# **ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

# **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**Khai phá dữ liệu trong doanh nghiệp**

**BÁO CÁO ĐỒ ÁN**

**HỆ THỐNG KHUYẾN NGHỊ MÔN HỌC CHO SINH VIÊN UIT TỪ DỮ LIỆU HỌC TẬP**

**Lớp:** DS317.P11

**GVHD:** ThS. Nguyễn Thị Anh Thư

**Nhóm 5:**

1. Phan Thanh Hải MSSV: 22520390

2. Trương Hồng Anh MSSV: 22520084

3. Trương Huỳnh Thúy An MSSV: 22520033

4. Nguyễn Hải Đăng MSSV: 22520189

5. Lê Trần Gia Bảo MSSV: 22520105

🙡🙢 Tp. Hồ Chí Minh, 12/2024 🙠🙣

# **CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

## **1. Giới thiệu**

Trong bối cảnh cách mạng công nghiệp 4.0, việc cá nhân hóa quá trình học tập thông qua hệ thống khuyến nghị môn học đang trở thành nhu cầu thiết yếu trong giáo dục đại học. Hệ thống này giúp sinh viên tối ưu hóa kết quả học tập và giảm thiểu nguy cơ rớt môn bằng cách phân tích dữ liệu học tập và đề xuất các môn học phù hợp với năng lực và sở thích cá nhân.

Đề tài "Hệ thống khuyến nghị môn học cho sinh viên UIT từ dữ liệu học tập" nhằm xây dựng một hệ thống thông minh, hỗ trợ sinh viên lập kế hoạch học tập hiệu quả hơn. Giải pháp này không chỉ giúp cải thiện trải nghiệm học tập của sinh viên mà còn góp phần nâng cao chất lượng giáo dục, tối ưu hóa quy trình đăng ký môn học trong bối cảnh giáo dục số hóa.

## **2. Mô tả đề tài**

### **2.1 Ngữ cảnh**

Trong bối cảnh giáo dục đại học hiện đại, sinh viên thường phải tự đưa ra quyết định chọn môn học dựa trên các yếu tố như chương trình đào tạo, sở thích cá nhân, và kết quả học tập trước đó. Tuy nhiên, tại Trường Đại học Công nghệ Thông tin (ĐHQG HCM), nhiều sinh viên gặp khó khăn trong việc lựa chọn môn học phù hợp với năng lực và mục tiêu học tập của bản thân.

Các khó khăn thường gặp bao gồm:

- Thiếu thông tin tổng quan về độ khó của môn học hoặc mức độ phù hợp với năng lực cá nhân.

- Không nắm rõ các mối liên hệ giữa môn học đã hoàn thành và các môn học tiếp theo trong chương trình đào tạo.

- Khó khăn trong việc dự đoán tác động của việc chọn môn học đến kết quả học tập tổng thể hoặc điểm trung bình học kỳ.

Bên cạnh đó, hệ thống đăng ký môn học hiện tại chủ yếu tập trung vào việc cung cấp thông tin cố định (như lịch học, mô tả môn học) mà chưa có khả năng cá nhân hóa, dẫn đến tình trạng sinh viên lựa chọn môn học một cách cảm tính, không tối ưu hóa được hiệu quả học tập.

Việc triển khai một hệ thống khuyến nghị môn học trong ngữ cảnh này không chỉ giúp giải quyết các khó khăn hiện hữu mà còn góp phần cải thiện trải nghiệm học tập của sinh viên, thúc đẩy khả năng lập kế hoạch học tập dài hạn một cách hiệu quả.

### **2.2 Đối tượng**

Đối tượng chính của đề tài là sinh viên đang theo học tại Trường Đại học Công nghệ Thông tin (ĐHQG HCM). Hệ thống được thiết kế để hỗ trợ sinh viên thuộc nhiều ngành học khác nhau trong việc lựa chọn môn học phù hợp dựa trên dữ liệu học tập cá nhân. Ngoài ra, hệ thống cũng có thể được mở rộng để phục vụ các giảng viên và cố vấn học tập trong việc hỗ trợ và theo dõi tiến độ học tập của sinh viên.

### **2.3 Phạm vi**

- Dữ liệu: Dữ liệu được sử dụng trong hệ thống bao gồm thông tin học tập lịch sử của sinh viên, thông tin về môn học và chương trình đào tạo của Trường Đại học Công nghệ Thông tin.

- Ứng dụng: Đề tài tập trung vào việc xây dựng hệ thống khuyến nghị gợi ý môn học cho sinh viên, áp dụng thử nghiệm trên dữ liệu của sinh viên UIT.

- Thời gian: Phạm vi nghiên cứu tập trung vào dữ liệu từ các học kỳ gần đây để đảm bảo tính cập nhật và liên quan của thông tin.

- Giới hạn: Hệ thống ban đầu chỉ hỗ trợ khuyến nghị môn học cho sinh viên trong khuôn khổ các ngành học tại UIT, chưa mở rộng ra các trường hoặc hệ thống giáo dục khác.

- Công nghệ: Hệ thống sẽ được triển khai dưới dạng website với công nghệ backend Django và frontend React, tích hợp các thuật toán học máy để đưa ra khuyến nghị chính xác.

## **3. Tổng quan**

### **3.1 Tính cấp thiết**

Trong quá trình học tập tại đại học, việc sinh viên chọn môn học không phù hợp với năng lực cá nhân là tình trạng phổ biến, dẫn đến chán nản, thiếu hứng thú và có nguy cơ rớt môn. Rớt môn không chỉ ảnh hưởng đến kết quả học tập, khiến sinh viên phải học lại, mà còn tác động tiêu cực đến tâm lý, giảm sự tự tin và làm suy giảm tinh thần học tập. Nghiêm trọng hơn, việc thiếu tín chỉ hoặc kéo dài thời gian tốt nghiệp có thể ảnh hưởng xấu đến cơ hội nghề nghiệp sau này.

Do đó, nhu cầu về một hệ thống khuyến nghị môn học là cần thiết hơn bao giờ hết. Hệ thống này có thể thực hiện tính toán và phân tích dữ liệu học tập của sinh viên, đưa ra các gợi ý môn học phù hợp với năng lực và sở thích cá nhân, giúp sinh viên lập kế hoạch học tập tối ưu và giảm thiểu nguy cơ rớt môn. Thông qua hệ thống, sinh viên có thể lựa chọn môn học phù hợp hơn, giảm áp lực tâm lý và cải thiện trải nghiệm học tập.

### **3.2 Tính mới**

Đề án không chỉ dừng lại ở việc khuyến nghị các môn học phù hợp mà còn cung cấp một website hỗ trợ sinh viên đăng ký môn học thông minh. Với các tính năng này, hệ thống khuyến nghị không chỉ giúp nâng cao chất lượng học tập mà còn đáp ứng được yêu cầu về tính minh bạch, tiết kiệm thời gian và chi phí học tập, mang lại trải nghiệm giáo dục toàn diện và thông minh cho sinh viên trong thời đại số hóa.

## **4. Mô tả bài toán**

### **Định nghĩa:** Bài toán xây dựng một hệ thống khuyến nghị môn học thông minh nhằm hỗ trợ sinh viên tối ưu hóa kết quả học tập. Hệ thống sử dụng dữ liệu lịch sử học tập của sinh viên, kết hợp với thông tin môn học, để gợi ý các môn phù hợp cho kỳ học tiếp theo.

### **Ý tưởng:** Hệ thống phân tích dữ liệu học tập bao gồm bảng điểm, lịch sử đăng ký môn học, và các yếu tố liên quan như tín chỉ, trạng thái môn học, ngành học, khóa học. Dựa vào các thuật toán học máy, hệ thống xác định mối quan hệ giữa các môn học và kết quả đạt được để đưa ra danh sách gợi ý cá nhân hóa.

### **Input:** Thông tin của sinh viên, bảng điểm các kỳ học trước và thông tin học kỳ sinh viên muốn được gợi ý, cụ thể:

- Thông tin sinh viên: mã số sinh viên, giới tính, khoa, hệ đào tạo, ngành học, niên khóa

- Bảng điểm của sinh viên (trước kỳ học gợi ý): mã môn học, số tín chỉ, điểm, trạng thái (bình thường, học cải thiện, học trả nợ), học kỳ, năm học của các môn.

- Kỳ học sinh viên muốn được gợi ý.

**Output:** Danh sách các môn học được khuyến nghị, bao gồm mã môn học, tên môn, số tín chỉ, và dự đoán về khả năng cải thiện điểm số khi chọn môn học đó.



## **5. Mục tiêu đề tài**

- Tìm hiểu và phân tích bộ dữ liệu UIT, tích hợp và xử lý bộ dữ liệu UIT

- Tìm hiểu các phương pháp dựa trên máy học nhằm xây dựng hệ thống phân tích và khuyến nghị môn học mà sinh viên nên đăng ký, dựa trên dữ liệu học tập cá nhân nhằm tối hóa kết quả học tập, hỗ trợ sinh viên lập kế hoạch học tập phù hợp với năng lực và nhu cầu của mình.​

- Thử nghiệm một số phương pháp học máy nhằm nâng cao độ chính xác của mô hình.

- Xây dựng demo hệ thống được dựng bằng django với react

## **6. Thách thức**

- Thu thập và xử lý dữ liệu: Bộ dữ liệu học tập của sinh viên thường không đồng nhất, có thể bị thiếu thông tin hoặc chứa dữ liệu không hợp lệ. Việc làm sạch, chuẩn hóa và tích hợp dữ liệu từ nhiều nguồn khác nhau là một thử thách lớn để đảm bảo chất lượng dữ liệu đầu vào.

- Phân tích và lựa chọn phương pháp: Lựa chọn phương pháp phù hợp để đảm bảo độ chính xác cao trong việc khuyến nghị môn học.

- Đảm bảo tính cá nhân hóa: Mỗi sinh viên có năng lực, sở thích và lộ trình học tập riêng. Do đó, hệ thống cần phân tích sâu sắc các yếu tố này để đưa ra khuyến nghị phù hợp, thay vì chỉ dựa vào các tiêu chí chung.

- Khả năng giải thích kết quả: Để hệ thống được sinh viên và nhà trường tin tưởng, các khuyến nghị cần dễ hiểu và có thể giải thích được. Điều này đòi hỏi xây dựng mô hình không chỉ chính xác mà còn minh bạch.

- Xây dựng giao diện thân thiện: Website hỗ trợ đăng ký môn học cần được thiết kế sao cho dễ sử dụng, trực quan, và phù hợp với nhu cầu của người dùng.

- Đảm bảo hiệu suất hệ thống: Hệ thống cần xử lý nhanh chóng, đặc biệt khi có lượng lớn sinh viên sử dụng đồng thời, để đảm bảo không ảnh hưởng đến trải nghiệm người dùng.

- Bảo mật dữ liệu: Vì dữ liệu cá nhân của sinh viên được sử dụng, đảm bảo bảo mật thông tin là một thách thức quan trọng, đòi hỏi các biện pháp bảo vệ dữ liệu mạnh mẽ.

- Đánh giá hiệu quả hệ thống: Khó khăn trong việc đo lường chính xác mức độ cải thiện điểm số và sự hài lòng của sinh viên đối với các khuyến nghị được đưa ra.

- Hạn chế tài nguyên: Việc xây dựng và vận hành một hệ thống với nhiều tính năng đòi hỏi nguồn lực tài chính, nhân lực và kỹ thuật đáng kể, đây có thể là rào cản trong giai đoạn phát triển.

# **CHƯƠNG II: PHƯƠNG PHÁP ĐỀ XUẤT**

## **1. Baseline 1 - Phương pháp ma trận tương đồng Cosine (Ph**ương pháp đề xuất)

Dữ liệu baseline 1 sử dụng bao gồm dữ liệu thông tin cá nhân sinh viên, dữ liệu mô tả môn học, dữ liệu kết quả học tập, dữ liệu thống kê độ phổ biến của các môn học, dữ liệu thống kê số môn học trung bình theo khoa và kỳ học **(tính từ kỳ học cần dự đoán trở về trước).**

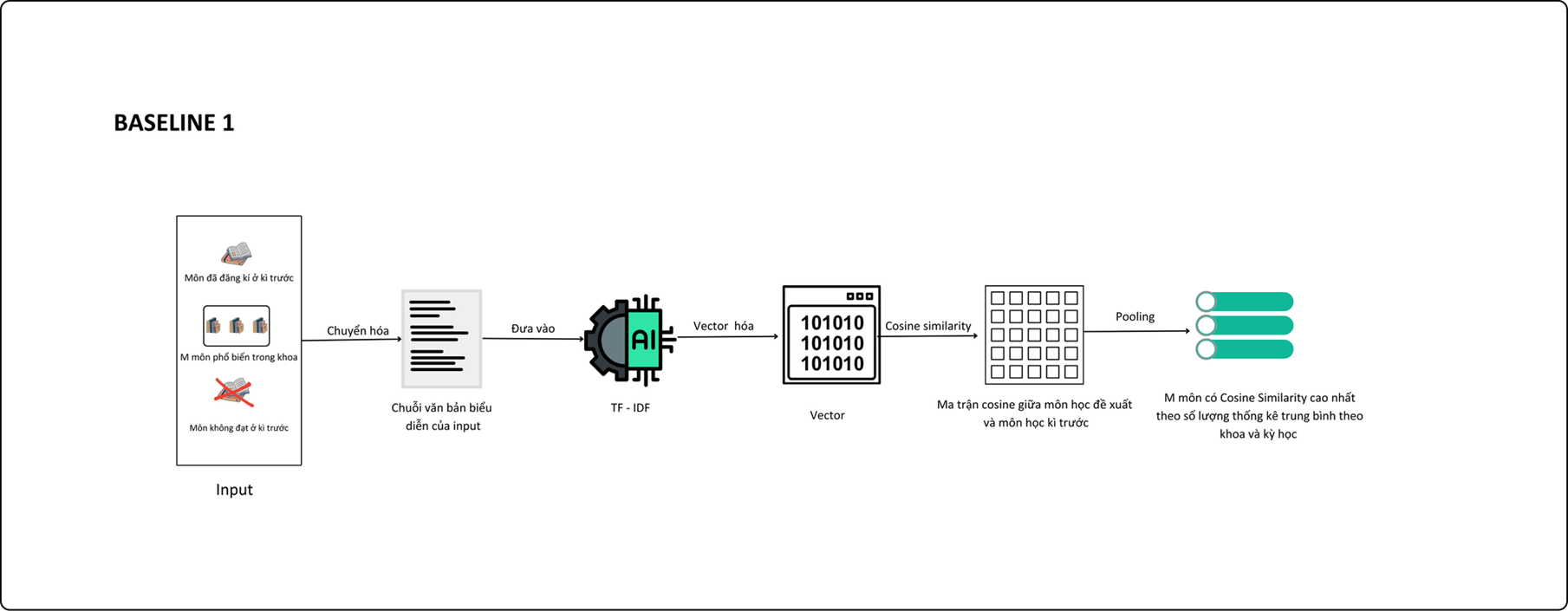
Quá trình chuẩn bị danh sách môn học bao gồm hai phần chính. Trước tiên, danh sách các môn học đã học được xây dựng dựa trên các môn mà sinh viên đã đăng ký ở kỳ học trước, giúp xác định lịch sử học tập của từng sinh viên. Tiếp theo, danh sách môn học đề xuất được tạo ra bằng cách kết hợp top **M** các môn phổ biến nhất trong khoa với các môn mà sinh viên chưa đạt yêu cầu **(điểm < 5)** ở kỳ trước. Điều này đảm bảo sinh viên có lộ trình học tập phù hợp, vừa củng cố kiến thức chưa đạt, vừa định hướng học các môn trọng tâm trong chương trình đào tạo. Lưu ý, các môn học trong danh sách đề xuất không trùng lặp với các môn học mà sinh viên đã đăng ký trong các kỳ học trước đó.

Quá trình vector hóa dữ liệu được thực hiện bằng cách kết hợp tất cả các đặc trưng liên quan đến môn học, bao gồm mã môn học **(mamh)**, ngành môn học **(nganhmh)**, loại môn học **(loaimh)**, và nhóm loại môn học **(nhomloaimh)** thành một chuỗi văn bản duy nhất. Chuỗi này đại diện cho thông tin tổng hợp về mỗi môn học, tạo điều kiện thuận lợi cho các bước xử lý tiếp theo. Sau đó, các đặc trưng này được chuyển đổi thành dạng vector bằng phương pháp **Word Embedding**, cụ thể là sử dụng kỹ thuật **TF-IDF**. Phương pháp này không chỉ giúp biểu diễn dữ liệu dưới dạng số hóa mà còn giữ lại mức độ quan trọng tương đối của từng đặc trưng trong tập dữ liệu.

Để đánh giá sự tương đồng giữa môn học được đề xuất và các môn đã học trong kỳ trước, sử dụng **ma trận tương đồng cosine (Cosine Similarity Matrix)**. Phương pháp này đo lường mức độ giống nhau giữa hai vector đặc trưng của các môn học. Trước tiên, các đặc trưng của môn học được mã hóa thành vector, sau đó tính giá trị **cosine similarity**, với giá trị nằm trong khoảng từ -1 đến 1. Giá trị càng gần 1 thể hiện mức độ tương đồng càng cao, giúp xác định các môn học phù hợp với sở thích và năng lực của sinh viên dựa trên lịch sử học tập.

Để tổng hợp **ma trận tương đồng cosine** và đánh giá độ phù hợp của các môn học đề xuất, áp dụng các **hàm tổng hợp (aggregate functions) như max, min, mean, và sum**. Hàm max giúp tìm mức độ tương đồng cao nhất giữa môn học đề xuất và bất kỳ môn đã học nào, thể hiện mối liên quan chặt chẽ nhất. Hàm min xác định mức tương đồng thấp nhất, làm nổi bật các khác biệt lớn nhất. Hàm mean tính giá trị trung bình, cung cấp cái nhìn tổng quan về mức độ phù hợp tổng thể, trong khi hàm sum cộng dồn toàn bộ giá trị tương đồng, phản ánh mức tương quan tích lũy. Việc sử dụng các hàm này giúp phân tích chi tiết và đưa ra quyết định dựa trên nhiều góc nhìn khác nhau.

Để gợi ý môn học phù hợp, chọn ra **N môn học (thống kê số môn học trung bình theo khoa và kỳ học)** có độ tương đồng cao nhất dựa trên **ma trận tương đồng cosine**. Sau khi tính toán mức độ tương đồng giữa các môn học đề xuất và môn đã học, sắp xếp các giá trị theo thứ tự giảm dần. Các môn học nằm trong **top N** sẽ được chọn làm gợi ý, vì chúng có mối liên quan chặt chẽ nhất với lịch sử học tập của sinh viên. Cách tiếp cận này giúp đảm bảo các môn học được đề xuất phù hợp với sở thích, năng lực, và định hướng học thuật của sinh viên.



*Baseline 1 - Phương pháp ma trận tương đồng Cosine*

## **2. Baseline 2 - Phương pháp ma trận tương đồng kết hợp thuật toán phân cụm K-Means**

Dữ liệu baseline 2 sử dụng bao gồm dữ liệu thông tin cá nhân sinh viên, dữ liệu mô tả môn học, dữ liệu kết quả học tập, dữ liệu thống kê độ phổ biến của các môn học, dữ liệu thống kê số môn học trung bình theo khoa và kỳ học **(tính từ kỳ học cần dự đoán trở về trước).**

**Phân cụm K-Means** có thể được áp dụng để phân chia sinh viên thành k nhóm dựa trên dữ liệu kết quả học tập trong quá khứ, chẳng hạn như điểm số, số tín chỉ hoàn thành, hoặc hiệu suất theo môn học. Quá trình này bao gồm việc chuyển đổi dữ liệu học tập thành các vector đặc trưng và áp dụng thuật toán **K-Means** để xác định các cụm sinh viên có đặc điểm tương đồng. Sau khi phân cụm, sinh viên đang được khuyến nghị sẽ được xếp vào một trong **k** nhóm bằng cách tính khoảng cách giữa vector đặc trưng của họ và tâm cụm **(centroid)** của các nhóm đã xác định. Việc này giúp cá nhân hóa các gợi ý môn học dựa trên nhóm sinh viên có kết quả học tập tương tự.

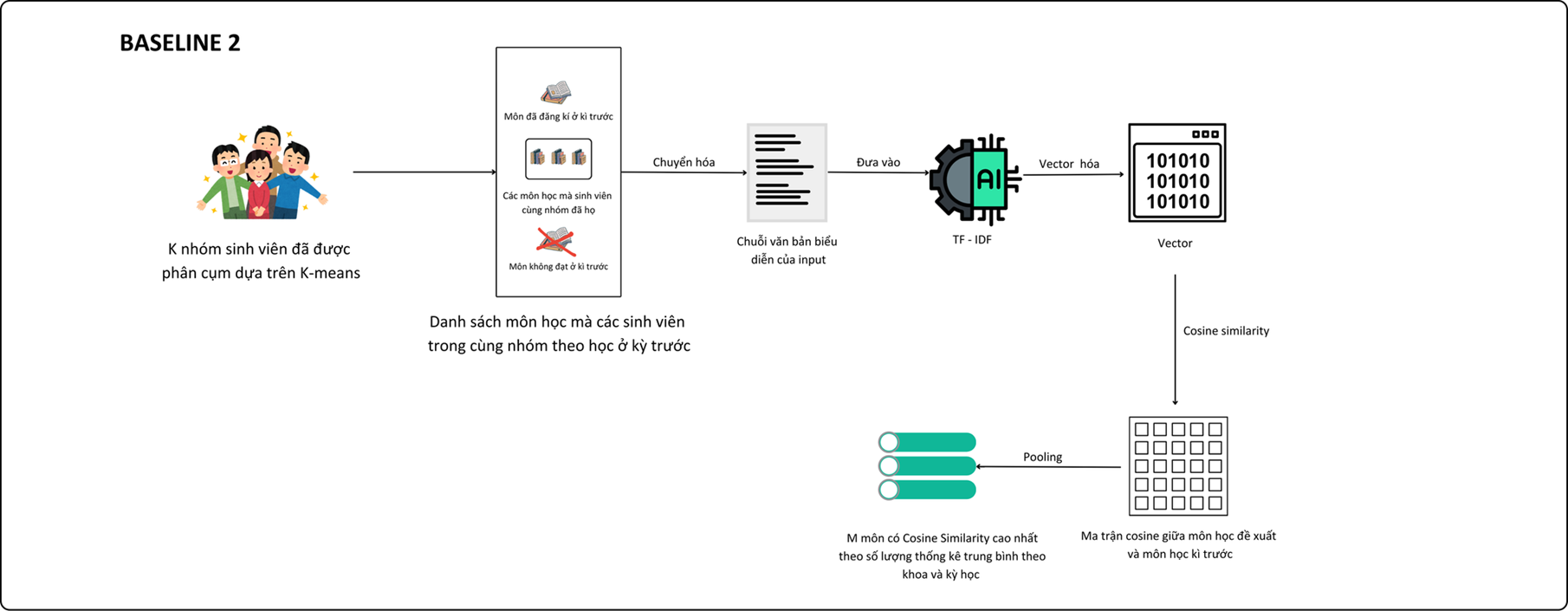
Để chuẩn bị danh sách môn học cho sinh viên, cần xây dựng hai danh sách chính. Thứ nhất, danh sách môn học đã học bao gồm các môn mà sinh viên đã đăng ký ở kỳ học trước, ghi nhận cả kết quả đạt và không đạt. Thứ hai, danh sách môn học đề xuất được xây dựng dựa trên hai nguồn: các môn học mà sinh viên trong cùng nhóm **(phân cụm)** đã hoàn thành ở kỳ trước, và các môn mà sinh viên không đạt **(điểm <5)** ở kỳ trước. Lưu ý, các môn học trong danh sách đề xuất không trùng lặp với các môn học mà sinh viên đã đăng ký trong các kỳ học trước đó.

Để vector hóa dữ liệu môn học, tất cả các đặc trưng như mã môn học **(mamh)**, ngành môn học **(nganhmh)**, loại môn học **(loaimh)**, và nhóm loại môn học **(nhomloaimh)** được ghép lại thành một chuỗi văn bản duy nhất đại diện cho mỗi môn học. Sau đó, phương pháp **Word Embedding** như **TF-IDF (Term Frequency-Inverse Document Frequency)** được áp dụng để chuyển đổi các chuỗi này thành các vector số. **TF-IDF** giúp biểu diễn mức độ quan trọng của từng từ trong mỗi đặc trưng, đồng thời giảm thiểu sự ảnh hưởng của các từ xuất hiện thường xuyên nhưng ít giá trị phân biệt. Vector hóa này là bước tiền xử lý cần thiết để sử dụng trong các thuật toán phân tích hoặc mô hình dự đoán.

Để đánh giá sự tương đồng giữa các môn học đề xuất và môn đã học kỳ trước, ma trận tương đồng cosine **(Cosine Similarity Matrix)** được sử dụng. Phương pháp này đo lường mức độ tương đồng giữa các vector đặc trưng của môn học, với giá trị cosine dao động từ -1 đến 1, trong đó 1 thể hiện sự tương đồng cao nhất. Đầu tiên, các đặc trưng môn học được vector hóa, sau đó tính **cosine similarity** giữa từng cặp môn học. Kết quả giúp xác định các môn học đề xuất có mức độ liên quan cao nhất với lịch sử học tập của sinh viên, hỗ trợ quá trình chọn lọc các gợi ý phù hợp.

Việc tổng hợp ma trận tương đồng cosine theo các môn học đề xuất bằng cách áp dụng các **hàm tổng hợp (aggregate functions) như max, min, mean và sum** mang lại cái nhìn khái quát và toàn diện về mức độ liên kết giữa các môn học. Cụ thể, hàm **max** xác định giá trị tương đồng lớn nhất giữa các cặp môn học, qua đó nhấn mạnh những mối quan hệ chặt chẽ nhất trong tập hợp. Ngược lại, hàm **min** làm rõ các cặp môn học có mức độ tương đồng thấp nhất, cho thấy những điểm ít liên quan nhất trong ma trận. Hàm **mean**, với vai trò tính giá trị trung bình của các độ tương đồng, cung cấp một chỉ số tổng quát về mức độ tương đồng trung bình, phản ánh mức độ đồng nhất chung giữa các môn học. Cuối cùng, hàm **sum** tổng hợp toàn bộ giá trị tương đồng, thể hiện mức độ liên kết tổng quát trong hệ thống các môn học được đề xuất.

Để đề xuất các môn học phù hợp cho sinh viên, một phương pháp hiệu quả là lựa chọn ra **N môn học (thống kê số môn học trung bình theo khoa và kỳ học)** có độ tương đồng cao nhất dựa trên **ma trận tương đồng cosine**. Cụ thể, các cặp môn học có giá trị tương đồng cao thể hiện mức độ liên quan mạnh mẽ về nội dung, kiến thức hoặc kỹ năng, từ đó tạo cơ sở để xác định những môn học bổ trợ lẫn nhau hoặc phù hợp với sở thích và nhu cầu học tập của sinh viên. Việc lựa chọn các môn học này có thể được thực hiện thông qua việc xếp hạng độ tương đồng của tất cả các môn trong ma trận và chọn ra **N** môn đứng đầu danh sách. Cách tiếp cận này không chỉ giúp cá nhân hóa lộ trình học tập mà còn tối ưu hóa khả năng tiếp cận và tiếp thu kiến thức liên ngành của sinh viên, đồng thời tăng tính hiệu quả trong việc xây dựng các chương trình học tập linh hoạt và phù hợp.



*Baseline 2 - Phương pháp ma trận tương đồng kết hợp thuật toán phân cụm K-Means*

## 3. Baseline 3 - Phương pháp ma trận tương đồng Cosine theo nhóm loại môn học kết hợp mô hình ngôn ngữ lớn PhoBERT

Dữ liệu baseline 3 sử dụng bao gồm dữ liệu thông tin cá nhân sinh viên, dữ liệu mô tả môn học, dữ liệu kết quả học tập, dữ liệu thống kê số môn học trung bình theo khoa, kỳ học và nhóm loại môn học **(tính từ kỳ học cần dự đoán trở về trước).**

Môn học được phân loại thành **5** nhóm chính: **Đại cương (ĐC), Tự chọn (TC), Cơ sở ngành (CSN), Chuyên ngành (CN), và Tốt nghiệp (TN).** Số lượng môn học **M** từ mỗi nhóm sẽ được xác định dựa trên thống kê trung bình, dựa vào dữ liệu cụ thể của từng khoa, kỳ học, và nhóm môn. Phương pháp này giúp đảm bảo việc phân bổ môn học hợp lý và phù hợp với nhu cầu đào tạo của từng chuyên ngành.

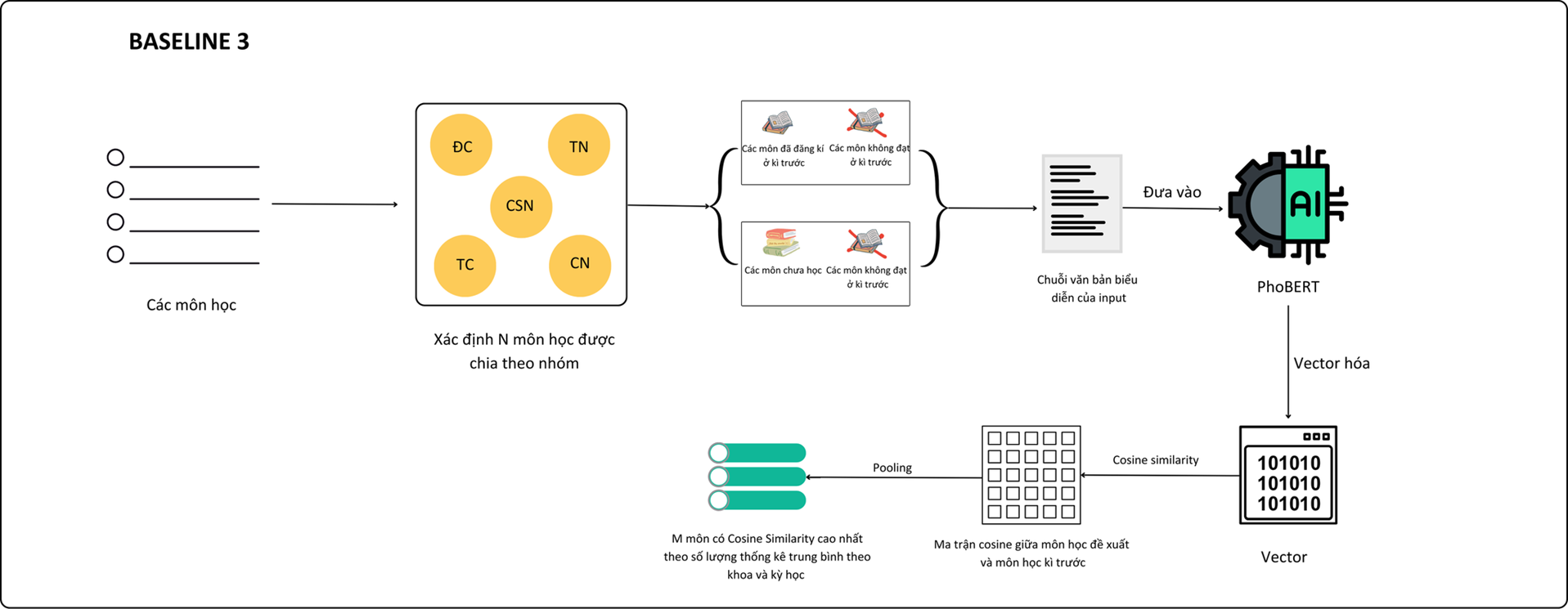
Danh sách môn học được chuẩn bị theo từng nhóm bao gồm: **Môn học đã học**, là các môn mà sinh viên đã đăng ký và hoàn thành trong các kỳ học trước; và **Môn học đề xuất**, bao gồm các môn học cùng nhóm mà sinh viên chưa đăng ký ở các kỳ trước, kết hợp với các môn mà sinh viên chưa đạt yêu cầu **(điểm dưới 5)** trong kỳ trước. Lưu ý, các môn học trong danh sách đề xuất không trùng lặp với các môn học mà sinh viên đã đăng ký trong các kỳ học trước đó.

Quá trình vector hóa dữ liệu bao gồm các bước xử lý đặc trưng: Đối với các đặc trưng liên tục như mã môn học **(mamh)** và số tín chỉ **(sotc)**, dữ liệu được chuẩn hóa để đưa về cùng thang đo. Các đặc trưng rời rạc như ngành môn học **(nganhmh)**, loại môn học **(loaimh)**, và nhóm loại môn học **(nhomloaimh)** được xử lý bằng phương pháp **One-hot Encoding**. Ngoài ra, nội dung môn học được tóm tắt bằng **Word Embedding** thông qua mô hình **PhoBERT** để trích xuất ý nghĩa ngữ nghĩa. Cuối cùng, tất cả các vector đặc trưng được chuẩn hóa về cùng thang đo nhằm đảm bảo tính đồng nhất và hỗ trợ so sánh hiệu quả.

Sự tương đồng giữa các môn học được đánh giá bằng cách sử dụng **ma trận tương đồng cosine** **(Cosine Similarity Matrix)**. Phương pháp này tính toán mức độ tương đồng giữa các vector đặc trưng của môn học đề xuất và các môn đã học ở kỳ trước, dựa trên góc giữa các vector trong không gian đặc trưng. Kết quả giúp xác định mức độ liên quan giữa các môn, hỗ trợ sinh viên lựa chọn môn học phù hợp với kiến thức nền tảng đã có.

Để tổng hợp **ma trận tương đồng cosine**, các **hàm tổng hợp (aggregate functions)** như **max**, **min**, **mean**, và **sum** được áp dụng. Các hàm này giúp tính toán các giá trị đại diện cho độ tương đồng giữa từng môn học đề xuất và toàn bộ các môn đã học trước đó. Kết quả tổng hợp cung cấp thông tin trực quan, hỗ trợ việc đánh giá mức độ phù hợp của từng môn học đề xuất dựa trên kiến thức nền tảng của sinh viên.

Dựa trên **ma trận tương đồng cosine** đã tổng hợp, hệ thống sẽ chọn ra **N** môn học có độ tương đồng cao nhất để đề xuất cho sinh viên. Những môn học này được đánh giá là phù hợp nhất với nền tảng kiến thức hiện tại của sinh viên, giúp tối ưu hóa quá trình học tập và nâng cao hiệu quả trong việc lựa chọn môn học.



*Baseline 3 - Phương pháp ma trận tương đồng Cosine theo nhóm loại môn học kết hợp mô hình ngôn ngữ lớn PhoBERT*

## 4. Baseline 4 - Phương pháp Re-rank ma trận tương đồng Cosine theo nhóm loại môn học kết hợp mô hình ngôn ngữ lớn PhoBERT

Dữ liệu baseline 4 sử dụng bao gồm dữ liệu thông tin cá nhân sinh viên, dữ liệu mô tả môn học, dữ liệu kết quả học tập, dữ liệu thống kê môn học theo độ phổ biến và theo điểm, dữ liệu thống kê số môn học trung bình theo khoa, kỳ học và nhóm loại môn học **(tính từ kỳ học cần dự đoán trở về trước).**

Môn học trong hệ thống đào tạo được phân loại thành **5** nhóm chính: **Đại cương (ĐC), Tự chọn (TC), Cơ sở ngành (CSN), Chuyên ngành (CN), và Tốt nghiệp (TN)**. Việc phân loại này giúp xác định rõ ràng cấu trúc các môn học trong chương trình đào tạo. Số lượng **N** môn học từ mỗi nhóm được xác định dựa trên thống kê trung bình theo các yếu tố như khoa, kỳ học, và nhóm môn. Phương pháp này giúp tối ưu hóa việc phân bổ môn học, đảm bảo tính cân đối và phù hợp với yêu cầu học tập của từng chuyên ngành và từng kỳ học.

Danh sách môn học được chuẩn bị theo từng nhóm loại môn học bao gồm hai loại chính: Môn học đã học và Môn học đề xuất. Môn học đã học là các môn mà sinh viên đã đăng ký và hoàn thành trong kỳ học trước. Trong khi đó, môn học đề xuất bao gồm các môn mà sinh viên trong cùng nhóm đã học ở kỳ trước, kết hợp với các môn mà sinh viên không đạt yêu cầu **(điểm dưới 5)**. Lưu ý, các môn học trong danh sách đề xuất không trùng lặp với các môn học mà sinh viên đã đăng ký trong các kỳ học trước đó.

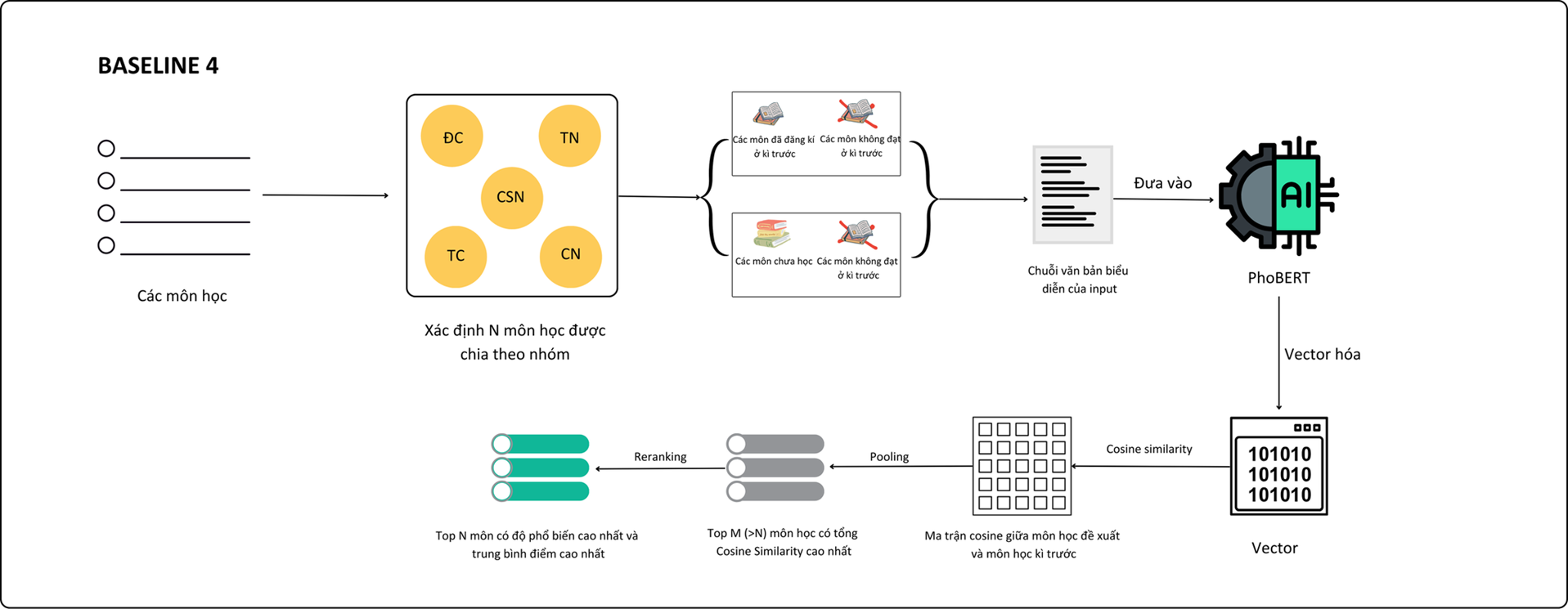
Quá trình vector hóa dữ liệu bao gồm nhiều bước xử lý để chuyển đổi các đặc trưng thành dạng có thể so sánh được. Đối với các đặc trưng liên tục như mã môn học **(mamh)** và số tín chỉ **(sotc)**, dữ liệu được chuẩn hóa để đưa về cùng một thang đo. Các đặc trưng rời rạc như ngành môn học **(nganhmh)**, loại môn học **(loaimh)**, và nhóm loại môn học **(nhomloaimh)** được mã hóa bằng phương pháp **One-hot Encoding**. Nội dung môn học được tóm tắt thông qua **Word Embedding** sử dụng mô hình **PhoBERT**, giúp chuyển đổi văn bản thành các vector có ý nghĩa ngữ nghĩa. Cuối cùng, tất cả các vector đặc trưng được chuẩn hóa để so sánh, đảm bảo tính nhất quán và hỗ trợ đánh giá chính xác mức độ tương đồng giữa các môn học.

Để đánh giá sự tương đồng giữa các môn học, chúng tôi sử dụng **ma trận tương đồng cosine (Cosine Similarity Matrix)**. Phương pháp này tính toán mức độ tương đồng giữa các vector đặc trưng của môn học đề xuất và các môn học đã hoàn thành trong kỳ trước. Tính toán dựa trên góc giữa các vector trong không gian đặc trưng, cho phép xác định mức độ liên quan giữa môn học đề xuất và các môn đã học. Điều này giúp hệ thống lựa chọn các môn học có nội dung và kiến thức tương tự, từ đó đề xuất các môn học phù hợp nhất với nền tảng kiến thức của sinh viên.

Sau khi tính toán ma trận tương đồng cosine, chúng tôi áp dụng các hàm tổng hợp **(aggregate functions)** như **max**, **min**, **mean**, và **sum** để tổng hợp độ tương đồng giữa các môn học đề xuất. Các hàm này giúp rút ra các giá trị đại diện cho mức độ tương đồng của từng môn học đề xuất với các môn đã học trong kỳ trước.

Dựa trên ma trận tương đồng và các chỉ số đánh giá, hệ thống sẽ chọn ra **M(> N)** môn học có độ tương đồng cao nhất với các môn học mà sinh viên đã học trong kỳ trước để đề xuất. Những môn học này được xác định là phù hợp nhất với nền tảng kiến thức của sinh viên, giúp họ dễ dàng tiếp cận và học tiếp các nội dung có liên quan, đồng thời cải thiện kết quả học tập. Việc chọn lọc các môn học có độ tương đồng cao không chỉ tối ưu hóa quá trình học tập mà còn giúp sinh viên có lộ trình học phù hợp và hiệu quả.

Sau chọn được danh sách môn học đề xuất từ ma trận tương đồng, danh sách **M** môn học đề xuất sẽ được sắp xếp lại và chọn ra **N** môn học dựa trên hai yếu tố chính: độ phổ biến và trung bình điểm. Môn học có độ phổ biến cao, tức là có nhiều sinh viên đăng ký và hoàn thành tốt, sẽ được ưu tiên đưa lên đầu danh sách. Đồng thời, những môn học có điểm trung bình cao cũng sẽ được xếp ưu tiên hơn, giúp sinh viên dễ dàng lựa chọn các môn học có chất lượng và khả năng hoàn thành tốt. Việc sắp xếp lại này nhằm tối ưu hóa việc lựa chọn môn học, giúp sinh viên dễ dàng đạt kết quả học tập cao nhất.



*Baseline 4 - Phương pháp Re-rank ma trận tương đồng Cosine theo nhóm loại môn học kết hợp mô hình ngôn ngữ lớn PhoBERT*

## 5. Baseline 5 - Phương pháp sử dụng mô hình học sâu MLP theo nhóm loại môn học

Dữ liệu baseline 5 sử dụng bao gồm dữ liệu thông tin cá nhân sinh viên, dữ liệu mô tả môn học, dữ liệu kết quả học tập, dữ liệu thống kê môn học theo độ phổ biến, dữ liệu thống kê số môn học trung bình theo khoa, kỳ học và nhóm loại môn học **(tính từ kỳ học cần dự đoán trở về trước).**

Môn học trong hệ thống đào tạo được phân loại thành **5** nhóm chính: **Đại cương (ĐC), Tự chọn (TC), Cơ sở ngành (CSN), Chuyên ngành (CN), và Tốt nghiệp (TN)**. Việc phân loại này giúp xác định rõ ràng cấu trúc các môn học trong chương trình đào tạo. Số lượng **N** môn học từ mỗi nhóm được xác định dựa trên thống kê trung bình theo các yếu tố như khoa, kỳ học, và nhóm môn. Phương pháp này giúp tối ưu hóa việc phân bổ môn học, đảm bảo tính cân đối và phù hợp với yêu cầu học tập của từng chuyên ngành và từng kỳ học.

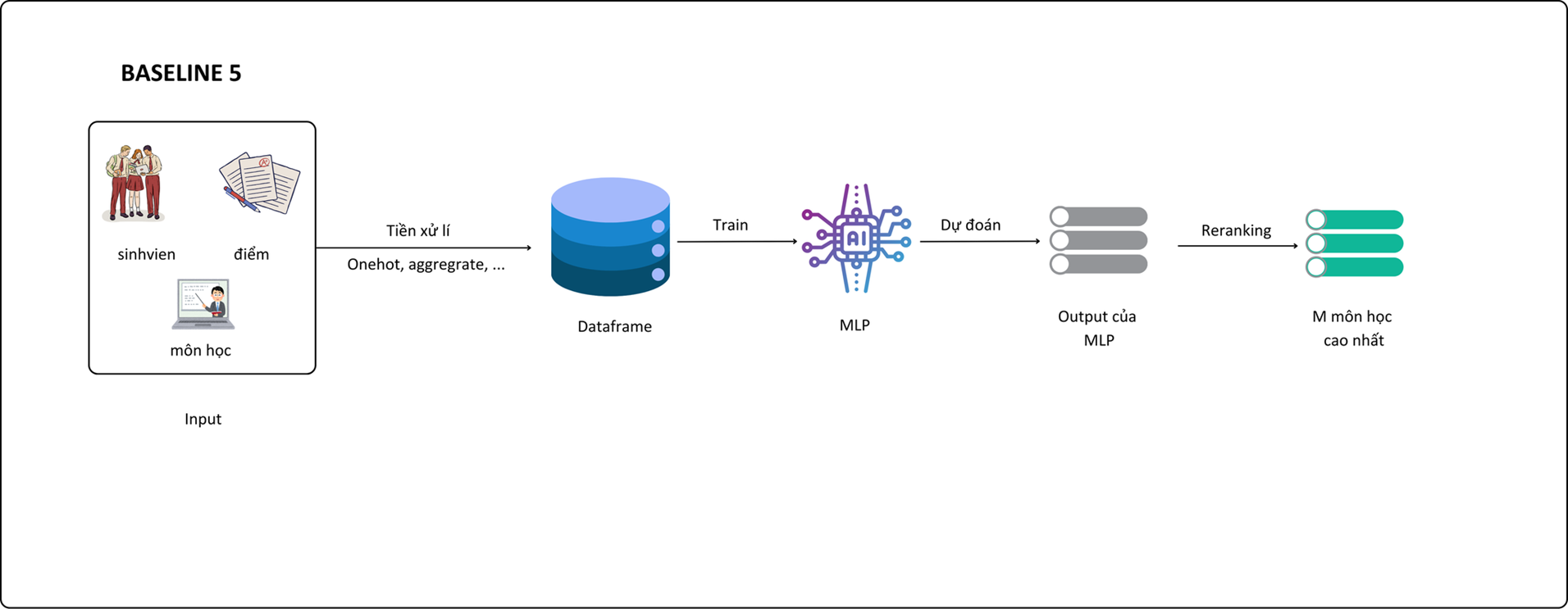
Danh sách môn học được phân chia theo hai nhóm chính. Thứ nhất, nhóm môn học đã học bao gồm các môn học mà sinh viên đã hoàn thành từ các kỳ học trước, giúp phản ánh tiến trình học tập và tích lũy tín chỉ. Thứ hai, nhóm môn học cần gợi ý bao gồm các môn học mà sinh viên chưa hoàn thành **(điểm dưới 5)** và các môn học được sinh viên khác cùng khoa lựa chọn nhiều trong các kỳ trước. Lưu ý, các môn học trong danh sách đề xuất không trùng lặp với các môn học mà sinh viên đã đăng ký trong các kỳ học trước đó.

Quá trình vector hóa dữ liệu được thực hiện nhằm chuyển đổi thông tin từ dạng thô sang dạng số, tối ưu hóa cho các mô hình phân tích và học máy. Đối với các đặc trưng liên tục, các thuộc tính như mã môn học **(mamh)** và số tín chỉ **(sotc)** được chuẩn hóa bằng phương pháp **Min-Max Scaling** để đưa giá trị về khoảng **[0, 1]**, đảm bảo đồng nhất về quy mô. Với các đặc trưng rời rạc, các thuộc tính như khoa **(khoa)** và nhóm loại môn học **(nhomloaimh)** được mã hóa bằng **One-Hot Encoding**, tạo ra các vector nhị phân đại diện cho từng giá trị riêng biệt. Đối với đặc trưng môn học, các thông tin như độ phổ biến **(dophobien\_scaled)** được tích hợp cùng các nhóm môn học đã mã hóa, tạo ra một biểu diễn toàn diện cho từng môn học. Cuối cùng, việc kết hợp đặc trưng giữa sinh viên và môn học được thực hiện, tạo ra các vector đầu vào đa chiều chứa thông tin tổng hợp, phục vụ cho việc xây dựng và huấn luyện mô hình phân tích dữ liệu. Phương pháp này đảm bảo tính toàn diện và tính tương thích cao giữa các loại dữ liệu khác nhau.

Mô hình **Multi-Layer Perceptron (MLP)** được sử dụng để dự đoán xác suất sinh viên sẽ vượt qua môn học. Kiến trúc của mô hình được thiết kế với các thành phần chính như sau: lớp đầu vào nhận số lượng đặc trưng đầu vào từ vector kết hợp của dữ liệu sinh viên và môn học. Tiếp theo là hai lớp ẩn với số lượng neuron lần lượt là **128** và **64**, sử dụng hàm kích hoạt **ReLU** nhằm đảm bảo tính phi tuyến và khả năng học biểu diễn đa dạng. Để giảm hiện tượng **overfitting**, một cơ chế **Dropout** với tỷ lệ **20%** được áp dụng sau mỗi lớp ẩn. Lớp đầu ra gồm một neuron duy nhất sử dụng hàm kích hoạt **sigmoid**, nhằm biểu diễn xác suất sinh viên vượt qua môn học.

Quá trình huấn luyện mô hình được thực hiện bằng cách chia dữ liệu thành hai tập riêng biệt: tập huấn luyện và tập kiểm định **(train/test)**, đảm bảo đánh giá chính xác hiệu suất của mô hình. Mô hình được tối ưu hóa với hàm mất mát **binary\_crossentropy** và **thuật toán** **Adam**, đảm bảo quá trình học diễn ra hiệu quả và hội tụ nhanh.

Quy trình gợi ý môn học được xây dựng nhằm cung cấp các lựa chọn phù hợp nhất cho sinh viên. Đầu tiên, hệ thống xác định nhóm loại môn học dựa trên các thống kê trung bình theo khoa, kỳ học, và số lượng môn học đã chọn của từng nhóm, giúp phân loại môn học thành các nhóm ưu tiên. Tiếp theo, danh sách môn học chưa học được lọc bằng cách loại bỏ các môn đã hoàn thành hoặc bị trùng lặp, tập trung vào các môn học mà sinh viên cần xem xét. Sau đó, mô hình **MLP (Multi-Layer Perceptron)** được sử dụng để dự đoán khả năng qua môn, bằng cách tính toán điểm dự kiến cho từng môn học trong danh sách. Cuối cùng, hệ thống sẽ gợi ý **Top-N** môn học bằng cách chọn ra **N** môn có điểm dự đoán cao nhất trong mỗi nhóm loại môn học.



*Baseline 5 - Phương pháp sử dụng mô hình học sâu MLP theo nhóm loại môn học*

# **CHƯƠNG III: THỰC NGHIỆM VÀ ĐÁNH GIÁ**

## **1. Bộ dữ liệu**

### 1.1. Nguồn dữ liệu

Bộ dữ liệu Education\_dataset\_V2 (UIT Dataset) được xây dựng từ thông tin thu thập từ sinh viên Trường Đại học Công nghệ Thông tin (UIT), thuộc Đại học Quốc gia TP.HCM, nhằm hỗ trợ nghiên cứu về hiệu suất và hành vi học tập của sinh viên. Dữ liệu bao gồm chi tiết kết quả học tập qua nhiều năm và học kỳ, tạo điều kiện cho việc phân tích các yếu tố ảnh hưởng đến kết quả học tập, đồng thời đề xuất các giải pháp nâng cao chất lượng giảng dạy.

Bộ dữ liệu được thu thập trong khoảng thời gian từ năm 2013 đến năm 2016. Education\_dataset\_V2 cung cấp thông tin đa dạng về sinh viên UIT bao gồm: thông tin cá nhân, kết quả học tập, điểm rèn luyện, chi tiết các điểm thành phần, điểm trung bình theo học kỳ / theo toàn khóa, các chứng chỉ ngoại ngữ, xếp loại anh văn, điểm thi tốt nghiệp trung học phổ thông, các giấy tờ hành chính liên quan, xếp loại học vụ, thông tin bảo lưu, thông tin tốt nghiệp, thông tin miễn giảm học phí.

Tuy nhiên, vì đặc trưng của bài toán khuyến nghị môn học chỉ sử dụng dữ liệu liên quan đến thông tin cá nhân sinh viên, kết quả học tập và thông tin mô tả các môn học nên dự án chỉ tập trung khai thác 3 bảng thuộc bộ dữ liệu Education\_dataset\_V2: bảng 01.sinhvien, bảng 02.diem, bảng 14.totnghiep

### **1.2. Phân tích và khai phá dữ liệu**

#### 1.2.1. Bảng 01.sinhvien

Bảng 01.sinhvien mô tả thông tin cá nhân của sinh viên trường Đại học Công nghệ khóa 8 - 14 như: giới tính, năm sinh, lớp, khoa, hệ đào tạo,...

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| id | ID sinh viên | Số nguyên | rời rạc |
| mssv | Mã số sinh viên | chuỗi | rời rạc |
| namsinh | Năm sinh | số nguyên | rời rạc |
| gioitinh | Giới tính | số nguyên | rời rạc |
| noisinh | Nơi sinh | chuỗi | rời rạc |
| lopsh | Lớp sinh hoạt | chuỗi | rời rạc |
| khoa | Khoa đào tạo | chuỗi | rời rạc |
| hedt | Hệ đào tạo | chuỗi | rời rạc |
| khoahoc | Khóa học | số nguyên | rời rạc |
| chuyennganh2 | Chuyên ngành 2 | chuỗi | rời rạc |
| tinhtrang | Tình trạng | số nguyên | rời rạc |
| diachi\_tinhtp | Địa chỉ (tỉnh thành phố) | chuỗi | rời rạc |
| Column1 | Thông tin công việc và nơi làm việc | chuỗi | rời rạc |
| \_1 | Chưa xác định | chuỗi | rời rạc |
| \_2 | Chưa xác định | chuỗi | rời rạc |
| … | Chưa xác định | chuỗi | rời rạc |
| \_\_55 | Chưa xác định | chuỗi | rời rạc |
| \_56 | Chưa xác định | chuỗi | rời rạc |

#### 1.2.2. Bảng 02.diem

Bảng 02.diem mô tả điểm các môn học của sinh viên từ năm 2013 đến năm 2016.

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| id | ID sinh viên | Số nguyên | rời rạc |
| mssv | Mã số sinh viên | chuỗi | rời rạc |
| mamh | Mã môn học | chuỗi | rời rạc |
| malop | Mã lớp | chuỗi | rời rạc |
| sotc | Số tín chỉ | số nguyên | rời rạc |
| namhoc | Năm học | số nguyên | rời rạc |
| hocky | Học kỳ | số nguyên | rời rạc |
| diem | Điểm | số nguyên | liên tục |
| trangthai | Trạng thái | số nguyên | rời rạc |
| mamh\_tt | Mã môn học tương đương | chuỗi | rời rạc |

#### 1.2.3. Bảng 14.totnghiep

Bảng thông tin xếp loại tốt nghiệp của sinh viên

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| id | ID sinh viên | Số nguyên | rời rạc |
| mssv | Mã số sinh viên | chuỗi | rời rạc |
| xeploai | Xếp loại tốt nghiệp | chuỗi | rời rạc |
| soquyetdinh | Số quyết định tốt nghiệp | chuỗi | rời rạc |
| ngaycapvb | Ngày cấp văn bằng | chuỗi | rời rạc |

### 1.3. Chất lượng dữ liệu

#### 1.3.1. Kiểm tra chất lượng dữ liệu

##### a. Độ chính xác

Đầu tiên sẽ thực hiện tính toán, kiểm tra tỷ lệ lỗi của dữ liệu và từ đó tính được độ chính xác của dữ liệu và các bước trên được tính theo công thức:

*Độ chính xác = 100% - tỷ lệ lỗi của dữ liệu*

| **Bảng 01.sinhvien** | | |
| --- | --- | --- |
| **Thuộc tính** | **Vấn đề** | **Tỷ lệ lỗi** |
| id | Không tồn tại dữ liệu sai | 0% |
| mssv | Không tồn tại dữ liệu sai | 0% |
| namsinh | Không tồn tại dữ liệu sai | 0% |
| gioitinh | Không tồn tại dữ liệu sai | 0% |
| noisinh | Không tồn tại dữ liệu sai | 0% |
| lopsh | Không tồn tại dữ liệu sai | 0% |
| khoa | Không tồn tại dữ liệu sai | 0% |
| hedt | Không tồn tại dữ liệu sai | 0% |
| khoahoc | Không tồn tại dữ liệu sai | 0% |
| chuyennganh2 | Không tồn tại dữ liệu sai | 0% |
| tinhtrang | Không tồn tại dữ liệu sai | 0% |
| diachi\_tinhtp | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 28.19% |
| Column1 | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| \_1 | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| \_2 | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| … | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| \_\_55 | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| \_56 | Dữ liệu không đồng nhất và không thể xác định được cho nên sẽ mặc định tất cả giá trị trong thuộc tính này là sai | 100% |
| **Bảng 02.diem** | | |
| **Thuộc tính** | **Vấn đề** | **Tỷ lệ lỗi** |
| id | Không tồn tại dữ liệu sai | 0% |
| mssv | Không tồn tại dữ liệu sai | 0% |
| mamh | Không tồn tại dữ liệu sai | 0% |
| malop | Không tồn tại dữ liệu sai | 0% |
| sotc | Không tồn tại dữ liệu sai | 0% |
| namhoc | Không tồn tại dữ liệu sai | 0% |
| hocky | Không tồn tại dữ liệu sai | 0% |
| diem | Không tồn tại dữ liệu sai | 0% |
| trangthai | Không tồn tại dữ liệu sai | 0% |
| mamh\_tt | Dữ liệu chưa đồng nhất các giá trị [' NULL', ' NULL);', ' NULL)'] | 99.85% |
| **Bảng 14.totnghiep** | | |
| **Thuộc tính** | **Vấn đề** | **Tỷ lệ lỗi** |
| id | Không tồn tại dữ liệu sai | 0% |
| mssv | Không tồn tại dữ liệu sai | 0% |
| xeploai | Dữ liệu chưa đồng nhất các giá trị [' Trung bình Khá', ' TB Khá', ' TB khá', ' Trung bình khá'] | 7.26% |
| soquyetdinh | Không tồn tại dữ liệu sai | 0% |
| ngaycapvb | Không tồn tại dữ liệu sai | 0% |

Từ những vấn đề và tỷ lệ lỗi riêng của từng thuộc tính, thực hiện tổng hợp thông tin tỷ lệ lỗi và độ chính xác của từng bảng. Có thể thấy bảng dữ liệu 01.sinhvien có tỷ lệ lỗi cao nhất và lên đến 83.02%.

| **Bảng dữ liệu** | **Tỷ lệ lỗi** | **Độ chính xác** |
| --- | --- | --- |
| 01.sinhvien | 83.02% | 16.98% |
| 02.diem | 9.99% | 90.01% |
| 14.totnghiep | 1.45% | 98.55% |

Bên cạnh đo đạc tỷ lệ lỗi và độ chính xác thì cũng cần kiểm tra độ lệch chuẩn và giá trị trung bình của các thuộc tính dữ liệu liên tục để xác minh dữ liệu có chính xác hay không, có hợp lệ hay không.

Ở bộ dữ liệu này chỉ tồn tại thuộc tính liên tục là ‘diem’ thuộc bảng dữ liệu 02.diem. Và kết quả độ lệch chuẩn là 2.38, giá trị trung bình là 6.33. Kết quả cho thấy hầu hết các điểm số tập trung quanh giá trị 6.33. Do thang điểm nằm trong khoảng từ 0 đến 10, giá trị trung bình 6.33 cho thấy mức độ khá ổn định và trên mức trung bình (5 điểm). Bên cạnh đó độ lệch chuẩn 2.38 được coi là khá lớn so với thang điểm 10, có sự khác biệt đáng kể giữa các điểm số của các sinh viên.

##### b. Tính đầy đủ

Quá trình đánh giá tính đầy đủ tập trung vào việc phát hiện các giá trị thiếu, giá trị không xác định, và các vấn đề về sự không đồng nhất của dữ liệu.

Tính toán tỷ lệ phần trăm dữ liệu thiếu so với tổng số dữ liệu và tính đầy đủ của dữ liệu với công thức:

*Tính đầy đủ của dữ liệu: = 100% - tỷ lệ dữ liệu thiếu của dữ liệu*

| **Bảng 01.sinhvien** | |
| --- | --- |
| **Thuộc tính** | **Tỷ lệ thiếu dữ liệu** |
| id | 0.25% |
| mssv | 0.25% |
| namsinh | 0.25% |
| gioitinh | 0.25% |
| noisinh | 0.25% |
| lopsh | 0.25% |
| khoa | 0.25% |
| hedt | 0.25% |
| khoahoc | 0.25% |
| chuyennganh2 | 0.25% |
| tinhtrang | 0.25% |
| diachi\_tinhtp | 0.25% |
| Column1 | 99.99% |
| \_1 | 99.99% |
| \_2 | 99.99% |
| … | 99.99% |
| \_\_55 | 99.99% |
| \_56 | 99.99% |
| **Bảng 02.diem** | |
| **Thuộc tính** | **Tỷ lệ thiếu dữ liệu** |
| id | 0.14% |
| mssv | 0.14% |
| mamh | 0.14% |
| malop | 0.14% |
| sotc | 0.14% |
| namhoc | 0.14% |
| hocky | 0.14% |
| diem | 0.14% |
| trangthai | 0.14% |
| mamh\_tt | 0.14% |
| **Bảng 14.totnghiep** | |
| **Thuộc tính** | **Tỷ lệ thiếu dữ liệu** |
| id | 0.11% |
| mssv | 0.11% |
| xeploai | 0.11% |
| soquyetdinh | 0.11% |
| ngaycapvb | 0.11% |

#### 

Bảng tổng hợp dữ liệu thiếu và tính đầy đủ của dữ liệu:

| **Bảng dữ liệu** | **Tỷ lệ dữ liệu thiếu** | **Tính đầy đủ** |
| --- | --- | --- |
| 01.sinhvien | 81.90% | 18.10% |
| 02.diem | 8.11% | 91.89% |
| 14.totnghiep | 0.11% | 99.89% |

Có thể thấy dữ liệu 01.sinhvien có tỷ lệ thiếu rất cao dẫn đến tính đầy đủ của dữ liệu rất thấp.

##### c. Tính hợp lệ

Thực hiện một số bước kiểm tra sau để xét dữ liệu có hợp lệ hay không:

- Kiểm tra tính chính quy: Sử dụng các kiểm tra thống kê như kiểm tra Shapiro-Wilk, v.v. để xác định xem dữ liệu có phân phối chính quy hay không.

- Kiểm tra ngoại lệ: Sử dụng các phương pháp như phân tích IQR (Interquartile Range) hoặc z-score để xác định các giá trị ngoại lệ trong dữ liệu.

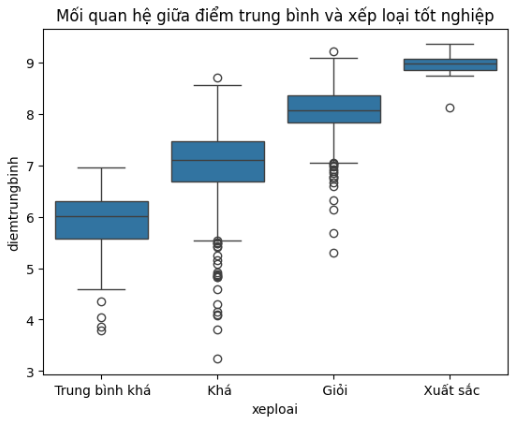
- Phân tích tương quan: Đánh giá mối liên hệ giữa các thuộc tính dữ liệu để xác định xem dữ liệu có hợp lý hay không.

Quá trình kiểm tra sẽ sử dụng thuộc tính diem trong dữ liệu 02.diem bên cạnh đó thực hiện tính toán điểm trung bình của các sinh viên đã tốt nghiệp để kiểm tra sự đồng nhất về kết quả xếp loại tốt nghiệp và điểm trung bình của sinh viên đó.

Kết quả thu được:

| **Thuộc tính** | **Kết quả kiểm định Kolmogorov Smirnov** | **Tỷ lệ dữ liệu ngoại lệ** |
| --- | --- | --- |
| diem | KS Stat = 0.11555456 | 6.49% |
| diemtrungbinh | KS Stat = 0.11801677 | 5.037% |

Kiểm định KS được sử dụng để kiểm tra xem một tập dữ liệu có tuân theo một phân phối lý thuyết (thường là phân phối chuẩn) hay không. Giá trị này cho biết mức độ khác biệt giữa phân phối của thuộc tính và phân phối chuẩn. KS Stat của cả 2 tương đối cao 0.1156 và 0.1180 cho thấy dữ liệu không tuân theo phân phối chuẩn. Có thể vấn đề là do điểm số còn nhiều ngoại lệ và nhiều dữ liệu sai quy định. Hệ số tương quan đã nói lên điều đó, hệ số tương quan giữa điểm trung bình và xếp loại tốt nghiệp: 0.7023. Đây là một hệ số khá thấp, trong khi xếp loại tốt nghiệp phải dựa vào điểm số. Cần xử lý vấn đề ngoại lệ để kết quả đạt được tốt hơn.



##### d. Tính duy nhất

Tính duy nhất sẽ kiểm tra tính trùng lặp của dữ liệu

Tỷ lệ trùng lặp trong dữ liệu được tính bằng công thức:

| **Bảng dữ liệu** | **Tỷ lệ dữ liệu thiếu** |
| --- | --- |
| 01.sinhvien | 0.24% |
| 02.diem | 0.136% |
| 14.totnghiep | 0.054% |

Kết quả cho thấy số lượng dữ liệu trùng không đáng lo ngại.

##### e. Tính nhất quán

Thực hiện lần lượt các thao tác kiểm tra như:

- Kiểm tra thống kê: Sử dụng các kiểm tra thống kê như kiểm tra chi-squared, kiểm tra t-test, v.v. để xác định xem có sự khác biệt đáng kể nào giữa các giá trị dữ liệu hay không.

Kết quả kiểm tra Chi-squared: mối quan hệ giữa khoa đào tạo và xếp loại tốt nghiệp của sinh viên: Chi2 = 32.7628930937618, p-value = 0.3329239711881565

Kết quả kiểm tra T-Test: Điểm trung bình của sinh viên đã tốt nghiệp và chưa tốt nghiệp: T-statistic: 40.9744083166 , p-value: 0.0000

Dựa trên kết quả kiểm định Chi-squared, không có mối liên hệ đáng kể giữa khoa đào tạo và xếp loại tốt nghiệp. Dữ liệu cho thấy xếp loại tốt nghiệp của sinh viên không phụ thuộc vào việc họ học ở khoa nào.

Dựa trên kết quả kiểm định T-test, ta có thể khẳng định rằng điểm trung bình của sinh viên đã tốt nghiệp khác với sinh viên chưa tốt nghiệp. Do T-statistic có giá trị lớn (40.97) và P-value nhỏ (0.0000), điều này cho thấy sự khác biệt là rất đáng kể.

- Kiểm tra số lượng mã số sinh viên có trong 3 bảng dữ liệu

Kết quả:

* Số sinh viên có mặt trong cả 3 tệp: 1805. Những sinh viên này có thông tin đầy đủ, nên có thể dùng họ để phân tích các yếu tố liên quan đến kết quả học tập, điểm số và khả năng tốt nghiệp.
* Số sinh viên chỉ có trong 01.sinhvien: 6491 trong 8316 chiếm tỷ lệ: 78.05%. Những sinh viên này chỉ có thông tin trong bảng sinh viên nhưng không có trong bảng điểm và bảng tốt nghiệp. Lý do có thể là: Họ chưa đăng ký môn học nào, nên không có trong bảng điểm. Họ chưa tốt nghiệp, nên không có trong bảng tốt nghiệp. Họ có thể là sinh viên mới nhập học, chưa có kết quả học tập nào.
* Số sinh viên chỉ có trong 02.diem: 2246 trong 99099 chiếm tỷ lệ: 2.27%. Những sinh viên này có dữ liệu điểm số trong bảng điểm, nhưng không có thông tin trong bảng sinh viên và không có trong bảng tốt nghiệp. Lý do có thể là: Họ là sinh viên đã rời khỏi trường, nên không còn thông tin trong bảng sinh viên. Những sinh viên này chưa tốt nghiệp, nên không có trong bảng tốt nghiệp.
* Số sinh viên chỉ có trong 14.totnghiep: 41 trong 1847 chiếm tỷ lệ: 2.22%. Những sinh viên này chỉ có thông tin trong bảng tốt nghiệp, nhưng không có trong bảng sinh viên và bảng điểm. Lý do có thể là: Sinh viên đã tốt nghiệp, nhưng dữ liệu của họ không còn trong bảng sinh viên và bảng điểm (do đã xóa dữ liệu cũ).

- Phân tích độ tương quan: Đánh giá mối liên hệ giữa các thuộc tính dữ liệu để xác định xem chúng có nhất quán với nhau hay không.

##### 

Từ biểu đồ ta có thể thấy:

Tương quan cao:

* namsinh và khoahoc: Hệ số tương quan 0.87 (rất mạnh).
  + Điều này hợp lý vì năm sinh và khóa học thường liên quan trực tiếp đến thời điểm nhập học.
* diem và xeploai: Hệ số tương quan 0.37 (mức độ vừa phải, dương).
  + Điểm số càng cao, xếp loại tốt nghiệp càng cao.
* sotc và namhoc: Tương quan 0.08 (yếu nhưng dương).
  + Sinh viên năm cao hơn có xu hướng tích lũy được nhiều tín chỉ hơn.

Tương quan nghịch:

* diem và trangthai: Hệ số tương quan -0.64 (mạnh, nghịch).
  + Khi điểm cao, trạng thái học tập (trangthai) có xu hướng ổn định hơn (ít bỏ học hoặc bị cảnh báo hơn).
* gioitinh và xeploai: Hệ số tương quan -0.13 (rất yếu, nghịch).
  + Ảnh hưởng của giới tính đến xếp loại là không đáng kể.

Tương quan yếu hoặc không đáng kể:

* gioitinh với các biến khác: Phần lớn hệ số tương quan rất gần 0, cho thấy giới tính không ảnh hưởng đáng kể đến các yếu tố như điểm số, trạng thái học tập, hoặc số tín chỉ.

##### f. Tính kịp thời

Chất lượng dữ liệu không thể đánh giá qua tính kịp thời vì bộ dữ liệu được thu thập từ năm 2013 đến năm 2016 và chưa được cập nhật tới thời điểm hiện tại.

#### 1.3.2. Nâng cao chất lượng dữ liệu

| **Bảng dữ liệu** | **Vấn đề** | **Giải pháp** |
| --- | --- | --- |
| 01. sinhvien | - Ở cột diachi\_tinhtp: Các ô có giá trị bị sai, giá trị đúng là tên các tỉnh, thành của Việt Nam. Nhiều ô lại có giá trị là xã, huyện, quận hay thậm chí các dữ liệu không liên quan khác.  - Ở các cột column1, \_1, \_2, \_3, \_4, \_5, \_6, \_7, \_8, \_9, \_10, \_11, \_12, \_13, \_14, \_15, \_16, \_17, \_18, \_19, \_20, \_21, \_22, \_23, \_24, \_25, \_26, \_27, \_28, \_29, \_30, \_31, \_32, \_33, \_34, \_35, \_36, \_37, \_38, \_39, \_40, \_41, \_42, \_43, \_44, \_45, \_46, \_47, \_48, \_49, \_50, \_51, \_52, \_53, \_54, \_55, \_56 dữ liệu sai rất nhiều và không đồng nhất. nên thực hiện việc xóa bỏ các cột này đi  - Xóa dữ liệu thiếu: xóa các mẫu mà tất cả các cột đều thiếu dữ liệu, xóa các cột column1, \_1, \_2, \_3, \_4, \_5, \_6, \_7, \_8, \_9, \_10, \_11, \_12, \_13, \_14, \_15, \_16, \_17, \_18, \_19, \_20, \_21, \_22, \_23, \_24, \_25, \_26, \_27, \_28, \_29, \_30, \_31, \_32, \_33, \_34, \_35, \_36, \_37, \_38, \_39, \_40, \_41, \_42, \_43, \_44, \_45, \_46, \_47, \_48, \_49, \_50, \_51, \_52, \_53, \_54, \_55, \_56 với tỷ lệ thiếu lên đến hơn 80%.  - Tồn tại những mẫu dữ liệu bị trùng. | - Xử lý dữ liệu sai: Ở cột diachi\_tinhtp: xóa bỏ các cột dữ liệu sai (giá trị không phải là 1 trong 63 tỉnh thành của Việt Nam) sau đó chỉnh lại định dạng chữ cho đúng quy định (tên tỉnh, thành phố phải viết in Hoa chữ cái đầu tiên của mỗi tiếng).  Xóa bỏ các cột: column1, \_1, \_2, \_3, \_4, \_5, \_6, \_7, \_8, \_9, \_10, \_11, \_12, \_13, \_14, \_15, \_16, \_17, \_18, \_19, \_20, \_21, \_22, \_23, \_24, \_25, \_26, \_27, \_28, \_29, \_30, \_31, \_32, \_33, \_34, \_35, \_36, \_37, \_38, \_39, \_40, \_41, \_42, \_43, \_44, \_45, \_46, \_47, \_48, \_49, \_50, \_51, \_52, \_53, \_54, \_55, \_56.  - Xử lý giá trị thiếu: xóa bỏ hết các mẫu có giá trị thiếu.  - Xử lý giá trị trùng lặp: Xóa những mẫu dữ liệu bị trùng |
| 02. diem | - Dữ liệu sai: ở cột mamh\_tt có các dữ liệu bị sai định dạng.  - Tồn tại nhiều dữ liệu thiếu.  - Tồn tại những mẫu dữ liệu bị trùng.  - Tồn tại dữ liệu ngoại lai. | - Thực hiện xử lý dữ liệu bằng cách thay thế dữ liệu sai.  - Xử lý giá trị thiếu: Xóa dữ liệu thiếu: xóa các mẫu mà tất cả các cột đều thiếu dữ liệu. thế dữ liệu thiếu: Với dữ liệu định tính (categorical): Các giá trị thiếu được thay bằng "NULL". Với dữ liệu định lượng (continuous): Giá trị thiếu ở cột 'diem' thiếu được thay bằng giá trị trung bình.  - Xử lý giá trị trùng lặp: Xóa những mẫu dữ liệu bị trùng  - Xử lý các giá trị ngoại lai (outliers): ở cột diem sử dụng IQR để xóa các dữ liệu ngoai lệ ấy đi |
| 14.totnghiep | - Tồn tại nhiều dữ liệu thiếu.  - Dữ liệu không nhất quán: ở cột xeploai, các giá trị [' Trung bình Khá', ' TB Khá', ' TB khá'] không nhất quán.  - Tồn tại những mẫu dữ liệu bị trùng. | - Xử lý giá trị thiếu: Xóa dữ liệu thiếu: xóa các mẫu mà tất cả các cột đều thiếu dữ liệu.  - Xử lý dữ liệu không nhất quán: ở cột xeploai, các giá trị [' Trung bình Khá', ' TB Khá', ' TB khá'] không nhất quán, xử lý vấn đề trên bằng cách thay thế các giá trị [' Trung bình Khá', ' TB Khá', ' TB khá'] thành ' Trung bình khá'  - Xử lý giá trị trùng lặp: Xóa những mẫu dữ liệu bị trùng |

#### 1.3.3. Đánh giá chất lượng dữ liệu

| **Phương pháp đánh giá** | **Dữ liệu ban đầu** | **Sau khi nâng cao chất lượng dữ liệu** |
| --- | --- | --- |
| Độ chính xác | - Dữ liệu 01.sinhvien:  Tỷ lệ lỗi: 83.02%  Độ chính xác: 16.98%  - Dữ liệu 02.sinhvien:  Tỷ lệ lỗi: 9.99%  Độ chính xác: 90.01%  Độ lệch chuẩn của cột diem: 2.38  Giá trị trung bình của cột diem: 6.33  - Dữ liệu 14.totnghiep:  Tỷ lệ lỗi: 1.45%  Độ chính xác: 98.55% | - Dữ liệu 01.sinhvien:  Tỷ lệ lỗi: 0%  Độ chính xác: 100%  - Dữ liệu 02.sinhvien:  Tỷ lệ lỗi: 0%  Độ chính xác: 100%  Độ lệch chuẩn của cột diem: 2.38  Giá trị trung bình của cột diem: 6.33  - Dữ liệu 14.totnghiep:  Tỷ lệ lỗi: 0%  Độ chính xác: 100% |
| Tính đầy đủ | - Dữ liệu 01.sinhvien: 18.10%  - Dữ liệu 02.sinhvien: 91.89%  - Dữ liệu 14.totnghiep: 99.89% | - Dữ liệu 01.sinhvien: 100%  - Dữ liệu 02.sinhvien: 100%  - Dữ liệu 14.totnghiep: 100% |
| Tính hợp lệ | 1. Tỷ lệ dữ liệu ngoại lệ  - Dữ liệu điểm: 6.49%  - Dữ liệu điểm trung bình: 5.037%  2. Kết quả kiểm định Kolmogorov Smirnov  - Dữ liệu điểm: KS Stat = 0.11555456  - Dữ liệu điểm trung bình: KS Stat = 0.11801677.  3. Hệ số tương quan giữa điểm trung bình và xếp loại tốt nghiệp: 0.7023 | 1. Tỷ lệ dữ liệu ngoại lệ  - Dữ liệu điểm: 1.137%  - Dữ liệu điểm trung bình: 5.037%  2. Kết quả kiểm định Kolmogorov Smirnov  - Dữ liệu điểm: KS Stat = 0.07666976, P-value = 0.0000  - Dữ liệu điểm trung bình: KS Stat = 0.07553649, P-value = 0.0000  3. Hệ số tương quan giữa điểm trung bình và xếp loại tốt nghiệp: 0.9029 |
| Tính duy nhất | - Tỷ lệ trùng trong student: 0.24%  - Tỷ lệ trùng trong score: 0.136%  - Tỷ lệ trùng trong graduate: 0.054% | - Tỷ lệ trùng trong student: 0%  - Tỷ lệ trùng trong score: 0%  - Tỷ lệ trùng trong graduate: 0% |
| Tính nhất quán | 1. Số lượng mã số sinh viên  Số sinh viên có mặt trong cả 3 tệp: 1805  - Số sinh viên chỉ có trong raw\_student\_df: 6491 trong 8316 chiếm tỷ lệ: 78.05%  - Số sinh viên chỉ có trong raw\_score\_df: 2246 trong 99099 chiếm tỷ lệ: 2.27%  - Số sinh viên chỉ có trong raw\_graduate\_df: 41 trong 1847 chiếm tỷ lệ: 2.22%  2. Kiểm tra Chi-squared:  mối quan hệ giữa khoa đào tạo và xếp loại tốt nghiệp của sinh viên:  Chi2 = 32.7628930937618  p-value = 0.3329239711881565  3. T-Test: Điểm trung bình của sinh viên đã tốt nghiệp và chưa tốt nghiệp  T-statistic: 40.9744083166 , p-value: 0.0000  4. Hệ số tương quan giữa điểm trung bình và xếp loại tốt nghiệp: 0.7023 | Hệ số tương quan giữa điểm trung bình và xếp loại tốt nghiệp: 0.9029 |

### 1.4. **Lựa chọn đặc trưng**

Đầu tiên, dữ liệu từ ba bảng sinhvien, diem, và totnghiep được nạp vào và nối lại với nhau dựa trên mã số sinh viên (mssv). Sau đó, các thuộc tính cần thiết cho phân tích được lựa chọn và dữ liệu được lọc theo trạng thái. Để đơn giản hóa phân tích, dữ liệu kỳ 3 được gộp vào kỳ 2 vì học kỳ 3 chỉ cho phép học các môn cần cải thiện điểm hoặc học lại các môn không đạt ở học kỳ 1 và học kỳ 2.

Dữ liệu được mở rộng bằng cách thêm thông tin chi tiết về các môn học từ hai bảng bổ sung: Danh-muc-mon-hoc và paraphrased\_tomtat. Bảng Danh-muc-mon-hoc cung cấp thông tin về đơn vị quản lý và loại môn học, trong khi bảng paraphrased\_tomtat chứa mô tả ngắn gọn về nội dung của từng môn học. Mô tả này được tiền xử lý bằng PhoBERT, một mô hình ngôn ngữ lớn tiếng Việt, để tạo ra các vector embedding đại diện cho ngữ nghĩa của từng môn học.

Để hỗ trợ phân tích sâu hơn, một số thuộc tính nâng cao được tạo ra. Thuộc tính **namhoc\_batdau** ghi lại năm học đầu tiên của mỗi sinh viên, **sohocky** tính toán số học kỳ đếm từ kỳ học đầu tiên ở năm học đầu tiên của mỗi sinh viên, **nhomloaimh** phân loại môn học thành 5 nhóm chính bao gồm: ĐC (Đại cương), CSN (Cơ sở ngành), CN (Chuyên ngành), TC (Tự chọn), TN (Tốt nghiệp) và hoanthanh đánh dấu kết quả học tập của sinh viên ở môn học đó đã đạt hay chưa với điều kiện điểm lớn hơn 5.

Cuối cùng, dữ liệu được sử dụng để thống kê một số đặc điểm của các môn học. Độ phổ biến được tính toán dựa trên số lượng sinh viên đăng ký môn học trong từng kỳ và khoa, sau đó được chuẩn hóa để so sánh giữa các môn. Điểm trung bình của môn học được tính toán theo kỳ, khoa và được chuyển đổi thành chỉ số "độ thành tích" để dễ dàng so sánh. Số lượng môn học trong mỗi nhóm (như chuyên ngành, cơ sở ngành, tự nhiên...) được thống kê theo kỳ và khoa.

Chi tiết cấu trúc đặc trưng được chọn của các bảng dữ liệu:

#### 1.4.1. Bảng processed\_student

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mssv | Mã số sinh viên | chuỗi | rời rạc |
| gioitinh | Giới tính | số nguyên | rời rạc |
| namhoc\_batdau | Năm học sinh viên bắt đầu tham gia học tập tại trường | số nguyên | liên tục |
| khoa | Khoa | chuỗi | rời rạc |
| hedt | Hệ đào tạo | chuỗi | rời rạc |
| nganhhoc | Ngành học | chuỗi | rời rạc |

#### 1.4.2. Bảng processed\_course

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mamh | Mã môn học | chuỗi | rời rạc |
| sotc | Số tín chỉ | số nguyên | liên tục |
| nganhmh | Ngành môn học | chuỗi | rời rạc |
| loaimh | Loại môn học | chuỗi | rời rạc |
| nhomloaimh | Nhóm loại môn học | chuỗi | rời rạc |

#### 1.4.3. Bảng processed\_score

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mssv | Mã số sinh viên | chuỗi | rời rạc |
| mamh | Mã môn học | chuỗi | rời rạc |
| nhomloaimh | Nhóm loại môn học | chuỗi | rời rạc |
| diem | Điểm môn học | số thực | liên tục |
| trangthai | Trạng thái môn học | chuỗi | rời rạc |
| namhoc | Năm học diễn ra | số nguyên | liên tục |
| hocky | Học kỳ diễn ra | số nguyên | liên tục |
| sohocky | Số học kỳ diễn ra | số nguyên | liên tục |
| hoanthanh | Sinh viên hoàn thành môn học hay không | boolean | rời rạc |

#### 1.4.4. Bảng onehot\_course

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mamh | Mã môn học | chuỗi | rời rạc |
| sotc | Số tín chỉ | số nguyên | liên tục |
| nganhmh (one-hot encoding) | Ngành môn học | boolean | rời rạc |
| loaimh (one-hot encoding) | Loại môn học | boolean | rời rạc |
| nhomloaimh (one-hot encoding) | Nhóm loại môn học | boolean | rời rạc |

#### 

#### 1.4.5. Bảng PhoBERT\_paraphased\_tomtat

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mamh | Mã môn học | chuỗi | rời rạc |
| embedding\_dim\_0 … embedding\_dim\_768 | PhoBERT Word Embedding vector | số thực | liên tục |

#### 1.4.6. Bảng subject\_popularity

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mamh | Mã môn học | chuỗi | rời rạc |
| khoa | Khoa | chuỗi | rời rạc |
| namhoc | Năm học | số nguyên | liên tục |
| sohocky | Số học kỳ | số nguyên | liên tục |
| sosv | Số sinh viên theo học | số nguyên | liên tục |
| tongsosv | Tổng số sinh viên | số nguyên | liên tục |
| dophobien | Độ phổ biến (sosv / tongsosv) | số thực | liên tục |
| dophobien\_scaled | Độ phổ biến được chuẩn hóa | số thực | liên tục |

#### 1.4.7. Bảng subject\_score

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| mamh | Mã môn học | chuỗi | rời rạc |
| khoa | Khoa | chuỗi | rời rạc |
| namhoc | Năm học | số nguyên | liên tục |
| sohocky | Số học kỳ | số nguyên | liên tục |
| sosv | Số sinh viên theo học | số nguyên | liên tục |
| diemtb | Điểm trung bình | số thực | liên tục |
| dothanhtich | Độ thành tích (diemtb / 10) | số thực | liên tục |
| dothanhtich\_scaled | Độ thành tích được chuẩn hóa | số thực | liên tục |

#### 1.4.8. Bảng group\_course

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| khoa | Khoa | chuỗi | rời rạc |
| namhoc | Năm học | số nguyên | liên tục |
| sohocky | Số học kỳ | số nguyên | liên tục |
| nhomloaimh | Nhóm loại môn học | chuỗi | rời rạc |
| somonhoc | Số môn học (số môn học trung bình thuộc từng nhóm loại môn học được làm tròn lên) | số nguyên | liên tục |

#### 1.4.9. Bảng group\_sum\_course

| **Trường dữ liệu** | **Mô tả** | **Kiểu dữ liệu** | **Kiểu biến** |
| --- | --- | --- | --- |
| khoa | Khoa | chuỗi | rời rạc |
| namhoc | Năm học | số nguyên | liên tục |
| sohocky | Số học kỳ | số nguyên | liên tục |
| somonhoc | Số môn học (số môn học trung bình được làm tròn lên) | số nguyên | liên tục |

## **2. Phương pháp đánh giá**

Để đánh giá hiệu quả của hệ thống gợi ý môn học, các chỉ số quan trọng được sử dụng bao gồm **Precision at K, Recall at K, và F1 Score**. **Precision at K** đo lường tỷ lệ các môn học được hệ thống khuyến nghị chính xác trong số **K** môn học đứng đầu danh sách gợi ý, phản ánh độ chính xác của hệ thống khi tập trung vào các đề xuất ưu tiên. **Recall at K** đo lường tỷ lệ các môn học được khuyến nghị chính xác trong tổng số các môn học mà sinh viên thực sự đăng ký, cho thấy mức độ bao phủ của hệ thống trong việc đề xuất các môn học phù hợp. Cuối cùng, **F1 Score** được tính toán như trung bình hài hòa giữa **Precision at K** và **Recall at K**, theo công thức:

Các chỉ số này giúp đánh giá toàn diện hiệu suất của hệ thống gợi ý, đảm bảo tính chính xác và khả năng hỗ trợ sinh viên một cách hiệu quả.

## **3. Triển khai thực nghiệm**

Trong kịch bản thực nghiệm, một tập dữ liệu bao gồm **505 sinh viên (đã tốt nghiệp)**, tất cả có dữ liệu đầy đủ với năm học bắt đầu vào **năm 2014** và các kỳ học từ **kỳ 2 đến kỳ 5**. Tiếp theo, **5 baseline** được áp dụng để thử nghiệm trên tập dữ liệu dự đoán các môn học từ **kỳ 2 đến kỳ 5**, nhằm đảm bảo tính đầy đủ và phản ánh chính xác xu hướng học tập của sinh viên. Dữ liệu đầu vào của các baseline được tổng hợp lại thành một DataFrame tổng thể chứa thông tin chi tiết về sinh viên, bao gồm các trường dữ liệu liên quan đến quá trình học tập, điểm số và các đặc điểm cá nhân. Với tổng cộng **51,950 dòng** và **17 cột**, bộ dữ liệu cung cấp cái nhìn toàn diện về tình hình học tập tại một trường đại học.

Dữ liệu được tổ chức dưới dạng bảng với các trường thể hiện các khía cạnh khác nhau của sinh viên và môn học. Các trường thông tin cá nhân như **mssv (Mã số sinh viên)**, **gioitinh (Giới tính)**, và **khoa (Khoa đào tạo)** là cơ sở để phân loại và quản lý sinh viên. Trong khi đó, các trường liên quan đến học tập như **mamh (Mã môn học)**, **diem (Điểm số)**, và **sotc (Số tín chỉ)** lại giúp theo dõi chi tiết quá trình học tập của từng sinh viên. Sự kết hợp giữa thông tin cá nhân và học tập tạo nên một nguồn dữ liệu đa chiều, phục vụ tốt cho việc phân tích và khai phá.

Các trường dữ liệu liên quan đến học tập như **namhoc (Năm học)**, **hocky (Học kỳ)**, và **sohocky (Số học kỳ)** cung cấp bức tranh rõ nét về tiến trình học tập của sinh viên qua thời gian. Cột **diem (Điểm)** là một biến số liên tục, đóng vai trò quan trọng trong việc đánh giá thành tích học tập. Điểm số này có thể được phân tích để tìm hiểu phân phối điểm, tỷ lệ đạt môn hoặc mối liên hệ với các yếu tố khác như **số tín chỉ (sotc)** hoặc **loại môn học (loaimh)**. Ngoài ra, các thông tin về **trạng thái học tập (trangthai, hoanthanh)** và **hệ đào tạo (hedt)** bổ trợ cho việc phân tích mức độ hoàn thành chương trình của sinh viên.

Bộ dữ liệu bao gồm nhiều trường định tính như **khoa, hedt, nganhmh (Ngành môn học)**, và **loaimh (Loại môn học)**. Những trường này có giá trị phân loại cao, cho phép nhóm sinh viên theo các tiêu chí cụ thể. Ví dụ, có thể phân tích tỷ lệ sinh viên theo từng khoa hoặc xem xét mối liên hệ giữa loại môn học và điểm số. Bên cạnh đó, cột **nhomloaimh (Nhóm loại môn học)** giúp phân nhóm môn học theo một hệ thống phân loại cụ thể, làm rõ hơn vai trò của từng môn trong chương trình đào tạo.

Các chỉ số đánh giá được sử dụng trong quá trình thử nghiệm bao gồm **macro precision at k, macro recall at k, và macro F1 score (kết quả trung bình của các kỳ học)**, giúp đo lường độ chính xác, độ bao phủ và sự cân bằng giữa hai yếu tố này trên toàn bộ các lớp. Sau khi thực hiện thử nghiệm ban đầu, các baseline được tinh chỉnh bằng phương pháp **GridSearch** để tối ưu hóa các siêu tham số, từ đó cải thiện hiệu suất của mô hình. Quy trình này giúp cải thiện khả năng dự đoán và đưa ra các gợi ý môn học phù hợp với từng sinh viên.

## **4. Kết quả thực nghiệm**

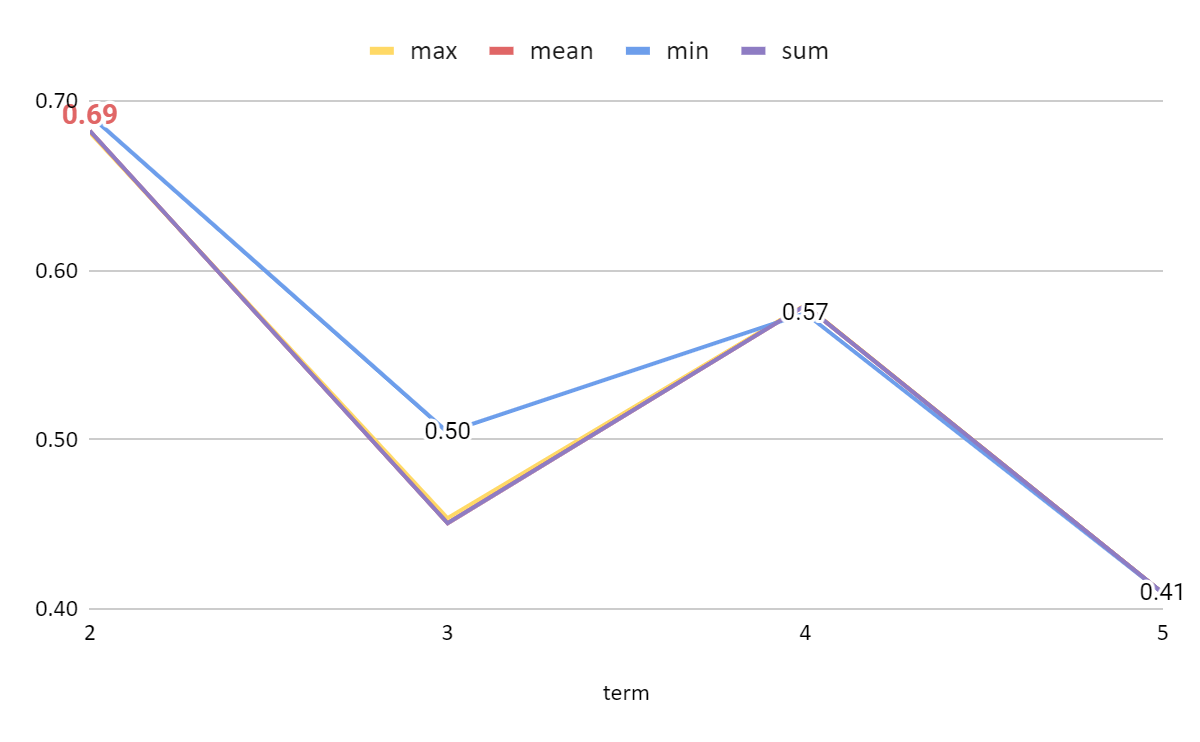
### 4.1. Baseline 1

Phương pháp **Grid Search** được sử dụng để thử qua tất cả tổ hợp siêu tham số của **Baseline 1** nhằm tìm ra tổ hợp có **Macro F1-score** cao nhất qua các kỳ. Các siêu tham số bao gồm: **pooling** và **top\_m**. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

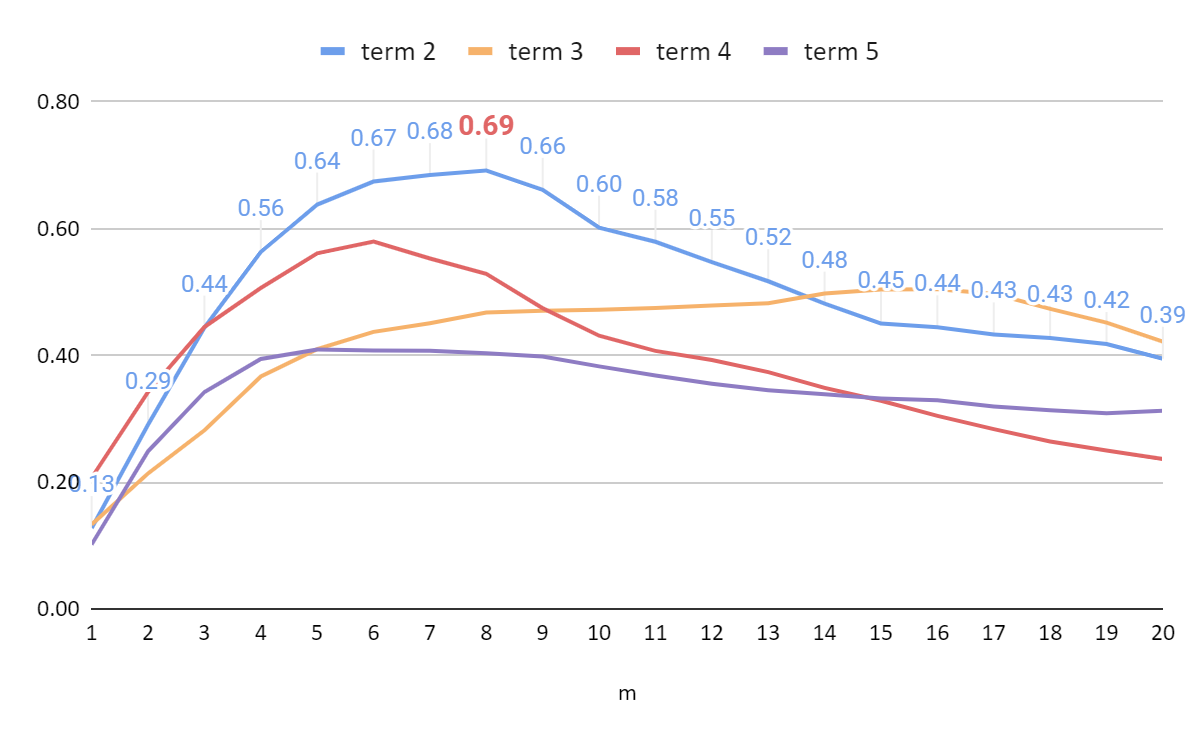
| **Siêu tham số** | **Ý nghĩa** | **Khoảng giá trị** | **Bước nhảy** |
| --- | --- | --- | --- |
| pooling | Hàm tổng hợp ma trận cosine | min, max, sum, mean |  |
| top\_m | Top **M** các môn phổ biến nhất trong khoa | [1, 20] | 1 |

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp Grid Search với **Baseline 1** cùng với dữ liệu trước và sau khi xử lý. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường **Macro F1-score cao nhất (Max Macro F1-score)**:

**Dữ liệu trước khi xử lý:**

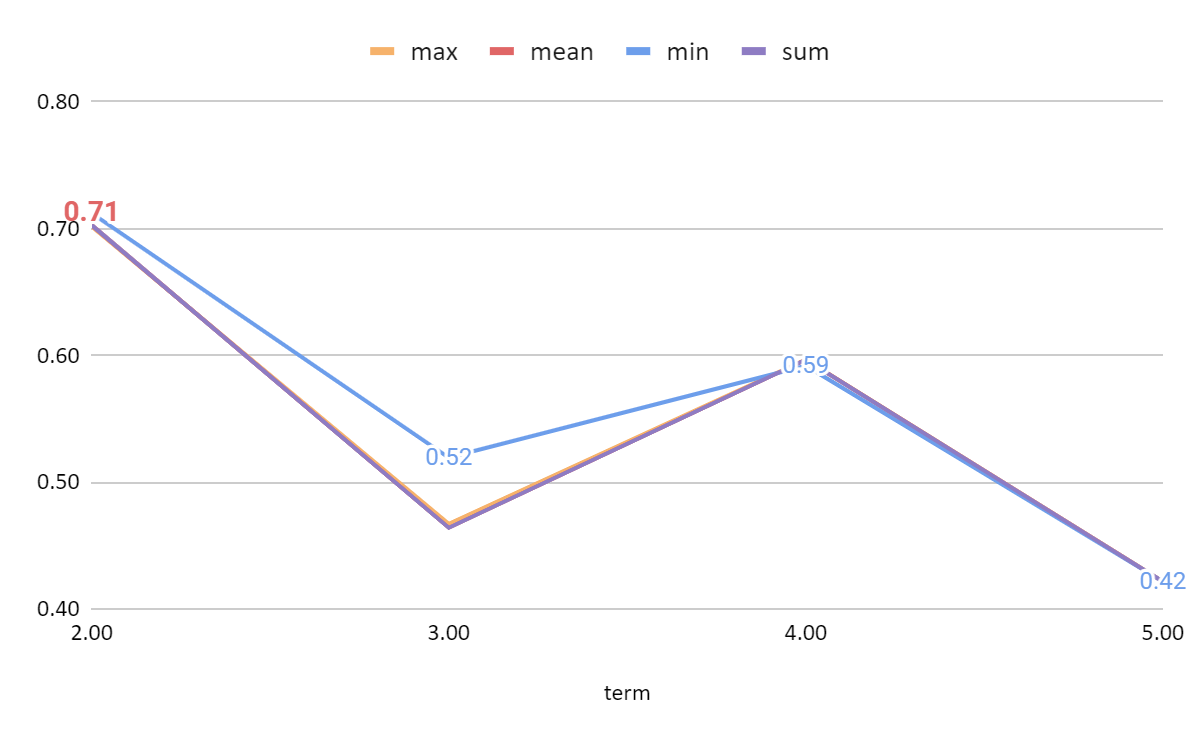


*Biểu đồ gridsearch tham số pooling qua các kỳ học*

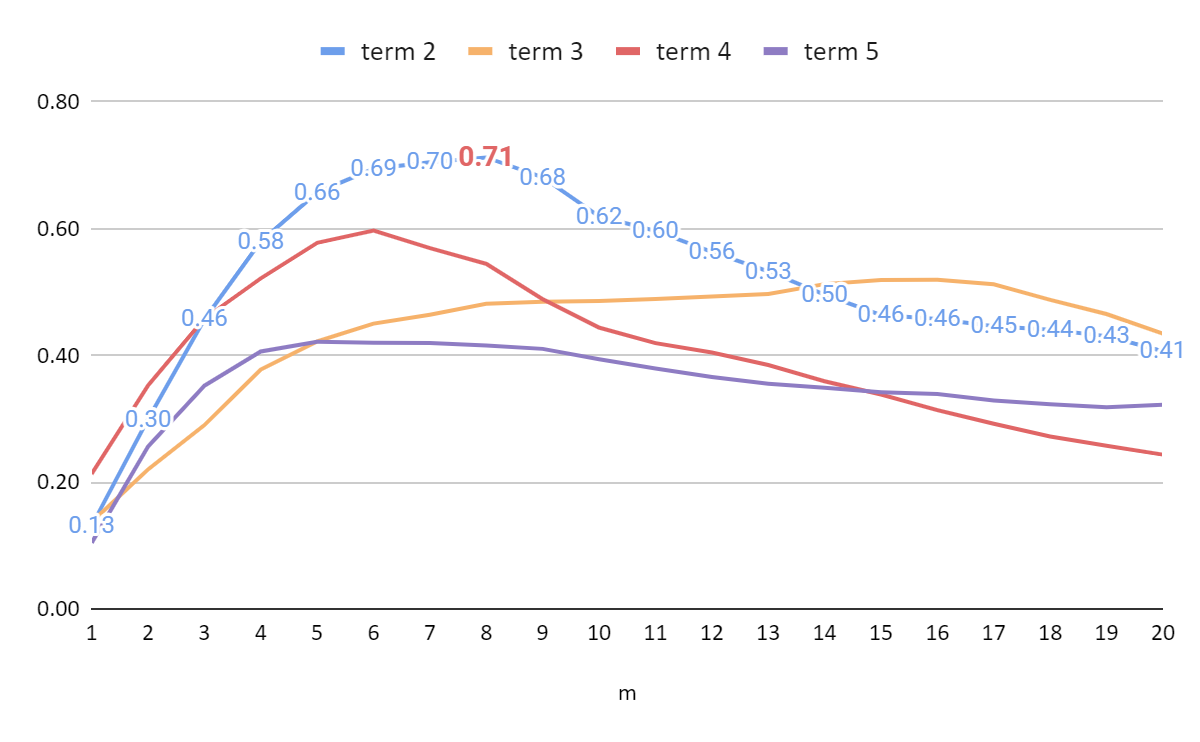


*Biểu đồ gridsearch tham số m qua các kỳ học*

**Dữ liệu sau khi xử lý:**

****

*Biểu đồ gridsearch tham số pooling qua các kỳ học*

**

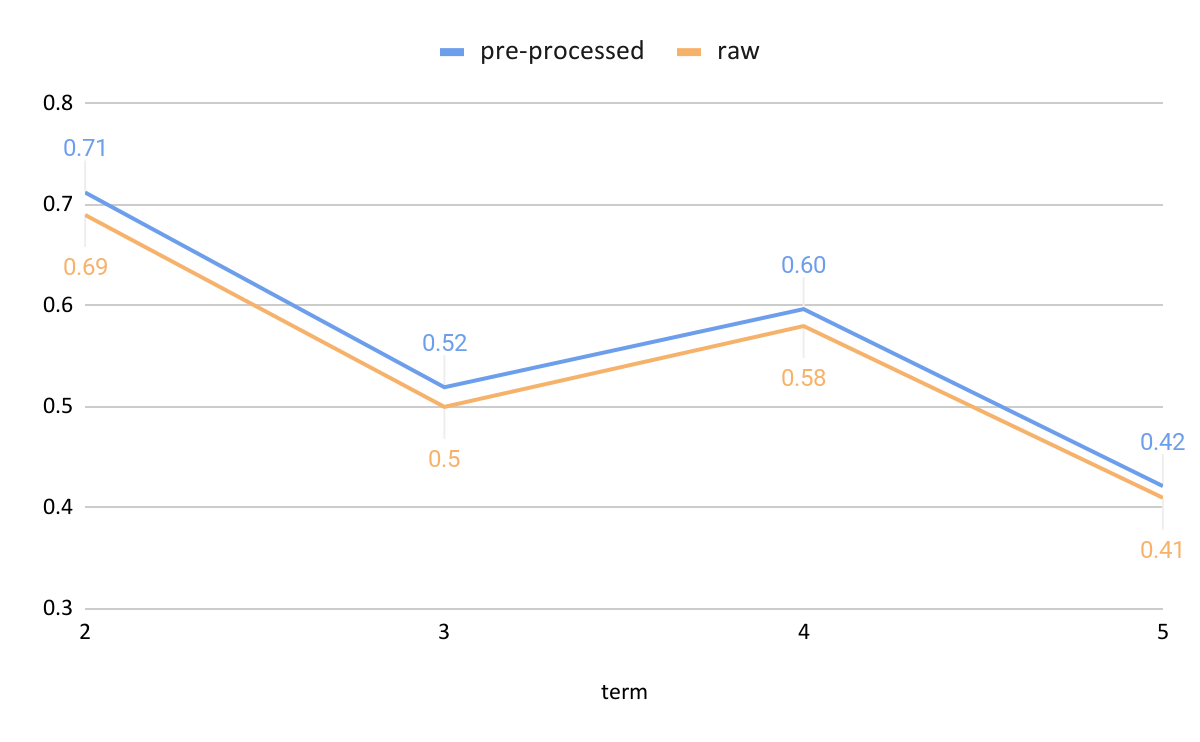
*Biểu đồ gridsearch tham số m qua các kỳ học*

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 1** đạt được **Macro F1-score cao nhất** với các chi tiết siêu tham số sau:

| **Siêu tham số** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| pooling | min | min |
| m | 8 | 8 |
| Macro F1-score cao nhất | **0.69** | **0.71** |

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 1** đạt được **Macro F1-score cao nhất** dao động qua các kỳ học chi tiết như sau:

| **Kỳ học** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| 2 | 0.69 | 0.71 |
| 3 | 0.50 | 0.52 |
| 4 | 0.58 | 0.60 |
| 5 | 0.41 | 0.42 |
| Macro F1-score trung bình | **0.55** | **0.56** |



*Biểu đồ dao động Macro F1-score cao nhất qua các kỳ học*

Qua quá trình thực nghiệm **Grid Search**, **Macro F1-score** của **Baseline 1** dao động lớn giữa các kỳ học. Chi tiết, ở kỳ học 2 **Macro F1-score** đạt kết quả cao nhất với lần lượt ở dữ liệu trước và sau khi xử lý là **0.69** và **0.71**; và giảm dần ở các kỳ học sau. Lý do cho hiện tượng trên là sự mất cân bằng dữ liệu giữa các kỳ học trong các năm học trước đó (trước năm 2014). Ở mặt khác, việc xử lý dữ liệu cũng góp phần gia tăng độ hiệu quả của **Baseline 1** khi tăng **Macro F1-score trung bình** của các kỳ học từ **0.55 lên 0.56 (tăng gần 3% lần).**

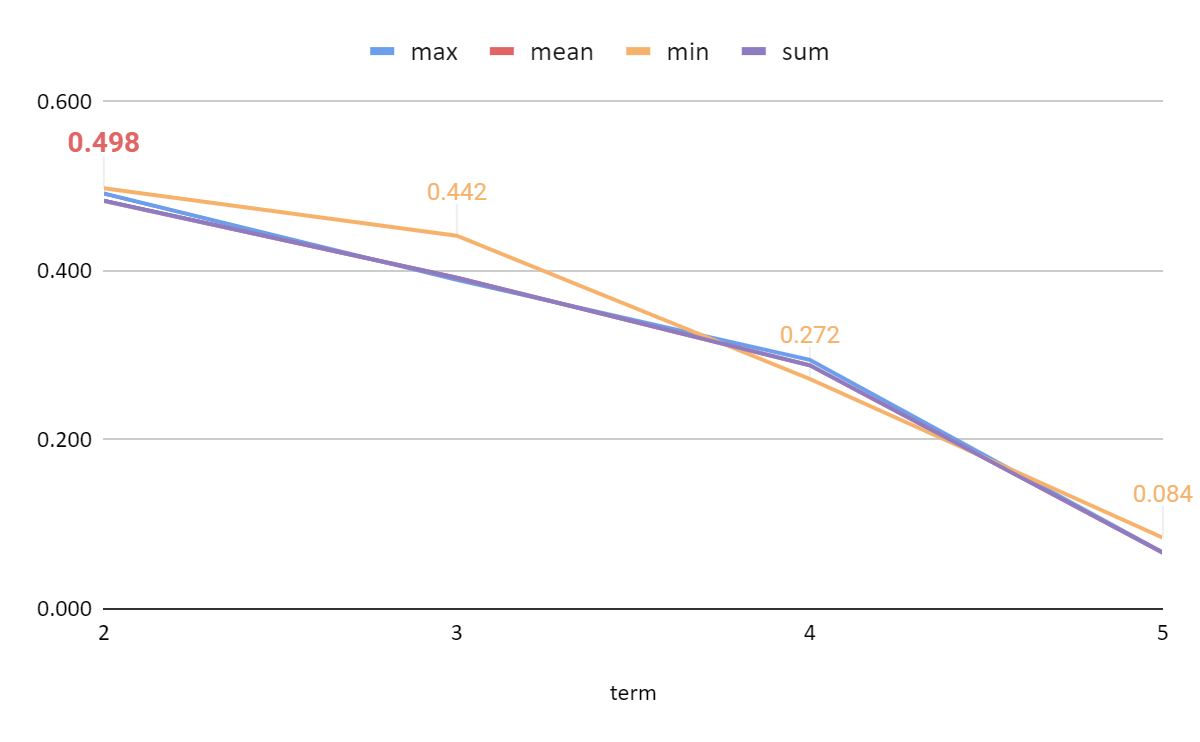
### 4.2. Baseline 2

Phương pháp **Grid Search** được sử dụng để thử qua tất cả tổ hợp siêu tham số của **Baseline 2** nhằm tìm ra tổ hợp có **Macro F1-score** cao nhất qua các kỳ. Các siêu tham số bao gồm: **pooling** và **k**. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

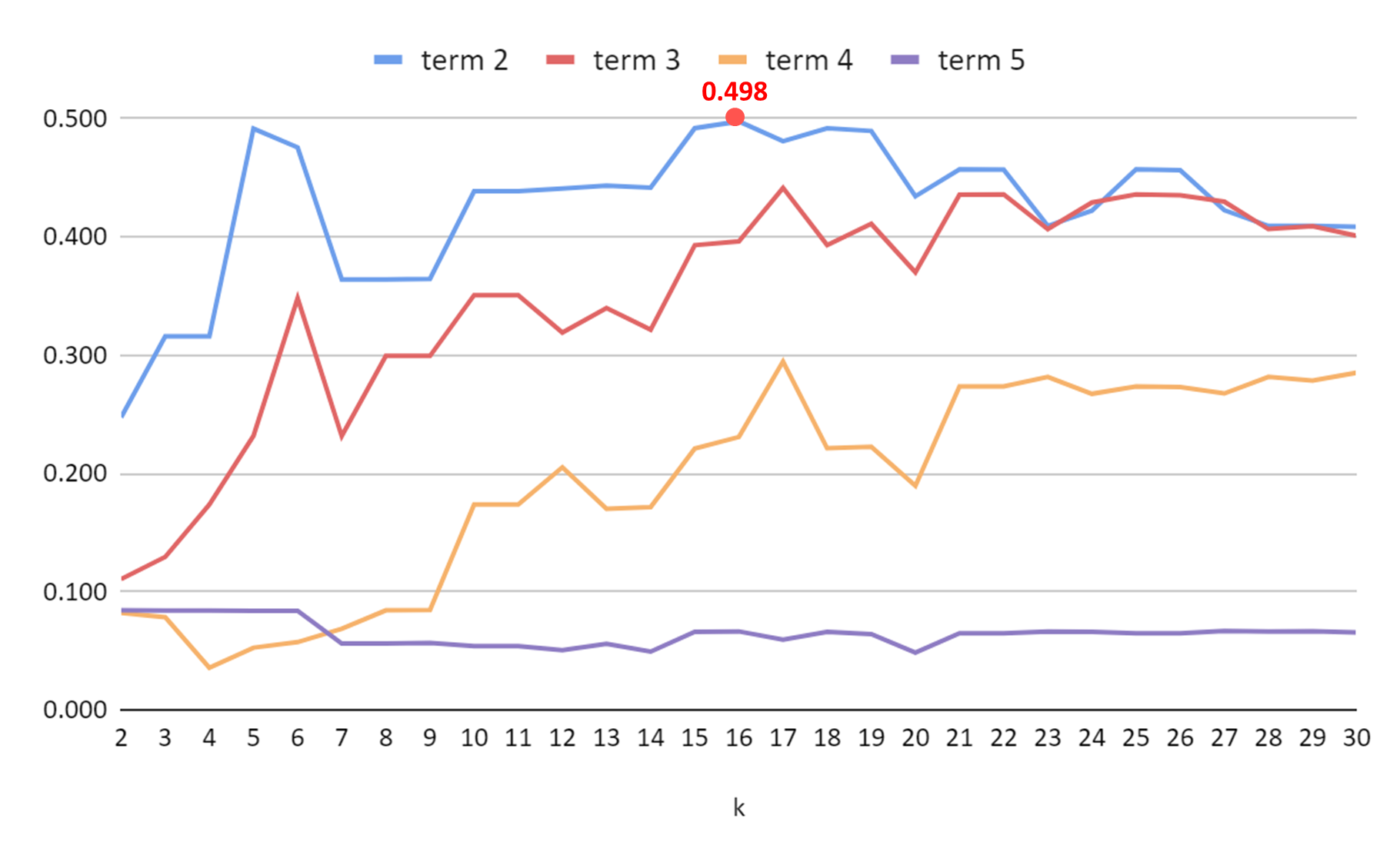
| **Siêu tham số** | **Ý nghĩa** | **Khoảng giá trị** | **Bước nhảy** |
| --- | --- | --- | --- |
| pooling | Hàm tổng hợp ma trận cosine | min, max, sum, mean |  |
| k | K nhóm phân cụm | [2, 30] | 1 |

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp Grid Search với **Baseline 2** cùng với dữ liệu trước và sau khi xử lý. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường **Macro F1-score cao nhất (Max Macro F1-score)**:

**Dữ liệu trước khi xử lý:**

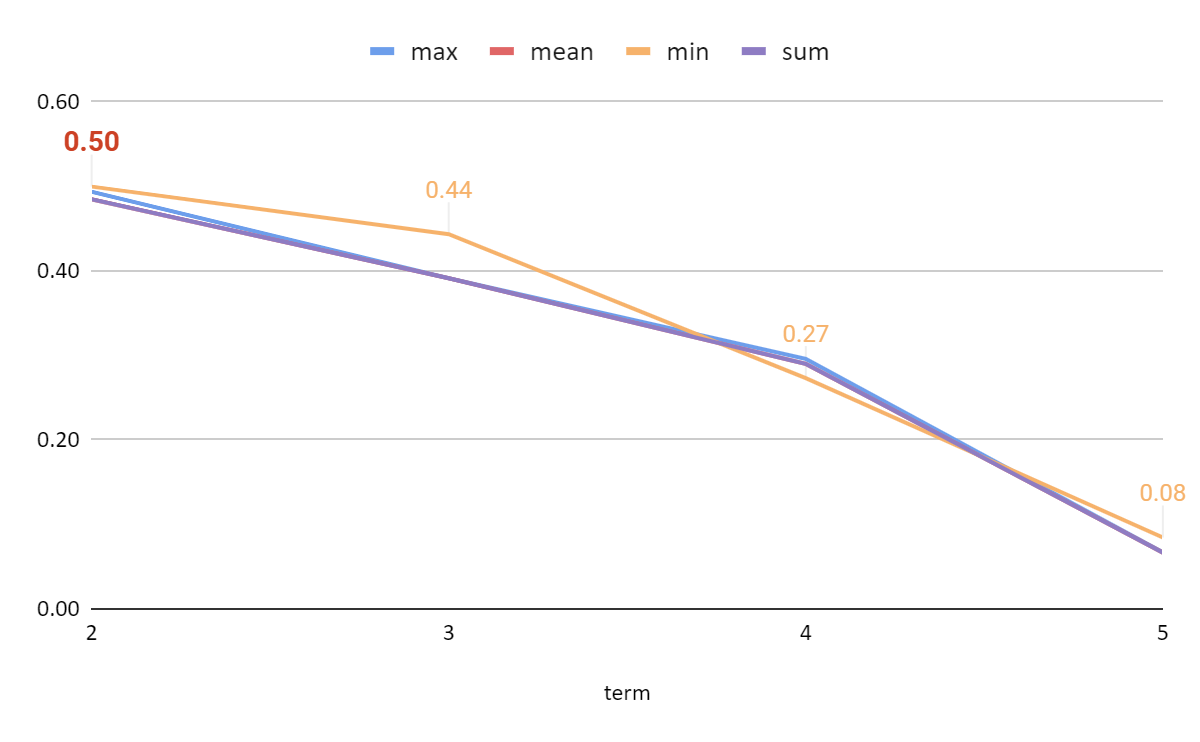


*Biểu đồ gridsearch tham số pooling qua các kỳ học*

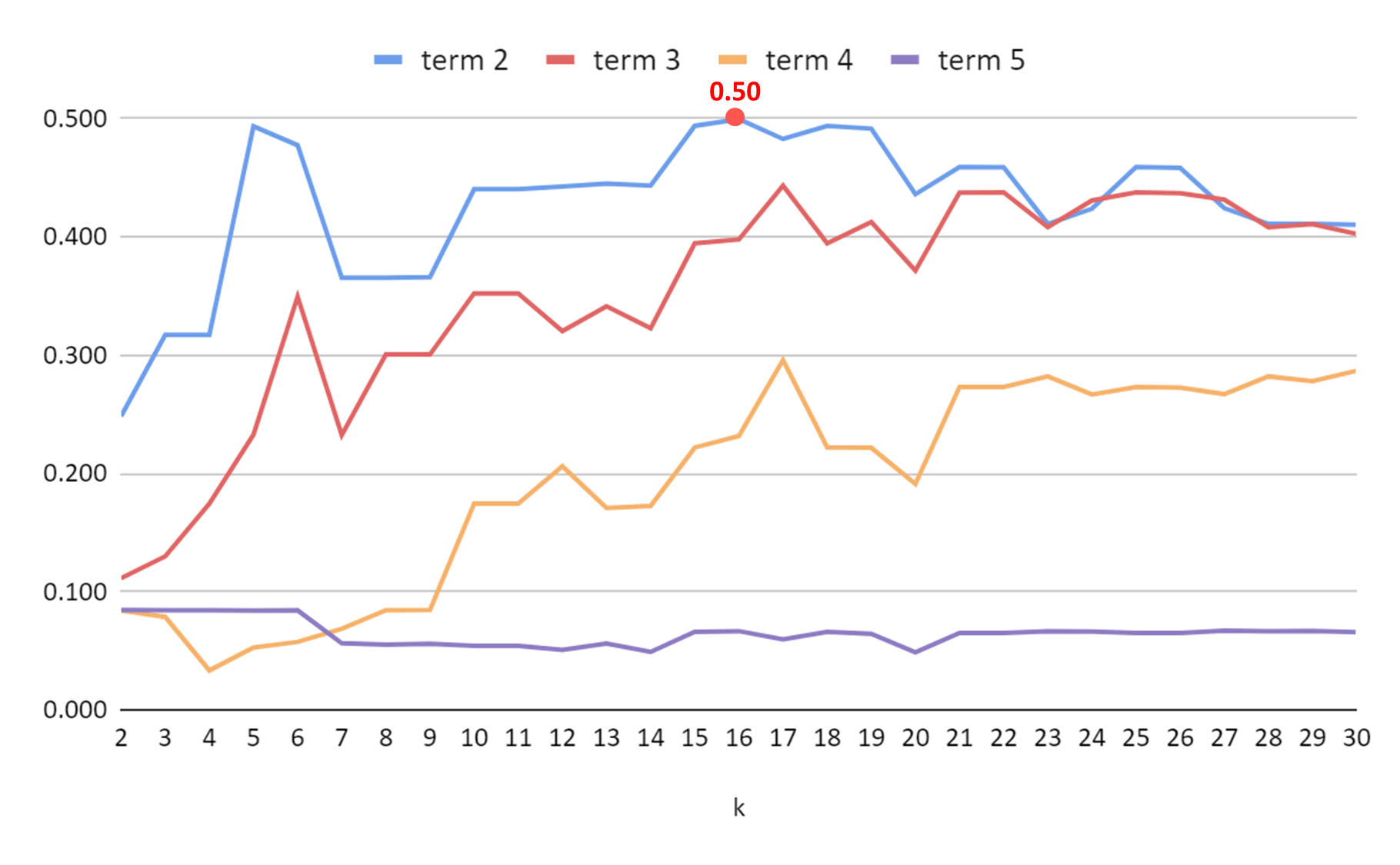


*Biểu đồ gridsearch tham số k qua các kỳ học*

**Dữ liệu sau khi xử lý:**

****

*Biểu đồ gridsearch tham số pooling qua các kỳ học*

**

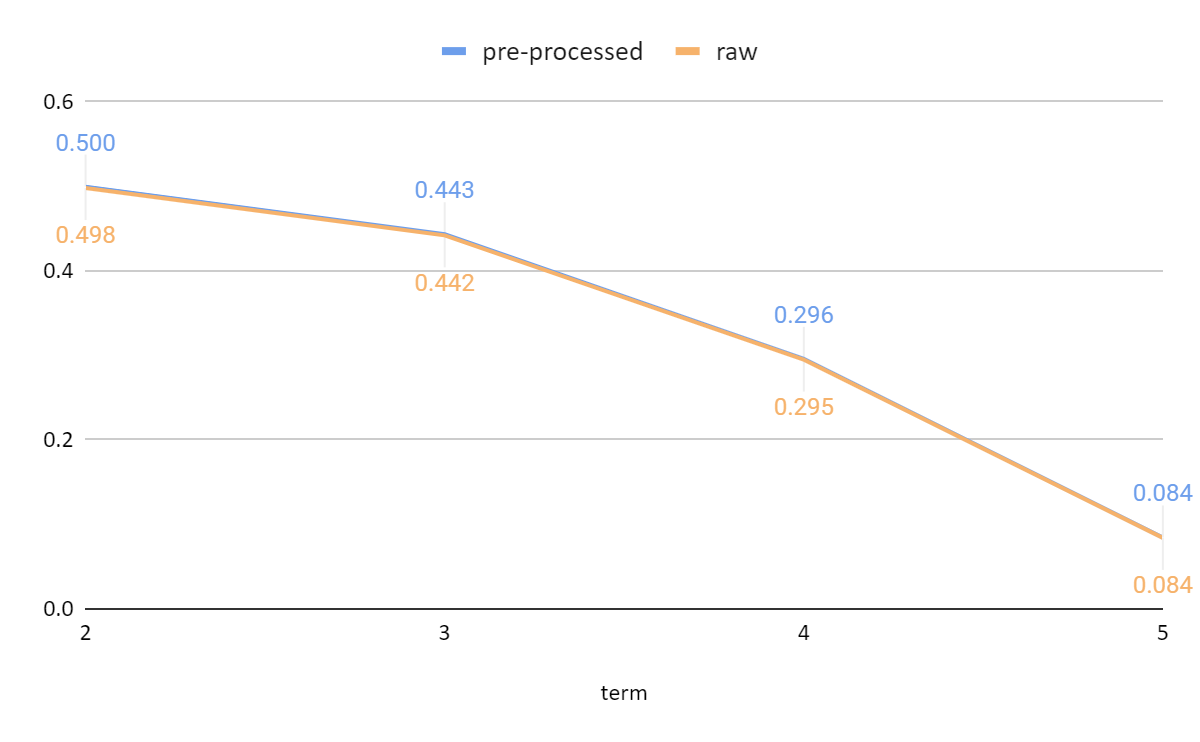
*Biểu đồ gridsearch tham số k qua các kỳ học*

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 2** đạt được **Macro F1-score cao nhất** với các chi tiết siêu tham số sau:

| **Siêu tham số** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| pooling | min | min |
| k | 16 | 16 |
| Macro F1-score cao nhất | **0.498** | **0.50** |

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 2** đạt được **Macro F1-score cao nhất** dao động qua các kỳ học chi tiết như sau:

| **Kỳ học** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| 2 | 0.498 | 0.50 |
| 3 | 0.442 | 0.443 |
| 4 | 0.295 | 0.296 |
| 5 | 0.084 | 0.084 |
| Macro F1-score trung bình | **0.330** | **0.331** |



*Biểu đồ dao động Macro F1-score cao nhất qua các kỳ học*

Qua quá trình thực nghiệm **Grid Search**, **Macro F1-score** của **Baseline 2** dao động lớn giữa các kỳ học. Chi tiết, ở kỳ học 2 **Macro F1-score** đạt kết quả cao nhất với lần lượt ở dữ liệu trước và sau khi xử lý là **0.498** và **0.50**; và giảm dần ở các kỳ học sau. Lý do cho hiện tượng trên là sự mất cân bằng dữ liệu giữa các kỳ học trong các năm học trước đó (trước năm 2014). Ở mặt khác, việc xử lý dữ liệu cũng gia tăng không đáng kể độ hiệu quả của **Baseline 2** khi tăng **Macro F1-score trung bình** của các kỳ học từ **0.330 lên 0.331 (tăng gần 0.3% lần).**

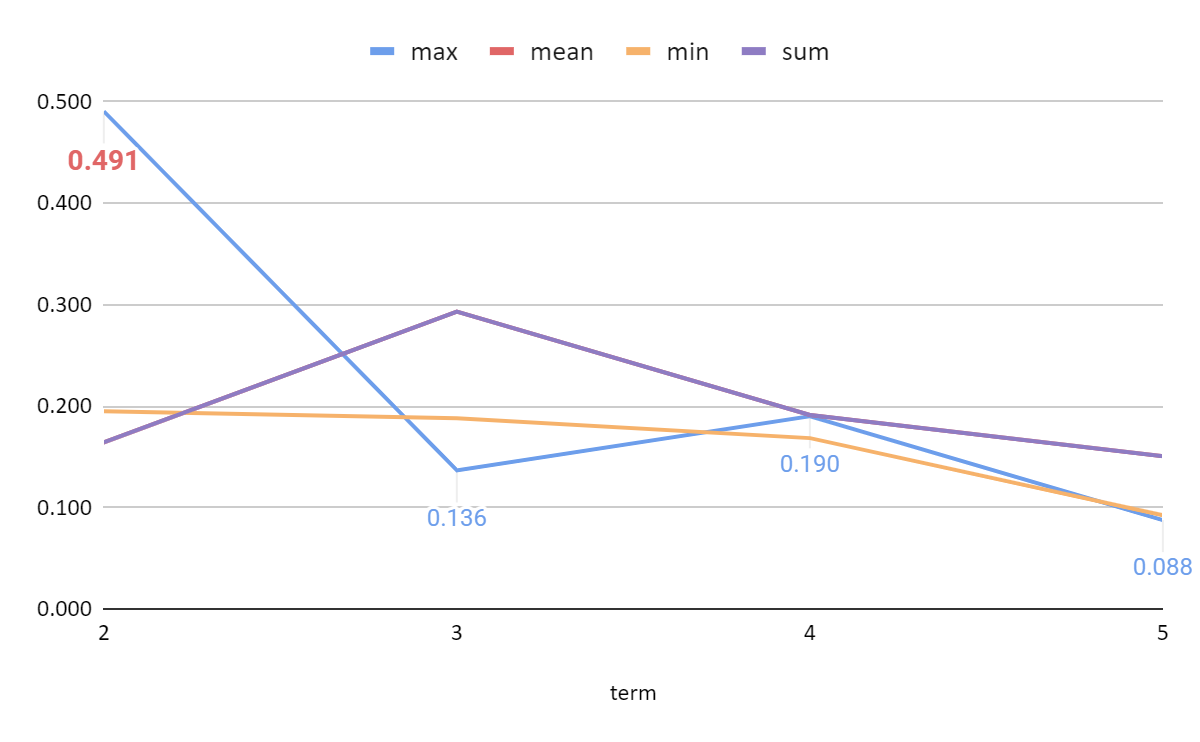
### 4.3. Baseline 3

Phương pháp **Grid Search** được sử dụng để thử qua tất cả tổ hợp siêu tham số của **Baseline 3** nhằm tìm ra tổ hợp có **Macro F1-score** cao nhất qua các kỳ. Siêu tham pooling là **pooling**. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho siêu tham số trên:

| **Siêu tham số** | **Ý nghĩa** | **Khoảng giá trị** | **Bước nhảy** |
| --- | --- | --- | --- |
| pooling | Hàm tổng hợp ma trận cosine | min, max, sum, mean |  |

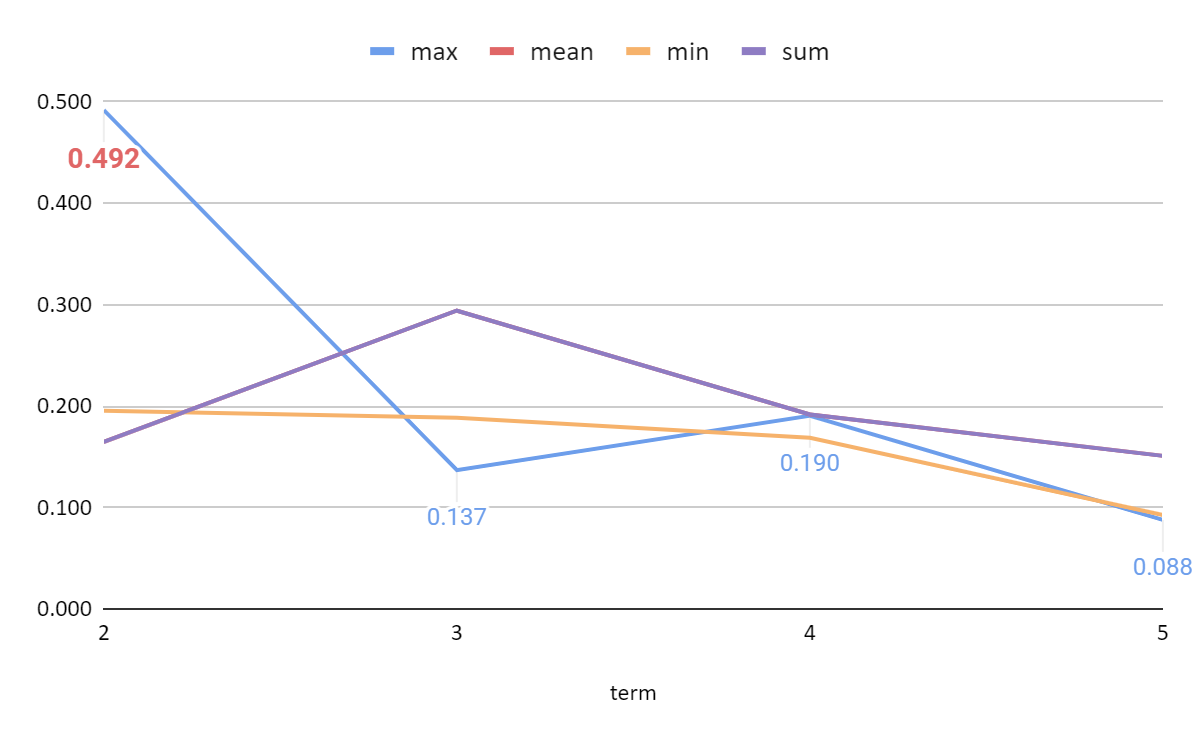
Các biểu đồ thể hiện quá trình thực nghiệm phương pháp Grid Search với **Baseline 3** cùng với dữ liệu trước và sau khi xử lý. Quá trình so sánh siêu tham số được thực hiện dựa trên hệ đo lường **Macro F1-score cao nhất (Max Macro F1-score)**:

**Dữ liệu trước khi xử lý:**

****

*Biểu đồ gridsearch tham số pooling qua các kỳ học*

**Dữ liệu sau khi xử lý:**

****

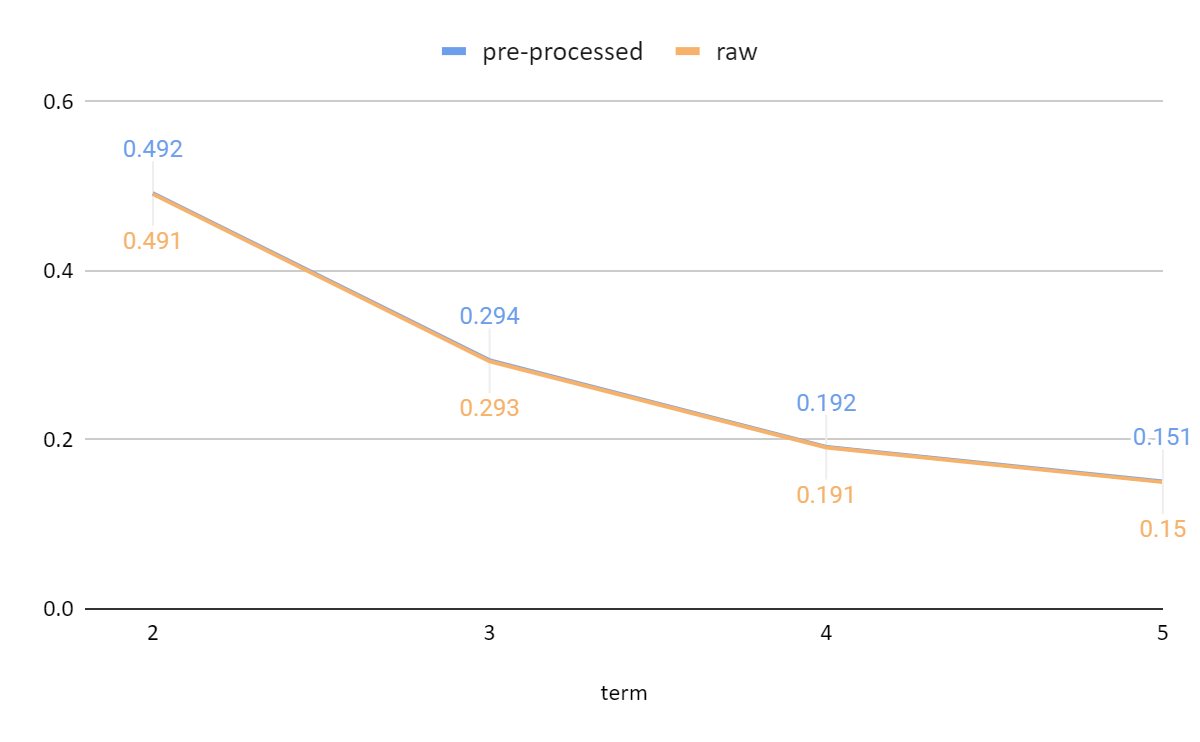
*Biểu đồ gridsearch tham số pooling qua các kỳ học*

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 3** đạt được **Macro F1-score cao nhất** với các chi tiết siêu tham số sau:

| **Siêu tham số** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| pooling | max | max |
| Macro F1-score cao nhất | **0.491** | **0.492** |

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 3** đạt được **Macro F1-score cao nhấ**t dao động qua các kỳ học chi tiết như sau:

| **Kỳ học** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| 2 | 0.491 | 0.492 |
| 3 | 0.293 | 0.294 |
| 4 | 0.191 | 0.192 |
| 5 | 0.150 | 0.151 |
| Macro F1-score trung bình | **0.281** | **0.282** |



*Biểu đồ dao động Macro F1-score cao nhất qua các kỳ học*

Qua quá trình thực nghiệm **Grid Search**, **Macro F1-score** của **Baseline 3** dao động lớn giữa các kỳ học. Chi tiết, ở kỳ học 2 **Macro F1-score** đạt kết quả cao nhất với lần lượt ở dữ liệu trước và sau khi xử lý là **0.491** và **0.492**; và giảm dần ở các kỳ học sau. Lý do cho hiện tượng trên là sự mất cân bằng dữ liệu giữa các kỳ học trong các năm học trước đó (trước năm 2014). Ở mặt khác, việc xử lý dữ liệu không góp phần gia tăng đáng kể độ hiệu quả của **Baseline 3** khi chỉ tăng **Macro F1-score trung bình** của các kỳ học từ **0.281 lên 0.282 (tăng gần 0.3% lần).**

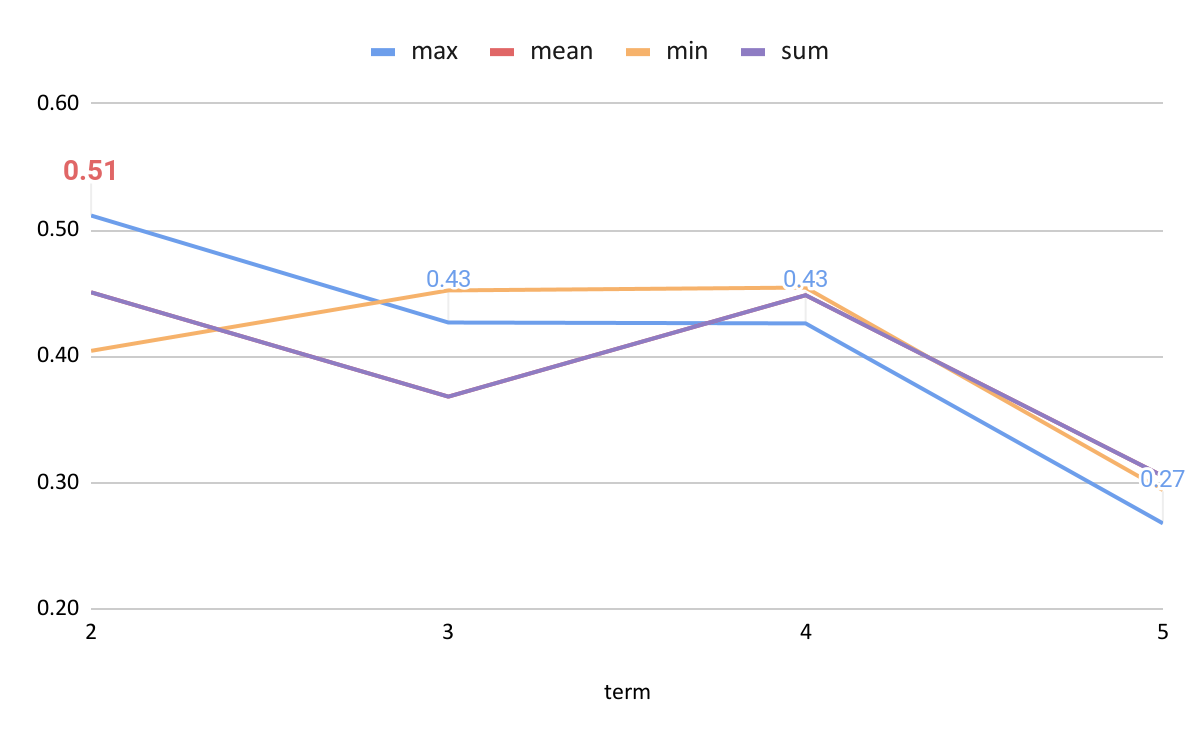
### 4.4. Baseline 4

Phương pháp **Grid Search** được sử dụng để thử qua tất cả tổ hợp siêu tham số của **Baseline 4** nhằm tìm ra tổ hợp có **Macro F1-score** cao nhất qua các kỳ. Các siêu tham số bao gồm: **pooling** và **top\_m**. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

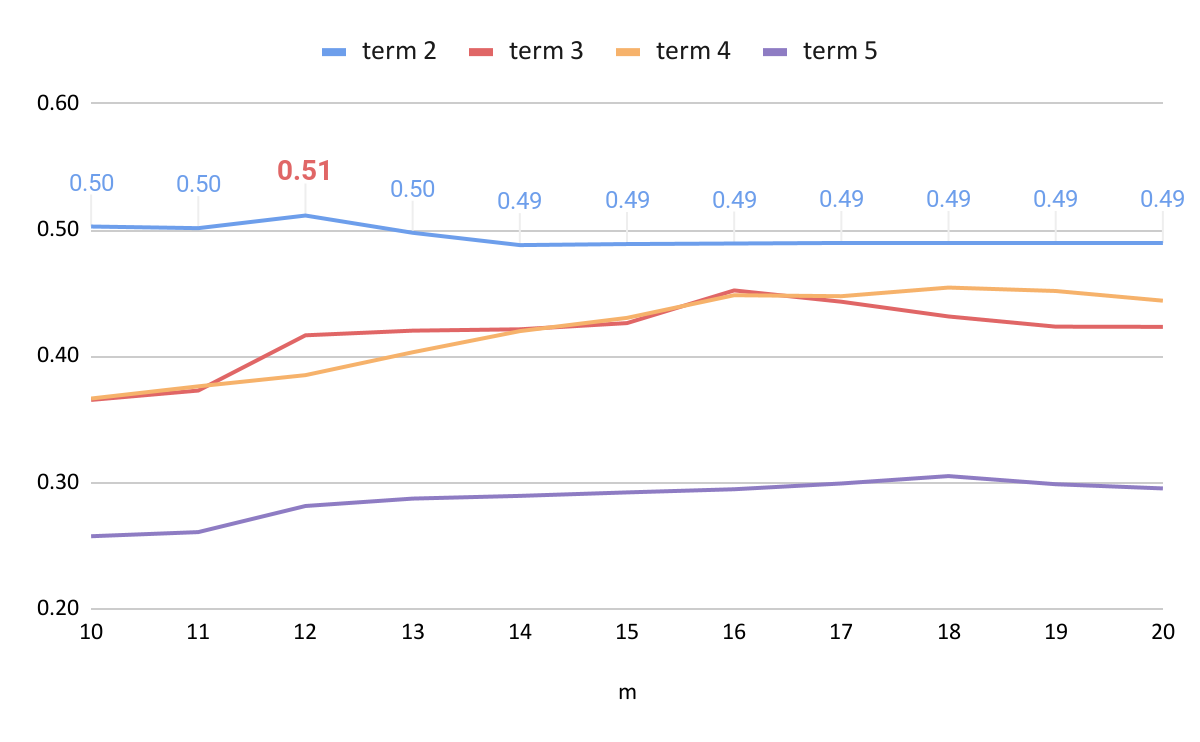
| **Siêu tham số** | **Ý nghĩa** | **Khoảng giá trị** | **Bước nhảy** |
| --- | --- | --- | --- |
| pooling | Hàm tổng hợp ma trận cosine | min, max, sum, mean |  |
| top\_m | Top **M** các môn học có **Cosine Similarity** sau tổng hợp cao nhất | [10, 20] | 1 |

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp Grid Search với **Baseline 4** cùng với dữ liệu trước và sau khi xử lý. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường **Macro F1-score cao nhất (Max Macro F1-score)**:

**Dữ liệu trước khi xử lý:**

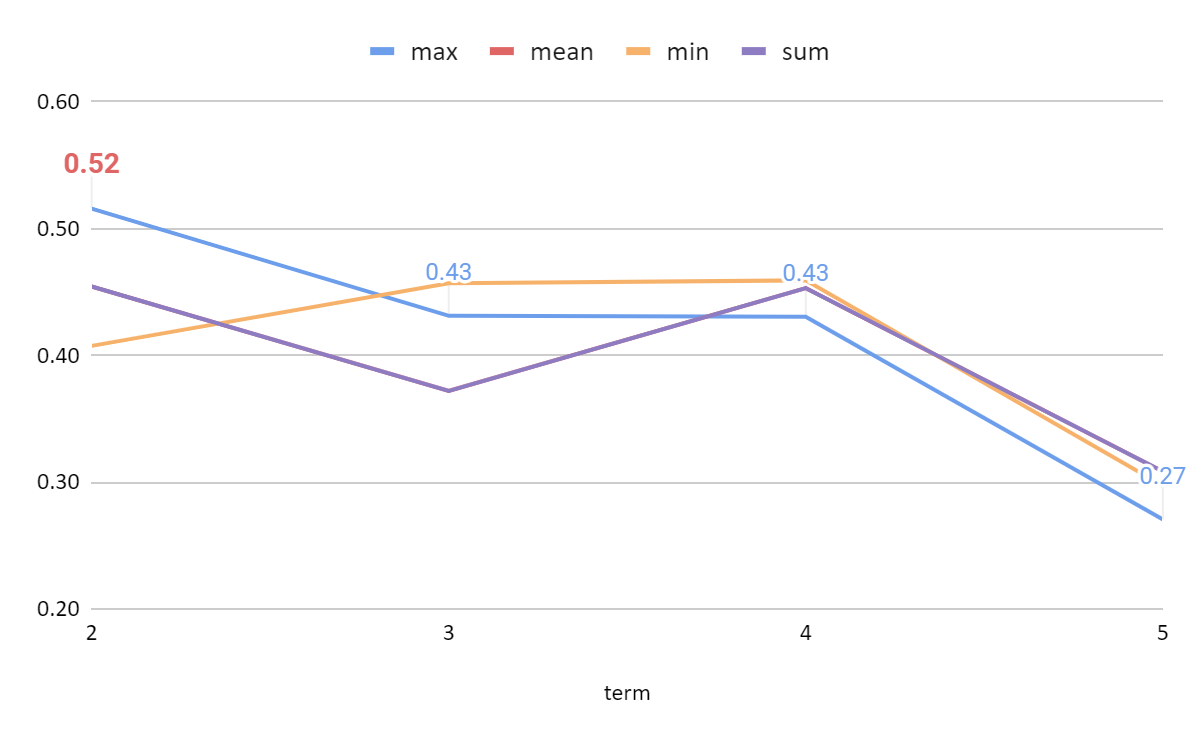


*Biểu đồ gridsearch tham số pooling qua các kỳ học*

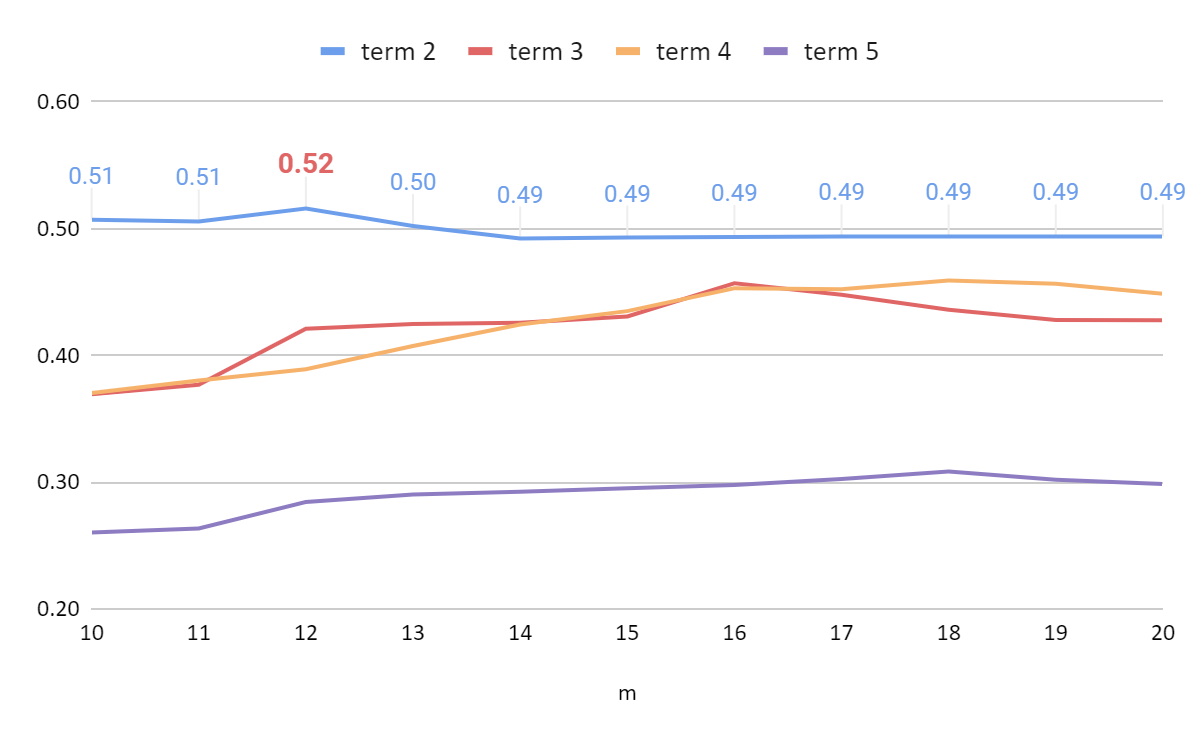


*Biểu đồ gridsearch tham số m qua các kỳ học*

**Dữ liệu sau khi xử lý:**

****

*Biểu đồ gridsearch tham số pooling qua các kỳ học*

**

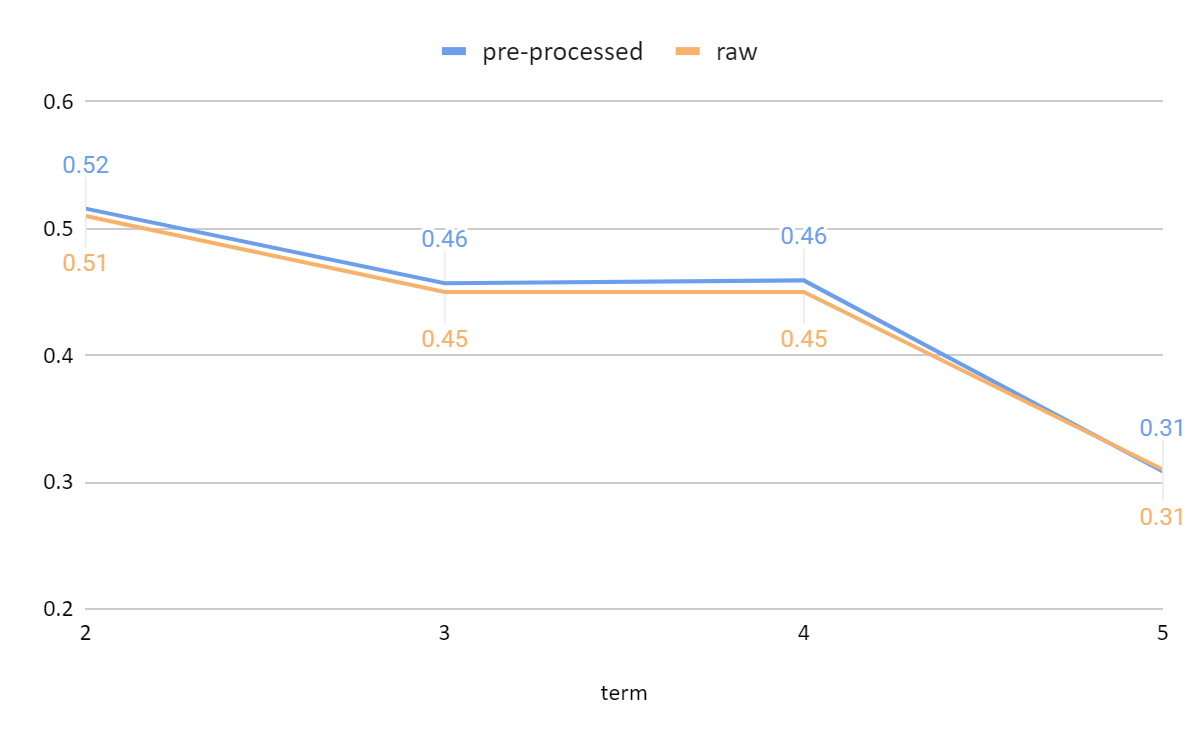
*Biểu đồ gridsearch tham số m qua các kỳ học*

Qua thực nghiệm bằng phương pháp Grid Search, **Baseline 4** đạt được **Macro F1-score** cao nhất với các chi tiết siêu tham số sau:

| **Siêu tham số** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| pooling | max | max |
| m | 12 | 12 |
| Macro F1-score cao nhất | **0.51** | **0.52** |

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 4** đạt được **Macro F1-score cao nhất** dao động qua các kỳ học chi tiết như sau:

| **Kỳ học** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| 2 | 0.51 | 0.52 |
| 3 | 0.45 | 0.46 |
| 4 | 0.45 | 0.46 |
| 5 | 0.31 | 0.31 |
| Macro F1-score trung bình | **0.43** | **0.44** |



*Biểu đồ dao động Macro F1-score cao nhất qua các kỳ học*

Qua quá trình thực nghiệm **Grid Search**, **Macro F1-score** của **Baseline 4** dao động lớn giữa các kỳ học. Chi tiết, ở kỳ học 2 **Macro F1-score** đạt kết quả cao nhất với lần lượt ở dữ liệu trước và sau khi xử lý là **0.51** và **0.52**; và giảm dần ở các kỳ học sau. Lý do cho hiện tượng trên là sự mất cân bằng dữ liệu giữa các kỳ học trong các năm học trước đó (trước năm 2014). Ở mặt khác, việc xử lý dữ liệu cũng góp phần gia tăng độ hiệu quả của Baseline 1 khi tăng Macro F1-score trung bình của các kỳ học từ **0.43 lên 0.44 (tăng hơn 2% lần).**

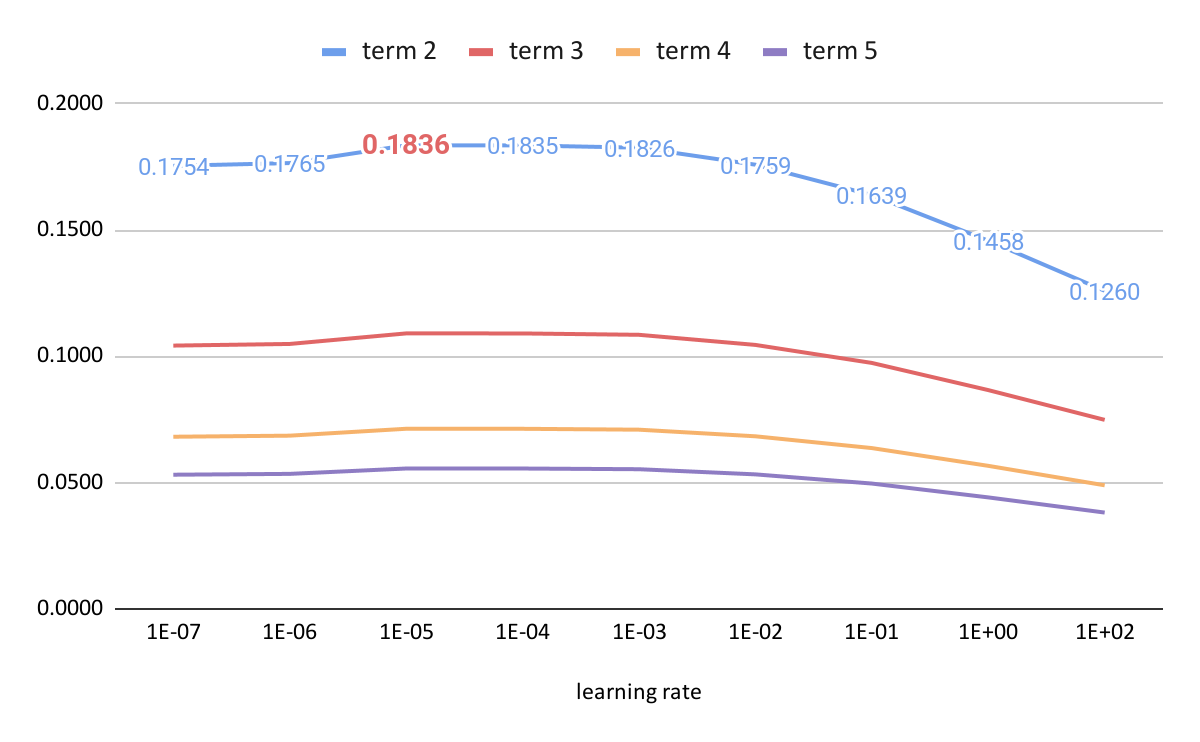
### 4.5. Baseline 5

Phương pháp **Grid Search** được sử dụng để thử qua tất cả tổ hợp siêu tham số của **Baseline 5** nhằm tìm ra tổ hợp có **Macro F1-score** cao nhất qua các kỳ. Các siêu tham số bao gồm: **learning rate, epoch** và **batch size**. Bảng dưới đây là các khoảng giá trị chi tiết áp dụng cho các siêu tham số trên:

| **Siêu tham số** | **Ý nghĩa** | **Khoảng giá trị** | **Bước nhảy** |
| --- | --- | --- | --- |
| learning rate | Tốc độ học của mô hình | 10e[n] n = [-7, 2] | 1 |
| epoch | Số vòng lặp mà mô hình huấn luyện toàn bộ dữ liệu | [1, 20] | 1 |
| batch size | Số lượng mẫu dữ liệu được sử dụng trong một lần cập nhật trọng số mô hình | 2^[n] n = [0, 8] | 1 |

Các biểu đồ thể hiện quá trình thực nghiệm phương pháp Grid Search với **Baseline 5** cùng với dữ liệu trước và sau khi xử lý. Quá trình so sánh các siêu tham số được thực hiện dựa trên hệ đo lường **Macro F1-score cao nhất (Max Macro F1-score)**:

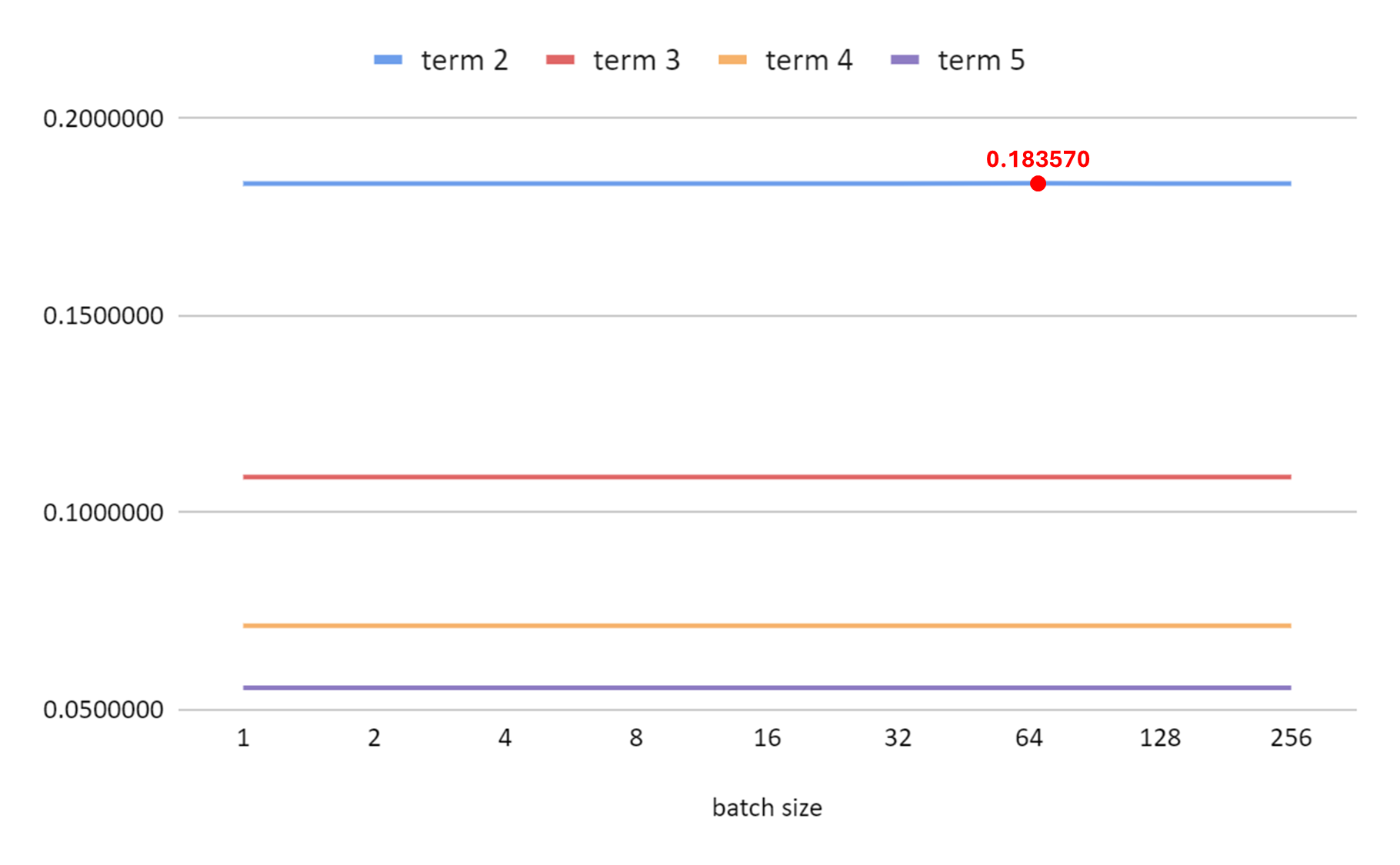
**Dữ liệu trước khi xử lý:**



*Biểu đồ gridsearch tham số learning rate qua các kỳ học*



*Biểu đồ gridsearch tham số epoch qua các kỳ học*

**

*Biểu đồ gridsearch tham số batch size qua các kỳ học*

**Dữ liệu sau khi xử lý:**

****

*Biểu đồ gridsearch tham số learning rate qua các kỳ học*

**

*Biểu đồ gridsearch tham số epoch qua các kỳ học*

**

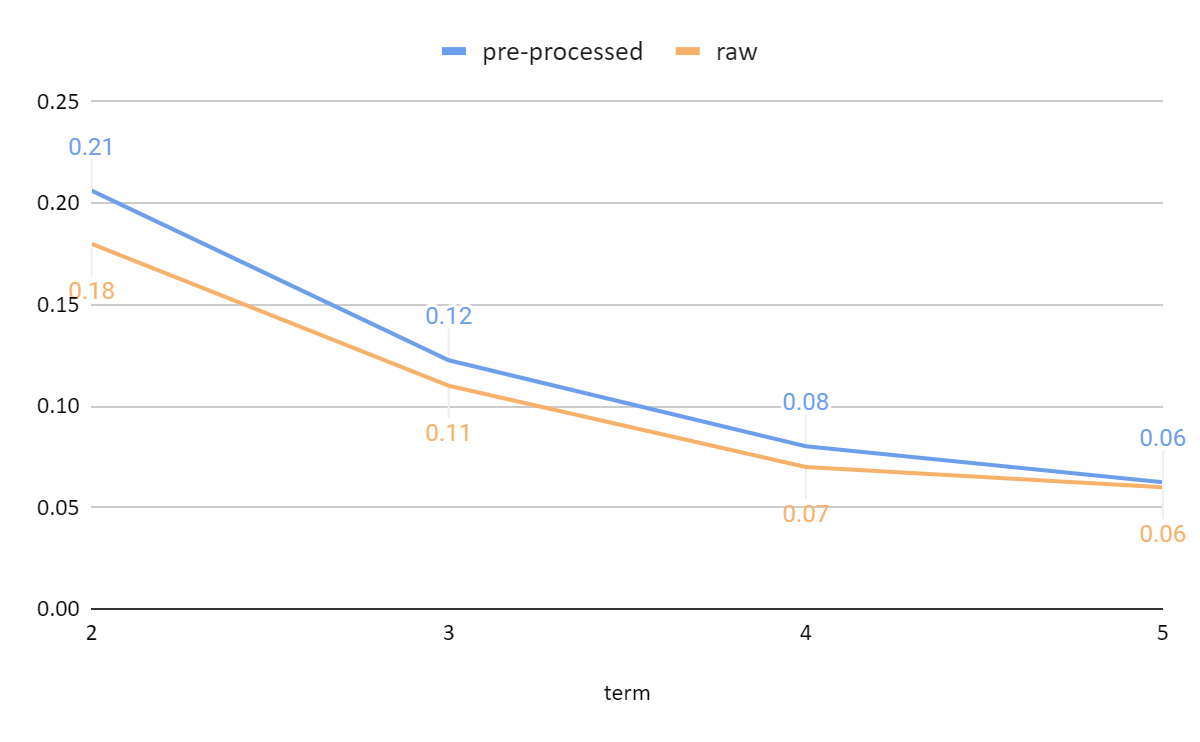
*Biểu đồ gridsearch tham số batch size qua các kỳ học*

Qua thực nghiệm bằng phương pháp Grid Search, **Baseline 5** đạt được **Macro F1-score** cao nhất với các chi tiết siêu tham số sau:

| **Siêu tham số** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| learning rate | 1e-5 | 1e-5 |
| epoch | 16 | 16 |
| batch size | 64 | 64 |
| Macro F1-score cao nhất | **0.183570** | **0.206247** |

Qua thực nghiệm bằng phương pháp **Grid Search**, **Baseline 5** đạt được **Macro F1-score cao nhất** dao động qua các kỳ học chi tiết như sau:

| **Kỳ học** | **Dữ liệu trước khi xử lý** | **Dữ liệu sau khi xử lý** |
| --- | --- | --- |
| 2 | 0.18 | 0.21 |
| 3 | 0.11 | 0.12 |
| 4 | 0.07 | 0.08 |
| 5 | 0.06 | 0.06 |
| Macro F1-score trung bình | **0.10** | **0.12** |



*Biểu đồ dao động Macro F1-score cao nhất qua các kỳ học*

Qua quá trình thực nghiệm **Grid Search**, **Macro F1-score** của **Baseline 5** dao động lớn giữa các kỳ học. Chi tiết, ở kỳ học 2 **Macro F1-score** đạt kết quả cao nhất với lần lượt ở dữ liệu trước và sau khi xử lý là **0.18** và **0.21**; và giảm dần ở các kỳ học sau. Lý do cho hiện tượng trên là sự mất cân bằng dữ liệu giữa các kỳ học trong các năm học trước đó (trước năm 2014). Ở mặt khác, việc xử lý dữ liệu cũng góp phần gia tăng độ hiệu quả của Baseline 1 khi tăng Macro F1-score trung bình của các kỳ học từ **0.10 lên 0.12 (tăng 20% lần).**

# **CHƯƠNG IV: PHÁT TRIỂN ỨNG DỤNG**

## **1.** Giới thiệu ứng dụng

### 1.1. Bài toán đặt ra

Trong bối cảnh hiện nay, việc hỗ trợ sinh viên lập kế hoạch học tập hiệu quả là một vấn đề thiết yếu đối với các cơ sở giáo dục. Để giải quyết bài toán này, hệ thống được xây dựng với mục tiêu cung cấp cho sinh viên UIT một công cụ khuyến nghị môn học dựa trên lịch sử học tập. Hệ thống sẽ phân tích dữ liệu học tập trong quá khứ để đề xuất các môn học phù hợp cho kỳ học tiếp theo, giúp sinh viên tối ưu hóa việc chọn môn học, nâng cao kết quả học tập và rút ngắn thời gian hoàn thành chương trình học. Đây không chỉ là một công cụ hỗ trợ cá nhân hóa mà còn góp phần giảm tải công việc tư vấn học tập của các giảng viên.

### 1.2. Phạm vi hệ thống

Hệ thống sẽ phục vụ trong việc khuyến nghị môn học và cung cấp các tính năng như truy vấn danh sách môn học khuyến nghị, danh sách các môn học đã đăng ký và chưa đăng ký. Ngoài ra, hệ thống được thiết kế để đảm bảo tính nhất quán và hiệu quả khi phục vụ đồng thời nhiều người dùng. Điều này đòi hỏi giao diện người dùng phải trực quan, dễ sử dụng, đồng thời các tính năng phải được xây dựng trên nền tảng công nghệ hiện đại nhằm đảm bảo tính bảo mật và ổn định của dữ liệu. Hệ thống còn hỗ trợ kết nối và tích hợp với các nguồn dữ liệu khác nhau để mở rộng khả năng xử lý và cải thiện chất lượng khuyến nghị.

### 1.3. Đối tượng sử dụng

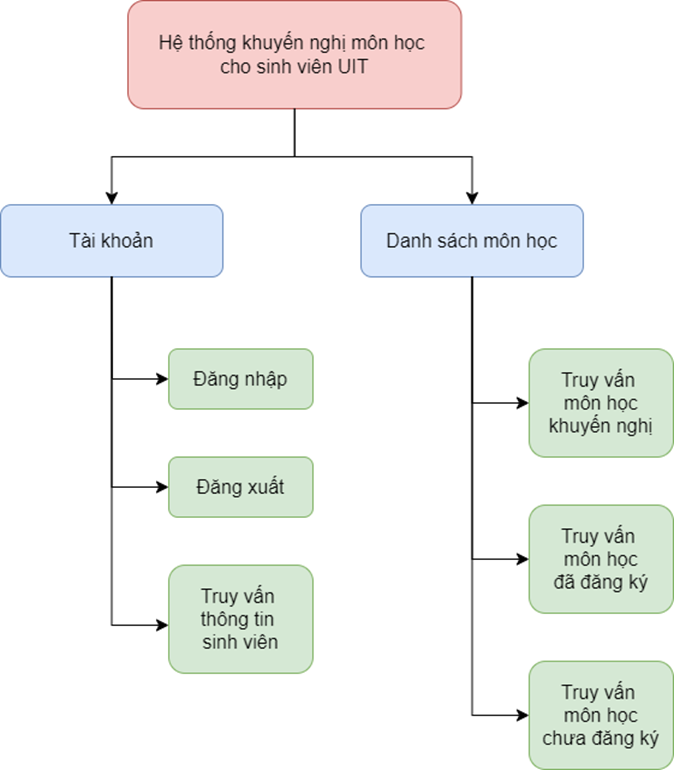
Hệ thống hướng đến hai nhóm đối tượng chính. Đối với **sinh viên UIT**, hệ thống sẽ là một công cụ hỗ trợ quan trọng, giúp họ lựa chọn môn học phù hợp với năng lực, sở thích và lộ trình học tập. Thông qua các khuyến nghị cá nhân hóa, sinh viên có thể tự tin hơn trong việc lên kế hoạch học tập và đạt được mục tiêu học thuật. Đối với **giảng viên và cố vấn học tập**, hệ thống cung cấp các công cụ để theo dõi tiến độ của sinh viên, từ đó đưa ra những tư vấn chính xác và hiệu quả hơn. Điều này không chỉ giảm thời gian xử lý thông tin thủ công mà còn cải thiện chất lượng hỗ trợ học tập.

### 1.4. Các ràng buộc giả định

Để đảm bảo hoạt động ổn định, hệ thống được thiết kế dựa trên một số giả định quan trọng. Thứ nhất, người dùng phải có thiết bị kết nối Internet và sử dụng trình duyệt web để truy cập hệ thống. Điều này đảm bảo khả năng hoạt động liên tục và tính linh hoạt của hệ thống. Thứ hai, người dùng cần được cấp tài khoản và phân quyền truy cập, nhằm đảm bảo tính bảo mật và kiểm soát thông tin. Cuối cùng, hệ thống cần cung cấp tài liệu hướng dẫn sử dụng chi tiết để người dùng hiểu rõ các chức năng, cơ chế hoạt động, cũng như cách khai thác hiệu quả dữ liệu và công cụ mà hệ thống mang lại. Những ràng buộc này là nền tảng để đảm bảo trải nghiệm người dùng tốt nhất và duy trì chất lượng dịch vụ.

## **2.** Mô tả tổng thể

2.1. Mô tả tổng thể hệ thống

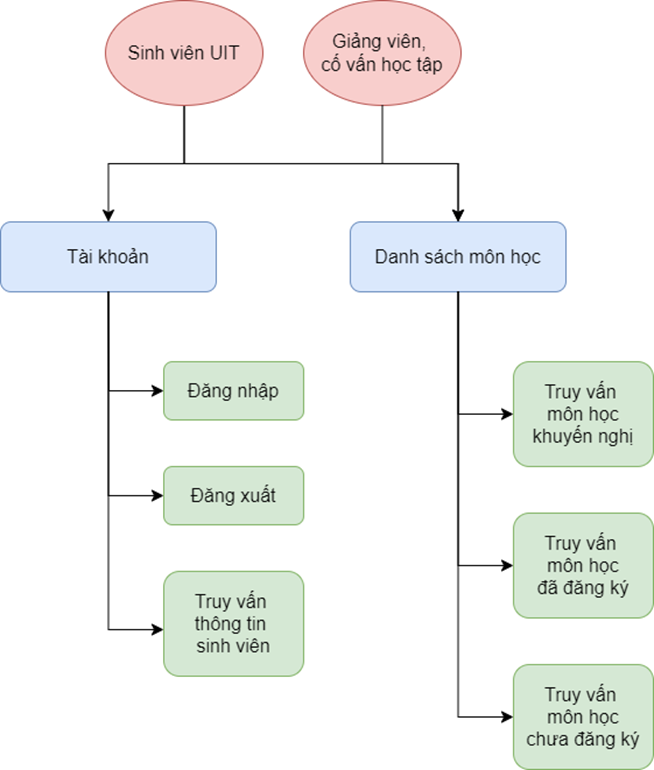


*Mô hình tổng thể hệ thống*

2.2. Chức năng hệ thống

| **STT** | **Tên chức năng** | **Công dụng** |
| --- | --- | --- |
| 1 | Đăng nhập | Người dùng có quyền hạn truy cập vào hệ  thống |
| 2 | Đăng xuất | Người dùng có quyền hạn đăng xuất khỏi hệ  thống bằng tay khi không còn nhu cầu sử  dụng hoặc tự động khi hết hạn phiên sử dụng |
| 3 | Truy vấn thông tin sinh viên | Tra cứu thông tin của sinh viên bao gồm: thông tin cá nhân, ngành học, khoa, hệ đào tạo,… |
| 4 | Truy vấn môn học khuyến nghị | Tra cứu danh sách môn học hệ thống khuyến nghị môn học cho sinh viên trong kỳ học tiếp theo |
| 5 | Truy vấn môn học đã đăng ký | Tra cứu danh sách môn học sinh viên đã đăng ký trong các kỳ học trước |
| 6 | Truy vấn môn học chưa đăng ký | Tra cứu danh sách môn học còn lại mà sinh viên chưa từng đăng ký |

2.3. Mô hình người dùng hệ thống



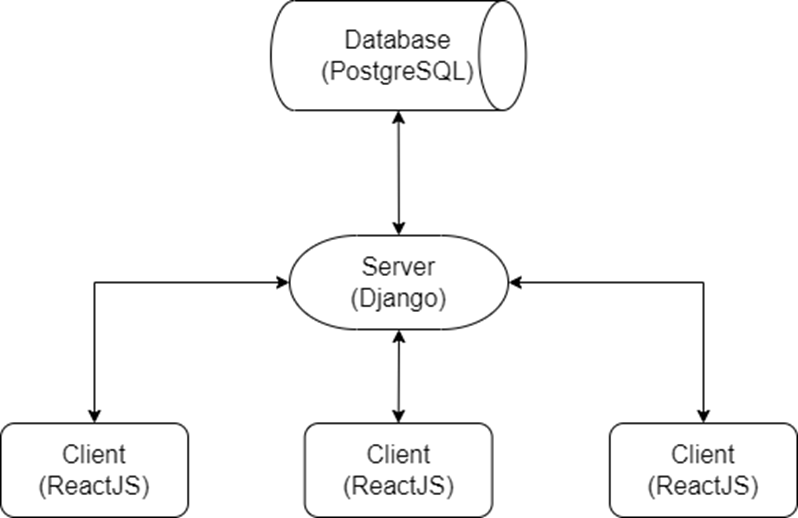
*Mô hình người dùng hệ thống*

## **3.** Thiết kế hệ thống

3.1. Kiến trúc hệ thống

Kiến trúc hệ thống được áp dụng vào đồ án là mô hình Client – Server. Mô hình Client – Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó. Client và Server về bản chất thì nó là 2 máy tính giao tiếp và truyền tải dữ liệu cho nhau:

* Máy tính đóng vai trò là máy khách – Client: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.
* Máy tính đóng vai trò là máy chủ – Server Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn. Máy chủ có khả năng lưu trữ, thay đổi và truy xuất dữ liệu tới cơ sở dữ liệu – Database.



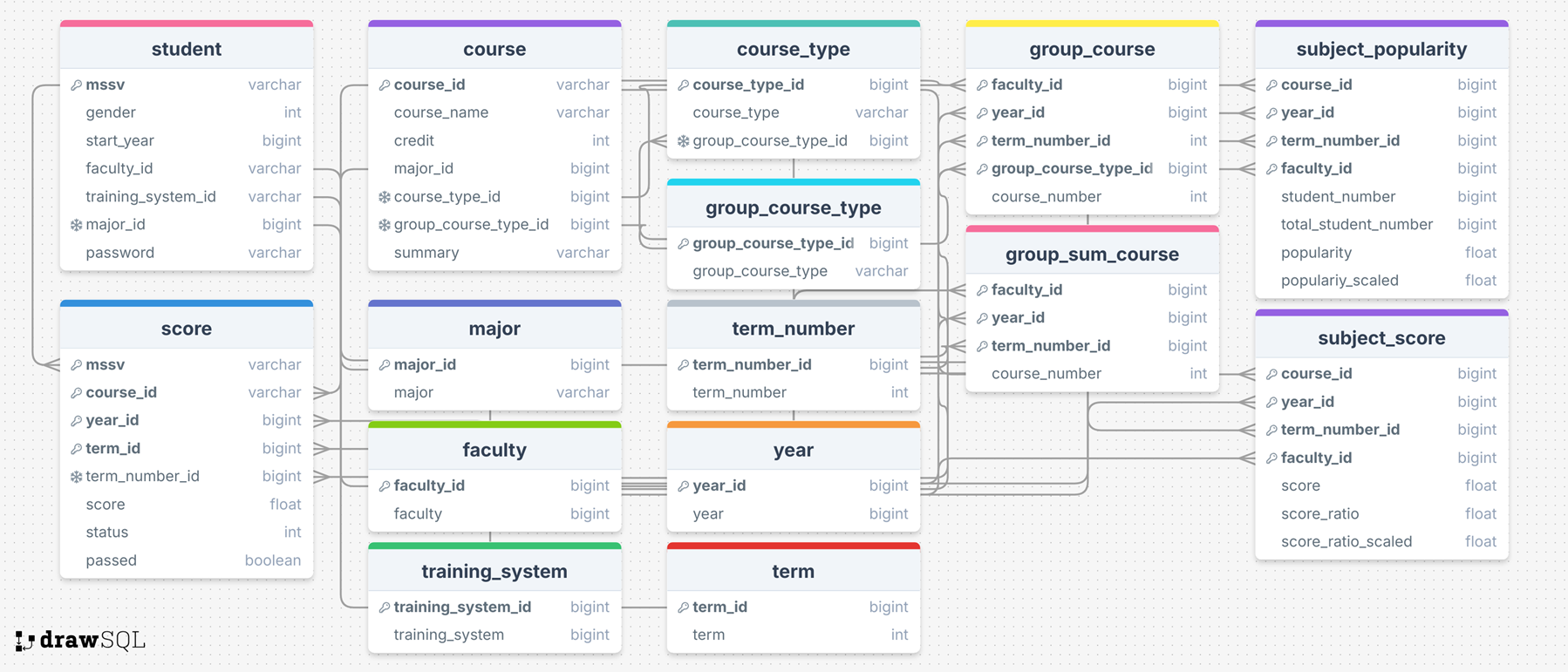
*Kiến trúc hệ thống Client - Server*

3.2. Mô tả các thành phần trong hệ thống

| **STT** | **Thành phần** | **Mô tả** |
| --- | --- | --- |
| 1 | Máy khách - Client  (React JS) | Client cung cấp giao diện người dùng, gửi yêu cầu đến Server và nhận phản hồi. |
| 2 | Máy chủ - Server  (Django) | Server kiểm tra, xử lý yêu cầu và truy cập Database để quản lý dữ liệu khi cần. |
| 3 | Cơ sở dữ liệu - Database (PostgreSQL) | Database lưu trữ dữ liệu dạng bảng liên kết, đảm bảo nhất quán và hỗ trợ quản lý dữ liệu. |

## **4.** Thiết kế cơ sở dữ liệu

### 4.1. Sơ đồ dữ liệu



*Sơ đồ dữ liệu*

### 4.2. Danh sách các bảng dữ liệu

| **STT** | **Bảng dữ liệu** | **Mô tả** |
| --- | --- | --- |
| 1 | student | Lưu trữ thông tin sinh viên |
| 2 | score | Lưu trữ thông tin kết quả học tập của sinh viên |
| 3 | course | Lưu trữ thông tin môn học |
| 4 | major | Lưu trữ thông tin ngành học |
| 5 | faculty | Lưu trữ thông tin khoa |
| 6 | training\_system | Lưu trữ thông tin hệ đào tạo |
| 7 | course\_type | Lưu trữ thông tin loại môn học |
| 8 | group\_course\_type | Lưu trữ thông tin nhóm loại môn học |
| 9 | year | Lưu trữ thông tin năm học |
| 10 | term | Lưu trữ thông tin kỳ học |
| 11 | term\_number | Lưu trữ thông tin số đếm kỳ học |
| 12 | group\_course | Lưu trữ thông tin thống kê số môn học theo năm học, số kỳ học, khoa và nhóm loại môn học |
| 13 | group\_sum\_course | Lưu trữ thông tin thống kê số môn học theo năm học, số kỳ học và khoa |
| 14 | subject\_popularity | Lưu trữ thông tin thống kê độ phổ biến các môn học |
| 15 | subject\_score | Lưu trữ thông tin độ thành tích theo điểm của các môn học |
| 16 | phobert\_paraphased\_tomtat | Lưu trữ PhoBERT embedding vector của tóm tắt đã được xử lý của các môn học |

### 4.3. Chi tiết các bảng dữ liệu

#### 4.3.1. Bảng student

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | mssv | varchar | Mã số sinh viên |
| 2 | gender | int | Giới tính |
| 3 | start\_year | int | Năm học bắt đầu |
| 4 | major\_id | varchar | Mã ngành học |
| 5 | faculty\_id | varchar | Mã khoa |
| 6 | training\_system\_id | varchar | Mã hệ đào tạo |
| 7 | password | varchar | Mật khẩu đăng nhập vào hệ thống |

#### 4.3.2. Bảng score

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | mssv | varchar | Mã số sinh viên |
| 2 | course\_id | varchar | Mã môn học |
| 3 | year\_id | varchar | Mã năm học |
| 4 | term\_id | varchar | Mã kỳ học |
| 5 | term\_number\_id | varchar | Mã số đếm kỳ học |
| 6 | score | float | Điểm số đạt được |
| 7 | status | int | Trạng thái môn học |
| 8 | passed | boolean | Đánh dấu môn học có hoàn thành hay không? |

#### 4.3.3. Bảng course

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | course\_id | varchar | Mã môn học |
| 2 | course\_name | varchar | Tên môn học |
| 3 | credit | int | Số tín chỉ |
| 4 | major\_id | varchar | Mã ngành học |
| 5 | course\_type\_id | varchar | Mã loại môn học |
| 6 | group\_course\_type\_id | varchar | Mã nhóm loại môn học |
| 7 | summary | varcar | Tóm tắt môn học |

#### 4.3.4. Bảng major

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | major\_id | varchar | Mã ngành học |
| 2 | major | varchar | Tên ngành học |

#### 4.3.5. Bảng faculty

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | faculty\_id | varchar | Mã khoa |
| 2 | faculty | varchar | Tên khoa |

#### 4.3.6. Bảng training\_system

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | training\_system\_id | varchar | Mã hệ đào tạo |
| 2 | training\_system | varchar | Tên hệ đào tạo |

#### 4.3.7. Bảng couse\_type

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | course\_type\_id | varchar | Mã loại môn học |
| 2 | course\_type | varchar | Tên loại môn học |
| 3 | group\_course\_type\_id | varchar | Mã nhóm loại môn học |

#### 4.3.8. Bảng group\_course\_type

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | group\_course\_type\_id | varchar | Mã nhóm loại môn học |
| 2 | group\_course\_typ | varchar | Tên nhóm loại môn học |

#### 4.3.9. Bảng year

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | year\_id | varchar | Mã năm học |
| 2 | year | int | Năm học |

#### 4.3.10. Bảng term

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | term\_id | varchar | Mã kỳ học |
| 2 | term | int | Kỳ học |

#### 4.3.11. Bảng term\_number

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | term\_number\_id | varchar | Mã số đếm kỳ học |
| 2 | term\_number | int | Số đếm kỳ học |

#### 4.3.12. Bảng group\_course

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | year\_id | varchar | Mã năm học |
| 2 | term\_number\_id | varchar | Mã kỳ học |
| 3 | faculty\_id | varchar | Mã khoa |
| 4 | group\_course\_type\_id | varchar | Mã nhóm loại môn học |
| 5 | course\_number | int | Trung bình số môn học sinh viên đăng ký |

#### 4.3.13. Bảng group\_sum\_course

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | year\_id | varchar | Mã năm học |
| 2 | term\_number\_id | varchar | Mã kỳ học |
| 3 | faculty\_id | varchar | Mã khoa |
| 4 | course\_number | int | Trung bình số môn học sinh viên đăng ký |

#### 4.3.14. Bảng subject\_popularity

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | year\_id | varchar | Mã năm học |
| 2 | term\_number\_id | varchar | Mã kỳ học |
| 3 | faculty\_id | varchar | Mã khoa |
| 4 | course\_id | varchar | Mã môn học |
| 5 | student\_number | int | Số sinh viên đăng ký môn học |
| 6 | total\_student\_number | int | Tổng số sinh viên |
| 7 | popularity | float | Độ phổ biến (tỉ lệ đăng ký) |
| 8 | popularity\_scaled | float | Độ phổ biến được chuẩn hóa |

#### 4.3.15. Bảng subject\_score

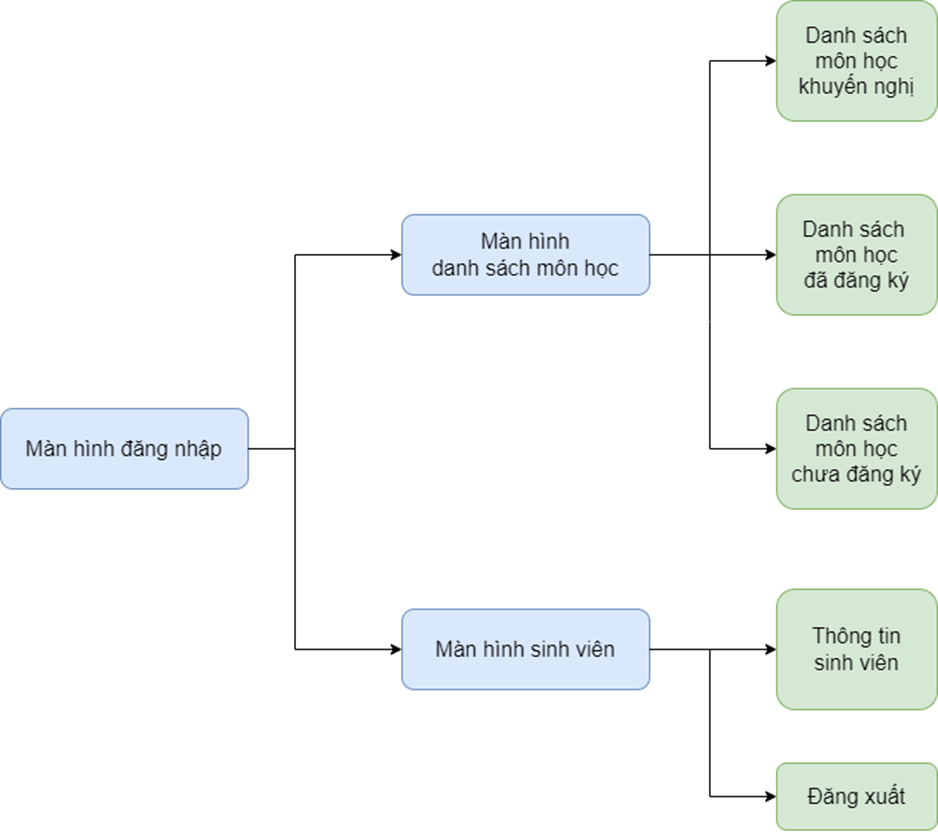
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | year\_id | varchar | Mã năm học |
| 2 | term\_number\_id | varchar | Mã kỳ học |
| 3 | faculty\_id | varchar | Mã khoa |
| 4 | course\_id | varchar | Mã môn học |
| 5 | score | float | Điểm môn học |
| 6 | score\_ratio | float | Độ thành tích (Tỉ lệ điểm) |
| 7 | score\_ratio\_scaled | float | Độ thành tích được chuẩn hóa |

#### 4.3.16. Bảng phobert\_paraphased\_tomtat

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | course\_id | varchar | Mã môn học |
| 2 - 769 | embedding\_dim\_[0-767] | float | PhoBERT Embedding Vector |

## 5. Thiết kế giao diện

### 5.1. Sơ đồ liên kết các màn hình



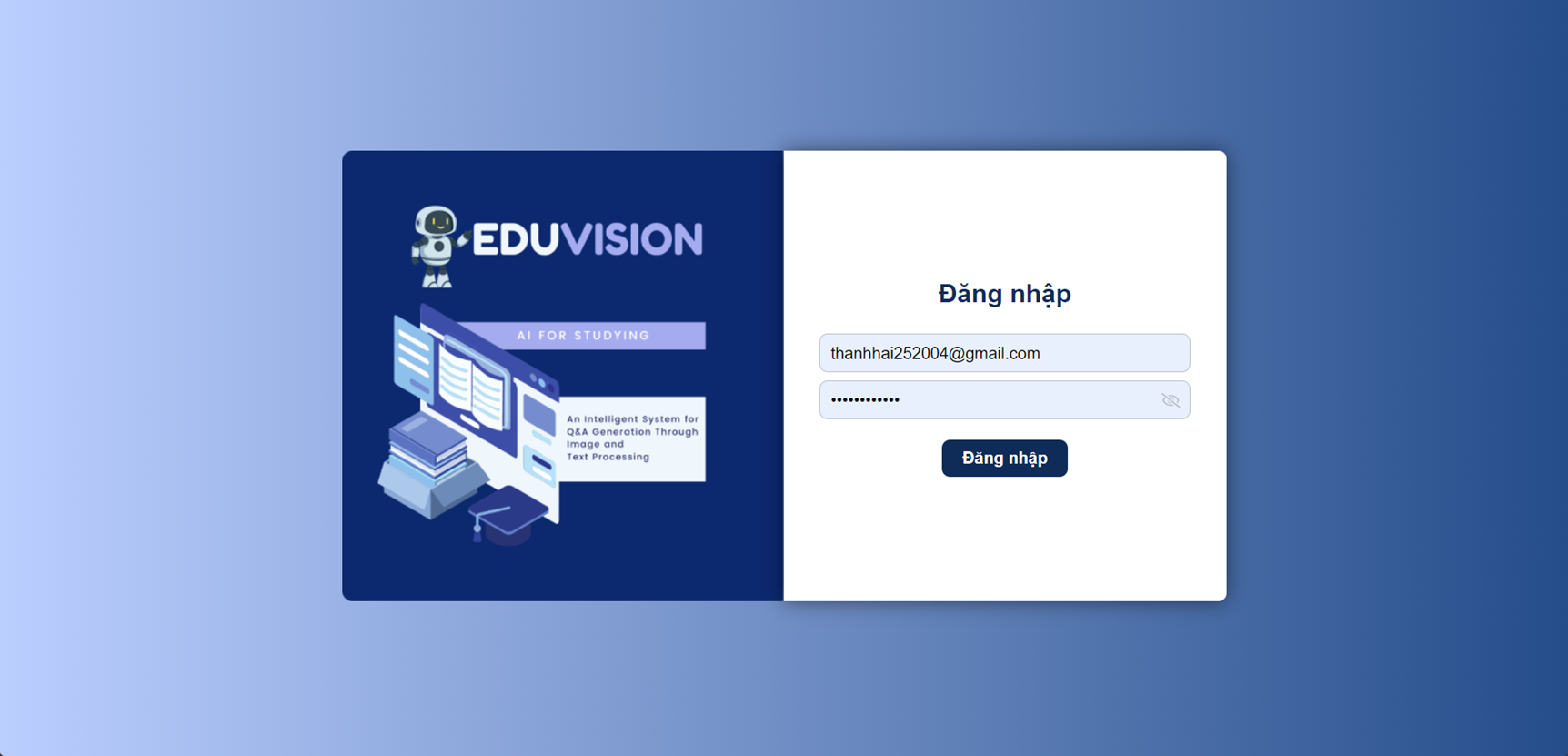
*Sơ đồ liên kết màn hình*

### 5.2. Danh sách các màn hình

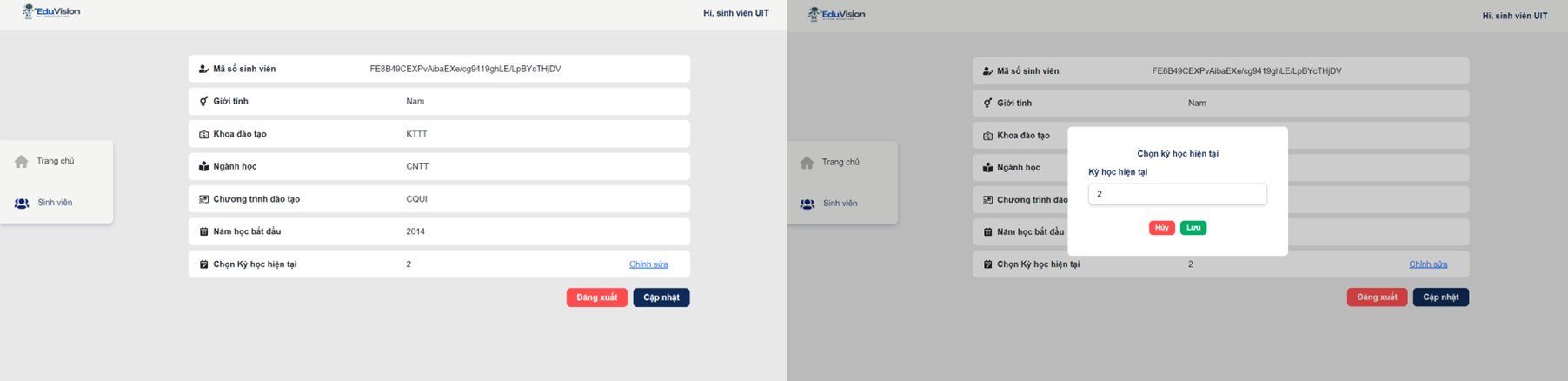
| **STT** | **Màn hình** | **Loại màn hình** | **Chức năng** |
| --- | --- | --- | --- |
| 1 | Đăng nhập | Nhập liệu | Đăng nhập vào hệ thống |
| 2 | Danh sách môn học | Tra cứu | Tra cứu thông các môn học khuyến nghị, đã đăng ký, chưa đăng ký |
| 3 | Sinh viên | Tra cứu  Nhập liệu | Tra cứu thông tin sinh viên  Nhập kỳ học tiếp theo  Đăng xuất khỏi hệ thống |

### 5.3. Giao diện màn hình

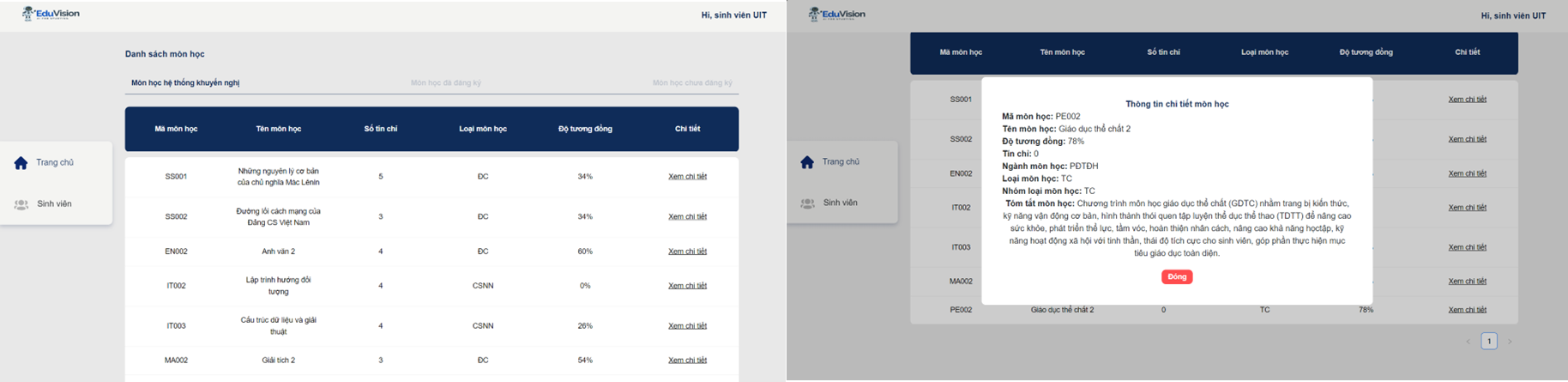
#### 5.3.1. Đăng nhập



#### 5.3.2. Danh sách môn học



#### 5.3.3. Sinh viên



# **CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **1. Kết luận**

Đồ án "Hệ thống khuyến nghị môn học cho sinh viên UIT từ dữ liệu học tập" đã giải quyết được nhu cầu cấp thiết trong việc cá nhân hóa lộ trình học tập của sinh viên thông qua việc phân tích dữ liệu học tập và áp dụng các phương pháp học máy tiên tiến. Hệ thống không chỉ giúp tối ưu hóa việc lựa chọn môn học mà còn hỗ trợ sinh viên lập kế hoạch học tập dài hạn, giảm thiểu nguy cơ rớt môn và cải thiện điểm số trung bình.

Các thuật toán khuyến nghị được đề xuất, như sử dụng ma trận tương đồng cosine, phân cụm K-Means, hoặc mô hình học sâu MLP, đều được áp dụng và thử nghiệm hiệu quả trên bộ dữ liệu thực tế từ sinh viên UIT. Quá trình triển khai không chỉ dừng lại ở việc xây dựng mô hình chính xác mà còn chú trọng đến việc tối ưu hóa giao diện và trải nghiệm người dùng thông qua nền tảng website tích hợp.

Đồng thời, đồ án cũng đã làm nổi bật những thách thức và giới hạn trong việc xử lý dữ liệu học tập không đồng nhất, đề xuất giải pháp nâng cao chất lượng dữ liệu, và chứng minh tính hiệu quả của các phương pháp học máy trong bối cảnh thực tế.

## **2. Hướng phát triển**

**Mở rộng phạm vi ứng dụng**:

Hệ thống hiện tại được thử nghiệm trên dữ liệu sinh viên UIT, nhưng có thể mở rộng để áp dụng cho các trường đại học khác hoặc các hệ thống giáo dục với chương trình đào tạo tương tự. Điều này đòi hỏi việc tùy chỉnh thuật toán và cấu trúc dữ liệu để phù hợp với đặc thù của từng tổ chức.

**Tích hợp thêm dữ liệu ngoại vi**:

Để nâng cao độ chính xác, có thể tích hợp dữ liệu từ các nguồn khác như đánh giá của giảng viên, phản hồi từ sinh viên về độ khó của môn học, hoặc dữ liệu thị trường lao động để liên kết lựa chọn môn học với cơ hội nghề nghiệp sau tốt nghiệp.

**Nâng cấp mô hình học máy**:

Áp dụng các mô hình tiên tiến hơn như Graph Neural Networks (GNN) để phân tích mối liên hệ giữa các môn học và hành vi học tập của sinh viên.

Kết hợp kỹ thuật học sâu với các mô hình ngôn ngữ lớn (như GPT) để xây dựng các khuyến nghị ngữ cảnh hóa hơn dựa trên mục tiêu và sở thích cá nhân.

**Cải thiện trải nghiệm người dùng**:

Phát triển thêm các tính năng như giải thích gợi ý môn học, cung cấp dự đoán rủi ro rớt môn, và đề xuất lộ trình học tập theo từng mục tiêu cụ thể như tốt nghiệp sớm hay cải thiện GPA.

Tối ưu hóa giao diện để thân thiện với người dùng hơn, tăng cường khả năng tương tác, và hỗ trợ đa ngôn ngữ.

**Đảm bảo tính bảo mật và riêng tư dữ liệu**:

Áp dụng các công nghệ bảo mật tiên tiến để bảo vệ dữ liệu cá nhân của sinh viên, đồng thời tuân thủ các quy định pháp luật về bảo mật dữ liệu như GDPR hoặc các quy định địa phương tương tự.

Github của nhóm: <https://github.com/group5-ds317/DS-Project>

# **TÀI LIỆU THAM KHẢO**

- Mondal, B., Patra, O., Mishra, S., & Patra, P. (2020). A course recommendation system based on grades.   
Retrieved from <https://www.researchgate.net/publication/342688390>

- Algarni, S., & Sheldon, F. (2023). Systematic review of recommendation systems for course selection. Machine Learning and Knowledge Extraction, 5(2), 560–596. <https://doi.org/10.3390/make50200334>

- Vedula Lakshmi Spandana, Joga Rohit Abhinav, Pothu Naga Sampath, Mohammed Musaiyab Ahamed. (2023). Content-based filtering and cosine similarity for course recommendation systems. Journal of Emerging Technologies and Innovative Research, 10(3). <http://www.jetir.org/papers/JETIR2303833.pdf>

- Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 1037–1042). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.findings-emnlp.92>

Liang, J., Sun, F., Zhang, L., & Jin, J. (2023). Discrete conditional diffusion for reranking in recommendation. arXiv preprint arXiv:2308.06982. <https://arxiv.org/abs/2308.06982>