021 trở đi.
- Đối với Machine Learning, nhóm sẽ tập trung vào hai chỉ số, accuracy – đánh gía độ chính xác của mô hình và precision - đảm bảo rằng mọi học sinh được phân loại chính xác trong nhóm của họ mà không có sự nhầm lẫn. Đây là kết quả trên tập test:

Thuật toán	Trước khi combine	Sau khi combine

	Accuracy	Precision		Accuracy	Precision	
XGBoost	95%	Class 0	97%	95%	Class 0	97%
		Class 1	91%		Class 1	91%
		Class 2	86%		Class 2	86%
		Class 3	93%		Class 3	93%
		Class 4	97%		Class 4	97%
RandomForest	96%	Class 0	97%	98%	Class 0	98%
		Class 1	93%		Class 1	96%
		Class 2	88%		Class 2	93%
		Class 3	93%		Class 3	96%
		Class 4	97%		Class 4	99%
AdaBoost	93%	Class 0	97%	94%	Class 0	96%
		Class 1	89%		Class 1	89%
		Class 2	85%		Class 2	85%
		Class 3	93%		Class 3	93%
		Class 4	97%		Class 4	97%
SVM	94%	Class 0	97%	86%	Class 0	93%
		Class 1	93%		Class 1	77%
		Class 2	87%		Class 2	64%
		Class 3	87%		Class 3	78%
		Class 4	95%		Class 4	89%

Nhận xét:

+ Có thể thấy, dữ liệu sau khi được combine giúp tăng hiệu suất của hầu hết các mô hình, đặc biệt là Random Forest và XGBoost. Vấn đề mất cân bằng dữ liệu chưa được giải quyết triệt để khi Class 2 thường có Precision thấp nhất.

- Để kiểm tra liệu mô hình có bị overfitting hay không, nhóm thực hiện k-fold cross validation:

Thuật toán	Trước khi combine		Sau khi combine	
	Average Accuracy	Mean Std	Average Accuracy	Mean Std
XGBoost	94.87%	0.0035	96.20%	0.0041
Random Forest	96.19%	0.0025	**98.04%**	0.0033
AdaBoost	93.32%	0.0059	94.86%	0.0048
SVM	94.29%	0.0046	94.03%	0.0037

Nhận xét:

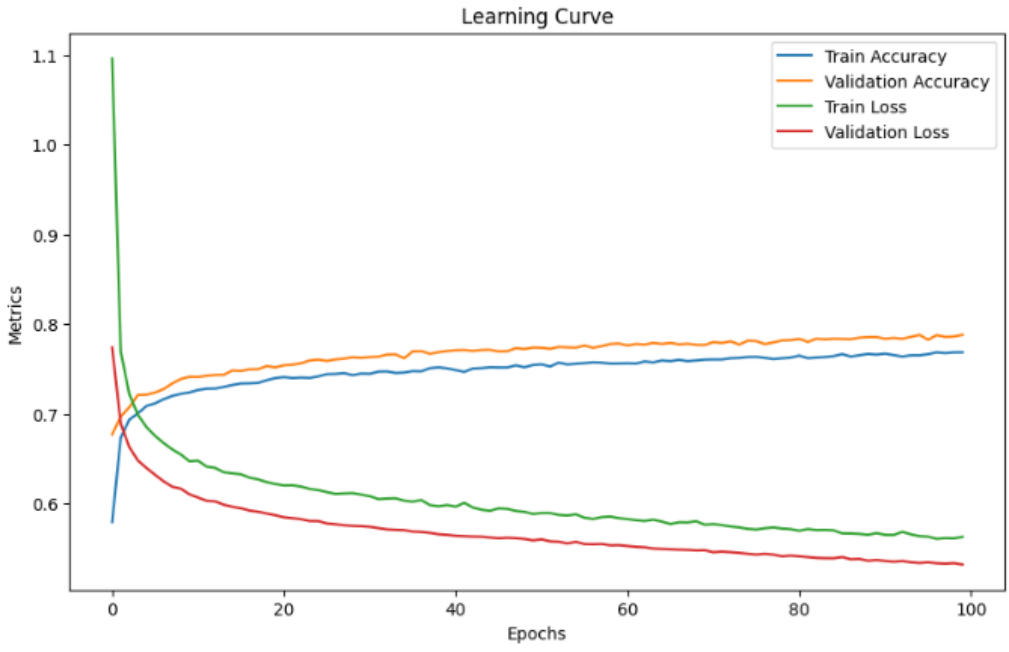
+ Nhìn chung, không có dấu hiệu overfitting đáng kể ở bất kỳ mô hình nào dựa trên độ lệch chuẩn thấp và sự cải thiện nhất quán của Average Accuracy trước và sau combine

+ Random Forest và XGBoost là hai thuật toán khai thác tốt nhất dữ liệu khi có độ chính xác cao và độ lệch chuẩn thấp và ổn định, tiếp theo là AdaBoost với hiệu suất tăng ít hơn.

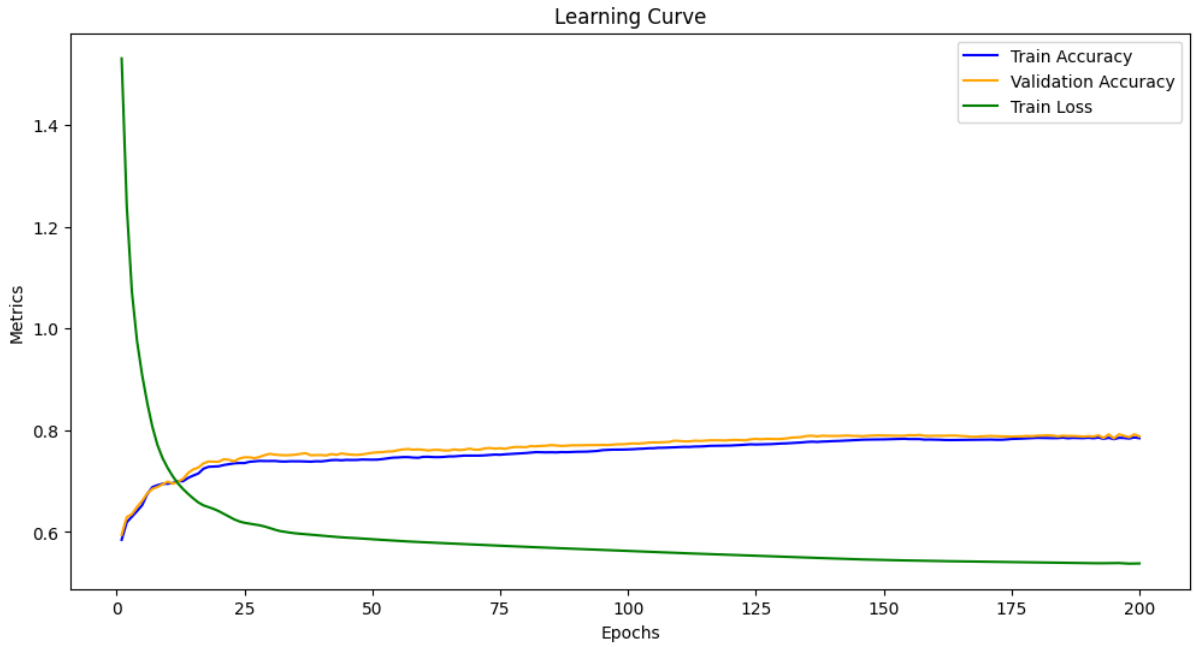
+ SVM có độ chính xác giảm, có thể là do thuật toán này chưa đủ hiệu quả so với các phương pháp học tăng cường trong việc khai thác cấu trúc và các đặc điểm phi tuyến phức tạp của dữ liệu mạng

- Đối với Deep Learning:

+ Feed Forward Neural Network



+ Graph Convolutional Network



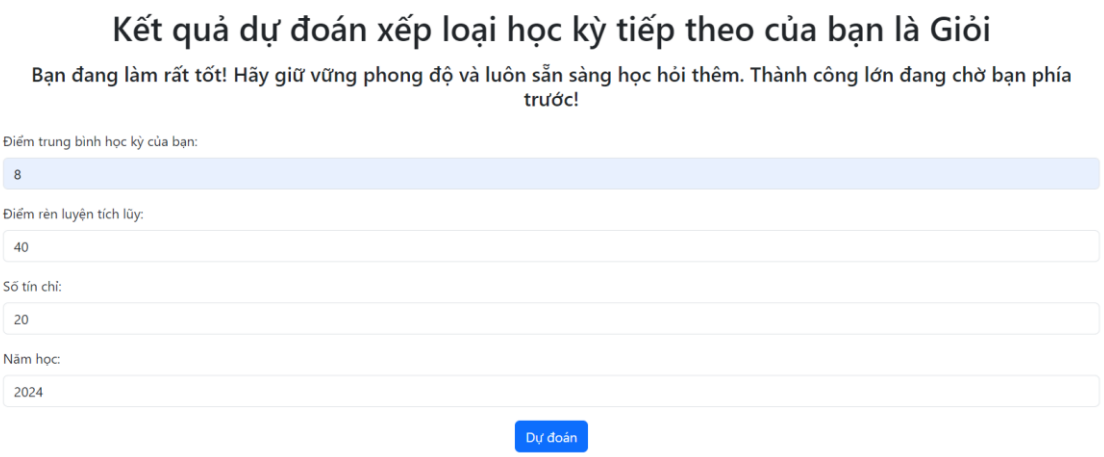
Nhận xét: Cả **Train Accuracy** và **Validation Accuracy** đều tăng nhanh ở các Epoch đầu và duy trì ổn định ở mức cao (Khoảng 79% ở GCN và 76% ở FNN) sau một số epoch (~50 epoch). Đây là dấu hiệu cho thấy mô hình học được tốt từ dữ liệu huấn luyện mà không có hiện tượng overfitting nghiêm trọng, do khoảng cách giữa hai độ chính xác này rất nhỏ. Đường **Validation Accuracy** gần như song song với đường **Train Accuracy** và không giảm ở các epoch sau. Điều này chứng tỏ mô hình không overfiting.

2.7. Chương trình demo

Đây là web giúp dự đoán kết quả học tập của sinh viên UIT trong kỳ học tiếp theo được xây dựng bằng Python (sử dụng Flask) Các yếu tố đầu vào bao gồm:

- Điểm trung bình học kỳ trước.
- Điểm rèn luyện tích lũy.
- Số tín chỉ đã hoàn thành.
- Năm học

Ứng dụng sẽ phân loại kết quả học tập của sinh viên thành các mức: Giỏi, Khá, Trung Bình và Yếu. Và cho những lời động viên tương ứng.


Kết quả dự đoán xếp loại học kỳ tiếp theo của bạn là Giỏi
Bạn đang làm rất tốt! Hãy giữ vững phong độ và luôn sẵn sàng học hỏi thêm. Thành công lớn đang chờ bạn phía trước!
Điểm trung bình học kỳ của bạn:
8
Điểm rèn luyện tích lũy:
40
Số tín chỉ:
20
Năm học:
2024
Dự đoán

CHƯƠNG 3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3.1. Kết luận

Kết quả từ mô hình dự đoán cho thấy độ chính xác cao trong việc dự đoán điểm số của sinh viên trong học kỳ tiếp theo. Nhờ vào việc sử dụng các thuật toán Machine Learning/Deep Learning, chúng em có thể dự đoán khá chính xác kết quả học tập của sinh viên dựa trên các yếu tố như điểm số học kỳ trước, số tín chỉ đã hoàn thành, và các yếu tố khác. Việc dự đoán này có thể giúp các giảng viên và trường học có chiến lược hỗ trợ kịp thời cho sinh viên.

3.2. Hướng phát triển

Trong tương lai, chúng em sẽ tiếp tục cải tiến mô hình dự đoán bằng cách thu thập thêm dữ liệu từ các nguồn khác như dữ liệu về hành vi học tập của sinh viên, các yếu tố ngoài học tập như sức khỏe hoặc các hoạt động ngoại khóa. Việc thử nghiệm với các mô hình học sâu phức tạp hơn có thể giúp nâng cao độ chính xác của dự đoán. Ngoài ra, ứng dụng của mô hình có thể được mở rộng để hỗ trợ các trường học trong việc lên kế hoạch giảng dạy và tư vấn cho sinh viên.

- **Cải thiện dữ liệu:** Có thể có những dữ liệu còn thiếu hoặc chưa đủ chi tiết. Việc thu thập thêm các dữ liệu mới sẽ giúp cải thiện độ chính xác.
- **Cải tiến mô hình:** Các mô hình hiện tại có thể chưa tối ưu hoàn toàn. Bạn có thể thử các thuật toán mới hoặc tinh chỉnh mô hình hiện tại.
- **Ứng dụng thực tiễn:** Đưa ra những ứng dụng thực tế của mô hình trong môi trường học thuật, chẳng hạn như việc dự đoán không chỉ điểm số mà còn các yếu tố như khả năng tốt nghiệp hay nhu cầu học thêm.
- **Khả năng mở rộng:** Có thể mở rộng mô hình cho nhiều trường học, nhiều ngành học khác nhau để tăng tính tổng quát và ứng dụng rộng rãi.

TÀI LIỆU THAM KHẢO

[1] J. Doe and J. Smith, " Feature Network Methods in Machine Learning and Applications," arXiv:2401.04874, 2024. Available: https://arxiv.org/abs/2401.04874

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, 2002, vol.16, pp. 321–357.

[3] Freund, Y, Schapire, R 1997, ‘A decision-theoretic generalization of on-line learning and an application to boosting’, J. Comput. Syst. Sci, 55:119–139.