

**Học viện Công nghệ Bưu chính viễn thông
Cơ sở tại thành phố Hồ Chí Minh**

Khoa: Công nghệ thông tin 2



Môn: KHAI PHÁ DỮ LIỆU
BÀI TẬP GIỮA KÌ

Đề tài:

Xây dựng kho dữ liệu dựa trên đặc điểm môn học, điểm số sinh viên đã học và khai phá kho dữ liệu gợi ý chọn chuyên ngành cho sinh viên

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Duy

Lớp: D21CQCNHT01-N

Nhóm: 5

Thành viên:

Nguyễn Thị Minh Thư - N21DCCN082

Nguyễn Thị Phương Thảo - N21DCCN078

Tô Phan Kiều Thương - N21DCCN184

Hồ Chí Minh, ngày 24 tháng 3 năm 2025

DANH MỤC HÌNH ẢNH

Hình 1: Thống kê cơ bản.....	16
Hình 2: Kiểm tra giá trị bị thiếu.....	16
Hình 3: Kiểm tra giá trị trùng lặp.....	17
Hình 4: Biểu đồ phân phối thuộc tính GPA của sinh viên	17
Hình 5: Biểu đồ phân phối thuộc tính điểm cuối kỳ của sinh viên.....	17
Hình 6: Biểu đồ phân phối thuộc tính điểm giữa kỳ của sinh viên	18
Hình 7: Biểu đồ phân phối thuộc tính điểm bài tập của sinh viên.....	18
Hình 8: Biểu đồ phân phối thuộc tính điểm chuyên cần của sinh viên	18
Hình 9: Biểu đồ phân phối xử lý capping GPA của sinh viên	19
Hình 10: Biểu đồ phân phối xử lý capping điểm cuối kỳ của sinh viên.....	19
Hình 11: Biểu đồ phân phối xử lý capping điểm giữa kỳ của sinh viên.....	19
Hình 12: Biểu đồ phân phối xử lý capping điểm bài tập của sinh viên.....	20
Hình 13: Biểu đồ phân phối xử lý capping điểm chuyên cần của sinh viên	20
Hình 14: Biểu đồ thể hiện phân bố chuyên ngành gợi ý.....	21
Hình 15: Tính điểm trung bình theo loại môn học	22
Hình 16: Kết quả điểm trung bình theo loại môn học của 5 sinh viên đầu danh sách	22
Hình 17: Biểu đồ phân phối điểm giữa 3 chuyên ngành theo loại môn học	23
Hình 18: Phân tích các kỹ năng trong bộ dữ liệu.....	23
Hình 19: Các kỹ năng xuất hiện trong bộ dữ liệu.....	24
Hình 20: Tổng hợp kỹ năng và lấy ra những kỹ năng phổ biến	24
Hình 21: Kỹ năng phổ biến ở các chuyên ngành trong bộ dữ liệu	25
Hình 22: Ma trận tương quan giữa các loại điểm	27
Hình 23: Biểu đồ phân phối F-value của các loại điểm.....	28
Hình 24: Phân phối Chi-square của các biến điểm.....	29
Hình 25: Biểu đồ phân phối tầm quan trọng tổng hợp của các biến điểm	30
Hình 26: Đoạn code xử lý nhóm bộ dữ liệu theo thuộc tính student_id và subject_code.....	32
Hình 27: Đoạn code xử lý điểm trung bình môn học theo loại môn học	34
Hình 28: Xây dựng vector kỹ năng với phân loại mức độ thành thạo.....	35
Hình 29: : Tính mức độ thành thạo dựa trên điểm trung bình	36
Hình 30: Tính phần trăm số lượng môn học theo phân loại	37
Hình 31: Gọi hàm để tính phần trăm số lượng theo từng nhóm chuyên ngành	38
Hình 32: Tính số lần học lại trung bình	38
Hình 33: Xử lý thuộc tính subject_type.....	39
Hình 34: Đoạn chương trình thực hiện chia bộ dữ liệu	40

Hình 35: Đoạn chương trình thực hiện việc chuẩn hóa dữ liệu.....	40
Hình 36: Đoạn chương trình định nghĩa các mô hình và tham số	41
Hình 37: Đoạn chương trình sử dụng GridSearchCV để tìm tham số tốt nhất cho mô hình.....	42
Hình 38: So sánh độ chính xác của các mô hình	44
Hình 39: Kết quả mô hình tốt nhất và chi tiết về các nhãn trong mô hình.....	45
Hình 40: Ma trận nhầm lẫn của mô hình	46
Hình 41: Biểu đồ thể hiện tầm quan trọng của các thuộc tính.....	48
Hình 42: Giao diện trang chủ của hệ thống gợi ý chuyên ngành	49
Hình 43: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (1)	50
Hình 44: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (2)	51
Hình 45: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (3)	52
Hình 46: Giao diện nhập thông tin điểm môn học của sinh viên	53
Hình 47: Giao diện kết quả dự đoán cho sinh viên.....	54

DANH MỤC BẢNG

Bảng 1: Các câu lệnh SQL phổ biến.....	9
Bảng 2: Các thành phần của Flask.....	10
Bảng 3 Tham số và kết quả tốt nhất trên các mô hình.....	44

MỤC LỤC

DANH MỤC HÌNH ẢNH	2
DANH MỤC BẢNG.....	4
MỤC LỤC	5
I. Giới thiệu đề tài	1
II. Tổng quan lý thuyết và khảo sát tài liệu.....	2
1. Khái niệm về kho dữ liệu, khai phá dữ liệu.....	2
1.1. Kho dữ liệu (Data Warehouse) trong giáo dục	2
1.2. Khai phá dữ liệu (Data Mining)	2
2. Các phương pháp	3
2.1. Các phương pháp dùng trong tiền xử lý dữ liệu	3
2.2. Các phương pháp dùng trong khai phá dữ liệu	4
2.3. Các phương pháp dùng trong huấn luyện mô hình	5
3. Công nghệ	8
3.1. Kho dữ liệu	8
3.2. Hệ thống.....	9
III. Xây dựng kho dữ liệu.....	11
1. Cấu trúc kho dữ liệu.....	11
2. Phân tích cấu trúc dữ liệu	14
2.1. Phân tích thông tin sinh viên	14
2.2. Phân tích thông tin môn học	14
2.3. Phân tích thông tin điểm số	14
2.4. Phân tích thông tin kỹ năng	15
2.5. Phân tích dữ liệu khuyến nghị:	15
IV. Khai phá bộ dữ liệu	15
1. Trực quan hóa và khai phá dữ liệu.....	15
1.1. Các thông số thống kê cơ bản.....	15
1.2. Kiểm tra giá trị bị thiếu:	16
1.3. Kiểm tra giá trị trùng lặp:	17

1.4. Kiểm tra giá trị ngoại lai.....	17
1.5. Phân tích phân bố chuyên ngành gọi ý (biến mục tiêu)	21
1.6. Phân tích điểm trung bình theo chuyên ngành	22
1.7. Phân tích kỹ năng của sinh viên	23
1.8. Phân tích tầm quan trọng của các biến điểm	27
2. Tiền xử lý và chuẩn hóa dữ liệu.....	31
2.1. Xử lý dữ liệu bị thiếu.....	31
2.2. Xử lý dữ liệu trùng.....	32
2.3. Xử lý các thuộc tính điểm.....	32
2.4. Xử lý thuộc tính subject_category, final_grade.....	33
2.5. Xử lý thuộc tính skill_list	34
2.6. Xử lý thuộc tính subject_name	36
2.7. Xử lý thuộc tính retake_count	38
2.8. Xử lý thuộc tính subject_type.....	38
V. Mô hình và thực nghiệm.....	39
1. Xây dựng mô hình	39
2. Thực nghiệm và phân tích.....	42
2.1. Trình bày kết quả	42
2.2. Đánh giá mô hình.....	45
3. Triển khai và Demo mô hình trên web	49
VI. Thảo luận	54
1. Ý nghĩa.....	54
2. Các yếu tố ảnh hưởng	55
3. Hạn chế và khuyến nghị cải tiến.....	55
VII. Kết luận	56
TÀI LIỆU THAM KHẢO.....	57

LỜI CẢM ƠN

Trong suốt quá trình thực hiện và hoàn thành đề án cơ sở, nhóm nghiên cứu luôn nhận được sự quan tâm, hướng dẫn tận tình từ các thầy cô, đặc biệt là từ thầy Nguyễn Ngọc Duy, giảng viên phụ trách môn Khai phá dữ liệu thuộc Khoa Công nghệ Thông tin. Nhóm xin gửi lời cảm ơn chân thành đến thầy, người đã tận tình chỉ bảo, định hướng và chia sẻ những kinh nghiệm quý báu, giúp nhóm có thể hoàn thành tốt nhiệm vụ của mình.

Nhóm cũng xin bày tỏ lòng biết ơn sâu sắc đến Ban Giám hiệu Học viện Công nghệ Bưu chính Viễn thông và Khoa Công nghệ Thông tin, nơi đã tạo điều kiện thuận lợi để nhóm được học tập, nghiên cứu trong một môi trường hiện đại với nguồn tài liệu phong phú và cập nhật. Những định hướng từ Khoa đã giúp nhóm có cơ hội tiếp cận và ứng dụng những kiến thức thực tiễn, đặc biệt trong lĩnh vực hệ thống thông tin quản lý.

Nhóm cũng xin gửi lời cảm ơn đến các bạn đồng môn đã luôn sẵn lòng chia sẻ, đóng góp ý kiến và hỗ trợ trong suốt quá trình làm việc. Dẫu đã cố gắng hết mình, nhưng bài đề án vẫn không tránh khỏi những thiếu sót. Nhóm mong nhận được góp ý từ các thầy cô và bạn bè để có thể hoàn thiện hơn, đồng thời học hỏi thêm nhiều bài học ý nghĩa cho chặng đường phía trước.

Một lần nữa, nhóm nghiên cứu xin chân thành cảm ơn!

I. Giới thiệu đề tài

Trong bối cảnh giáo dục hiện đại, việc định hướng nghề nghiệp và chọn lựa chuyên ngành học phù hợp đóng vai trò then chốt đối với sự phát triển bền vững của sinh viên cũng như nguồn nhân lực cho xã hội. Nhiều sinh viên gặp khó khăn trong việc lựa chọn chuyên ngành do thiếu thông tin tổng hợp về đặc điểm các môn học và kết quả học tập của mình. Chính vì vậy, đề tài “*Xây dựng kho dữ liệu về đặc điểm môn học và điểm số của sinh viên – Khai phá dữ liệu để gợi ý chọn chuyên ngành cho sinh viên*” được nghiên cứu nhằm giải quyết bài toán này.

Đề tài tập trung xây dựng một cơ sở dữ liệu toàn diện, trong đó lưu trữ các thông tin liên quan đến đặc điểm của các môn học (bao gồm lý thuyết, thực hành, thiết kế, kỹ thuật, công nghệ, ...) cùng với điểm số của sinh viên. Từ kho dữ liệu này, các kỹ thuật khai phá dữ liệu sẽ được áp dụng để phân tích, rút ra những mẫu số liệu, từ đó đưa ra các gợi ý định hướng chọn chuyên ngành học phù hợp với năng lực và sở thích của từng sinh viên. Qua đó, đề tài không chỉ giúp sinh viên có được cái nhìn tổng quan về năng lực bản thân mà còn hỗ trợ các nhà quản lý giáo dục trong công tác định hướng và xây dựng chương trình đào tạo.

Mục tiêu của đề tài bao gồm:

- **Xây dựng kho dữ liệu:** Tích hợp các thông tin về sinh viên, đặc điểm môn học (lý thuyết, kỹ thuật, công nghệ) và điểm số.
- **Khai phá dữ liệu:** Áp dụng các thuật toán khai phá (phân loại, phân nhóm, khai thác quy tắc kết hợp) để tìm ra các mẫu mối quan hệ giữa kết quả học tập và lựa chọn chuyên ngành.
- **Xây dựng hệ thống gợi ý:** Phát triển một hệ thống hỗ trợ định hướng chuyên ngành dựa trên kết quả phân tích, giúp sinh viên lựa chọn ngành học phù hợp. Ở đây tập trung phân tích 3 ngành chính: Công nghệ phần mềm, An toàn thông tin và Trí tuệ nhân tạo.

Như vậy, đề tài hứa hẹn mang lại giá trị thiết thực cho sinh viên và các nhà quản lý giáo dục, tạo điều kiện cho việc định hướng nghề nghiệp một cách khoa học và chính xác hơn.

II. Tổng quan lý thuyết và khảo sát tài liệu

1. Khái niệm về kho dữ liệu, khai phá dữ liệu

1.1. Kho dữ liệu (Data Warehouse) trong giáo dục

Khái niệm: Kho dữ liệu là một hệ thống lưu trữ tập trung, tích hợp dữ liệu từ nhiều nguồn khác nhau với mục tiêu phục vụ cho các *hoạt động phân tích* và *ra quyết định*. Trong bối cảnh giáo dục, kho dữ liệu giúp tổng hợp thông tin về sinh viên, các môn học, điểm số và các kỹ năng liên quan, từ đó tạo nên cơ sở dữ liệu có chất lượng để hỗ trợ việc phân tích và gợi ý chọn chuyên ngành.

Một kho dữ liệu hiệu quả thường bao gồm các thành phần như quá trình trích xuất, chuyển đổi và tải (ETL), cơ sở dữ liệu lưu trữ (thường là quan hệ hoặc NoSQL) và các công cụ truy vấn, phân tích dữ liệu. Đối với đề tài này, cấu trúc kho dữ liệu được xây dựng dựa trên bảng dữ liệu tích hợp các thông tin sinh viên, môn học, điểm số và kỹ năng, giúp xác định mối liên hệ giữa các yếu tố học tập với định hướng ngành nghề.

1.2. Khai phá dữ liệu (Data Mining)

Khái niệm: Khai phá dữ liệu là quá trình trích xuất các thông tin ẩn chứa trong dữ liệu thông qua việc áp dụng các thuật toán và kỹ thuật phân tích, nhằm phát hiện ra các mẫu, xu hướng và mối quan hệ hữu ích. Mục tiêu chính của khai phá dữ liệu trong đề tài là tìm ra các mối liên hệ giữa đặc điểm của môn học (như loại môn, số tín chỉ, tỷ lệ lý thuyết – thực hành, kỹ năng phát triển) với kết quả học tập của sinh viên (điểm số, số lần học lại, ...) để từ đó đưa ra các gợi ý chọn chuyên ngành phù hợp.

Các thuật toán phổ biến được ứng dụng trong khai phá dữ liệu bao gồm:

- Phân loại (Classification): Xác định lớp (chuyên ngành) mà sinh viên có khả năng phù hợp dựa trên các đặc trưng học tập.
- Phân nhóm (Clustering): Nhóm các sinh viên có đặc điểm học tập tương đồng nhằm nhận diện các nhóm học tập khác nhau.
- Quy tắc kết hợp (Association Rule Mining): Phân tích các mẫu kết hợp giữa các môn học, điểm số và kỹ năng nhằm phát hiện ra các mối quan hệ tiềm ẩn.

- Học máy (Machine Learning): Áp dụng các mô hình học máy nhằm dự đoán chuyên ngành phù hợp dựa trên dữ liệu quá khứ.

2. Các phương pháp

2.1. Các phương pháp dùng trong tiền xử lý dữ liệu

a. Phương pháp IQR (Interquartile Range) – Phát hiện giá trị ngoại lai

Mục tiêu: Xác định và xử lý các giá trị ngoại lai (outliers) trong dữ liệu, giúp tăng độ chính xác của mô hình.

Nguyên lý hoạt động:

- Q1 (Quartile 1): Phân vị thứ 25% – giá trị nhỏ hơn 25% dữ liệu.
- Q3 (Quartile 3): Phân vị thứ 75% – giá trị nhỏ hơn 75% dữ liệu.
- $IQR = Q3 - Q1$: Khoảng tứ phân vị.
- Một giá trị được xem là ngoại lai nếu:

$$< Q1 - 1.5 \times IQR \text{ hoặc } > Q3 + 1.5 \times IQR$$

Cách xử lý: Loại bỏ (removal) hoặc giới hạn (capping) các giá trị vượt ngưỡng trên hoặc dưới để đảm bảo phân bố dữ liệu hợp lý.

b. Phương pháp Capping (Giới hạn) trong xử lý giá trị ngoại lai

Định nghĩa: Phương pháp capping (hay còn gọi là winsorization) là kỹ thuật xử lý outliers bằng cách thay thế các giá trị ngoại lai bằng giá trị ngưỡng thay vì loại bỏ chúng hoàn toàn khỏi tập dữ liệu.

Nguyên lý hoạt động: Thay vì loại bỏ các giá trị ngoại lai, phương pháp capping giữ lại số lượng mẫu dữ liệu nhưng thay thế giá trị của outliers bằng một giá trị giới hạn

- Xác định ngưỡng trên và dưới:
 - Ngưỡng dưới: $Q1 - 1.5 \times IQR$
 - Ngưỡng trên: $Q3 + 1.5 \times IQR$
- Thay thế giá trị:
 - Các giá trị $<$ ngưỡng dưới được thay thế bằng giá trị ngưỡng dưới
 - Các giá trị $>$ ngưỡng trên được thay thế bằng giá trị ngưỡng trên

Ứng dụng trong đề tài: Phát hiện và xử lý giá trị ngoại lai trong dữ liệu. Giúp làm sạch dữ liệu, loại bỏ các giá trị bất thường để đảm bảo chất lượng đầu vào cho các mô hình phân tích.

2.2. Các phương pháp dùng trong khai phá dữ liệu

a. Phương pháp F-test (ANOVA một chiều – Analysis of Variance)

Mục tiêu: Kiểm tra xem giá trị trung bình của một đặc trưng liên tục có sự khác biệt đáng kể giữa các nhóm phân loại (chuyên ngành) hay không.

Nguyên lý hoạt động:

- $F\text{-value} = (\text{Phương sai giữa các nhóm}) / (\text{Phương sai trong nhóm})$
- F-value càng lớn \rightarrow sự khác biệt giữa các nhóm càng rõ rệt \rightarrow đặc trưng có khả năng phân loại cao.

Giải thích kết quả:

- Nếu F-value cao: Điều này cho thấy giá trị trung bình của biến phân tích có sự khác biệt rõ rệt giữa các nhóm. Nói cách khác, biến này có khả năng phân biệt tốt giữa các nhóm đối tượng, và là một biến quan trọng trong phân tích dữ liệu.
- Nếu F-value thấp: Cho thấy giá trị trung bình của biến phân tích gần tương đương ở tất cả các nhóm. Biến này không có đóng góp đáng kể trong việc phân biệt giữa các nhóm, do đó ít giá trị sử dụng trong mô hình phân tích hoặc phân loại.

Ứng dụng trong đề tài: Đánh giá mức độ khác biệt giữa các nhóm. Giúp xác định biến nào có khả năng phân biệt tốt giữa các nhóm đối tượng, từ đó chọn đặc trưng phù hợp cho mô hình phân loại.

b. Phương pháp Chi-square test (Kiểm định Chi bình phương cho biến rời rạc hóa)

Mục tiêu: Kiểm định mối quan hệ giữa hai biến rời rạc, thường dùng để đánh giá mức độ liên quan giữa đặc trưng và nhãn phân loại (chuyên ngành).

Nguyên lý hoạt động:

- Xây dựng bảng tần suất (contingency table) giữa các nhóm điểm rời rạc và chuyên ngành.

- So sánh tần suất quan sát thực tế với tần suất kỳ vọng nếu hai biến độc lập.
- Tính toán giá trị Chi-square theo công thức:

$$X^2 = \sum \frac{(O - E)^2}{E}$$

O: tần suất quan sát thực tế, E: tần suất kỳ vọng.

Giải thích kết quả:

- Nếu giá trị Chi-square lớn: Cho thấy có mối liên hệ chặt chẽ giữa hai biến rời rạc. Biến phân tích có khả năng phân biệt tốt giữa các nhóm, và do đó có giá trị cao trong phân loại hoặc dự đoán.
- Nếu giá trị Chi-square nhỏ: Cho thấy hai biến gần như độc lập, không có mối liên hệ đáng kể. Biến này không có ý nghĩa phân loại, ít đóng vai trò trong mô hình phân tích dữ liệu.

Ứng dụng trong đề tài: Kiểm tra mối liên hệ giữa các biến rời rạc. Giúp phát hiện sự liên quan giữa đặc trưng và biến mục tiêu, hỗ trợ trong việc chọn lọc đặc trưng có ý nghĩa thống kê cao.

2.3. Các phương pháp dùng trong huấn luyện mô hình

a. Phương pháp Logistic Regression

Logistic Regression là một thuật toán học có giám sát (supervised learning) được sử dụng phổ biến trong các bài toán phân loại (classification). Thuật toán không dự đoán giá trị liên tục như hồi quy tuyến tính mà dự đoán xác suất của một mẫu thuộc vào một lớp nhất định, sau đó dùng ngưỡng (thường là 0.5) để quyết định phân loại.

Thuật toán xây dựng một hàm tuyến tính kết hợp các đặc trưng đầu vào, sau đó dùng hàm sigmoid (logistic function) để chuyển đổi đầu ra thành xác suất:

$$P(Y = 1|X) = \frac{1}{1 + e^{-z}} \quad , \text{ trong đó } z = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n$$

Kết quả dự đoán là:

- $P \geq 0.5 \rightarrow \text{Lớp 1}$
- $P < 0.5 \rightarrow \text{Lớp 0}$

Trong bài toán nhiều lớp (Multinomial Logistic Regression), thuật toán mở rộng để dự đoán xác suất cho từng lớp và chọn lớp có xác suất cao nhất.

Logistic Regression thường được dùng trong:

- Phân loại nhị phân (có/không, đạt/trượt, phù hợp/không phù hợp)
- Dự đoán hành vi, xu hướng (ví dụ: chọn chuyên ngành, dự đoán rủi ro, ...)

Ưu điểm:

- Dễ hiểu, dễ triển khai.
- Cho kết quả dưới dạng xác suất.
- Hiệu quả với dữ liệu tuyến tính.

Nhược điểm:

- Không phù hợp với dữ liệu phi tuyến phức tạp.
- Dễ bị ảnh hưởng nếu các đặc trưng có quan hệ chặt chẽ (đa cộng tuyến).

b. Phương pháp K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) là một thuật toán học có giám sát (supervised learning) đơn giản nhưng hiệu quả, được sử dụng cho cả bài toán phân loại và hồi quy. Thuật toán này dựa trên giả định rằng các mẫu có đặc điểm tương tự sẽ có nhãn tương tự nhau.

Nguyên lý hoạt động: K-Nearest Neighbors hoạt động dựa trên giả định rằng các điểm dữ liệu gần nhau có đặc điểm tương tự. Khi cần dự đoán nhãn cho một mẫu mới, thuật toán sẽ tính khoảng cách giữa mẫu đó và tất cả các điểm trong tập huấn luyện, sau đó chọn ra K điểm gần nhất. Dựa vào đa số nhãn của K điểm này (hoặc trung bình giá trị đối với bài toán hồi quy), thuật toán quyết định nhãn cho mẫu mới.

Công thức tính khoảng cách:

Khoảng cách Euclidean giữa hai điểm $A(x_1, x_2)$ và $B(y_1, y_2)$ được tính bằng:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ngoài ra có thể sử dụng khoảng cách Manhattan:

$$d(A, B) = |x_2 - x_1| + |y_2 - y_1|$$

Ưu điểm:

- Dễ hiểu và dễ triển khai.
- Không cần giả định về phân phối dữ liệu.

- Linh hoạt với nhiều loại dữ liệu khác nhau.

Nhược điểm:

- Tính toán khoảng cách cho toàn bộ dữ liệu có thể chậm với tập dữ liệu lớn.
- Nhạy cảm với dữ liệu nhiễu và dữ liệu không cân bằng.
- Hiệu suất giảm khi số lượng đặc trưng quá nhiều (Curse of Dimensionality).

c. Phương pháp Random Forest

Random Forest là một thuật toán học có giám sát dựa trên phương pháp ensemble, trong đó sử dụng nhiều cây quyết định (decision trees) để đưa ra dự đoán chính xác và ổn định hơn. Thuật toán này xây dựng các cây quyết định từ các mẫu dữ liệu ngẫu nhiên khác nhau (bagging) và sử dụng một tập con đặc trưng ngẫu nhiên cho mỗi cây, giúp giảm hiện tượng overfitting.

Nguyên lý hoạt động: Random Forest tạo ra nhiều cây quyết định từ các mẫu dữ liệu và đặc trưng được lấy ngẫu nhiên, sau đó kết hợp các dự đoán của từng cây (thường bằng cách bỏ phiếu cho phân loại hoặc trung bình cho hồi quy) để cho ra kết quả cuối cùng.

Ưu điểm:

- Nhờ vào cơ chế ensemble, kết quả thường ổn định và chính xác.
- Việc kết hợp nhiều cây giúp giảm hiện tượng quá khớp (overfitting) của từng cây riêng lẻ.
- Hiệu quả với các bài toán phức tạp và tập dữ liệu lớn.

Nhược điểm:

- Tốn nhiều thời gian và tài nguyên khi huấn luyện với số lượng cây lớn.
- Do kết hợp nhiều cây, khó xác định ảnh hưởng của từng đặc trưng so với các mô hình đơn giản hơn.

d. Phương pháp Neural Network

Neural Network (Mạng nơ-ron nhân tạo) là một thuật toán học có giám sát được truyền cảm hứng từ cấu trúc của não người, sử dụng các lớp nơ-ron liên kết với nhau để học và biểu diễn các mối quan hệ phức tạp giữa các đặc trưng của dữ liệu.

Nguyên lý hoạt động: Mạng nơ-ron gồm nhiều lớp (input, hidden, output) với các nơ-ron được kết nối với nhau bằng các trọng số. Dữ liệu được đưa vào lớp đầu vào, sau đó lan truyền qua các lớp ẩn, nơi các hàm kích hoạt (activation functions) như

ReLU, sigmoid hay tanh được sử dụng để tạo ra các đặc trưng phi tuyến, cuối cùng kết quả được tính toán tại lớp đầu ra.

Ưu điểm:

- Phù hợp với các bài toán phức tạp như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên.
- Không cần phải thiết kế thủ công các đặc trưng.
- Có thể mở rộng với nhiều kiến trúc khác nhau (CNN, RNN, LSTM, v.v.).

Nhược Điểm:

- Huấn luyện mô hình sâu đòi hỏi thời gian và phần cứng mạnh.
- Mô hình thường được coi là "hộp đen" do khó xác định cụ thể vai trò của từng nơ-ron.
- Cần các biện pháp như dropout, regularization để tránh hiện tượng quá khớp.

3. Công nghệ

3.1. Kho dữ liệu

MySQL – Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System) là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, phổ biến và mạnh mẽ, sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) để quản lý dữ liệu.

Đặc điểm nổi bật:

- Tốc độ truy xuất dữ liệu nhanh, ổn định.
- Hỗ trợ các giao dịch (transaction) và toàn vẹn dữ liệu.
- Có thể tích hợp dễ dàng với các ứng dụng web.
- Được sử dụng bởi nhiều hệ thống lớn như WordPress, Facebook, Google.

Cấu trúc cơ bản:

- Database: Tập hợp các bảng (tables).
- Table: Lưu trữ dữ liệu dưới dạng hàng (rows) và cột (columns).
- Primary Key: Xác định duy nhất mỗi bản ghi trong bảng.
- Foreign Key: Tạo mối liên kết giữa các bảng.
- Index: Tăng tốc độ tìm kiếm và truy vấn dữ liệu.

Câu lệnh SQL phổ biến:

Câu lệnh	Mục đích
SELECT	Truy vấn dữ liệu
INSERT	Thêm dữ liệu
UPDATE	Cập nhật dữ liệu
DELETE	Xóa dữ liệu
JOIN	Kết hợp nhiều bảng
GROUP BY, ORDER BY	Gom nhóm, sắp xếp dữ liệu

Bảng 1: Các câu lệnh SQL phổ biến

Ứng dụng thực tế:

- Hệ thống quản lý thông tin sinh viên, nhân sự, sản phẩm, đơn hàng, ...
- Lưu trữ dữ liệu người dùng cho các ứng dụng web.
- Phân tích dữ liệu trong các hệ thống báo cáo.

3.2. Hệ thống

JavaScript – Ngôn ngữ lập trình phía client (Front-End) là một ngôn ngữ lập trình mạnh mẽ dùng để tạo các chức năng tương tác trên trang web. Đây là công nghệ cốt lõi của lập trình web hiện đại, cùng với HTML và CSS.

Đặc điểm nổi bật:

- Chạy trực tiếp trên trình duyệt (client-side).
- Thao tác và thay đổi nội dung HTML/CSS động.
- Tương tác thời gian thực với server thông qua Ajax / Fetch API.
- Hỗ trợ lập trình hướng đối tượng và lập trình hàm.

Các thành phần chính:

- DOM Manipulation: Thay đổi giao diện trực tiếp từ JS.
- Events Handling: Xử lý sự kiện như click, nhập liệu, cuộn chuột.
- AJAX / Fetch API: Giao tiếp bất đồng bộ với server.

- ES6+ Features:
 - let, const: Khai báo biến mới.
 - Arrow function: Cách viết hàm ngắn gọn.
 - Destructuring, Template literals, Spread operators, ...

Các thư viện / Framework phổ biến:

- jQuery: Đơn giản hóa thao tác DOM.
- ReactJS / VueJS / AngularJS: Xây dựng ứng dụng web động theo SPA (Single Page Application).

Ứng dụng thực tế:

- Giao diện tương tác người dùng (UX).
- Thực hiện các thao tác như xác thực dữ liệu, gửi form, hiển thị bảng dữ liệu.
- Giao tiếp với Backend mà không cần tải lại trang.

Flask – Web Framework nhẹ của Python là một micro-framework viết bằng Python dùng để xây dựng các ứng dụng web hoặc RESTful API. Mặc dù nhẹ, Flask vẫn đủ mạnh mẽ để xây dựng các hệ thống web hoàn chỉnh.

Đặc điểm nổi bật:

- Đơn giản, dễ học, linh hoạt.
- Hỗ trợ đầy đủ routing, template, form, session, middleware.
- Tích hợp dễ dàng với CSDL như MySQL, PostgreSQL, SQLite.
- Phù hợp cho cả dự án nhỏ và mở rộng thành ứng dụng lớn.

Thành phần chính:

Thành phần	Vai trò
app.route()	Định nghĩa đường dẫn URL và hàm xử lý
Jinja2	Template engine để render HTML động
request, session, redirect, url_for	Quản lý luồng dữ liệu từ phía người dùng
Flask-MySQL, SQLAlchemy	Kết nối và thao tác với CSDL

Bảng 2: Các thành phần của Flask

Xây dựng RESTful API:

Flask dễ dàng xây dựng API trả về JSON để phục vụ frontend hoặc mobile apps.

Ứng dụng thực tế:

- Xây dựng hệ thống quản lý dữ liệu nội bộ (dashboard, admin panel).
- Triển khai hệ thống phân tích dữ liệu.
- Cung cấp backend cho frontend như ReactJS hoặc mobile app.

III. Xây dựng kho dữ liệu

1. Cấu trúc kho dữ liệu

Kho dữ liệu của đề tài được xây dựng dựa trên bảng dữ liệu chính có tên ***student_major_recommendation***. Bảng này được thiết kế nhằm tích hợp đầy đủ các thông tin liên quan đến sinh viên, môn học, điểm số, kỹ năng phát triển và gợi ý chọn chuyên ngành. Cấu trúc bảng được định nghĩa theo các nhóm thông tin chính như sau:

Thông tin sinh viên:

- **student_id**: Định danh duy nhất của sinh viên, dùng để theo dõi dữ liệu của mỗi cá nhân.
- **student_name**: Tên của sinh viên, phục vụ cho việc hiển thị và định danh.
- **student_current_semester**: Học kỳ hiện tại, biểu thị số kỳ học mà sinh viên đã hoàn thành.
- **student_current_gpa**: Điểm trung bình hiện tại, là chỉ số tổng quát đánh giá năng lực học tập.

Thông tin môn học:

- **subject_name**: Tên môn học, giúp xác định môn học nào đang được phân tích.
- **subject_credits**: Số tín chỉ của môn học, thể hiện tầm quan trọng và khối lượng kiến thức.
- **subject_category**: Nhóm môn học, với các giá trị được định nghĩa là theory, technique, và tool nhằm phân loại theo tính chất của môn (lý thuyết, kỹ thuật, công nghệ).

- **subject_type**: Loại môn học được phân chia thành core (môn chuyên ngành) và general (môn cơ bản chung).
- **no_theory**: Số giờ lý thuyết của môn học.
- **no_practice**: Số giờ thực hành của môn học.
- Các phần trăm như **attendance_percentage**, **midterm_percentage**, **final_percentage**, **assignment_percentage**: Xác định tỷ lệ phần trăm trọng số của từng thành phần điểm (chuyên cần, giữa kỳ, cuối kỳ và bài tập) trong môn học.

Thông tin điểm số:

- **final_grade**: Điểm tổng kết cuối cùng của môn học.
- **midterm_grade**: Điểm giữa kỳ, cho biết khả năng tiếp thu kiến thức ban đầu.
- **attendance_grade**: Điểm chuyên cần, phản ánh sự tham gia của sinh viên trong quá trình học.
- **assignment_grade**: Điểm bài tập, thể hiện khả năng áp dụng kiến thức.
- **retake_count**: Số lần học lại, đánh giá mức độ khó khăn của môn học đối với sinh viên.

Thông tin kỹ năng:

- **skill_list**: Danh sách các kỹ năng chính mà môn học giúp phát triển. Ví dụ, với môn “Đại số” các kỹ năng như “Tư duy logic”, “Giải quyết vấn đề” và “Tư duy toán học” được liệt kê.

Dữ liệu khuyến nghị:

- **recommendation_major**: Chuyên ngành được gợi ý cho sinh viên. Các giá trị khả dĩ bao gồm “Công nghệ phần mềm”, “An toàn thông tin” và “Trí tuệ nhân tạo”. Đây là cột target dùng cho các mô hình học máy nhằm dự đoán lựa chọn chuyên ngành phù hợp.

Ngoài ra, lưu ý rằng đối với bộ dữ liệu tự sinh này một sinh viên có thể học 16 môn thuộc chương trình đào tạo trước khi phân chuyên ngành, với các môn học như: Đại số, Giải tích, Tiếng Anh, Xác suất thống kê, Xử lý tín hiệu số, Ngôn ngữ lập trình C++, Kỹ năng thuyết trình, Kỹ năng làm việc nhóm, Cấu trúc dữ liệu và giải thuật, Cơ sở dữ liệu, Lập trình hướng đối tượng, Nhập môn trí tuệ nhân tạo, Mạng máy

tính, An toàn hệ thống thông tin, Hệ điều hành (an toàn) và Nhập môn công nghệ phần mềm.

Lưu ý: Kho dữ liệu này là tự sinh, được tổng hợp từ nhiều nguồn khác nhau, với mục tiêu tạo ra một hệ thống dữ liệu đồng nhất phục vụ cho việc huấn luyện và triển khai mô hình gợi ý chuyên ngành.

Môn học cơ sở chung (áp dụng cho tất cả các chuyên ngành):

- Giải tích
- Đại số
- Xác suất thống kê
- Tiếng Anh
- Cơ sở dữ liệu
- Kỹ năng làm việc nhóm
- Kỹ năng thuyết trình

Chuyên ngành An toàn thông tin:

- An toàn và bảo mật hệ thống thông tin
- Mạng máy tính
- Hệ điều hành

Chuyên ngành Công nghệ phần mềm:

- Nhập môn công nghệ phần mềm
- Ngôn ngữ lập trình C++
- Lập trình hướng đối tượng
- Cấu trúc dữ liệu và giải thuật

Chuyên ngành Trí tuệ nhân tạo:

- Nhập môn trí tuệ nhân tạo
- Xác suất thống kê (có thể xem là bắt buộc ở đây)
- Xử lý tín hiệu số (*nếu chương trình đi theo hướng AI ứng dụng tín hiệu*)

2. Phân tích cấu trúc dữ liệu

2.1. Phân tích thông tin sinh viên

student_id: Là khóa chính của bảng, giúp phân biệt từng sinh viên. Việc sử dụng định danh duy nhất cho phép liên kết thông tin học tập của cùng một sinh viên qua nhiều môn học.

student_name, student_current_semester, student_current_gpa: Các thông tin này không chỉ phục vụ mục đích hiển thị mà còn đóng vai trò quan trọng trong việc đánh giá năng lực học tập tổng thể và khả năng định hướng chuyên ngành.

Ví dụ: số kỳ học đã hoàn thành và điểm trung bình (GPA) có thể ảnh hưởng đến việc dự đoán khả năng thành công ở các chuyên ngành yêu cầu kiến thức nền tảng mạnh.

2.2. Phân tích thông tin môn học

subject_name, subject_credits: Xác định tên và trọng lượng của môn học, cho thấy tầm quan trọng của môn trong chương trình đào tạo. Số tín chỉ càng cao thường đồng nghĩa với nội dung học tập phong phú và yêu cầu kiến thức chuyên sâu.

subject_category phân loại môn học theo tính chất: theory (tập trung vào lý thuyết), technique (kỹ thuật, cân bằng giữa lý thuyết và thực hành) và tool (công nghệ, thực hành nhiều hơn).

subject_type cho biết môn học thuộc nhóm core (môn chuyên ngành, có ảnh hưởng lớn tới định hướng chuyên môn) hay general (môn chung, cung cấp kiến thức nền tảng).

no_theory, no_practice và các tỷ lệ phần trăm: Đây là các chỉ số định lượng giúp đánh giá mức độ lý thuyết và thực hành trong từng môn học.

Ví dụ: môn “Tiếng Anh” có tỷ lệ lý thuyết cao (30%) và không có phần thực hành (0%), trong khi các môn kỹ thuật thường có tỷ lệ lý thuyết và thực hành cân bằng hơn.

2.3. Phân tích thông tin điểm số

Các cột **final_grade, midterm_grade, attendance_grade, assignment_grade** cung cấp thông tin chi tiết về hiệu quả học tập của sinh viên ở từng môn.

retake_count: Giúp xác định mức độ khó của môn học đối với sinh viên; số lần học lại cao có thể là chỉ số cho thấy môn học đòi hỏi kỹ năng hoặc kiến thức chuyên sâu.

2.4. Phân tích thông tin kỹ năng

skill_list: Lưu trữ danh sách các kỹ năng mà môn học hướng tới phát triển.

Đây là thông tin quan trọng trong việc liên kết kết quả học tập với nhu cầu của các chuyên ngành. Ví dụ, môn “Cấu trúc dữ liệu và giải thuật” giúp phát triển các kỹ năng “Lập trình”, “Tư duy thuật toán” và “Tối ưu hóa” rất cần thiết cho ngành Công nghệ phần mềm.

2.5. Phân tích dữ liệu khuyến nghị:

recommendation_major: Là cột target trong mô hình học máy.

Việc phân tích mối liên hệ giữa các thuộc tính như điểm số, số tín chỉ, tỷ lệ lý thuyết – thực hành, kỹ năng phát triển và các đặc điểm của sinh viên sẽ hỗ trợ trong việc đưa ra dự đoán về chuyên ngành phù hợp.

Các giá trị gợi ý bao gồm “Công nghệ phần mềm”, “An toàn thông tin” và “Trí tuệ nhân tạo”, phản ánh các hướng phát triển nghề nghiệp dựa trên kết quả học tập và năng lực của sinh viên.

Từ phân tích trên, cấu trúc dữ liệu được thiết kế có tính đồng nhất và tích hợp cao, tạo nền tảng vững chắc cho quá trình khai phá dữ liệu. Các thuộc tính được lựa chọn và phân loại rõ ràng giúp dễ dàng trong việc trích xuất đặc trưng, phát hiện mối quan hệ giữa các biến để đi đến việc xây dựng mô hình gợi ý chuyên ngành cho sinh viên.

IV. Khai phá bộ dữ liệu

1. Trực quan hóa và khai phá dữ liệu

1.1. Các thông số thống kê cơ bản

Dữ liệu được sử dụng trong đề tài bao gồm 93.532 quan sát (hàng) và 22 thuộc tính (cột), thể hiện thông tin của 6.684 sinh viên theo học từ 12 - 16 môn khác nhau. Mỗi quan sát đại diện cho một sự kết hợp “sinh viên – môn học”, trong đó ghi nhận đầy đủ dữ liệu về điểm số (giữa kỳ, cuối kỳ, bài tập, chuyên cần), số giờ lý thuyết và thực hành, tỉ lệ thành phần điểm cũng như số lần học lại.

```

: print(f"Shape của dataset: {df.shape}")
  print(f"Số lượng sinh viên: {df['student_id'].nunique()}")
  print(f"Số lượng môn học: {df['subject_code'].nunique()}")
  print(f"Số lượng chuyên ngành gợi ý: {df['recommendation_major'].nunique()}")
  print(f"Danh sách chuyên ngành: {df['recommendation_major'].unique()}")

Shape của dataset: (93532, 22)
Số lượng sinh viên: 6684
Số lượng môn học: 16
Số lượng chuyên ngành gợi ý: 3
Danh sách chuyên ngành: ['An toàn thông tin' 'Công nghệ phần mềm' 'Trí tuệ nhân tạo']

```

Hình 1: Thống kê cơ bản

Sử dụng thư viện pandas, ta có thể nạp dữ liệu vào một DataFrame và quan sát nhanh các chỉ số thống kê cơ bản bằng hàm describe(). Từ kết quả thống kê được hiển thị trong hình trên, có thể nhận thấy miền giá trị của các đặc trưng trải rộng và khá đa dạng. Hơn nữa, với nhiều thuộc tính khác nhau (như điểm số, giờ lý thuyết – thực hành, tỉ lệ thành phần điểm), việc phân tích sâu và thực hiện các bước tiền xử lý sẽ trở nên cần thiết để đảm bảo hiệu quả của các mô hình học máy ở giai đoạn tiếp theo.

1.2. Kiểm tra giá trị bị thiếu:

```
df.isnull().sum().sort_values(ascending=False)
```

student_id	0
student_name	0
skill_list	0
retake_count	0
attendance_grade	0
assignment_grade	0
midterm_grade	0
final_grade	0
assignment_percentage	0
final_percentage	0
midterm_percentage	0
attendance_percentage	0
no_practice	0
no_theory	0
subject_type	0
subject_category	0
subject_credits	0
subject_name	0
subject_code	0
student_current_gpa	0
student_current_semester	0
recommendation_major	0
dtype: int64	

Hình 2: Kiểm tra giá trị bị thiếu

Dựa vào kết quả trả về, ta thấy được bộ dữ liệu không có giá trị bị thiếu ở bất kỳ thuộc tính nào.

1.3. Kiểm tra giá trị trùng lặp:

```
df.duplicated().sum()
```

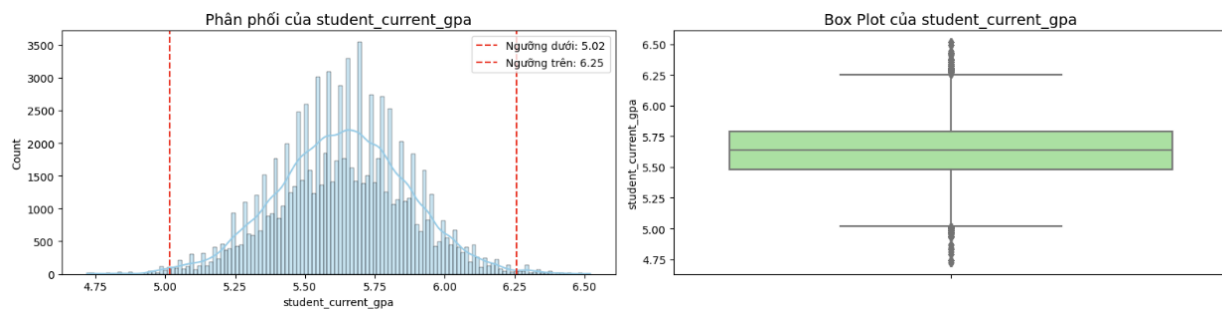
0

Hình 3: Kiểm tra giá trị trùng lặp

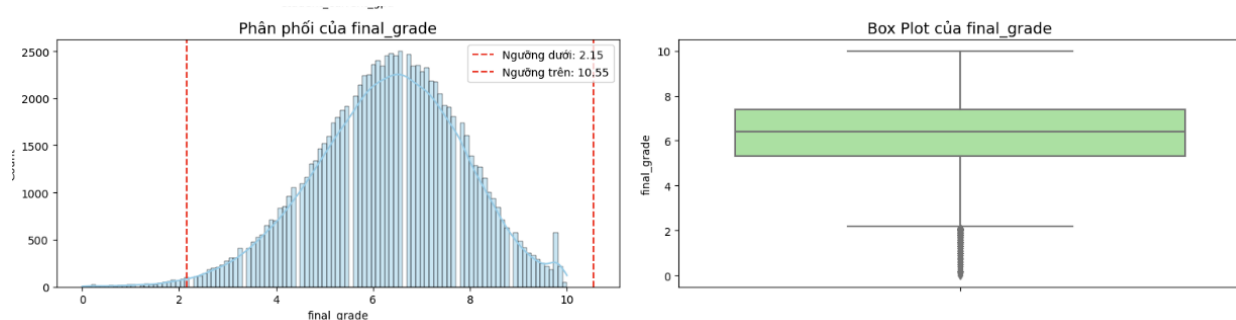
Ở bước này, có thể nhận xét bộ dữ liệu không có bất kỳ bản ghi nào trùng lặp.

1.4. Kiểm tra giá trị ngoại lai

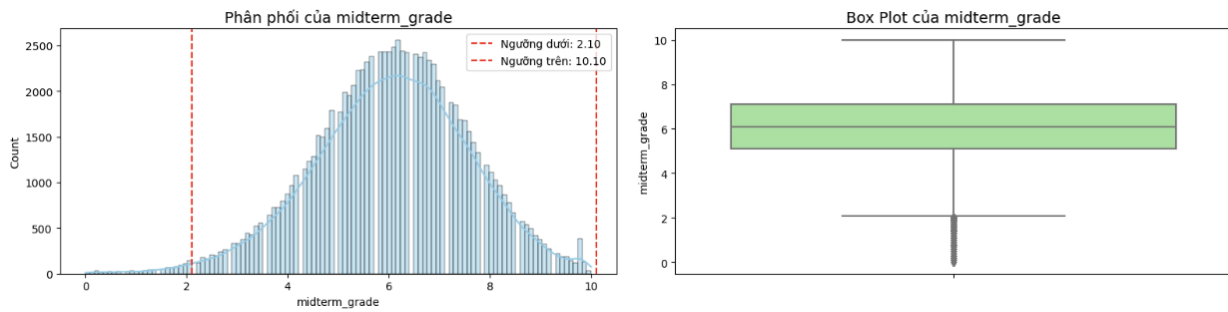
Ta tiến hành dùng phương pháp IQR (**Interquartile Range**) để xác định các giá trị ngoại lai cho các thuộc tính *student_current_gpa*, *final_grade*, *midterm_grade*, *assignment_grade*, *attendance_grade*.



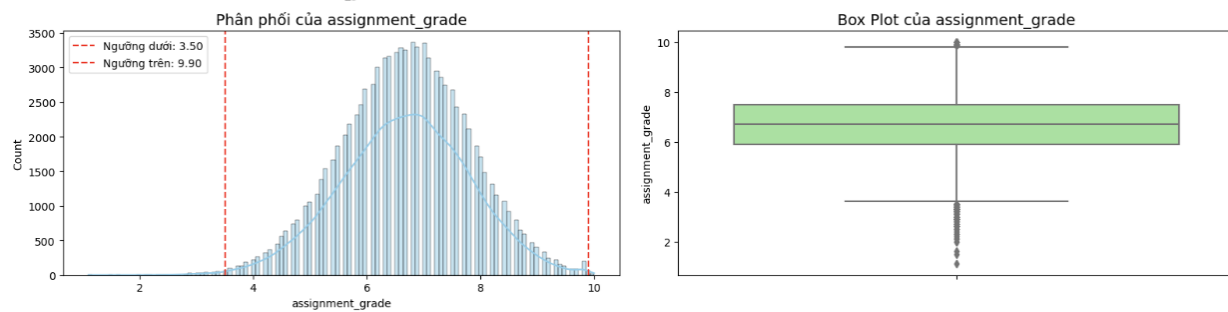
Hình 4: Biểu đồ phân phối thuộc tính GPA của sinh viên



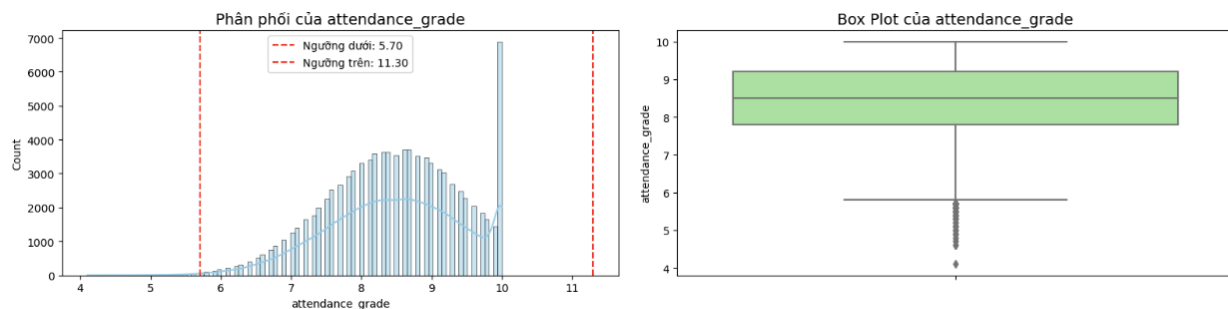
Hình 5: Biểu đồ phân phối thuộc tính điểm cuối kỳ của sinh viên



Hình 6: Biểu đồ phân phối thuộc tính điểm giữa kỳ của sinh viên



Hình 7: Biểu đồ phân phối thuộc tính điểm bài tập của sinh viên

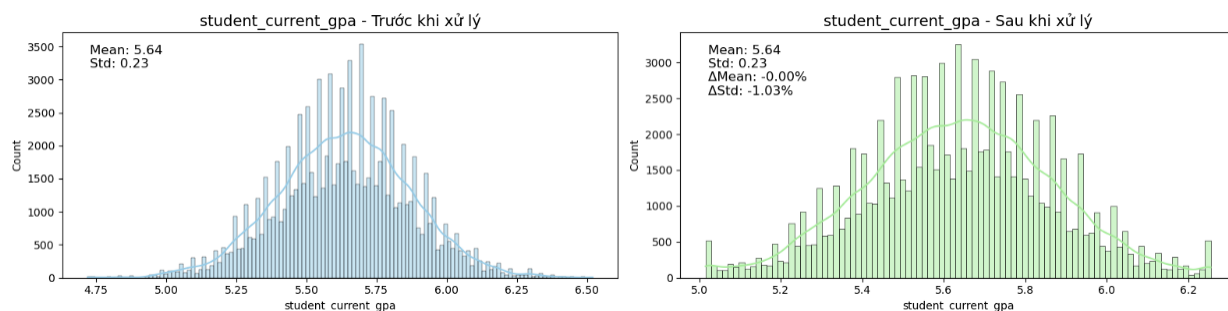


Hình 8: Biểu đồ phân phối thuộc tính điểm chuyên cần của sinh viên

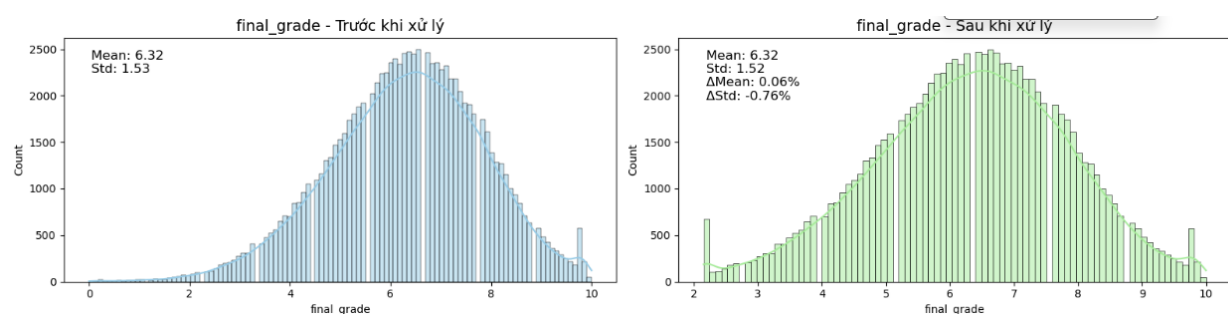
Sau khi sử dụng **IQR** để xác định giới hạn của các giá trị ngoại lai, biểu đồ cho thấy phần lớn các biến (điểm giữa kỳ, điểm bài tập, điểm cuối kỳ, GPA, điểm chuyên cần) tập trung trong khoảng giá trị phổ biến, song vẫn tồn tại một số quan sát vượt quá ngưỡng trên hoặc dưới. Cụ thể, một vài giá trị của **attendance_grade** cao hơn nhiều so với hầu hết dữ liệu còn lại, trong khi số ít giá trị khác lại thấp hơn đáng kể.

Tương tự, **midterm_grade** và **assignment_grade** cũng xuất hiện một số điểm bất thường. Kết quả này khẳng định sự hiện diện của ngoại lai trong tập dữ liệu và gợi ý rằng cần có bước xử lý hoặc xem xét phù hợp (loại bỏ, giới hạn, hoặc chuyển đổi giá trị) trước khi tiến hành bước tiếp theo.

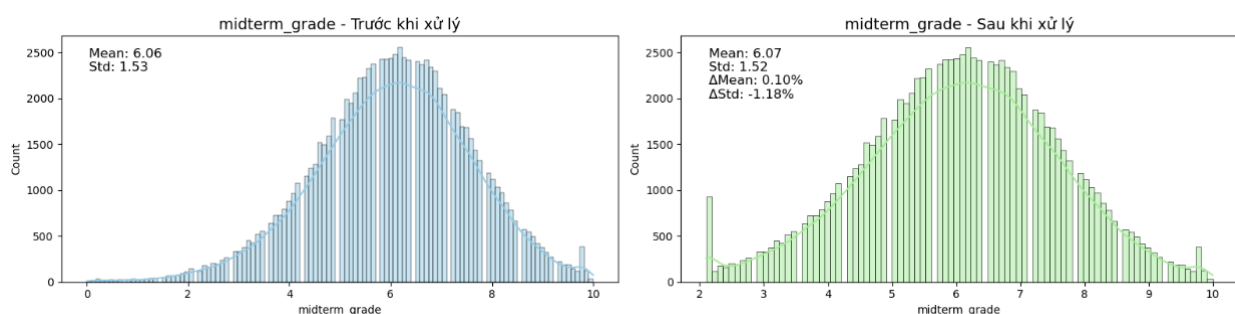
Vì thế, để xử lý các giá trị ngoại lai còn lại, ta áp dụng **phương pháp cắt bỏ (capping)**, tức là giới hạn (cap) các quan sát vượt quá một ngưỡng nhất định về đúng ngưỡng đó.



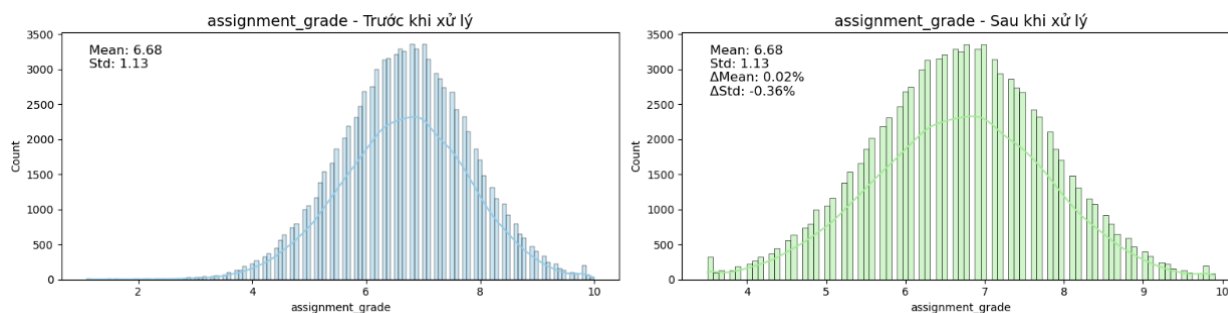
Hình 9: Biểu đồ phân phối xử lý capping GPA của sinh viên



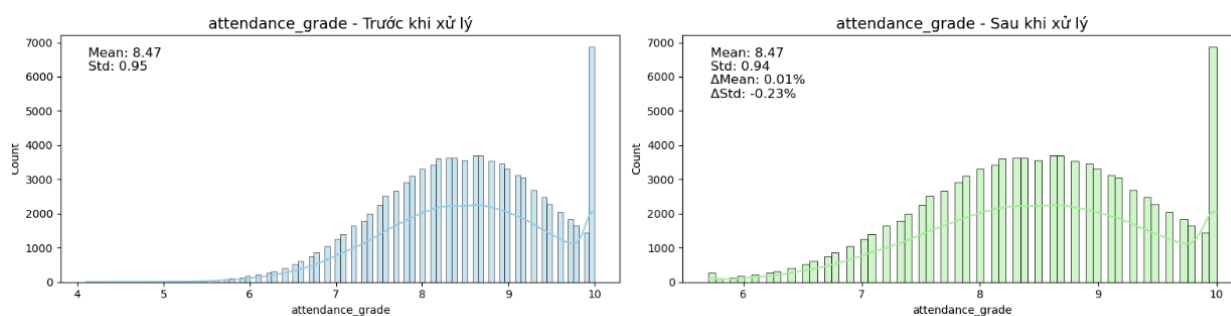
Hình 10: Biểu đồ phân phối xử lý capping điểm cuối kỳ của sinh viên



Hình 11: Biểu đồ phân phối xử lý capping điểm giữa kỳ của sinh viên



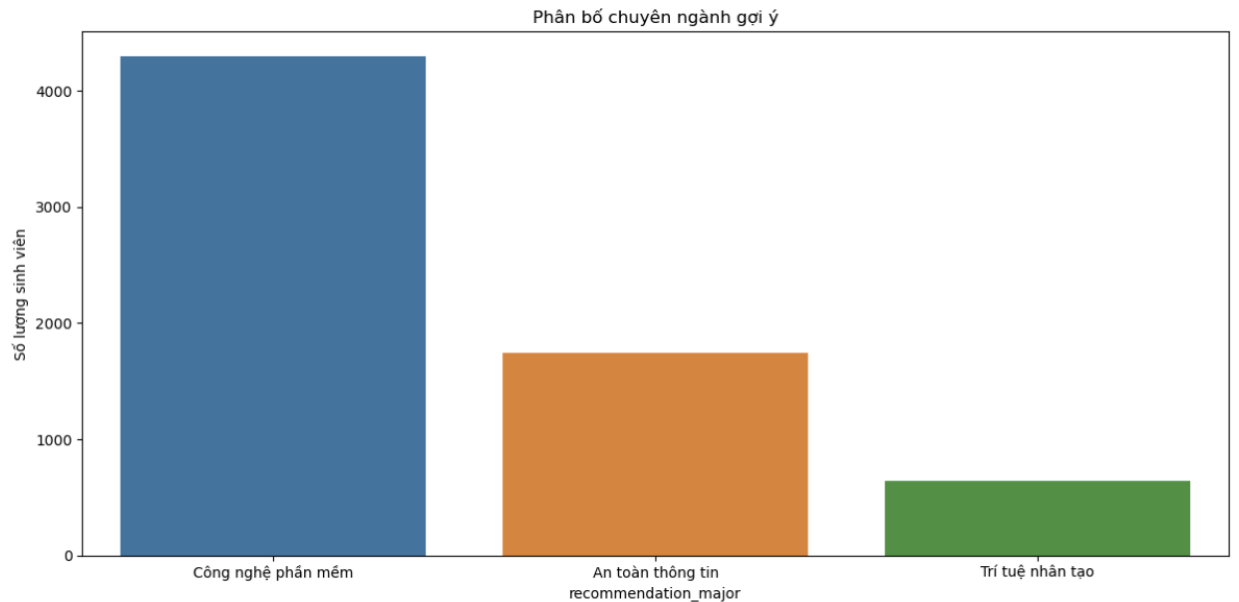
Hình 12: Biểu đồ phân phối xử lý capping điểm bài tập của sinh viên



Hình 13: Biểu đồ phân phối xử lý capping điểm chuyên cần của sinh viên

Sau khi áp dụng capping, kết quả cho thấy phân bố của các đặc trưng (chẳng hạn **student_current_gpa**, **final_grade**, **midterm_grade**, **assignment_grade**, **attendance_grade**) trở nên ổn định hơn. Cụ thể, độ lệch chuẩn của một số đặc trưng giảm nhẹ, phản ánh việc loại bớt những điểm quá cao hoặc quá thấp. Bên cạnh đó, giá trị trung bình ít bị kéo về phía các ngoại lai, giúp phân bố dữ liệu sát với thực tế hơn.

1.5. Phân tích phân bố chuyên ngành gợi ý (biến mục tiêu)



Hình 14: Biểu đồ thể hiện phân bố chuyên ngành gợi ý

Từ biểu đồ, có thể thấy được chuyên ngành Công nghệ phần mềm chiếm số lượng sinh viên áp đảo nhất (khoảng 4200 sinh viên), gấp hơn 2 lần so với chuyên ngành An toàn thông tin (khoảng 1750 sinh viên). Trí tuệ nhân tạo có số lượng sinh viên ít nhất, chỉ khoảng 600 sinh viên.

Sự chênh lệch lớn này cho thấy Công nghệ phần mềm đang là chuyên ngành được gợi ý nhiều nhất trong hệ thống, trong khi Trí tuệ nhân tạo có thể là chuyên ngành mới hoặc có tiêu chí gợi ý khắt khe hơn so với hai chuyên ngành còn lại.

Có một số dự đoán về nguyên nhân có thể giải thích tại sao số lượng sinh viên có sự phân hóa đối với từng ngành trên như sau:

Thứ nhất, Công nghệ phần mềm đang có nhu cầu tuyển dụng cao, nhiều cơ hội việc làm, và chương trình đào tạo thường tập trung vào các môn lập trình, phát triển hệ thống, khiến số sinh viên được gợi ý chuyên ngành này áp đảo.

Thứ hai, An toàn thông tin đòi hỏi kiến thức chuyên sâu về bảo mật, mã hóa và kỹ năng phân tích rủi ro, nên số lượng sinh viên đáp ứng đủ yêu cầu hoặc có hứng thú theo đuổi ít hơn.

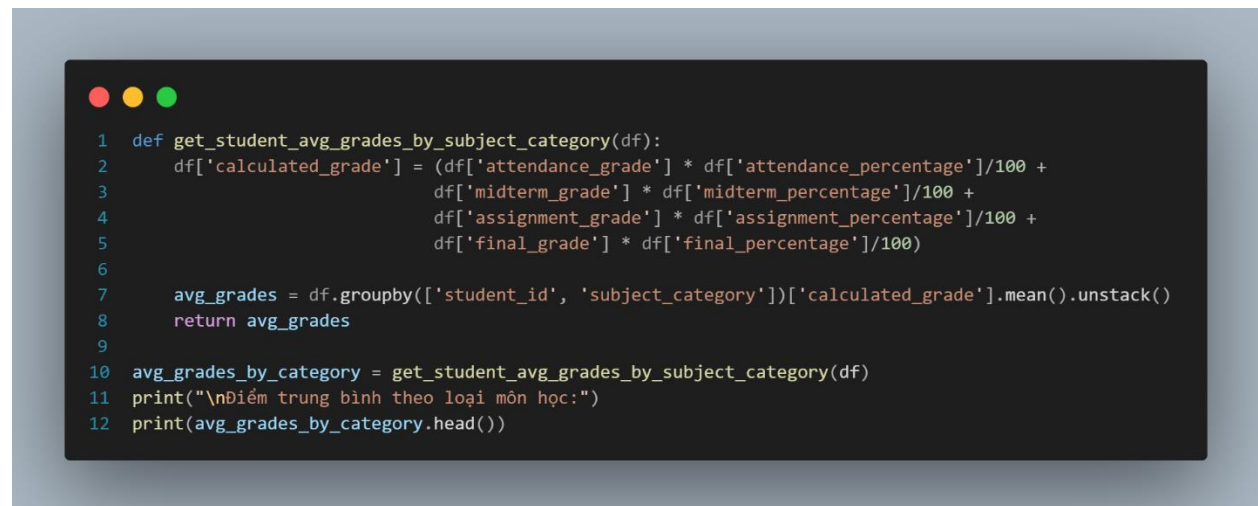
Thứ ba, Trí tuệ nhân tạo là lĩnh vực mới, thường gắn với các môn học phức tạp về toán, thống kê và học máy, khiến số sinh viên được gợi ý ngành này trở nên giới hạn.

Cuối cùng, sở thích và định hướng nghề nghiệp cá nhân cũng góp phần định hình tỷ lệ phân bố giữa ba chuyên ngành, dẫn đến sự chênh lệch trong dữ liệu thu thập.

Để tìm hiểu kỹ hơn về sự phụ thuộc của các thuộc tính khác đối với biến mục tiêu, ta tiếp tục tiến hành phân tích sâu hơn về từng nhóm thuộc tính.

1.6. Phân tích điểm trung bình theo chuyên ngành

Ta tiến hành tính toán điểm trung bình của sinh viên theo loại môn học (lý thuyết, kỹ thuật, công nghệ)



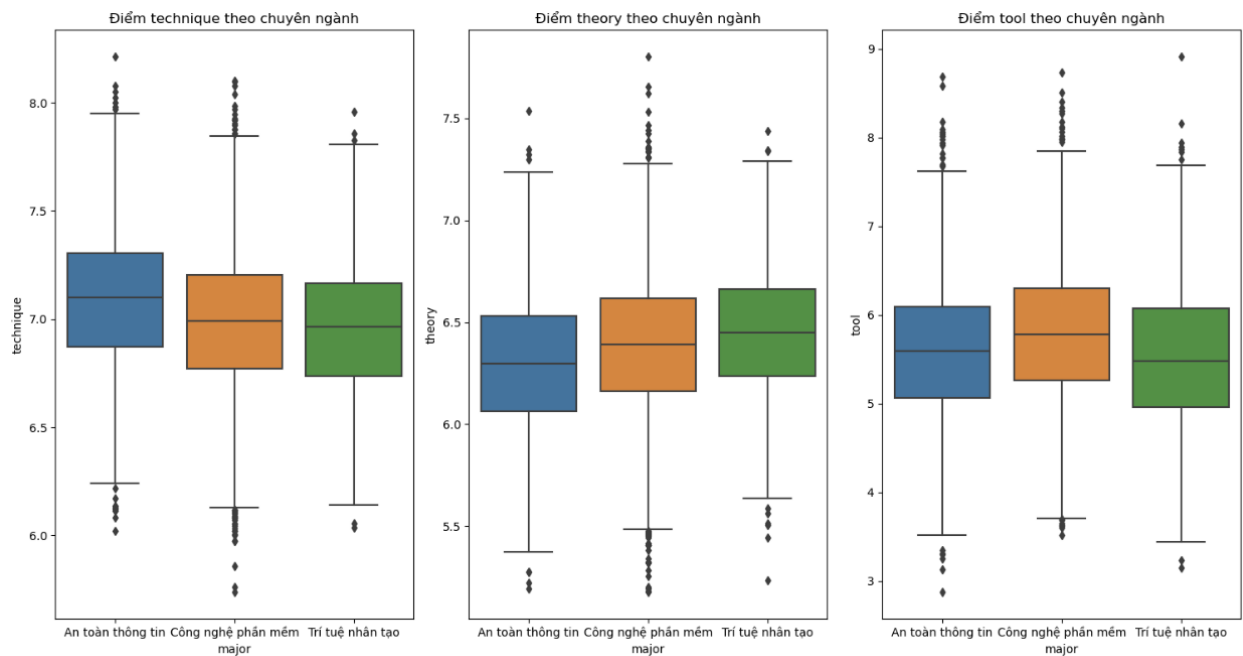
```
1 def get_student_avg_grades_by_subject_category(df):
2     df['calculated_grade'] = (df['attendance_grade'] * df['attendance_percentage']/100 +
3                             df['midterm_grade'] * df['midterm_percentage']/100 +
4                             df['assignment_grade'] * df['assignment_percentage']/100 +
5                             df['final_grade'] * df['final_percentage']/100)
6
7     avg_grades = df.groupby(['student_id', 'subject_category'])['calculated_grade'].mean().unstack()
8     return avg_grades
9
10 avg_grades_by_category = get_student_avg_grades_by_subject_category(df)
11 print("\nĐiểm trung bình theo loại môn học:")
12 print(avg_grades_by_category.head())
```

Hình 15: Tính điểm trung bình theo loại môn học

```
Điểm trung bình theo loại môn học:
subject_category  technique    theory  tool
student_id
10000              6.860000    6.130714  4.32
10001              6.832500    6.345000  4.47
10002              7.108333    6.413333  5.23
10003              6.613333    6.481429  5.61
10004              6.311667    6.301250  4.69
```

Hình 16: Kết quả điểm trung bình theo loại môn học của 5 sinh viên đầu danh sách

Sau khi tính được điểm trung bình cho từng loại môn học (lý thuyết, kỹ thuật, công nghệ), ta tiến hành so sánh phân phối điểm giữa ba chuyên ngành được gợi ý.



Hình 17: Biểu đồ phân phối điểm giữa 3 chuyên ngành theo loại môn học

Nhìn vào 3 biểu đồ Box Plot, có thể thấy **Trí tuệ nhân tạo** nổi trội ở nhóm môn lý thuyết, với mức điểm nhỉnh hơn so với hai chuyên ngành còn lại. **Công nghệ phần mềm** thể hiện thế mạnh ở nhóm môn công nghệ (tool), đồng thời đạt kết quả kỹ thuật (technique) tương đối cao. Trong khi đó, **An toàn thông tin** lại xếp sau hai chuyên ngành kia ở cả hai nhóm môn công nghệ và kỹ thuật, cho thấy mức độ tiếp cận thực hành và ứng dụng của nhóm sinh viên này còn hạn chế hơn.

1.7. Phân tích kỹ năng của sinh viên

```

1  vectorizer = CountVectorizer(token_pattern=r'^\s+')
2
3  all_skills = []
4  for skills in df['skill_list'].dropna().unique():
5      all_skills.extend([skill.strip() for skill in skills.split(',')])
6  unique_skills = sorted(set(all_skills))
7
8  print(f"\nTổng số kỹ năng duy nhất: {len(unique_skills)}")
9  print("Một số kỹ năng phổ biến:")
10 print(unique_skills[:20])

```

Hình 18: Phân tích các kỹ năng trong bộ dữ liệu

Ta tiến hành tổng hợp tất cả những kỹ năng mà sinh viên được học từ môn học, cho ra danh sách kỹ năng từ bộ dữ liệu gốc. Có tất cả 37 kỹ năng.

Tổng số kỹ năng duy nhất: 37

Một số kỹ năng phổ biến:

['Bảo mật', 'Giao thức mạng', 'Giao tiếp', 'Giải quyết vấn đề', 'Kiểm thử', 'Làm việc nhóm', 'Lập trình', 'Lập trình OOP', 'Lập trình hệ thống', 'Machine Learning', 'Mã hóa', 'Mô hình hóa', 'Ngoại ngữ', 'Phân tích', 'Phân tích dữ liệu', 'Phân tích rủi ro', 'Phân tích tín hiệu', 'Process', 'Quy trình phát triển', 'Quản lý dữ liệu']

Hình 19: Các kỹ năng xuất hiện trong bộ dữ liệu

Sau khi đã tổng hợp được tất cả kỹ năng, ta tiến hành phân tích kỹ năng theo chuyên ngành.

```
1 def get_skills_by_major(df):
2     skills_by_major = {}
3
4     for major in df['recommendation_major'].unique():
5         major_skills = []
6         for skills in df[df['recommendation_major'] == major]['skill_list'].dropna():
7             major_skills.extend([skill.strip() for skill in skills.split(',')])
8
9         skill_counts = pd.Series(major_skills).value_counts()
10        skills_by_major[major] = skill_counts
11
12    return skills_by_major
13
14 skills_by_major = get_skills_by_major(df)
15
16 for major, skills in skills_by_major.items():
17     print(f"\nTop 10 kỹ năng phổ biến cho chuyên ngành {major}:")
18     print(skills.head(10))
```

Hình 20: Tổng hợp kỹ năng và lấy ra những kỹ năng phổ biến

Top 10 kỹ năng phổ biến cho chuyên ngành An toàn thông tin:

Giao tiếp	4484
Giải quyết vấn đề	4438
Lập trình	4135
Bảo mật	3472
Tư duy toán học	3046
Thuyết trình	2991
Tư duy logic	2916
Xử lý dữ liệu	2802
Phân tích rủi ro	1745
Mã hóa	1745

Name: count, dtype: int64

Top 10 kỹ năng phổ biến cho chuyên ngành Công nghệ phần mềm:

Lập trình	11801
Giải quyết vấn đề	11406
Giao tiếp	11304
Xử lý dữ liệu	7673
Tư duy logic	7665
Thuyết trình	7539
Tư duy toán học	7480
Bảo mật	7259
Kiểm thử	4258
Quản lý dự án	4258

Name: count, dtype: int64

Top 10 kỹ năng phổ biến cho chuyên ngành Trí tuệ nhân tạo:

Giao tiếp	1667
Giải quyết vấn đề	1638
Lập trình	1609
Xử lý dữ liệu	1236
Tư duy toán học	1127
Thuyết trình	1105
Tư duy logic	1083
Bảo mật	968
Tư duy phân tích	642
Machine Learning	642

Name: count, dtype: int64

Hình 21: Kỹ năng phổ biến ở các chuyên ngành trong bộ dữ liệu

Dựa trên kết quả tách và đếm tần suất xuất hiện của các kỹ năng trong danh sách môn học, có thể thấy cả 3 chuyên ngành đều đề cao các kỹ năng nền tảng như **Giao tiếp**, **Giải quyết vấn đề**, và **Lập trình** – đây là bộ ba kỹ năng phổ biến nhất, đồng thời kỹ năng mềm như **Thuyết trình** và **Tư duy logic** cũng xuất hiện thường xuyên. Điều này phản ánh yêu cầu chung của lĩnh vực CNTT về khả năng làm việc nhóm, trao đổi thông tin và tư duy giải quyết vấn đề.

Bên cạnh đó, mỗi chuyên ngành vẫn có những kỹ năng đặc trưng riêng:

- **An toàn thông tin** nổi bật với kỹ năng **Bảo mật**, **Phân tích rủi ro** và **Mã hóa**, xếp cao trong top 10 kỹ năng phổ biến của ngành này nhưng lại ít xuất hiện ở các ngành khác.

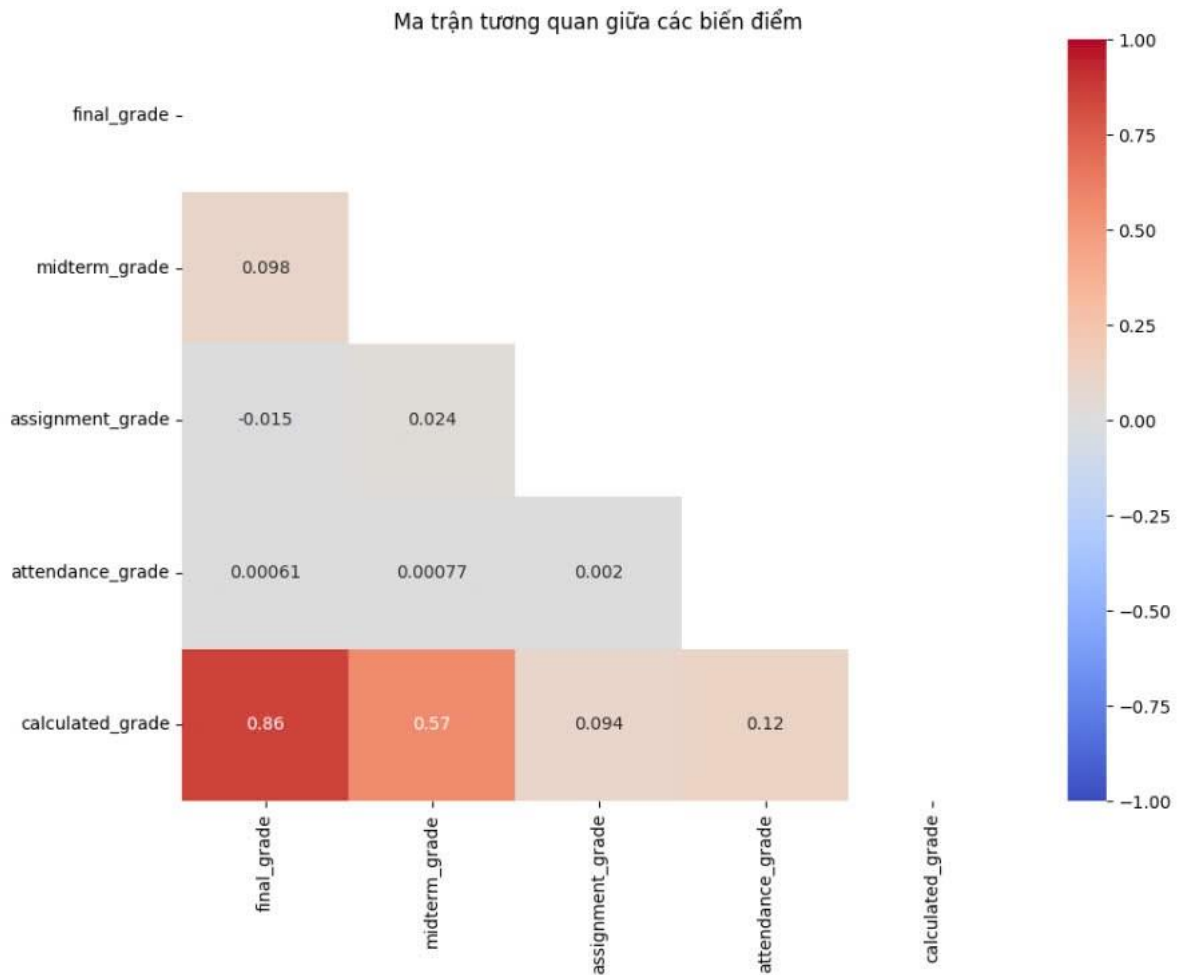
- **Công nghệ phần mềm** có thế mạnh về **Kiểm thử**, **Quản lý dự án** và một số kỹ năng liên quan đến phát triển quy mô lớn, đồng thời sở hữu số lượng sinh viên có kỹ năng **Lập trình** nhiều nhất.
- **Trí tuệ nhân tạo** đòi hỏi kiến thức chuyên sâu về **Machine Learning** và **Tư duy phân tích**, phù hợp với các môn học liên quan đến xử lý dữ liệu và xây dựng mô hình.

Về mức độ phổ biến, **Công nghệ phần mềm** có số lượng kỹ năng được đề cập lớn nhất, thể hiện quy mô của ngành trong dữ liệu. Mặc dù kỹ năng **Bảo mật** đứng thứ 4 trong An toàn thông tin, số tuyệt đối sinh viên có kỹ năng này ở Công nghệ phần mềm vẫn cao hơn do tổng lượng sinh viên lớn.

Cuối cùng, tầm quan trọng của **kỹ năng mềm** (đặc biệt là **Giao tiếp** và **Thuyết trình**) cũng được khẳng định, khi chúng nằm trong nhóm đầu ở cả ba chuyên ngành, cho thấy vai trò không thể thiếu của năng lực giao tiếp trong bất kỳ lĩnh vực kỹ thuật nào.

1.8. Phân tích tầm quan trọng của các biến điểm

a. Phân tích tương quan giữa các biến điểm



Hình 22: Ma trận tương quan giữa các loại điểm

Dựa trên ma trận tương quan, có thể nhận thấy:

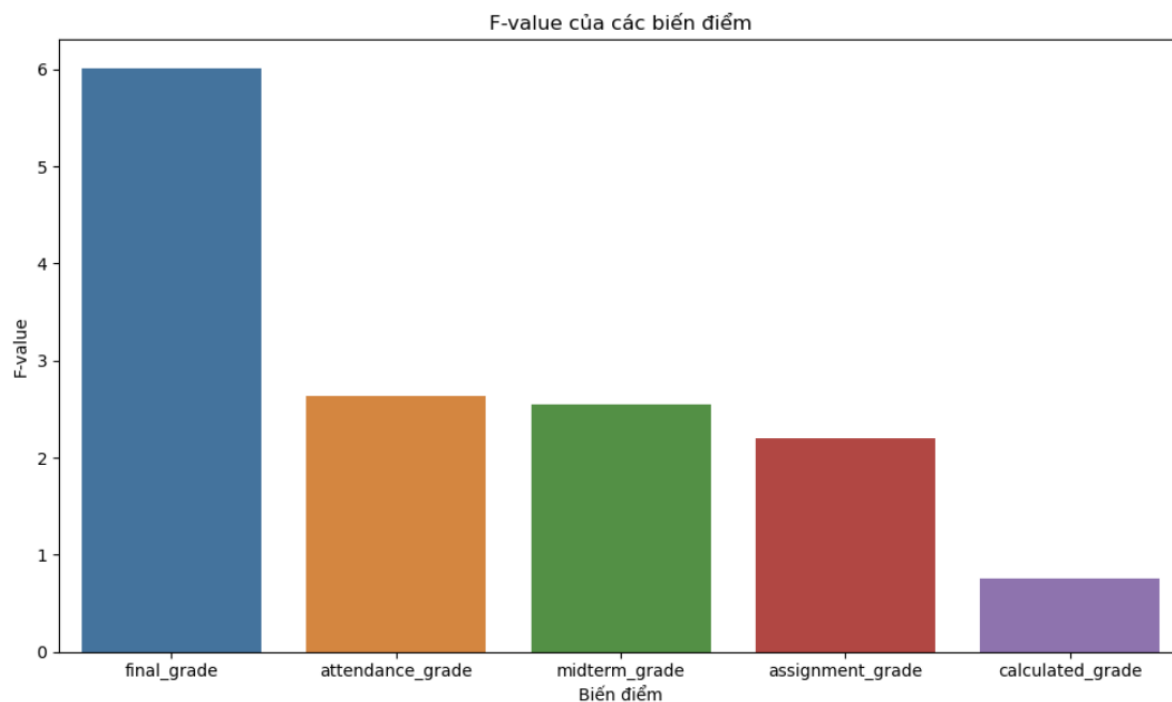
- **final_grade** và **calculated_grade** có tương quan mạnh nhất ($r=0.86$), phản ánh việc điểm cuối kỳ chiếm tỉ trọng lớn trong công thức tính điểm tổng hợp.
- **midterm_grade** có mối liên hệ trung bình với **calculated_grade** ($r=0.57$), nhưng gần như không tương quan với **final_grade** ($r=0.098$).
- **assignment_grade** chỉ tương quan vừa phải với **final_grade** ($r=0.45$) và rất yếu với **calculated_grade** ($r=0.094$).
- **attendance_grade** hầu như không tương quan với các biến khác (tất cả giá trị đều dưới 0.2), cho thấy điểm chuyên cần ít ảnh hưởng đến điểm cuối kỳ hay điểm tổng hợp.

Như vậy, điểm tổng hợp (**calculated_grade**) phụ thuộc chủ yếu vào điểm cuối kỳ và phần nào vào điểm giữa kỳ, trong khi điểm chuyên cần gần như độc lập với các thành phần đánh giá còn lại.

b. Phân tích phương sai (ANOVA ngầm định)

Phương pháp F-test (ANOVA một chiều) so sánh tỷ lệ giữa phương sai giữa các nhóm chuyên ngành và phương sai trong cùng nhóm, từ đó đưa ra chỉ số F-value cho từng biến. Một F-value cao cho biết biến đó có khả năng phân biệt giữa các nhóm chuyên ngành một cách rõ ràng.

Ở đây, ta tiến hành tính toán F-value để đánh giá khả năng phân biệt giữa các ngành.



Hình 23: Biểu đồ phân phối F-value của các loại điểm

Từ kết quả F-test, ta thu được rằng:

Thứ nhất, kết quả cho thấy **final_grade** có F-value cao nhất, phản ánh khả năng phân biệt vượt trội giữa các nhóm chuyên ngành. Điều này gợi ý rằng sinh viên thuộc ba chuyên ngành (Công nghệ phần mềm, An toàn thông tin, Trí tuệ nhân tạo) có sự khác biệt đáng kể về điểm cuối kỳ.

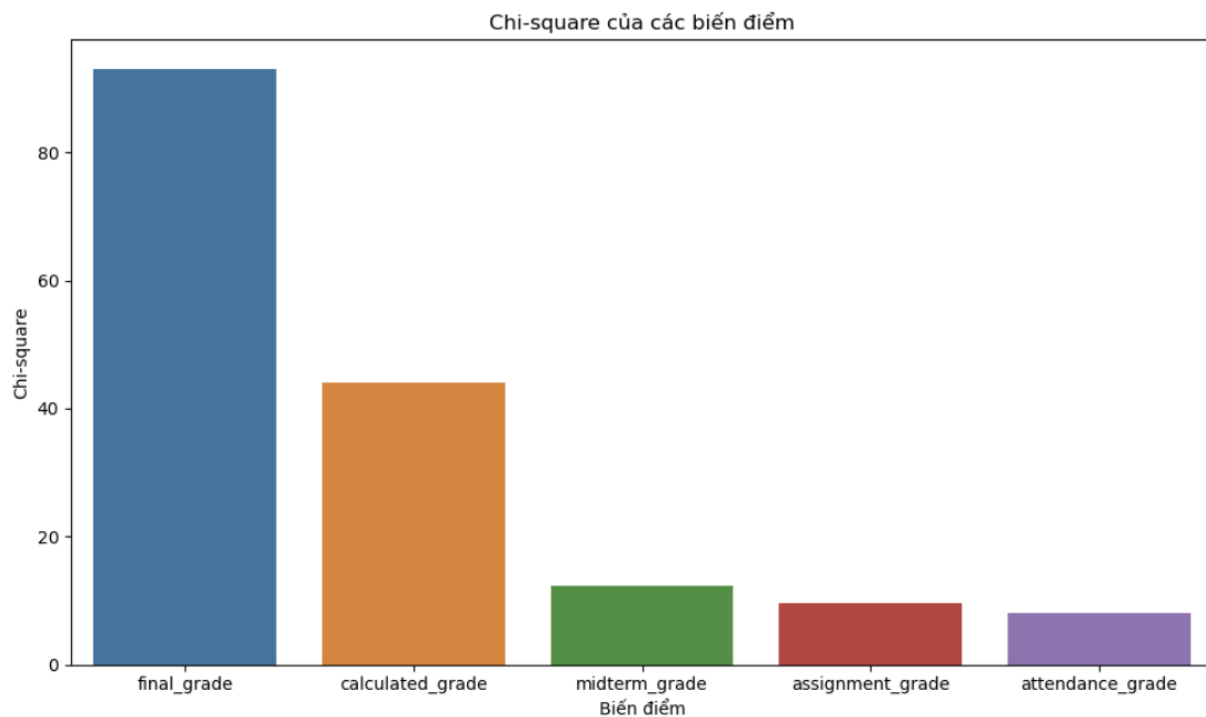
Thứ hai, **attendance_grade** (điểm chuyên cần) xếp thứ hai, cho thấy mức độ tham gia của sinh viên cũng góp phần không nhỏ trong việc phân biệt chuyên ngành. Sự khác biệt về tính kỷ luật, mức độ tuân thủ quy định có thể thay đổi theo định hướng ngành học.

Thứ ba, **midterm_grade** và **assignment_grade** có F-value ở mức trung bình, phản ánh sự chênh lệch giữa các chuyên ngành vẫn có nhưng không rõ rệt bằng điểm cuối kỳ hay điểm chuyên cần.

Thứ tư, **calculated_grade** lại có F-value thấp nhất, dù nó là điểm tổng hợp từ nhiều thành phần. Có thể do việc trung bình hóa khiến các chênh lệch bị “làm phẳng”, làm giảm khả năng phân biệt giữa các chuyên ngành.

c. Chi-square test cho biến rời rạc hóa

Phương pháp Chi-square (Chi-square test) thường được áp dụng khi muốn kiểm định mối liên hệ giữa hai biến phân loại. Trong trường hợp này, để đánh giá xem các biến điểm (vốn là dữ liệu liên tục) có liên quan đến chuyên ngành được gợi ý - **recommendation_major** - hay không, ta cần phân chia điểm thành những khoảng (bins), biến chúng thành dữ liệu rời rạc. Sau đó, việc xây dựng bảng liên hợp và tính chỉ số Chi-square cho phép xác định mức độ liên kết giữa từng biến điểm đã rời rạc hóa với chuyên ngành.



Hình 24: Phân phối Chi-square của các biến điểm

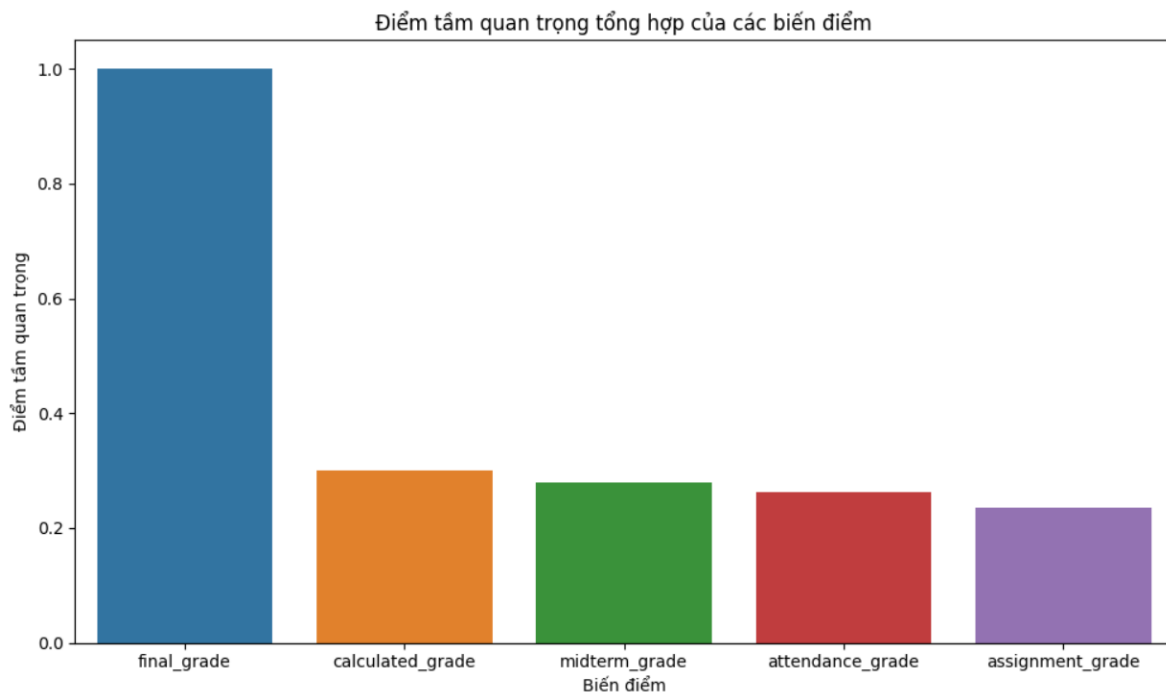
Dựa trên biểu đồ Chi-square:

- **final_grade** đạt giá trị Chi-square cao nhất, thể hiện mức độ liên quan mạnh nhất với chuyên ngành được gợi ý khi so sánh giữa các nhóm điểm cuối kỳ.

- **calculated_grade** xếp thứ hai, gợi ý rằng khi chuyển thành dạng rời rạc, điểm tổng hợp vẫn có khả năng phân biệt đáng kể giữa các chuyên ngành, dù không bằng **final_grade**.
- **midterm_grade**, **assignment_grade** và **attendance_grade** có Chi-square thấp hơn rõ rệt, cho thấy khi được chia thành các khoảng, chúng ít thể hiện sự khác biệt rõ ràng giữa các chuyên ngành.

d. Tổng hợp điểm chỉ số tầm quan trọng của các biến điểm

Từ kết quả phân tích được qua 2 phương pháp F-value và Chi-square, mỗi biến điểm được tính **điểm tổng hợp** bằng cách chuẩn hóa F-value và Chi-square về cùng thang [0,1], rồi lấy trung bình hai giá trị này. Qua đó, ta có một thước đo thống nhất phản ánh khả năng phân biệt giữa các chuyên ngành của từng biến điểm, xét theo cả hai góc độ liên tục và rời rạc.



Hình 25: Biểu đồ phân phối tầm quan trọng tổng hợp của các biến điểm

Kết quả cho thấy:

- **final_grade** có điểm tầm quan trọng cao nhất, khẳng định tính vượt trội của điểm cuối kỳ trong việc phân tách chuyên ngành.
- **calculated_grade** xếp thứ hai, cho thấy điểm tổng hợp cũng có đóng góp đáng kể khi xem xét dưới hai phương pháp.

- **midterm_grade**, **attendance_grade**, và **assignment_grade** lần lượt có điểm tầm quan trọng thấp hơn, phản ánh việc chúng không tạo ra sự khác biệt lớn như **final_grade** hay **calculated_grade**.

Giải thích cho lý do cho kết quả trên:

Thứ nhất, điểm thi cuối kỳ (final_grade) phản ánh khá sát năng lực cá nhân của sinh viên. Đây thường là bài đánh giá tổng hợp, yêu cầu sinh viên nắm vững toàn bộ kiến thức của môn học. Các thành phần khác như điểm chuyên cần (attendance_grade) hay điểm bài tập (assignment_grade) có thể bị ảnh hưởng bởi yếu tố làm việc nhóm, sao chép hoặc điểm danh hộ, nên không thể hiện hoàn toàn khả năng thực sự của từng cá nhân.

Thứ hai, tính phân hóa của điểm cuối kỳ thường cao hơn so với các thành phần khác. Ở nhiều môn học, điểm chuyên cần hoặc điểm bài tập có thể có phân phối khá hẹp, vì đa số sinh viên đều đạt mức điểm tương đối cao. Ngược lại, điểm cuối kỳ lại có sự chênh lệch rõ hơn giữa người học tốt và người học kém, nên khi so sánh giữa các chuyên ngành, biến này giúp nhận diện sự khác biệt mạnh mẽ hơn. Trong khi đó, **calculated_grade** (điểm tổng hợp) bị “trung bình hóa” bởi nhiều thành phần, có thể làm giảm khả năng phân biệt này.

Thứ ba, mối liên hệ với yêu cầu chuyên ngành cũng được thể hiện rõ qua điểm thi cuối kỳ. Bài thi cuối kỳ thường đánh giá những kỹ năng cốt lõi và kiến thức chuyên sâu của môn học, vốn tương ứng với đòi hỏi của từng chuyên ngành (ví dụ: kỹ năng lập trình, tư duy thuật toán, bảo mật, v.v.). Do đó, sự khác biệt trong năng lực cốt lõi của sinh viên sẽ bộc lộ rõ ràng hơn ở điểm cuối kỳ, giúp biến này có mức tầm quan trọng cao khi phân biệt giữa các nhóm chuyên ngành.

Kết luận: Điểm cuối kỳ (**final_grade**) tỏ ra hiệu quả hơn trong việc xây dựng các đặc trưng huấn luyện mô hình, mặc dù về mặt lý thuyết, **điểm tổng kết** (**calculated_grade**) được kỳ vọng phản ánh toàn diện quá trình học tập của sinh viên.

2. Tiền xử lý và chuẩn hóa dữ liệu

2.1. Xử lý dữ liệu bị thiếu

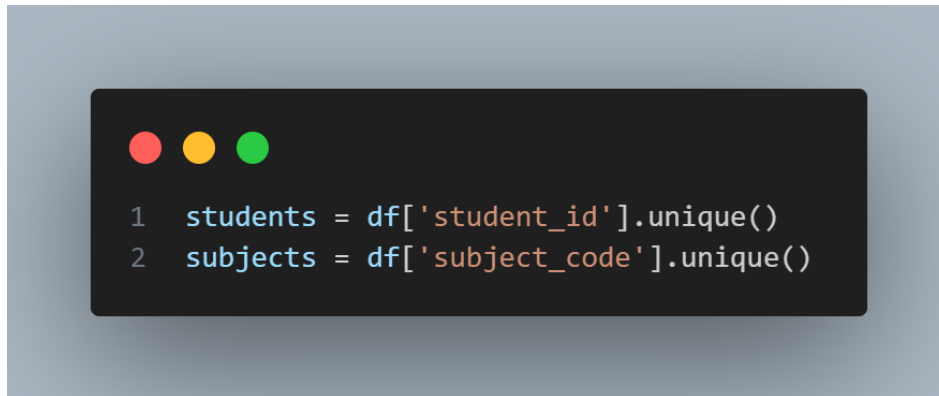
Với bộ dữ liệu được đưa ra ở trên ta có thể thấy không có giá trị bị thiếu đã được chứng minh ở giai đoạn “Kiểm tra giá trị bị thiếu” trong Mục IV nên bước này ta sẽ bỏ qua.

2.2. Xử lý dữ liệu trùng

Với bộ dữ liệu được đưa ra ở trên ta có thể thấy không có giá trị trùng lặp đã được chứng minh ở giai đoạn “Kiểm tra giá trị trùng lặp” trong Mục IV nên bước này ta sẽ bỏ qua.

Từ cấu trúc dữ liệu đã thu thập ở trên, ta thực hiện trích xuất đặc trưng từ những thuộc tính của bộ dữ liệu.

Thực hiện nhóm dữ liệu theo thuộc tính `student_id` và `subject_code`



Hình 26: Đoạn code xử lý nhóm bộ dữ liệu theo thuộc tính `student_id` và `subject_code`

2.3. Xử lý các thuộc tính điểm

Các thuộc tính điểm: *final grade*, *midterm grade*, *assignment grade*, *attendance grade*

Thay vì tính toán điểm tổng kết cuối cùng của một môn học thì ta thực hiện việc tách các thang điểm vì các lý do sau:

- Các thang điểm phản ánh rõ từng khía cạnh năng lực học tập của sinh viên.

Ta có thể nói mỗi thành phần điểm thể hiện một kỹ năng khác nhau:

- *Final grade* đánh giá tư duy tổng hợp và khả năng thi cử.
- *Midterm grade* phản ánh khả năng tiếp thu kiến thức giai đoạn đầu.
- *Assignment grade* thể hiện năng lực thực hành, nghiên cứu độc lập.
- *Attendance grade* thể hiện thái độ học tập và tính kỷ luật.

→ Gộp lại sẽ làm mất đi thông tin cụ thể của từng kỹ năng.

- Tránh mất thông tin chi tiết trong quá trình xử lý dữ liệu, từ đó giúp khai thác dữ liệu tốt hơn.

Khi gộp điểm tổng kết, các đặc trưng thành phần bị ẩn đi. Hai sinh viên có thể có cùng điểm tổng kết, nhưng khác biệt hoàn toàn về khả năng và phong cách học tập. Việc giữ lại từng thang điểm giúp mô hình hiểu sâu hơn về sinh viên.

Ví dụ: Điểm assignment cao → có xu hướng phù hợp chuyên ngành thực hành (kỹ thuật, lập trình), ...

Giữ nguyên từng thang điểm là cách làm khoa học, giàu thông tin và linh hoạt, giúp mô hình và quá trình phân tích dữ liệu trở nên hiệu quả hơn, hỗ trợ tốt cho việc gợi ý chuyên ngành học phù hợp, ...

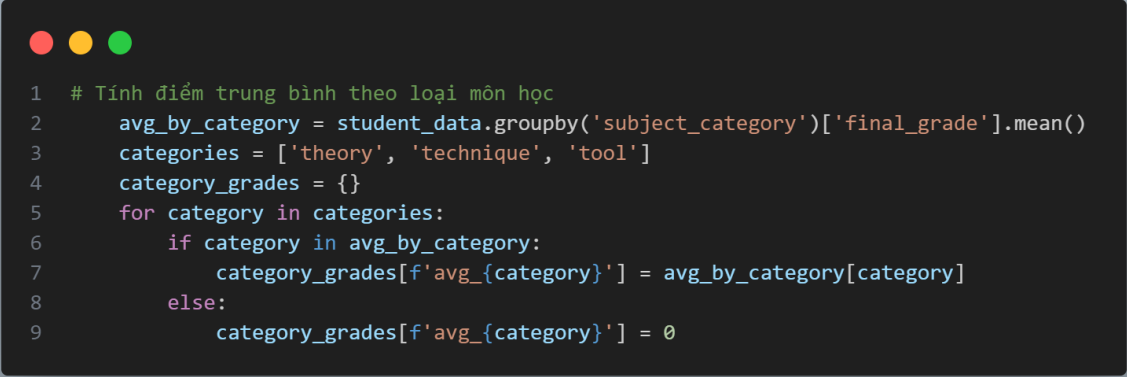
Bên cạnh đó, ta có thể thấy mức độ ảnh hưởng của các thang điểm có sự khác biệt

2.4. Xử lý thuộc tính `subject_category`, `final_grade`

Việc xử lý này là để tính điểm trung bình môn học theo loại môn học. Ta có thể thấy điểm trung bình môn học theo loại môn học (lý thuyết - theory, công nghệ - technique, công cụ - tool) có sự khác biệt giữa các chuyên ngành như đã được thể hiện ở mục 1.6 “Phân tích điểm trung bình theo chuyên ngành” của phần IV. Trong mục 1.6 cho ta thấy **Trí tuệ nhân tạo** nổi trội ở nhóm môn lý thuyết, với mức điểm nhỉnh hơn so với hai chuyên ngành còn lại. **Công nghệ phần mềm** thể hiện thế mạnh ở nhóm môn công nghệ (tool), đồng thời đạt kết quả kỹ thuật (technique) tương đối cao. Trong khi đó, **An toàn thông tin** lại xếp sau hai chuyên ngành kia ở cả hai nhóm môn công nghệ và kỹ thuật, cho thấy mức độ tiếp cận thực hành và ứng dụng của nhóm sinh viên này còn hạn chế hơn.

Vậy nên yếu tố điểm trung bình môn học theo loại môn học là một trong những yếu tố có thể ảnh hưởng đến mục tiêu gợi ý chuyên ngành.

Dưới đây là cách xử lý để tìm ra điểm trung bình môn học theo loại môn học.



```

1 # Tính điểm trung bình theo loại môn học
2 avg_by_category = student_data.groupby('subject_category')['final_grade'].mean()
3 categories = ['theory', 'technique', 'tool']
4 category_grades = {}
5 for category in categories:
6     if category in avg_by_category:
7         category_grades[f'avg_{category}'] = avg_by_category[category]
8     else:
9         category_grades[f'avg_{category}'] = 0

```

Hình 27: Đoạn code xử lý điểm trung bình môn học theo loại môn học

2.5. Xử lý thuộc tính skill_list

Bên cạnh điểm số, độ thành thạo kỹ năng cũng là một yếu tố quan trọng để gợi ý chuyên ngành. Việc học các môn học giúp cho sinh viên tích lũy được một số kỹ năng và cũng có thể củng cố nhưng kỹ năng đã học chắc chắn hơn.

Một số lý do yếu tố kỹ năng có thể ảnh hưởng đến việc chọn chuyên ngành:

- *Xác định điểm mạnh chuyên môn cá nhân:* Thống kê các kỹ năng giúp phát hiện sinh viên giỏi ở nhóm kỹ năng nào (ví dụ: phân tích dữ liệu, lập trình, bảo mật, AI...), từ đó định hướng chuyên ngành phù hợp.
- *Bổ sung góc nhìn kỹ năng bên cạnh điểm số:* Chuyên ngành không chỉ phụ thuộc vào điểm môn học mà còn dựa vào kỹ năng tích lũy.
- *Phân tích mức độ thành thạo:* Việc tính toán mức độ thành thạo từng kỹ năng (dựa vào số lần xuất hiện và điểm trung bình) giúp đánh giá chính xác khả năng của sinh viên trong từng mảng kỹ năng.

Với một số lý do đã nêu ở trên, nên ta sẽ thực hiện việc trích xuất đặc trưng của các kỹ năng bằng cách tính độ thành thạo của các kỹ năng như sau:

Phân tích trường skill_list từ mỗi môn học, tách các kỹ năng riêng lẻ. Với mỗi kỹ năng, thu thập:

- Tất cả các điểm số môn học liên quan đến kỹ năng đó
- Số lần kỹ năng xuất hiện trong các môn học

Tính điểm trung bình cho mỗi kỹ năng. Phân loại mức độ thành thạo dựa trên điểm trung bình:

- Giỏi (3): điểm trung bình ≥ 8
- Khá (2): điểm trung bình ≥ 6 và < 8
- Trung bình (1): điểm trung bình ≥ 3 và < 6
- Yếu (0): điểm trung bình < 3

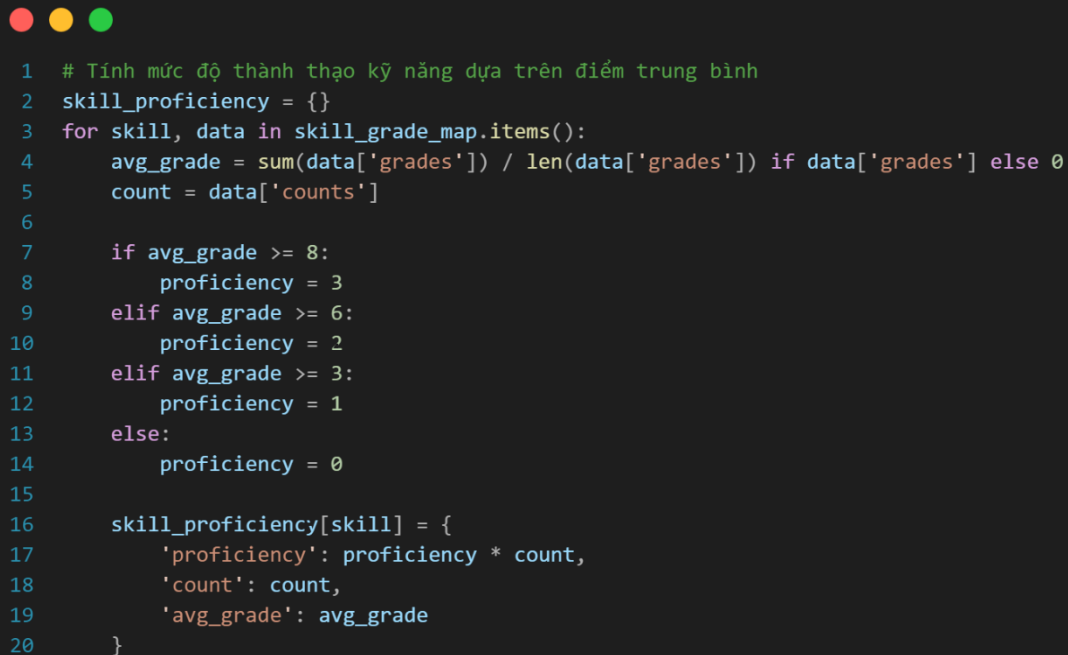
Tính điểm thành thạo tổng hợp: mức độ thành thạo \times số lần xuất hiện. Sắp xếp kỹ năng theo mức độ thành thạo, số lần xuất hiện và điểm trung bình (giảm dần). Từ đó, ghi lại các đặc trưng cho mỗi kỹ năng:

- Số lần xuất hiện: skill_{tên_kỹ_năng}
- Mức độ thành thạo: skill_{tên_kỹ_năng}_level
- Điểm trung bình: skill_{tên_kỹ_năng}_grade

Dưới đây là đoạn code xử lý độ thành thực của các kỹ năng đã được học từ các môn học.

```
1 # Tạo vector kỹ năng với phân loại mức độ thành thạo
2 skill_grade_map = {}
3
4 for _, row in student_data.iterrows():
5     final_grade = row["final_grade"]
6     if not pd.isna(row["skill_list"]):
7         for skill in [s.strip() for s in row["skill_list"].split(",")]:
8             if skill not in skill_grade_map:
9                 skill_grade_map[skill] = {"grades": [], "counts": 0}
10            skill_grade_map[skill]["grades"].append(final_grade)
11            skill_grade_map[skill]["counts"] += 1
```

Hình 28: Xây dựng vector kỹ năng với phân loại mức độ thành thạo



```

1 # Tính mức độ thành thạo kỹ năng dựa trên điểm trung bình
2 skill_proficiency = {}
3 for skill, data in skill_grade_map.items():
4     avg_grade = sum(data['grades']) / len(data['grades']) if data['grades'] else 0
5     count = data['counts']
6
7     if avg_grade >= 8:
8         proficiency = 3
9     elif avg_grade >= 6:
10        proficiency = 2
11    elif avg_grade >= 3:
12        proficiency = 1
13    else:
14        proficiency = 0
15
16    skill_proficiency[skill] = {
17        'proficiency': proficiency * count,
18        'count': count,
19        'avg_grade': avg_grade
20    }

```

Hình 29: : Tính mức độ thành thạo dựa trên điểm trung bình

2.6. Xử lý thuộc tính `subject_name`

Có một nhận định như sau: “Một sinh viên có tỷ lệ điểm cao ở nhóm môn Toán – Vật lý có khả năng phù hợp với các chuyên ngành kỹ thuật, trong khi sinh viên có tỷ lệ điểm cao ở nhóm môn Ngữ văn – Lịch sử có thể phù hợp với chuyên ngành xã hội – nhân văn.” Từ nhận định này ta có thể thấy có một mối quan hệ giữa các môn học ảnh hưởng nhiều đến việc lựa chọn chuyên ngành đó và phân lớp điểm số của môn học đó (trung bình, giỏi, xuất sắc). Ta phân loại các phân lớp như sau: Trung bình (3 – 5 điểm), Giỏi (6 – 7 điểm), Xuất sắc (8 – 10 điểm).

Từ mối quan hệ trên, ta có thể suy luận ra một số yếu tố có thể ảnh hưởng đến việc lựa chọn chuyên ngành như sau:

- Điểm trung bình của các môn học ảnh hưởng nhiều đến chuyên ngành đó theo phân lớp điểm số.
- Phần trăm số lượng các môn học ảnh hưởng nhiều đến chuyên ngành đó theo phân lớp điểm số.

Với hai ý kiến trên, ta sẽ thực hiện theo yếu tố thứ 2 bởi vì yếu tố thứ 2 không bị ảnh hưởng nhiều bởi yếu tố này có tính ổn định hơn yếu tố 1 khi bộ dữ liệu còn chưa

chính xác. Trong khi đó, yếu tố thứ 1 dễ bị ảnh hưởng nếu khi bộ dữ liệu chưa chính xác.

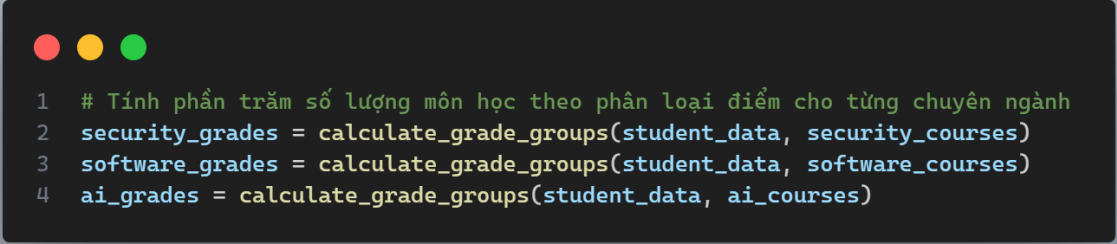
Ví dụ: Giả thuyết chỉ có một môn ảnh hưởng đến chuyên ngành; Với nhiều sinh viên có điểm môn học A thuộc phân lớp Xuất sắc nhưng đều thuộc 8 điểm thì khi tính điểm trung bình sẽ là 8; Với nhiều sinh viên có điểm môn học A thuộc phân lớp Xuất sắc nhưng đều thuộc 10 điểm thì khi tính điểm trung bình sẽ là 10;

Cả hai trường hợp ta có thể thấy có sự chênh lệch đáng kể nếu như bộ dữ liệu chưa chính xác hoặc chưa cân bằng. Còn nếu như tính theo phần trăm số lượng thì ta có thể thấy là trong cả hai trường hợp thì đều có giá trị là 1.

Tiếp theo, ta sẽ thực hiện việc tính toán phần trăm số lượng môn học theo phân lớp điểm số.

```
1  # Tính số lượng và phần trăm môn học theo phân loại điểm cho từng chuyên ngành
2  def calculate_grade_groups(student_df, courses):
3      course_df = student_df[student_df["subject_name"].isin(courses)]
4
5      total_subjects = len(course_df)
6      if total_subjects == 0:
7          return 0, 0, 0, 0.0, 0.0, 0.0
8
9      # Đếm số lượng môn theo nhóm điểm
10     count_average = len(course_df[(course_df["final_grade"] >= 3) & (course_df["final_grade"] < 6)])
11     count_good = len(course_df[(course_df["final_grade"] >= 6) & (course_df["final_grade"] < 8)])
12     count_excellent = len(course_df[course_df["final_grade"] >= 8])
13
14     # Tính phần trăm theo từng nhóm
15     percent_average = (count_average / total_subjects) * 100
16     percent_good = (count_good / total_subjects) * 100
17     percent_excellent = (count_excellent / total_subjects) * 100
18
19     return (
20         count_average, count_good, count_excellent,
21         percent_average, percent_good, percent_excellent,
22     )
```

Hình 30: Tính phần trăm số lượng môn học theo phân loại



```

1 # Tính phần trăm số lượng môn học theo phân loại điểm cho từng chuyên ngành
2 security_grades = calculate_grade_groups(student_data, security_courses)
3 software_grades = calculate_grade_groups(student_data, software_courses)
4 ai_grades = calculate_grade_groups(student_data, ai_courses)

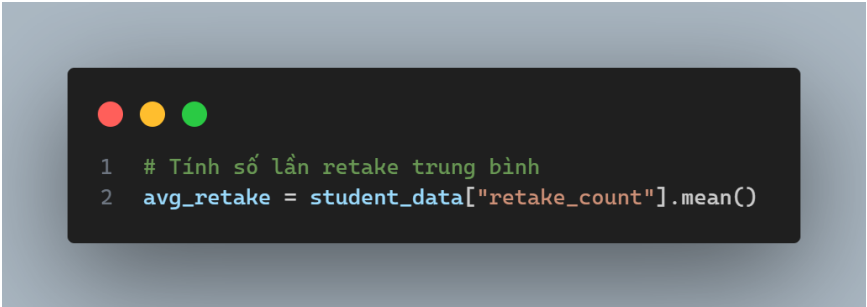
```

Hình 31: Gọi hàm để tính phần trăm số lượng theo từng nhóm chuyên ngành

2.7. Xử lý thuộc tính `retake_count`

Tính trung bình số lần thi lại trên tất cả các môn học là chỉ số định lượng phản ánh khả năng nắm bắt kiến thức và hiệu quả học tập của sinh viên. Chỉ số này giúp xác định sinh viên có thành tích ổn định và hiểu bài tốt (số lần thi lại thấp) hay gặp khó khăn trong học tập (số lần thi lại cao). Từ đó, nó là cơ sở định lượng hỗ trợ gợi ý chuyên ngành phù hợp: sinh viên có số lần thi lại thấp có khả năng theo kịp các chuyên ngành đòi hỏi kiến thức nền tảng vững chắc, trong khi sinh viên có số lần thi lại cao cần cân nhắc lại định hướng học tập của mình.

Ta thực hiện việc tính trung bình số lần thi lại trên tất cả các môn.



```

1 # Tính số lần retake trung bình
2 avg_retake = student_data["retake_count"].mean()

```

Hình 32: Tính số lần học lại trung bình

2.8. Xử lý thuộc tính `subject_type`

Trong khai phá dữ liệu giáo dục, việc trích xuất đặc trưng từ thuộc tính `subject_type` – bao gồm hai giá trị: `core` (môn chuyên ngành) và `general` (môn đại cương) – đóng vai trò quan trọng trong gợi ý chuyên ngành cho sinh viên. Việc phân loại này phản ánh mục tiêu và định hướng của các môn học trong chương trình đào tạo.

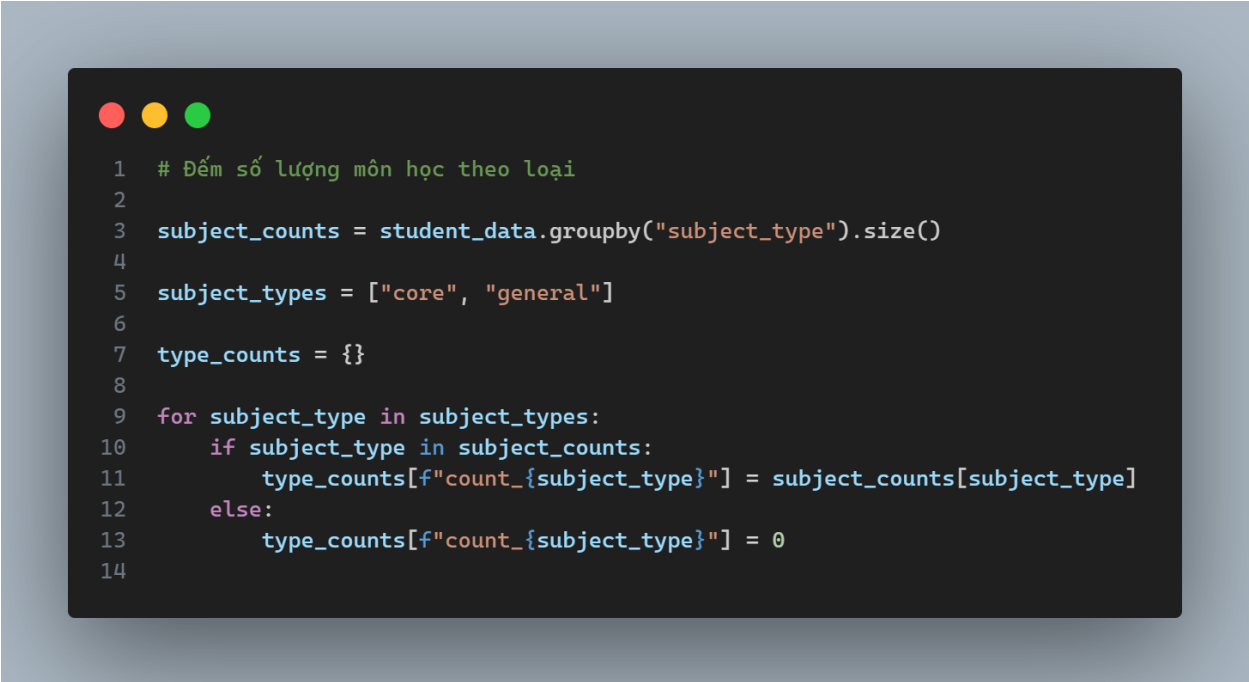
Việc tính toán số lượng môn học `core` và `general` mà sinh viên đã học giúp xây dựng các đặc trưng đơn giản nhưng giàu ý nghĩa, thể hiện rõ xu hướng học tập và mức độ định hướng chuyên ngành. Sinh viên học nhiều môn `core` thường có xu hướng theo

đuôi chuyên ngành cụ thể, trong khi sinh viên học chủ yếu môn general có thể chưa xác định rõ định hướng.

Các đặc trưng này không chỉ dễ hiểu, dễ xử lý, mà còn phù hợp với nhiều mô hình học máy như hồi quy logistic, cây quyết định, k-láng giềng gần nhất (kNN) trong phân loại hoặc gợi ý chuyên ngành. Khi được tích hợp vào hệ thống gợi ý, chúng cung cấp thông tin hữu ích để xác định chuyên ngành phù hợp cho từng sinh viên.

Tóm lại, đặc trưng từ `subject_type` góp phần nâng cao hiệu quả của hệ thống gợi ý chuyên ngành, giúp sinh viên có định hướng học tập phù hợp với năng lực và sở thích cá nhân.

Dưới đây là đoạn xử lý:



```
1  # Đếm số lượng môn học theo loại
2
3  subject_counts = student_data.groupby("subject_type").size()
4
5  subject_types = ["core", "general"]
6
7  type_counts = {}
8
9  for subject_type in subject_types:
10     if subject_type in subject_counts:
11         type_counts[f"count_{subject_type}"] = subject_counts[subject_type]
12     else:
13         type_counts[f"count_{subject_type}"] = 0
14
```

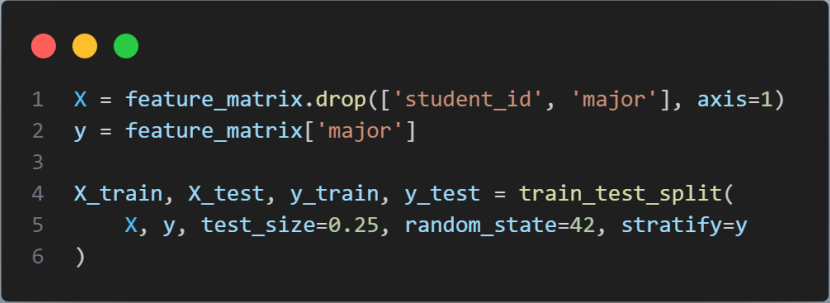
Hình 33: Xử lý thuộc tính `subject_type`

V. Mô hình và thực nghiệm

1. Xây dựng mô hình

Trong đề tài này, ta sẽ sử dụng nhiều phương pháp để thực hiện việc khai phá dữ liệu để gợi ý chuyên ngành cho sinh viên, đó là Logistic Regression, KNN, Radom Forest, Neural Network.

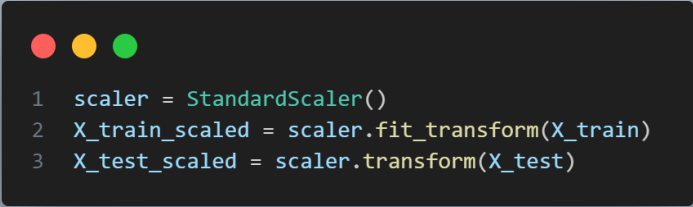
Việc đầu tiên khi xây dựng mô hình, ta sẽ thực hiện việc định nghĩa bộ dữ liệu và chia bộ dữ liệu thành hai phần là train và test.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains six lines of Python code for splitting a dataset.

```
1 X = feature_matrix.drop(['student_id', 'major'], axis=1)
2 y = feature_matrix['major']
3
4 X_train, X_test, y_train, y_test = train_test_split(
5     X, y, test_size=0.25, random_state=42, stratify=y
6 )
```

Hình 34: Đoạn chương trình thực hiện chia bộ dữ liệu

Tiếp theo đó là thực hiện chuẩn hóa dữ liệu sử dụng phương pháp Standard Scaler.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains three lines of Python code for standardizing the data using StandardScaler.

```
1 scaler = StandardScaler()
2 X_train_scaled = scaler.fit_transform(X_train)
3 X_test_scaled = scaler.transform(X_test)
```

Hình 35: Đoạn chương trình thực hiện việc chuẩn hóa dữ liệu

Sau khi chuẩn bị xong ta định nghĩa mô hình và các tham số cho từng phương pháp như sau:

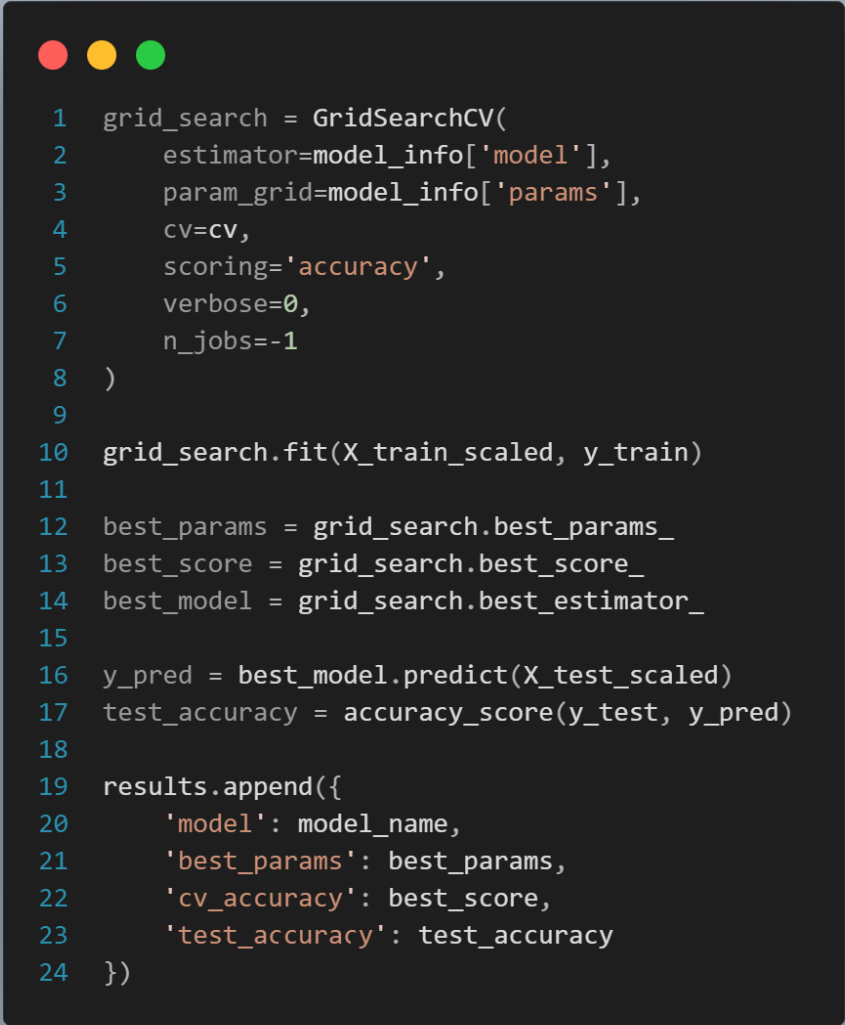
```

1 # Định nghĩa các mô hình và tham số tương ứng để thử nghiệm
2 models = {
3     'Logistic Regression': {
4         'model': LogisticRegression(random_state=42),
5         'params': {
6             'C': [0.01, 0.1, 1.0, 10.0, 100.0],
7             'solver': ['liblinear', 'lbfgs', 'saga'],
8             'max_iter': [1000],
9             'multi_class': ['auto', 'ovr', 'multinomial']
10        }
11    },
12    'Random Forest': {
13        'model': RandomForestClassifier(random_state=42),
14        'params': {
15            'n_estimators': [200],
16            'max_depth': [30],
17            'min_samples_split': [10],
18            'min_samples_leaf': [4]
19        }
20    },
21    'KNN': {
22        'model': KNeighborsClassifier(),
23        'params': {
24            'n_neighbors': [3, 5, 7, 9, 11],
25            'weights': ['uniform', 'distance'],
26            'metric': ['euclidean', 'manhattan', 'minkowski']
27        }
28    },
29    'Neural Network': {
30        'model': MLPClassifier(random_state=42),
31        'params': {
32            'hidden_layer_sizes': [(50, ), (100, ), (50, 50), (100, 50)],
33            'activation': ['relu', 'tanh'],
34            'alpha': [0.0001, 0.001, 0.01],
35            'learning_rate': ['constant', 'adaptive'],
36            'max_iter': [500]
37        }
38    }
39 }

```

Hình 36: Đoạn chương trình định nghĩa các mô hình và tham số

Tiếp đó, sử dụng công cụ GridSearchCV để thực hiện việc tìm tham số tốt nhất đối với mô hình từng loại phương pháp như hình ảnh dưới đây thể hiện



```

1  grid_search = GridSearchCV(
2      estimator=model_info['model'],
3      param_grid=model_info['params'],
4      cv=cv,
5      scoring='accuracy',
6      verbose=0,
7      n_jobs=-1
8  )
9
10 grid_search.fit(X_train_scaled, y_train)
11
12 best_params = grid_search.best_params_
13 best_score = grid_search.best_score_
14 best_model = grid_search.best_estimator_
15
16 y_pred = best_model.predict(X_test_scaled)
17 test_accuracy = accuracy_score(y_test, y_pred)
18
19 results.append({
20     'model': model_name,
21     'best_params': best_params,
22     'cv_accuracy': best_score,
23     'test_accuracy': test_accuracy
24 })

```

Hình 37: Đoạn chương trình sử dụng GridSearchCV để tìm tham số tốt nhất cho mô hình

Sau khi thực hiện một số vòng lặp nhất định để tìm tham số tốt nhất thì lưu kết quả lưu lại mô hình tốt nhất cũng như đã thực hiện xong việc huấn luyện mô hình.

2. Thực nghiệm và phân tích

2.1. Trình bày kết quả

Sau khi thực hiện huấn luyện các mô hình thì ta được tham số và kết quả tốt nhất trên từng mô hình như sau:

Mô hình Logistic Regression và Neural Network cho độ chính xác cao nhất:

- Cả hai mô hình đều đạt độ chính xác rất cao, gần như tương đương nhau, dao động xấp xỉ 0.95 – 0.9803 ở cả hai chỉ số cv_accuracy (độ chính xác trên tập huấn luyện chéo) và test_accuracy (độ chính xác trên tập kiểm tra).
- Điều này cho thấy mức độ tổng quát hóa tốt và ít xảy ra overfitting hoặc underfitting.

Mô hình Random Forest có độ chính xác thấp hơn một chút:

- Độ chính xác chỉ ở mức xấp xỉ 0.89, thấp hơn so với Logistic Regression và Neural Network.
- Tuy nhiên, khoảng cách giữa cv_accuracy và test_accuracy là nhỏ, chứng tỏ mô hình vẫn có khả năng tổng quát ổn định.

Mô hình KNN có độ chính xác thấp nhất:

- Cả hai chỉ số accuracy đều chỉ khoảng 0.81 – 0.82, cho thấy KNN không hoạt động tốt bằng các mô hình còn lại trên bộ dữ liệu này.
- Điều này có thể do KNN không phù hợp với dữ liệu có độ phức tạp cao hoặc phân bố không đồng đều, hoặc do bộ dữ liệu cần chuẩn hóa tốt hơn cho KNN hoạt động hiệu quả.

model	cv_accuracy	test_accuracy	best_parameter
<i>Logistic Regression</i>	0.974264	0.980251	'C': 10.0 'max_iter': 1000 'multi_class': 'auto' 'solver': 'lbfgs'
<i>Neural Network</i>	0.958307	0.959306	'activation': 'tanh' 'alpha': 0.0001 'hidden_layer_sizes': (50,) 'learning_rate': 'constant' 'max_iter': 500
<i>Random Forest</i>	0.891682	0.891083	'max_depth': 30 'min_samples_leaf': 4 'min_samples_split': 10 'n_estimators': 200
<i>KNN</i>	0.807099	0.813285	'activation': 'tanh' 'alpha': 0.0001

			'hidden_layer_sizes': (50,) 'learning_rate': 'constant' 'max_iter': 500
--	--	--	---

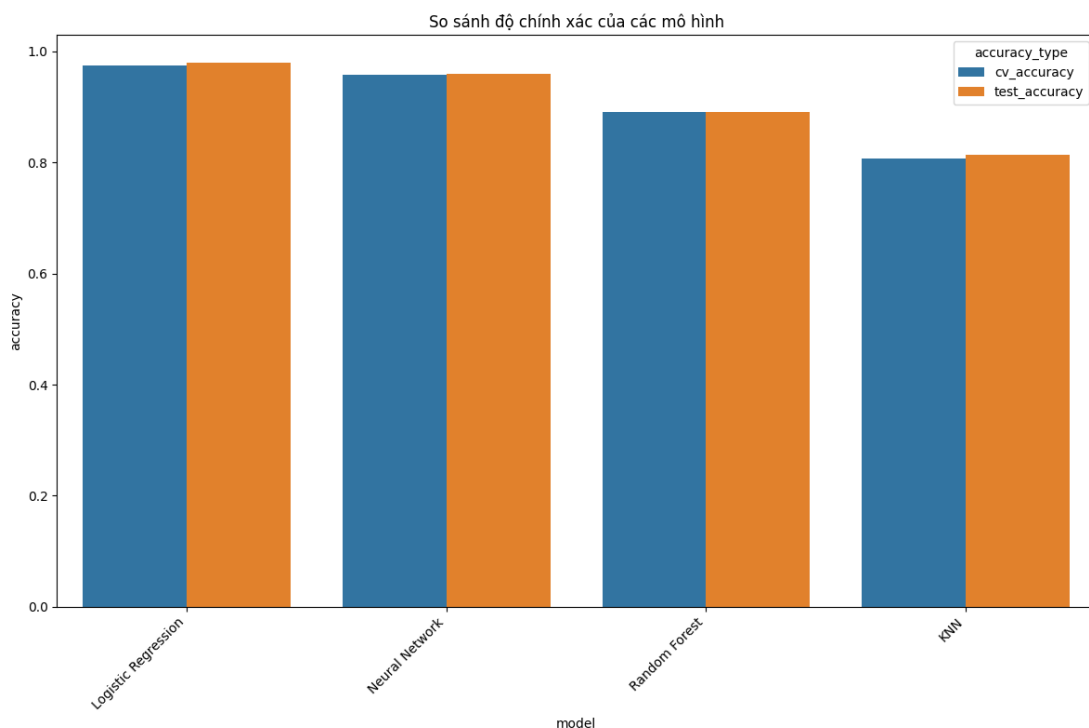
Bảng 3: Tham số và kết quả tốt nhất trên các mô hình

Từ bảng trên ta có thể thấy, kết quả accuracy của phương pháp Logistic Regression là cho kết quả tốt nhất 0.980251 cùng với tham số tốt nhất là $C = 10.0$, $\text{max_iter} = 1000$, $\text{multi_class} = \text{auto}$, $\text{solver} = \text{lbfgs}$. Trong đó tham số C là tham số regularization, max_iter là số vòng lặp tối đa, multi_class là tham số thiết lập phương pháp xử lý bài toán phân loại nhiều lớp, solver là tham số thuật toán tối ưu hóa dùng để tìm trọng số tốt nhất.

Hình bên dưới thể hiện độ tin cậy của bộ dữ liệu train và bộ dữ liệu test trên từng phương pháp. Ta cũng có thể thấy sự chênh lệch giữa CV Accuracy và Test Accuracy nhỏ ở tất cả các mô hình

=> Điều này chứng tỏ quá trình huấn luyện và đánh giá được thực hiện tốt, không có dấu hiệu overfitting rõ rệt.

=> Các mô hình có tính ổn định cao giữa huấn luyện và kiểm tra.



Hình 38: So sánh độ chính xác của các mô hình

2.2. Đánh giá mô hình

Sau khi cho ra kết quả thì ta thực hiện việc đánh giá chi tiết dựa trên từng nhãn của bộ dữ liệu.

Mô hình tốt nhất: Logistic Regression

Độ chính xác: 0.9803

Báo cáo phân loại của mô hình tốt nhất:

	precision	recall	f1-score	support
An toàn thông tin	0.98	0.97	0.97	437
Công nghệ phần mềm	0.98	0.99	0.99	1074
Trí tuệ nhân tạo	0.96	0.94	0.95	160
accuracy			0.98	1671
macro avg	0.98	0.97	0.97	1671
weighted avg	0.98	0.98	0.98	1671

Hình 39: Kết quả mô hình tốt nhất và chi tiết về các nhãn trong mô hình

Từ hình trên ta có một số nhận xét như sau:

- Mô hình Logistic Regression đạt độ chính xác tổng thể cao (Accuracy = 98.03%), cho thấy khả năng phân loại tốt giữa các chuyên ngành.
- Hiệu suất theo từng chuyên ngành:
- Công nghệ phần mềm có hiệu suất phân loại cao nhất:

Precision: 0.98 | Recall: 0.97 | F1-score: 0.97 => Điều này cho thấy mô hình rất ít nhầm lẫn khi phân loại chuyên ngành này.

- An toàn thông tin cũng đạt hiệu suất tốt:

Precision: 0.98 | Recall: 0.99 | F1-score: 0.99 => Mô hình có khả năng nhận diện chính xác các sinh viên thuộc chuyên ngành này.

- Trí tuệ nhân tạo có kết quả thấp hơn so với hai chuyên ngành còn lại:

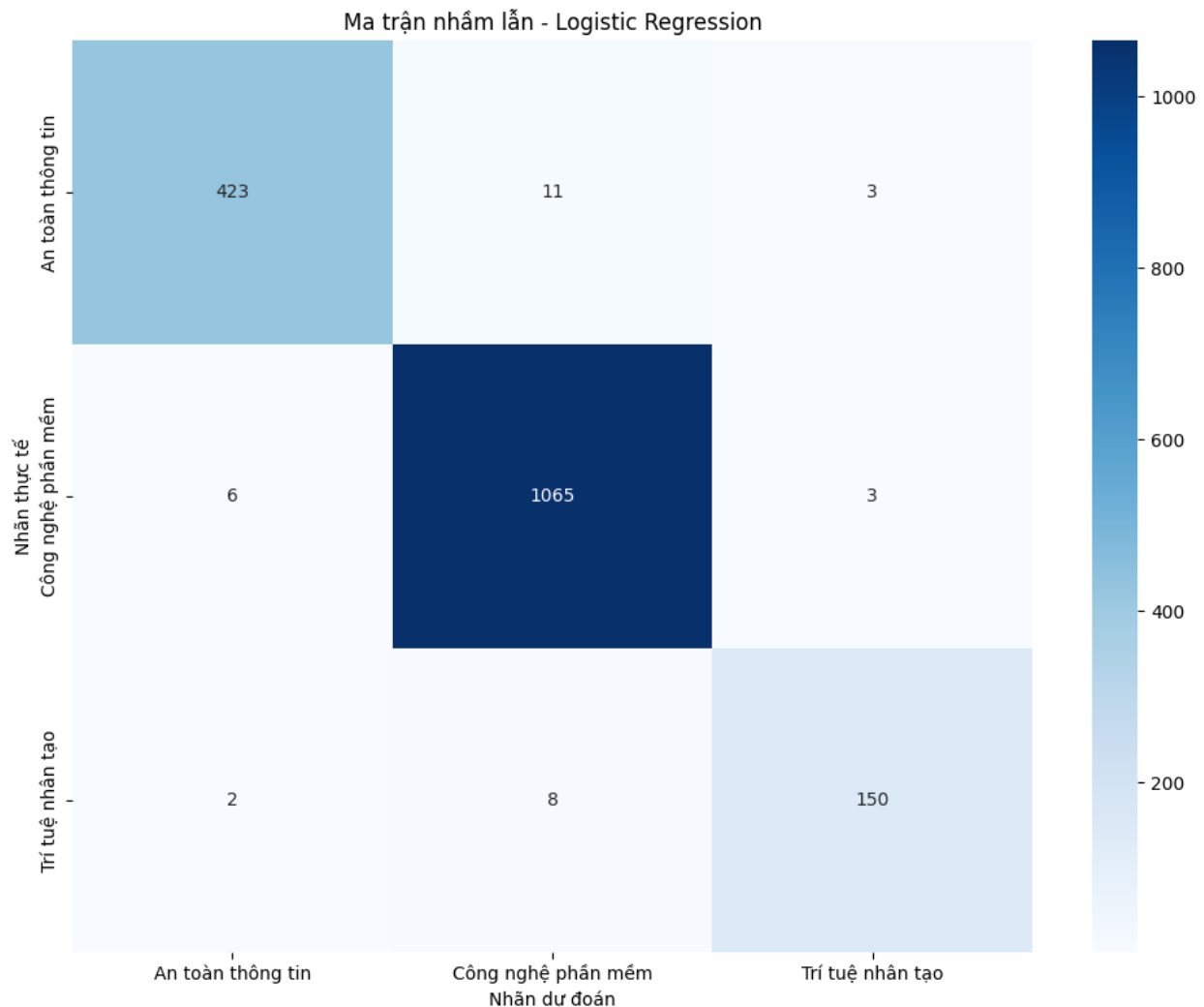
Precision: 0.96 | Recall: 0.94 | F1-score: 0.95 => Tuy vẫn đạt hiệu suất tốt, nhưng có thể thấy mô hình gặp khó khăn hơn trong việc nhận diện đúng sinh viên thuộc chuyên ngành này, có thể do số lượng dữ liệu thấp (support = 160).

- Chỉ số trung bình:

Macro average (trung bình không trọng số): F1-score = 0.97

Weighted average (trung bình có trọng số theo số mẫu): F1-score = 0.98 → Cả hai chỉ số đều cho thấy hiệu suất tổng thể rất tốt và cân bằng giữa các lớp.

Bên cạnh đó, ta cũng có ma trận nhầm lẫn của mô hình như sau.



Hình 40: Ma trận nhầm lẫn của mô hình

Ta có một số đánh giá và kết luận như sau:

Lớp "An toàn thông tin" (Hàng 1):

- Dự đoán đúng: 423
- Nhầm thành "Công nghệ phần mềm": 11
- Nhầm thành "Trí tuệ nhân tạo": 3 → Mô hình phân loại tốt, nhưng vẫn có một vài nhầm lẫn nhỏ.

Lớp "Công nghệ phần mềm" (Hàng 2):

- Dự đoán đúng: 1065 (rất cao)
- Nhầm thành "An toàn thông tin": 6
- Nhầm thành "Trí tuệ nhân tạo": 3 → Lớp này có độ chính xác rất cao, ít bị nhầm.

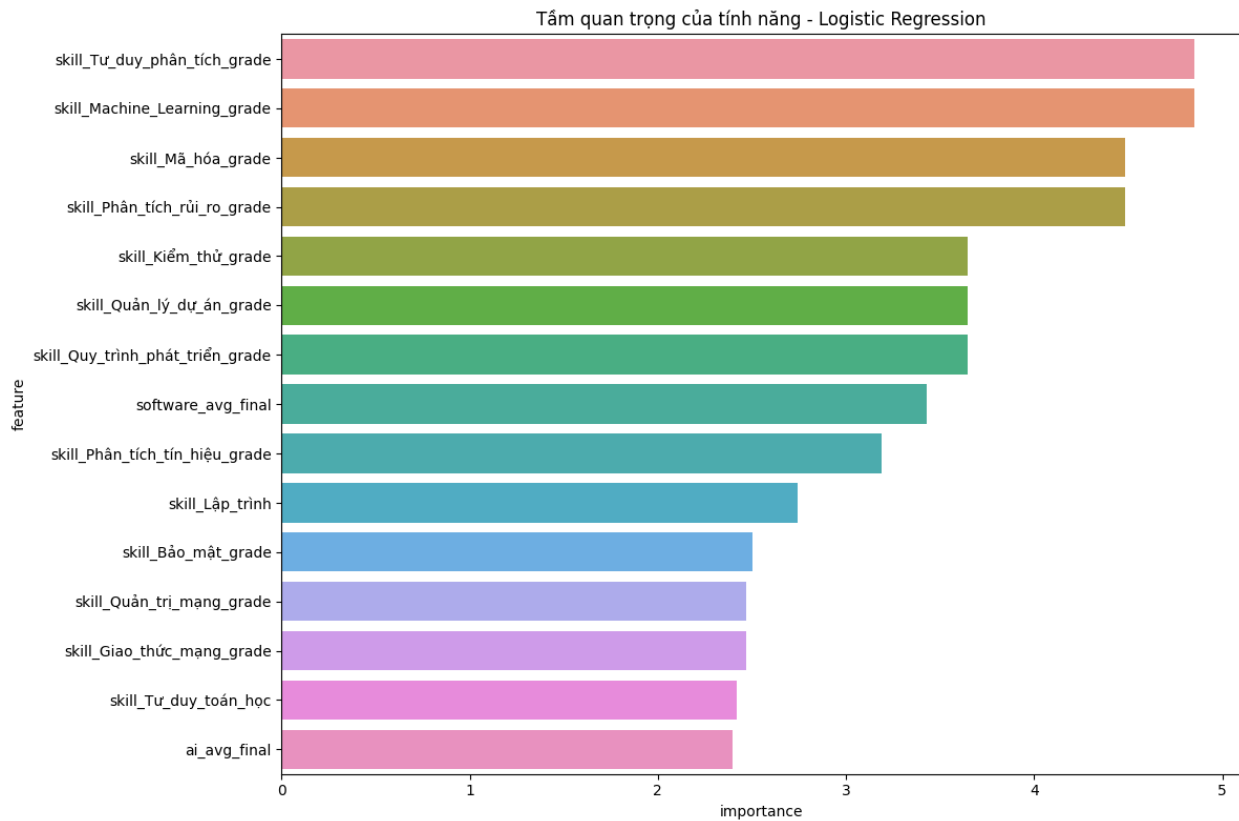
Lớp "Trí tuệ nhân tạo" (Hàng 3):

- Dự đoán đúng: 151
- Nhầm thành "An toàn thông tin": 2
- Nhầm thành "Công nghệ phần mềm": 8 → Hiệu suất chấp nhận được, nhưng còn hiện tượng bị nhầm nhiều hơn so với hai lớp trên.

Kết luận:

- Mô hình Logistic Regression hoạt động tốt, đặc biệt là với lớp "Công nghệ phần mềm".
- Một chút nhầm lẫn xảy ra ở các lớp còn lại, nhưng sai số là nhỏ.
- Dễ dàng nhận thấy mô hình có khả năng phân biệt rõ ràng giữa các lớp chuyên ngành.
- Không có hiện tượng mất cân bằng nghiêm trọng hay sai lệch quá mức.

Tiếp theo sẽ thực hiện việc đánh giá tầm quan trọng của các thuộc tính trong bộ dữ liệu đối với mô hình.



Hình 41: Biểu đồ thể hiện tầm quan trọng của các thuộc tính

Ta có một số nhận xét như sau:

Các đặc trưng có tầm quan trọng cao nhất:

- skill_Tư_duy_phân_tích_grade
- skill_Machine_Learning_grade
- skill_Phân_tích_rủi_ro_grade
- skill_Mã_hóa_grade

→ Đây là các yếu tố mà mô hình Logistic Regression đánh giá có ảnh hưởng mạnh mẽ đến khả năng phân loại.

Các đặc trưng có tầm quan trọng trung bình:

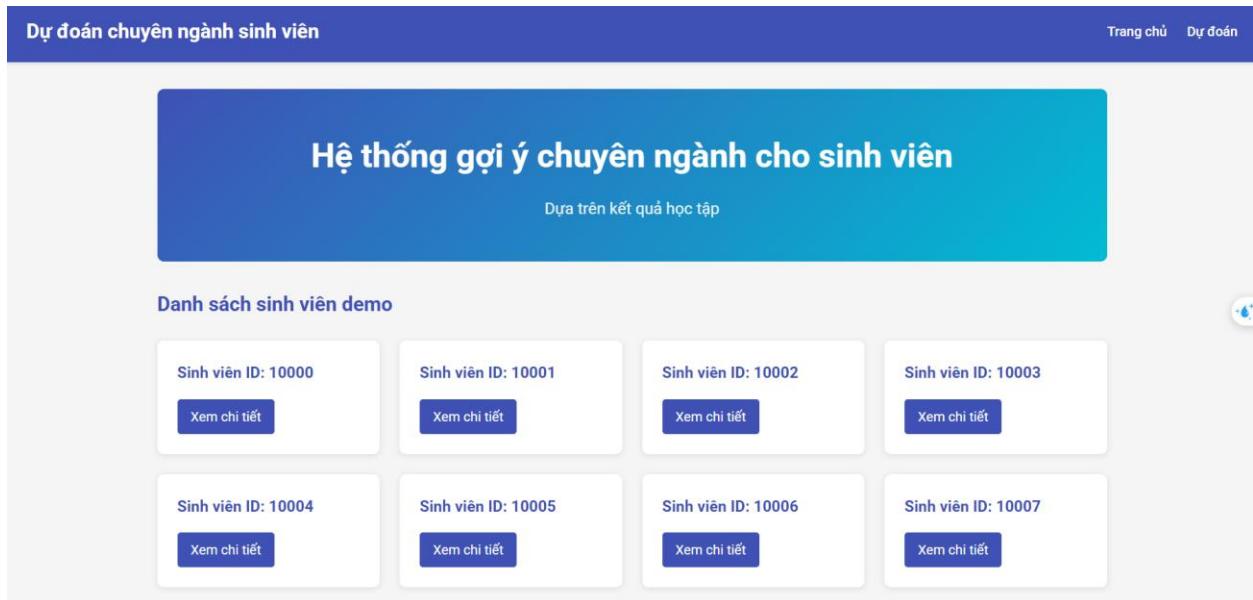
- skill_Quy_trình_phát_triển_grade, skill_Kiểm_thử_grade, skill_Quản_lý_dự_án_grade, skill_Bảo_mật_grade, software_avg_final, v.v.
- Những yếu tố này cũng có ảnh hưởng nhưng không mạnh bằng nhóm đầu.

Đặc trưng có tầm quan trọng thấp nhất:

- skill_Quản_trị_mạng_grade, skill_Giao_thức_mạng_grade, security_avg_final... → Mức ảnh hưởng đến mô hình là nhỏ, có thể ít đóng vai trò phân biệt giữa các nhãn (class).

3. Triển khai và Demo mô hình trên web

Để thân thiện với sinh viên, ta sẽ thiết kế một hệ thống hỗ trợ giao diện thân thiện người dùng như sau.



Hình 42: Giao diện trang chủ của hệ thống gợi ý chuyên ngành

Trong giao diện trang chủ, có một số trường hợp thử nghiệm trong bộ dữ liệu để sinh viên có thể xem xét, đánh giá hệ thống gợi ý chuyên ngành. Sinh viên có thể bấm vào nút xem chi tiết để thực hiện việc xem.

ID: 10001

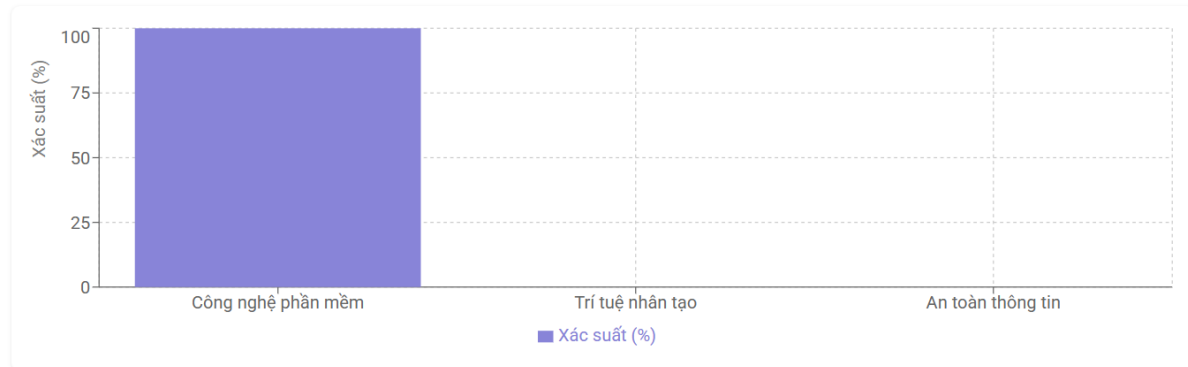
Tên: Ngô Văn Tuấn

Học kỳ hiện tại: 1

GPA: 5.57

Chuyên ngành hiện tại: Công nghệ
phần mềm

Gợi ý chuyên ngành



1. Công nghệ phần mềm

Xác suất: 99.95%

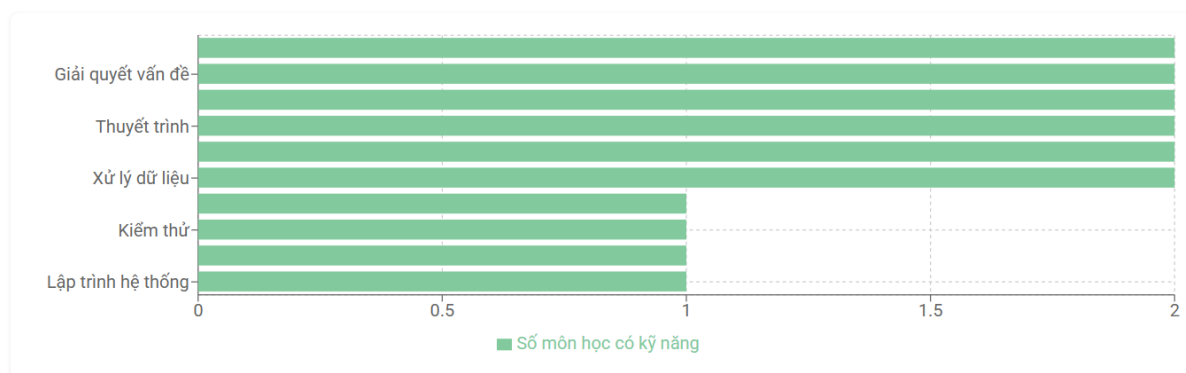
2. Trí tuệ nhân tạo

Xác suất: 0.05%

Hình 43: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (1)

Trong giao diện chi tiết, có thể thấy xác suất phù hợp của từng chuyên ngành đối với sinh viên đó.

Kỹ năng nổi bật



Gợi ý môn học

Môn học mới nên học

Ngôn ngữ lập trình C++	Điểm TB ngành: 5.7
Kỹ năng làm việc nhóm	Điểm TB ngành: 7.0
Mạng máy tính	Điểm TB ngành: 6.8

Hình 44: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (2)

Bên cạnh việc thể hiện trực quan sự phù hợp của từng chuyên ngành đối với sinh viên thì hệ thống còn thể hiện một số kỹ năng nổi bật của sinh viên thông qua việc cung cấp điểm số của các môn học và một số gợi ý về mục tiêu cần đạt được của các môn học chưa được học để phù hợp với chuyên ngành đã gợi ý, điểm số của các môn đã học và một số môn học cần cải thiện để phù hợp với chuyên ngành đã được gợi ý.

Môn học cần cải thiện

Lập trình hướng đối tượng	Điểm của bạn: 3.4 Điểm TB ngành: 5.7
Tiếng Anh	Điểm của bạn: 4.8 Điểm TB ngành: 6.0

Danh sách môn học đã học

Mã môn	Tên môn	Loại môn	Điểm
ELE1330	Xử lý tín hiệu số	technique	8.7
INT1303	An toàn và bảo mật hệ thống thông tin	technique	8.7
INT1306	Cấu trúc dữ liệu và giải thuật	theory	8.4
INT1313	Cơ sở dữ liệu	technique	7.0
SKD1101	Kỹ năng thuyết trình	technique	6.8
BAS1226	Xác suất thống kê	theory	6.2
INT1304	Thiết kế và phân tích thuật toán	theory	6.4

Hình 45: Giao diện chi tiết dự đoán chuyên ngành của một sinh viên (3)

Bên cạnh một số mẫu trường hợp thì sinh viên cũng có thể thực hiện việc nhập thông tin của cá nhân để thực hiện việc dự đoán chuyên ngành.

Danh sách môn học (1/3)

Đại số

BAS1201

3 tín chỉ | core | theory

Xóa

Giải tích

BAS1203

3 tín chỉ | core | theory

Thêm

Tiếng Anh

BAS1156

2 tín chỉ | general | theory

Thêm

Xác suất thống kê

BAS1226

3 tín chỉ | core | theory

Thêm

Cấu trúc dữ liệu và giải thuật

INT1306

3 tín chỉ | core | theory

Thêm

Nhập môn trí tuệ nhân tạo

INT1341

3 tín chỉ | core | theory

Thêm

← Trang trước

Trang 1 / 3

Trang tiếp →

Môn học đã thêm (1)

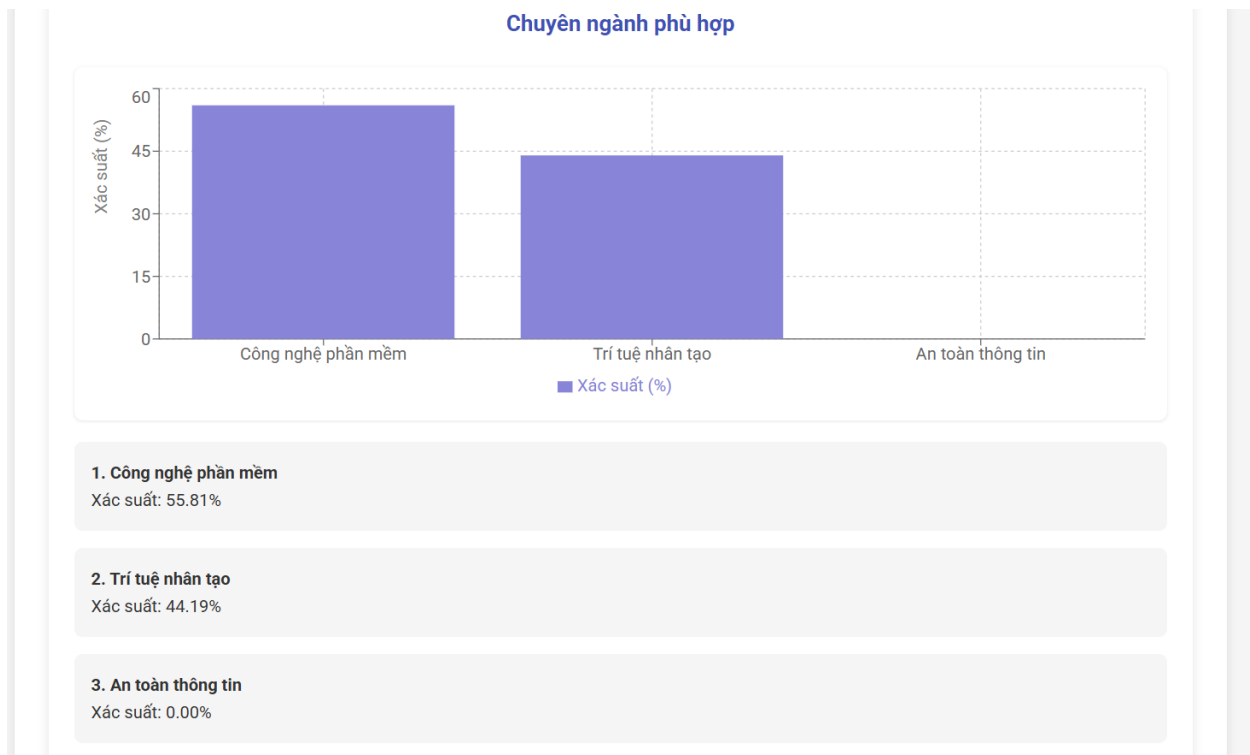
Môn học	Điểm cuối kỳ	Điểm giữa kỳ	Điểm bài tập	Điểm chuyên cần	Số lần học lại	Thao tác
Đại số BAS1201	3	3	3	3	0	Xóa

Dự đoán chuyên ngành

Hình 46: Giao diện nhập thông tin điểm môn học của sinh viên

Hệ thống việc lập những môn học mà sinh viên có trong chương trình đào tạo trước khi thực hiện việc chia chuyên ngành. Sinh viên có thể nhấn chọn vào môn học đó để thực hiện việc nhập điểm số của cá nhân. Sau khi kết thúc việc nhập dữ liệu sinh viên cần nhấn nút dự đoán chuyên ngành để hệ thống thực hiện việc gợi ý chuyên ngành.

Sau khi nhấn nút dự đoán thì hệ thống sẽ thể hiện kết quả thống như trong phần mẫu trường hợp thử nghiệm ở giao diện trang chủ.



Hình 47: Giao diện kết quả dự đoán cho sinh viên

Hệ thống gợi ý cho sinh viên mức độ phù hợp của từng chuyên ngành đối với sinh viên đó, mức độ kỹ năng của sinh viên đó đạt được, một số môn học cần học để phù hợp với chuyên ngành được gợi ý và một số môn học cần cải thiện điểm số.

Hệ thống sẽ cho ra được kết quả tốt nhất khi sinh viên đó đã hoàn thành hết những môn học trước khi thực hiện việc chọn chuyên ngành và ít chính xác hơn đối với trường hợp sinh viên chưa hoàn thành đủ các môn học.

VI. Thảo luận

1. Ý nghĩa

Hỗ trợ định hướng chuyên ngành cho sinh viên: Dựa trên kết quả học tập và đặc điểm môn học, hệ thống có thể đưa ra gợi ý chuyên ngành phù hợp với năng lực và sở thích của sinh viên.

Tối ưu hóa quá trình tư vấn học tập: Giảm tải công việc tư vấn thủ công, giúp cố vấn học tập và nhà trường đưa ra quyết định chính xác hơn.

Nâng cao chất lượng đào tạo: Sinh viên được học đúng chuyên ngành phù hợp, từ đó phát huy năng lực, nâng cao tỷ lệ tốt nghiệp đúng hạn và tìm được việc làm sau ra trường.

2. Các yếu tố ảnh hưởng

Chất lượng dữ liệu đầu vào: Bao gồm điểm số, thông tin môn học, thông tin sinh viên phải đảm bảo đầy đủ, chính xác và đồng nhất.

Sự đa dạng của môn học và chuyên ngành: Cần xây dựng mối liên hệ giữa đặc điểm môn học (lý thuyết, thực hành, kỹ năng mềm, ...) với các chuyên ngành cụ thể.

Thuật toán khai phá dữ liệu: Việc lựa chọn mô hình phân tích dữ liệu phù hợp (ví dụ: phân cụm, phân loại, cây quyết định, mạng nơron nhân tạo) ảnh hưởng lớn đến hiệu quả gợi ý.

Cập nhật dữ liệu thường xuyên: Dữ liệu sinh viên liên tục thay đổi, do đó kho dữ liệu và mô hình khai phá cần được cập nhật định kỳ.

3. Hạn chế và khuyến nghị cải tiến

Yếu tố định lượng:

Hệ thống chỉ dựa trên điểm số mà bỏ qua các yếu tố như sở thích cá nhân, tính cách và đam mê của sinh viên. Điều này có thể dẫn đến việc đánh giá không đầy đủ năng lực và khả năng phù hợp của sinh viên đối với chuyên ngành. Cần bổ sung thêm các yếu tố phi định lượng để cải thiện độ chính xác của mô hình.

Khả năng phản ánh năng lực:

Điểm cao không nhất thiết đồng nghĩa với năng lực phù hợp cho một chuyên ngành, đặc biệt là đối với các ngành đòi hỏi kỹ năng mềm hoặc tư duy sáng tạo. Việc chỉ sử dụng điểm số có thể bỏ sót những khía cạnh quan trọng khác của năng lực cá nhân. Do đó, cần bổ sung thêm các yếu tố đánh giá khác để dự đoán một cách toàn diện hơn.

Chất lượng bộ dữ liệu:

Bộ dữ liệu hiện tại chưa đủ lớn và chất lượng, dẫn đến hệ thống thiếu chính xác trong việc gợi ý chuyên ngành. Cần cải thiện bộ dữ liệu thông qua việc thu thập thêm thông tin, đảm bảo dữ liệu được chuẩn hóa, đầy đủ và cập nhật thường xuyên.

Phản hồi từ người dùng:

Để nâng cao hiệu quả của mô hình gợi ý, cần thu thập phản hồi từ sinh viên sau khi họ lựa chọn chuyên ngành. Những phản hồi này sẽ giúp điều chỉnh và cải thiện mô hình trong tương lai, đảm bảo nó ngày càng phù hợp và chính xác hơn với thực tế.

VII. Kết luận

Việc xây dựng kho dữ liệu dựa trên đặc điểm môn học, điểm số sinh viên đã học và khai phá dữ liệu để gợi ý chọn chuyên ngành là một hướng đi thiết thực và có tính ứng dụng cao trong giáo dục đại học hiện nay. Đề tài không chỉ giúp tăng cường hiệu quả tư vấn học tập, mà còn định hướng đúng đắn cho sinh viên lựa chọn chuyên ngành phù hợp với năng lực và sở thích cá nhân, từ đó nâng cao chất lượng đào tạo và hiệu quả đầu ra cho nhà trường.

Tuy nhiên, để đảm bảo tính hiệu quả và thực tiễn, hệ thống cần chú trọng đến chất lượng dữ liệu đầu vào, đa dạng yếu tố phân tích, cũng như lựa chọn mô hình khai phá dữ liệu phù hợp. Đồng thời, việc cải tiến hệ thống thông qua phản hồi người dùng, cập nhật dữ liệu liên tục và tích hợp các yếu tố định tính là điều cần thiết để nâng cao độ chính xác và tính cá nhân hóa trong các gợi ý chuyên ngành.

Đề tài hứa hẹn sẽ là một công cụ hỗ trợ đắc lực cho công tác quản lý đào tạo và định hướng nghề nghiệp, góp phần hiện đại hóa hoạt động giáo dục trong thời đại chuyển đổi số.

TÀI LIỆU THAM KHẢO

Flask. (2024). *Flask Documentation (Version 3.0.x)*. Pallets Projects.
<https://flask.palletsprojects.com/en/3.0.x/>

Mozilla Developer Network (MDN). (2024). *JavaScript Guide*.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2023). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. <https://scikit-learn.org/stable/modules/ensemble.html#random-forests>

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

Scikit-learn. (2024). *Logistic Regression*. https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

Chollet, F. (2023). *Deep Learning with Python* (2nd ed.). Manning Publications.

TensorFlow. (2024). *TensorFlow Documentation*. <https://www.tensorflow.org/>

Agresti, A. (2019). *An Introduction to Categorical Data Analysis* (3rd ed.). Wiley.

Montgomery, D. C. (2019). *Design and Analysis of Experiments* (10th ed.). Wiley.

Towards Data Science. (2023). *Outlier Detection with Capping and Winsorization*.
<https://towardsdatascience.com/handling-outliers-with-winsorization-abb2629f7a6d>

Towards Data Science. (2023). *Detecting Outliers using the IQR method in Python*.
<https://towardsdatascience.com/identifying-and-handling-outliers-using-python-1c6ecb6c10c0>

Oracle. (2024). *MySQL Documentation*. <https://dev.mysql.com/doc/>